

ГОУВПО «Донецкий национальный технический университет»

Институт компьютерных наук и технологий

Факультет интеллектуальных систем и программирования

Кафедра «Программная инженерия» им. Л.П. Фельдмана

Программная инженерия:

**методы и технологии разработки информационно-
вычислительных систем**

(ПИИВС-2022)

**Сборник материалов IV Международной научно-практической
конференции**

г. Донецк

29-30 ноября 2022 года

Донецк – 2022

Программная инженерия: методы и технологии разработки информационно-вычислительных систем (ПИИВС-2022): сборник научных трудов IV Международной научно-практической конференции, Том 1. 29-30 ноября 2022 г. – Донецк, ГОУВПО «Донецкий национальный технический университет», 2022. – 251 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;
- интеллектуальные системы, анализ данных и распознавание образов;
- компьютерное моделирование, 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 образования и науки ДНР. В организации конференции приняли участие: Министерство образования и науки ДНР, ГОУВПО «Донецкий национальный технический университет», ФГБОУ ВПО «Ульяновский государственный технический университет» (г. Ульяновск), ФГАОУ ВО «НИУ Московский институт электронной техники» (МИЭТ, г. Москва).

В первом томе сборника научных трудов представлено 47 докладов сотрудников академических институтов и высших учебных заведений, а также специалистов других научных организаций из Донецкой Народной Республики, Российской Федерации, Республики Беларусь, Израиля.

ГОУВПО «Донецкий национальный технический университет», 2022

ОГЛАВЛЕНИЕ

ПЛЕНАРНЫЕ ДОКЛАДЫ	8
А.Я. Аноприенко (Донецк, ДонНТУ) СИСТЕМОДИНАМИКА КОМПЬЮТЕРНОЙ И ПРОГРАММНОЙ ИНЖЕНЕРИИ: ВЫЗОВЫ И ПЕРСПЕКТИВЫ	8
Б.А. Кулик (Санкт-Петербург, Институт Проблем Машиноведения РАН) РАСШИРЕНИЕ ВОЗМОЖНОСТЕЙ ЛОГИЧЕСКОГО АНАЛИЗА ЗА СЧЕТ УТОЧНЕНИЯ ИНТЕРПРЕТАЦИИ ИСЧИСЛЕНИЯ ПРЕДИКАТОВ	9
А.Я. Фридман (Институт информатики и математического моделирования ФИЦ КНЦ РАН) СИТУАЦИОННОЕ МОДЕЛИРОВАНИЕ БЕЗОПАСНОСТИ КРИТИЧЕСКИХ ИНФРАСТРУКТУР	10
В.Н. Штепа (Республика Беларусь, УО «Полесский государственный университет») СТРУКТУРА И ФУНКЦИОНАЛ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЙ РЕШЕНИЙ В ВОДООТВЕДЕНИИ	16
Григорьев А.В. (Донецк, ДонНТУ) СЕМИОТИЧЕСКАЯ МОДЕЛЬ И СИТУАЦИОННОЕ УПРАВЛЕНИЕ КАК ФОРМА ПРЕДСТАВЛЕНИЯ И СЕМАНТИКА КОНЦЕПТУАЛЬНОЙ МОДЕЛИ ФИЗИЧЕСКИХ ПРЕДМЕТНЫХ ОБЛАСТЕЙ САПР	17
СЕКЦИЯ 1. «ПРОГРАММНЫЕ СИСТЕМЫ, ЯЗЫКИ И ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ, ТЕСТИРОВАНИЕ И КАЧЕСТВО ПО»	18
М.Ю. Копачев (Рыбинск, Рыбинский государственный авиационный технический университет имени П.А. Соловьева) ЭКСПЕРИМЕНТАЛЬНАЯ ОЦЕНКА БЫСТРОДЕЙСТВИЯ СОРТИРОВОК JAVA STANDARD LIBRARY	18
А.О. Истягин, И.А. Коломойцева (Донецк, ДонНТУ) АЛГОРИТМ ПОСТРОЕНИЯ WEB-API СРЕДСТВАМИ ПЛАТФОРМЫ .NET	25
С.В. Щедрин (Донецк, ДонНТУ), Д.Г. Кривошея (Академия Министерства внутренних дел Донецкой Народной Республики имени Ф.Э. Дзержинского), Д.В. Харламов (Донецк, ДонНТУ) РАЗРАБОТКА ШАБЛОНА МОБИЛЬНОГО ЛЕНДИНГ-ПРИЛОЖЕНИЯ ДЛЯ ANDROID....	31
В.В. Данилевич, О.В. Рычка (Донецк, ДонНТУ) МОДЕЛЬ ВЫПОЛНЕНИЯ КОДА В СРЕДЕ CLR	39
А.Д. Суров, Д.Г. Раннев, А.В. Чернышова (Донецк, ДонНТУ) ВЫБОР АРХИТЕКТУРЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ СОЗДАНИЯ ОБРАЗОВАТЕЛЬНОГО ПОРТАЛА	46
В.Ю. Олейник, А.В. Чернышова (Донецк, ДонНТУ) РАЗРАБОТКА И РАЗВЕРТЫВАНИЕ TELEGRAM БОТА ДЛЯ ОПОВЕЩЕНИЯ О НОВОСТЯХ ДОННТУ	53

О.В. Морозова (Донецк, ДонНТУ) РАЗРАБОТКА СЕРВИСА ОПРЕДЕЛЕНИЯ МЕСТОПОЛОЖЕНИЯ ЧЛЕНОВ ГРУППЫ С ДОВЕРИТЕЛЬНЫМИ ОТНОШЕНИЯМИ.....	59
СЕКЦИЯ 2. «ИНФОРМАЦИОННЫЕ СИСТЕМЫ, БАЗЫ ДАННЫХ, БЕЗОПАСНОСТЬ И ЗАЩИТА ДАННЫХ»	66
В.В. Бондаренко, А.В. Чернышова (Донецк, ДонНТУ) ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ ПРОТОКОЛОВ ПЕРЕДАЧИ ДАННЫХ, ИСПОЛЬЗУЮЩИХ КРИПТОГРАФИЧЕСКУЮ ЗАЩИТУ	66
А.В. Чернышова, А.А. Афанасьева (Донецк, ДонНТУ) МЕТОДЫ БЕЗОПАСНОЙ АУТЕНТИФИКАЦИИ В КЛИЕНТ-СЕРВЕРНЫХ ПРИЛОЖЕНИЯХ.....	74
О.Г. Артеменко (Донецк, ДонНТУ) ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ И ПРОИЗВОДИТЕЛЬНОСТИ БАЗ ДАННЫХ ПУТЁМ ОПТИМИЗАЦИИ SQL-ЗАПРОСОВ	81
А.В. Боднар, В.А. Коломойцев (Донецк, ДонНТУ) ОБЪЕКТНЫЙ АНАЛИЗ И МОДЕЛИРОВАНИЕ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УЧЕТА ДЕЯТЕЛЬНОСТИ ПРЕДПРИЯТИЯ	88
СЕКЦИЯ 3. «ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ, АНАЛИЗ ДАННЫХ И РАСПОЗНАВАНИЕ ОБРАЗОВ»	89
О.С. Леонова, И.А. Коломойцева (Донецк, ДонНТУ) РАЗРАБОТКА РЕКОМЕНДАТЕЛЬНОЙ СИСТЕМЫ НА ОСНОВЕ КОНТЕНТНО- ОРИЕНТИРОВАННОЙ ФИЛЬТРАЦИИ ДЛЯ КНИЖНОГО ИНТЕРНЕТ-МАГАЗИНА.....	89
И.В. Савицкая, Р.В. Лазарский (Донецк, ДонНТУ) ПРОГНОЗИРОВАНИЕ СТОИМОСТИ НЕДВИЖИМОСТИ НА ОСНОВЕ МОДЕЛИ ЛИНЕЙНОЙ РЕГРЕССИИ	94
И.А. Коломойцева (Донецк, ДонНТУ) АЛГОРИТМ ФОРМИРОВАНИЯ ТЕЗАУРУСА ДЛЯ РАСШИРЕНИЯ ЗАПРОСА ИНФОРМАЦИОННО-ПОИСКОВОЙ СИСТЕМЫ НА ПРИМЕРЕ ПРЕДМЕТНОЙ ОБЛАСТИ «ПРОГРАММИРОВАНИЕ».....	99
Д.В. Дручевский, О.И. Федяев (Донецк, ДонНТУ) РАСПОЗНАВАНИЕ ЛИЦ ЧЕЛОВЕКА СТАТИСТИЧЕСКИМИ МЕТОДАМИ ИЗ ПАКЕТА MATLAB	105
Т.Г. Дмитрюк (Донецк, ДонНТУ) АЛГОРИТМИЧЕСКОЕ ОБЕСПЕЧЕНИЕ СППР ОПТИМАЛЬНОГО ПЛАНИРОВАНИЯ ПРОИЗВОДСТВЕННО-ЛОГИСТИЧЕСКОЙ ДЕЯТЕЛЬНОСТИ ПРЕДПРИЯТИЯ	110
А.А. Суханов, Н.М. Ткачѳв, О.И. Федяев (Донецк, ДонНТУ) ВИЗУАЛЬНЫЙ КОНТРОЛЬ ПРИСУТСТВИЯ СТУДЕНТОВ В АУДИТОРИИ НА ОСНОВЕ ГЛУБОКОЙ НЕЙРОННОЙ СЕТИ	111

О.А. Криводубский, Е.Н. Павлюк (Донецк, ДонНТУ) ИНФОРМАЦИОННО ТЕХНОЛОГИЧЕСКИЕ СИСТЕМЫ ПОДГОТОВКИ ПРИНИМАЕМЫХ РЕШЕНИЙ ПО ЗАПРОСАМ И РЕАЛИЗАЦИИ ГОРНО-ШАХТНОГО ОБОРУДОВАНИЯ.....	116
СЕКЦИЯ 4. «КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ, СИСТЕМЫ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ, КОМПЬЮТЕРНОЙ ГРАФИКИ И ОБРАБОТКИ ИЗОБРАЖЕНИЙ».	
С.В. Иваница, В.А. Мишустин (Донецк, ДонНТУ) СОВРЕМЕННОЕ СОСТОЯНИЕ МЕТОДОВ КОМПЬЮТЕРНОЙ ОБРАБОТКИ ИНФОРМАЦИИ ПРИ ВИЗУАЛИЗАЦИИ ОБЪЕКТОВ	117
О.А. Серёженко, С.А. Зори (Донецк, ДонНТУ) ПРОБЛЕМАТИКА ПОИСКА ИЗОБРАЖЕНИЙ ПО ИХ СОДЕРЖИМОМУ И ПОДХОДЫ К ЕЁ РАЗРЕШЕНИЮ	125
Н.А. Бездетный, С.А. Зори (Донецк, ДонНТУ) РАЗРАБОТКА МОДЕЛИ И СИМУЛЯЦИИ РАСПРОСТРАНЕНИЯ ЭПИДЕМИИ ВИРУСА С ИСПОЛЬЗОВАНИЕМ ИГРОВОГО ДВИЖКА UNITY	133
И.Н. Паламарь, А.И. Гагарина (Рыбинск, Рыбинский государственный авиационный технический университет имени П.А. Соловьева) МЕТОДИКА ИССЛЕДОВАНИЯ НЕЙРОСЕТЕВОЙ ТЕХНОЛОГИИ РЕАЛИЗАЦИИ E2E СИСТЕМ СИНТЕЗА СИГНАЛА	134
Б.Ю. Кныш И.Ю. Анохина (Донецк, ДонНТУ) КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ МЕТОДОМ МОНТЕ-КАРЛО.....	140
Н.А. Задорина, В.В. Непомилуев (Рыбинск, Рыбинский государственный авиационный технический университет имени П.А. Соловьева) ПРИМЕНЕНИЕ МОДЕЛИРОВАНИЯ ПРОЦЕССА ИНДИВИДУАЛЬНОГО ПОДБОРА ДЕТАЛЕЙ ПРИ СБОРКЕ ВЫСОКОТОЧНЫХ ИЗДЕЛИЙ.....	146
Н.А. Задорина, А.А. Малышева (Рыбинск, Рыбинский государственный авиационный технический университет имени П.А. Соловьева) РАЗРАБОТКА И ИССЛЕДОВАНИЕ МОДЕЛЕЙ ПРОЦЕССА РАСПРОСТРАНЕНИЯ ВРЕДОНОСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В КОРПОРАТИВНЫХ СЕТЯХ....	155
Д.В. Бельков, В.И. Зензеров, В.Н. Лавренюк (Донецк, ДонНТУ) САМООРГАНИЗАЦИЯ АВТОМОБИЛЬНОГО ПОТОКА В КРИТИЧЕСКОЕ СОСТОЯНИЕ	161
Д.В. Бельков, Д.Д. Лосев (Донецк, ДонНТУ) АНАЛИЗ ТРАФИКА БЕСПРОВОДНОЙ СЕТИ	169
Д.В. Бельков, Д.Г. Рубанов (Донецк, ДонНТУ) МОДЕЛЬ СТРУКТУРЫ УГОЛЬНОГО ПЛАСТА.....	177
Д.В. Бельков, М.И. Трушкин (Донецк, ДонНТУ) ИССЛЕДОВАНИЕ ВИДЕО ТРАФИКА КОМПЬЮТЕРНОЙ СЕТИ.....	184

В.Н. Беловодский, С.Л. Букин (Донецк, ДонНТУ) ОДИН ПОДХОД К ФОРМИРОВАНИЮ ПОЛИГАРМОНИЧЕСКИХ КОЛЕБАНИЙ В ВИБРАЦИОННЫХ МАШИНАХ С ИНЕРЦИОННЫМ ВОЗБУЖДЕНИЕМ.....	191
З.Е. Филер (г. Нетания, Израиль) НЕРАВЕНСТВА В КОМПЛЕКСНОЙ ОБЛАСТИ И ГИПОТЕЗА РИМАНА	198
СЕКЦИЯ 5. «ПРОЕКТИРОВАНИЕ ПРОГРАММНЫХ И КОМПЬЮТЕРНЫХ СИСТЕМ, СРЕДСТВА АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ ПО И СИСТЕМ».....	199
А.В. Григорьев, Е.С. Бондаренко (Донецк, ДонНТУ) КЛАССИФИКАЦИЯ МОДЕЛЕЙ САПР С ПОМОЩЬЮ МАШИННОГО ОБУЧЕНИЯ.....	199
А.Д. Стальнов, А.В. Григорьев (Донецк, ДонНТУ) ПРОБЛЕМАТИКА АДАПТИВНОСТИ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ В ОБЛАСТИ НЕЙРОСЕТЕВЫХ ТЕХНОЛОГИЙ.....	205
Д.А. Савельева, А.В. Григорьев (Донецк, ДонНТУ) О ЗАДАЧЕ ПРОЕКТИРОВАНИЯ УНИВЕРСАЛЬНОГО КОНСТРУКТОРА НЕЙРОННЫХ СЕТЕЙ С ВОЗМОЖНОСТЬЮ ЭКСПОРТА И ИМПОРТА МОДЕЛЕЙ РАЗЛИЧНЫХ ФОРМАТОВ	214
Н.А. Горбунов, А.В. Григорьев (Донецк, ДонНТУ) ДИНАМИЧНЫЙ РАСЧЕТ ТВЕРДОТЕЛЬНОЙ ДЕТАЛИ С ПРИМЕНЕНИЕМ МЕТОДА КОНЕЧНЫХ ЭЛЕМЕНТОВ.....	221
А.В. Гончаров, А.В. Григорьев (Донецк, ДонНТУ) АНАЛИЗ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ЛОГИСТИКИ В РАЗРЕЗЕ МОДИФИКАЦИИ И СИНТЕЗА ТРЕБОВАНИЙ ДЛЯ СОЗДАНИЯ УНИВЕРСАЛЬНОГО ОПТИМИЗИРОВАННОГО ПРОГРАММНОГО ПРОДУКТА.....	230
В.Г. Шапалин, Д.В. Николаенко (Донецк, ДонНТУ), И.А. Янковский (Республика Беларусь, УО «Полесский государственный университет») ПРОЕКТИРОВАНИЕ КОМПЬЮТЕРНОЙ СИСТЕМЫ РАСПОЗНАВАНИЯ ЛИЦ	236
Р.В. Мальчева, А.И. Кобыляцкий (Донецк, ДонНТУ), Л.П. Володько (Республика Беларусь, УО «Полесский государственный университет») ВЫБОР СРЕДСТВ РЕАЛИЗАЦИИ КИБЕРБЕЗОПАСНОСТИ СИСТЕМЫ «УМНЫЙ ДОМ»	241
С.И. Изосимова, В.Н. Пигуз (Донецк, ГУ «Институт проблем искусственного интеллекта») КОМПЬЮТЕРНАЯ СИСТЕМА ПСИХОФИЗИОЛОГИЧЕСКОГО ДИАГНОСТИРОВАНИЯ, ИНТЕЛЛЕКТУАЛЬНО-ДУХОВНОЙ РЕАБИЛИТАЦИИ И БЕЗМЕДИКАМЕНТОЗНОЙ ТЕРАПИИ	246
О.В. Рычка (Донецк, ДонНТУ) ПРАКТИЧЕСКОЕ ИСПОЛЬЗОВАНИЕ МЕТОДОВ ПОИСКА И КОРРЕКТИРОВКИ АНОМАЛИЙ ДЛЯ ПОСТРОЕНИЯ ТОЧНЫХ ПРОГНОЗОВ	252
Л.О. Воробьёв, А.В. Григорьев (Донецк, ДонНТУ)	

ОБЗОР ИНСТРУМЕНТОВ РАЗРАБОТКИ ДИАЛОГОВЫХ АГЕНТОВ С БАЗАМИ ЗНАНИЙ 253

ПЛЕНАРНЫЕ ДОКЛАДЫ

УДК 001.51:122

Системодинамика компьютерной и программной инженерии: вызовы и перспективы

А. Я. Аноприенко

Донецкий национальный технический университет
anoprien@ya.ru, , SPIN-код: 4819-8590

Аноприенко А.Я. Системодинамика компьютерной и программной инженерии: вызовы и перспективы. Статья подготовлена по материалам пленарного доклада международной научной конференции и посвящена системодинамике техносферы, которая рассматривается в качестве нового научного направления. Особое внимание уделено её важнейшим составляющим - системодинамикам компьютерной и программной инженерии. Рассмотрено действие закона Мура в современных условиях. На основе проведенных исследований выполнен анализ основных вызовов, закономерностей и перспектив развития в этой области.

Ключевые слова: системодинамика, компьютерная инженерия, программная инженерия, закон Мура, экспоненциальный рост, закономерности развития.

Anoprienko Alexander. System dynamics of computer and software engineering: challenges and prospects. The article is prepared based on the materials of the plenary report of the international scientific conference and is devoted to the system dynamics of the Technosphere, which is considered as a new scientific direction. Particular attention is paid to its most important components - the system dynamics of computer and software engineering. The operation of Moore's Law in modern conditions is considered. Based on the studies carried out, an analysis of the main challenges, patterns and development prospects in this area are carried out.

Key words: system dynamics, computer engineering, software engineering, Moore's Law, exponential growth, patterns of development.

**Полный текст статьи опубликован в выпуске № 3 (29), 2022 г.
научного журнала «Информатика и кибернетика»**

Расширение возможностей логического анализа за счет уточнения интерпретации исчисления предикатов

Б.А. Кулик

Институт Проблем Машиноведения РАН, Санкт-Петербург
ba-kulik@yandex.ru

Одним из вариантов интерпретации исчисления предикатов является представление предикатов и логических формул в виде отношения, в котором областью истинности (domain) всех переменных является одно и то же множество. В эту интерпретацию внесены следующие изменения: 1) разным переменным соответствуют разные области их изменения; 2) n -местные отношения представлены не как множества кортежей значений переменных, а как объединения декартовых произведений. Обосновано, что математической моделью измененной интерпретации является алгебра кортежей. Такое изменение интерпретации позволяет существенно расширить аналитические возможности логического анализа, в частности, решить задачу вычисления следствий с заданными свойствами.

Ключевые слова: интерпретация, логический анализ, аксиоматический подход, математическая логика, теория множеств, алгебра множеств, алгебра кортежей.

Kulik B. A. Expanding the possibilities of logical analysis by clarifying the interpretation of predicate calculus. One of the variants of interpretation of predicate calculus is the representation of predicates and logical formulas in the form of a relation in which the domain of all variables is the same set. The following changes have been made to this interpretation: 1) different variables correspond to different their domains; 2) n -place relations are represented not as sets of tuples of variable values, but as unions of Cartesian products. It is proved that the mathematical model of the modified interpretation is the n -tuple algebra. Such a change in interpretation makes it possible to significantly expand the analytical capabilities of logical analysis, in particular, to solve the problem of calculating consequences with specified properties.

Key words: interpretation, logical analysis, axiomatic approach, mathematical logic, set theory, algebra of sets, n -tuple algebra

**Полный текст статьи опубликован в выпуске № 3 (29), 2022 г.
научного журнала «Информатика и кибернетика»**

Ситуационное моделирование безопасности критических инфраструктур

А.Я. Фридман

д.т.н., профессор, Институт информатики и математического моделирования ФИЦ КНЦ РАН,
fridman@iimm.ru, OrcID: 0000-0003-2408-6892, SPIN-код: 1512-1669

Фридман А.Я. Ситуационное моделирование безопасности критических инфраструктур. Представлена концепция решения задач превентивной аналитики безопасности структурных проектов развития промышленно-природных систем (ППС) различного масштаба в рамках ранее разработанной автором ситуационной системы моделирования (ССМ), которая предназначена для количественного сопоставления альтернативных вариантов реализации структуры ППС. Новизна предлагаемого подхода состоит в интеграции средств изучения слабо формализованных сложных нестационарных пространственных объектов в рамках причинно-следственных моделей и комплексном использовании экспертных знаний для формирования критериев и выбора предпочтительных структур ППС, которые при необходимости можно изучать более детально в имитационном режиме. Система моделирования обеспечивает возможность поиска и сценарного прогнозирования «узких мест» в структуре ППС с возможностью оценки ущербов от зависимых отказов и обоснования рациональных мер по их предупреждению с помощью специализированных цифровых двойников компонентов ППС.

Ключевые слова: промышленно-природная система, превентивная аналитика в задачах безопасности, ситуационный цифровой двойник.

Введение

Предпосылки проекта:

- профилактика нештатных и чрезвычайных ситуаций (НЧС) не менее важна, чем оперативное реагирование на них;
- самые значительные ущербы наносят зависимые (сложные, комплексные) отказы (ЗСКО) [1], порождающие цепочки событий, которые инициируют НЧС;

Цель проекта: обеспечение комплексной интеллектуальной обработки разнородной информации о статических и динамических характеристиках потенциально опасных объектов и их взаимосвязях для выявления возможностей ЗСКО.

Объекты моделирования – промышленно-природные системы (ППС) – территории, экосистемы, технологии и т.д., чьи свойства существенно зависят от географических координат и от времени.

Актуальность задачи вызвана усложнением и взаимопроникновением ППС; ростом неопределенностей в мире из-за сложности и ускорения развития; увеличением доступности данных в связи с быстрым развитием Интернета Вещей (IoT). Особенно важна поставленная проблема, очевидно, для так называемых критических инфраструктур [2], «сбой в работе которых в любом случае отразится на здоровье, безопасности и благосостоянии граждан страны».

Метод решения: превентивная аналитика с целью поиска и сценарного прогнозирования «узких мест» в структуре взаимосвязанных ППС с возможностью оценки ущербов от ЗСКО и обоснования рациональных мер по их предупреждению.

Новизна подхода состоит в интеграции средств изучения слабо формализованных сложных нестационарных пространственных объектов и комплексном использовании экспертных знаний для формирования критериев и осуществления выбора предпочтительных вариантов реализации структуры ППС, изучаемых более детально в режиме имитационного моделирования.

Возможности реализации проекта определяются, естественно, финансовыми и календарными ограничениями, но также связаны с дополнением рождающегося Интернета вещей новым сектором – Интернетом моделей (цифровых двойников) вещей (IoMoT – Internet of Models of Things), то есть с переходом от IoT к IoTaM (Internet of Things and Models). Для повышения безопасности, целесообразно до встраивания новых устройств и систем в реальный мир встроить их модели в мир виртуальный.

С учетом изложенного выше, в качестве среды реализации описанной концепции превентивной аналитики безопасности предлагается использовать ранее разработанную с участием и под руководством автора ситуационную систему моделирования (ССМ) [3-5]. Обоснование такого предложения проводится в следующем разделе настоящей работы.

Обзор основных идей ССМ

Ситуационная система моделирования [3-5] конкретизирует идеи ситуационного управления [6] для задачи моделирования ППС. Поскольку существование подобных объектов сегодня в первую очередь определяется способами управления ими, ППС моделируются в ССМ в рамках структурного подхода [7]. Первоначально они рассматривались как организационные иерархии [8], позднее было предложено обобщение разработанного метода ситуационного моделирования для сетцентрических структур [9, 10].

Как и во многих современных системах моделирования (ближайшим аналогом ССМ по функциональности представляется интеллектуальная СППР «Ресурс-действие-объект» (РДО) [11]), ССМ базируется на информационной модели предметной области, в ССМ эта модель называется ситуационной концептуальной моделью (СКМ). Качественное различие между ССМ и аналогами состоит в том, что:

- ситуационная концептуальная модель ППС строится с использованием новых формальных определений базовых понятий ситуационного управления (ситуация, классификация, пополнение и обобщение ситуаций, сценарий и т.д.); эти определения соответствуют требованиям системного подхода [12], что обеспечило автоматизацию всех этапов обработки ситуаций и выполнения имитационного моделирования;

- модель ППС допускает оперативное изменение по мере появления новых знаний о характеристиках объекта моделирования, причем в ходе таких изменений выполняется детализированный анализ корректности и consistency модели, что позволяет не потерять ее обзорность и управляемость;

- в модели реализованы унифицированные процедуры расчетно-логической обработки разнородной информации о свойствах ППС, вырабатываемой расчетными модулями, а также встроенными в ССМ геоинформационной (ГИС) и экспертной (ЭС) системами; это облегчает сопоставление ситуаций в статике и динамике и создает возможность традиционной для интеллектуальных систем генерации «быстрых прототипов» компонентов ППС;

- моделирование проводится в обобщенном пространстве состояний ППС, метрика которого строится с учетом весовых коэффициентов, отражающих мнения экспертов о степени опасности того или иного значения переменной состояния; на этой основе модели опасных и чрезвычайных ситуаций рассматриваются как расширения моделей нормального функционирования на области значений переменных, которые выходят за пределы диапазонов безопасной работы ППС;

- в модели не применяется нечеткая и другие неклассические логики, что поддерживает возможность направленного перебора ситуаций с целью выявления предпосылок ЗСКО и анализа эффективности мер по предупреждению таких отказов.

Теперь кратко рассмотрим структуру СКМ и организацию моделирования.

В СКМ допускаются три типа сущностей: *объекты*, каждый из которых соответствует некоторой структурной компоненте ППС; *ресурсы*, моделирующие все допустимые (как материальные, так и информационные) связи между элементами ППС, и *процессы*, формализующие все преобразования ресурсов из одного типа в другой. Для унификации процедур обработки каждому ресурсу приписывается *домен* – упорядоченный по степени желательности (расстоянии от номинального или идеального значения) список значений этого ресурса, включающий как допустимые (безопасные), так и опасные (нежелательные и критические) значения.

Первый этап построения СКМ – конструирование дерева объектов, изоморфного (с заданной степенью детализации) организационной структуре ППС. В ходе разработки этого дерева не только формируется общая структура СКМ, но и решаются следующие основные задачи:

- привязка СКМ к карте, для чего в каждой ветви дерева один или несколько объектов атрибутируются списком ГИС-элементов, формирующих их графическое представление; таким образом создается и поддерживается при любых корректировках модели биективное однозначное соответствие между концептуальной и географической спецификациями ППС для автоматизации процедур работы с графическими параметрами объектов (измерения таких параметров и выдачи результатов их расчета на карту);

- в модель вводятся предназначенные для сопоставления *альтернативные структуры* реализации ППС либо декомпозицией объектов на подобъекты с помощью отношения классификации (ИЛИ), либо заданием нескольких вариантов получения входных наборов ресурсов, требующихся для работы процессов или объектов;

- специфицируются критерии качества работы объектов, для чего каждому объекту назначается некоторый *обобщенный критерий качества* (ОКК) [13], формализующий требования суперобъекта к заданному набору параметров данного объекта (эти параметры выбираются из набора выходных ресурсов объекта и формализуют вектор скалярных критериев качества (СКК) его работы) путем назначения номинального значения и допустимого отклонения текущего значения от номинального для каждого СКК. Значение ОКК вычисляется как взвешенная сумма значений всех СКК, причем весами служат обратные величины допусков на отклонение соответствующего СКК от его номинального значения. Это представляется логичным для учета предпочтений ЛПР: чем важнее для него данный СКК, тем меньшее отклонение допустимо для этого СКК.

Разработанный ОКК, во-первых, непосредственно отражает требования вышестоящего ЛПР к данному объекту ППС, а во-вторых, имеет еще одно привлекательное свойство: он принимает нулевое значение только когда ОПР идеально соответствует требованиям ЛПР, то есть все СКК принимают номинальные значения, и значение ОКК равно единице, если все СКК находятся на границе заданного им допустимого отклонения от

идеала. Это свойство (назовем его *условием нормировки ОКК*) значительно упрощает поиск источника проблем в работе ППС: этим источником является объект, который находится на самом нижнем уровне поддерева объектов, входящих в исследуемый фрагмент, и имеет значение своего ОКК существенно больше единицы.

Предложенный способ формирования ОКК показал свою эффективность не только при оценке текущего состояния ППС, но и при планировании и координации взаимодействий нескольких ЛПР, ответственных за работу различных компонентов одного ППС [14].

Способ компьютерной реализации сущностей СКМ конкретизируется заданием каждому из них *исполнителя* (имитатора, в последнее время для программ, выполняющих ту же функцию, используют термин *цифровые двойники* [15]). В ССМ исполнители процессов устанавливают способ доступа и тип имитирующего программного модуля, исполнители ресурсов задают способ получения их значений в виде временных рядов (из БД, вычислением заданной функции и т.п.), а исполнители объектов организуют процедуры получения и выдачи их графических характеристик. Имитационный режим, в конечном счете, сводится к последовательности вызовов исполнителей процессов согласно моделируемому сценарию с предварительной подготовкой нужных для этого исходных данных и автоматическим синтезом баз данных для хранения полученных результатов. При наличии корректной исходной информации при этом решается задача Коши для обобщенного многомерного разностного уравнения [16], образованного уравнениями вызываемых программных модулей-исполнителей процессов и продукционными правилами, если исполнителем каких-либо процессов назначена ЕС ССМ.

Для реализации *ситуационного подхода* в СКМ применяются следующие основные понятия.

Факт есть минимальная порция информации в СКМ, задающая подписание значений из домена одного ресурса, желательных или, наоборот, запрещенных для текущего состояния ППС (факты «за» и факты «против» соответственно). *Исходная ситуация* есть набор фактов, интересующих ЛПР и задаваемых им как цель текущего сеанса имитации. Затем система анализирует эту ситуацию с учетом зависимостей, имеющих в СКМ, и расширяет ее до *полной ситуации* (ПС), которая включает исходную ситуацию и моделируется некоторым связным фрагментом СКМ, позволяющим организовать имитацию. Корневой объект полной ситуации называется *объектом принятия решений* (ОПР), его место в дереве объектов показывает организационный (административный) уровень принятия решений в данной ПС. По построению любая ПС может включать несколько альтернативных структур реализации соответствующего ей фрагмента ППС, каждая такая альтернатива не содержит вариантов реализации и именуется *достаточной ситуацией* (ДС), поскольку в ней достаточно данных для автоматической генерации исполнительской среды и начала имитации. Различные ДС при том же ОПР можно сравнивать между собой в статике по значению ОКК этого ОПР либо в динамике на некотором интервале времени. Последовательность таких ДС реализует возможный *сценарий* функционирования фрагмента ППС.

Основная операция обработки ситуаций в ССМ, как и в методе ситуационного управления вообще, – *классификация ситуаций*. Идея этой операции в ССМ следующая. Естественно полагать, что наиболее важен для ЛПР тот СКК ОПР, который меньше всего отличается от требований ЛПР, то есть его текущее значение вносит минимальный вклад в ОКК (назовем этот критерий *доминирующим* в текущей ситуации), то к одному классу относятся все ДС с одинаковым доминирующим критерием, а в одном классе лучше та ДС, для которой вклад доминирующего СКК в ОКК меньше. Тогда *оптимальной* в каждом классе будет ДС в наименьшем вкладе доминирующего критерия в ОКК. Именно такая структура рекомендуется ЛПР для реализации, если требуется перевести модель ППС из текущего класса ситуаций в новый.

Принятие управленческих решений в ССМ производится путем изменения структуры рассматриваемого фрагмента или всей СКМ в целом для того, чтобы перевести процесс работы этого фрагмента из текущего класса ситуаций в новый, выбранный ЛПР. В таком случае ССМ должна предложить ЛПР предпочтительную для него структуру из нового класса.

Описанная выше схема классификации ситуаций непосредственно по ОКК хорошо работает при оперативном управлении функционированием ППС, но имеет существенный недостаток при стратегическом моделировании: в ней результаты классификации зависят от значений переменных состояния, которые в реальности повторяются крайне редко. Поскольку при стратегическом моделировании предпочтительные ДС выбираются из архива ситуаций, требовалось уменьшить эту зависимость, для чего разработана процедура *когнитивной классификации ситуаций* [17] на базе нормализованной модели Тверски [18], которая была модифицирована для иерархических систем. В отличие от «классической» классификации, которая не различает представителей внутри класса, когнитивная классификация вводит понятия «наиболее типичного» представителя каждого класса – прототипа (наиболее известные примеры: поэт – Пушкин, фрукт – яблоко) и меру типичности остальных представителей этого класса как расстояние до прототипа. Предложенная нами иерархическая модель Тверски может применяться для иерархических структур, поскольку учитывает «затухание» влияния различий между представителями по мере понижения уровня иерархии, на котором имеются эти различия. На основе полученной классификации выполняется необходимая в ситуационном управлении [6] операция обобщения *описания ситуаций* с целью сохранения в архиве ситуаций только наиболее эффективных из них.

Ситуационный анализ безопасности ППС в ССМ

Как уже упоминалось, ситуационное моделирование ППС в ССМ ведется в обобщенном пространстве состояний [19], позволяющем одинаково работать с переменными любых типов. В этом пространстве опасные и аварийные (критические) значения переменных состояния «расширяют» список допустимых значений каждой переменной (*безопасный диапазон* – safety range, SR), что дает возможность единообразно исследовать нормальные режимы работы ППС и НЧС, при необходимости с заменой исполнителей элементов ППС, когда имеет место переход данных элементов из нормального режима функционирования в нештатный или аварийный. Для классификации режимов работы введены следующие типы сценариев функционирования ППС на заданном интервале времени: допустимый (запрещенный) – не содержащий (содержащий) критических значений переменных состояния; безопасный (опасный) – не содержащий (содержащий) недопустимых значений таких переменных. Для нормальной работы ППС синтезируются безопасные допустимые сценарии, предпочтительный выбирается из них по минимуму суммы обобщенных затрат [3] на функционирование ОНР.

Анализ НЧС и эффективности мер по минимизации ущерба от них выполняется перебором опасных сценариев, могущих появиться при реализации того или иного набора доступных по финансированию мер, и выбором среди этих сценариев имеющего минимальную сумму обобщенных затрат при отсутствии критических значений переменных состояния ОНР и объектов из подчиненного ему фрагмента СКМ.

Для автоматизации процедур синтеза сценариев работы ППС с учетом аспектов надежности и безопасности в ССМ на основе метода логико-вероятностного анализа (ЛВА) [20] разработаны логические модели надежности и безопасности ППС [3]. Существенное отличие предложенных логических моделей от моделей ЛВА состоит в степени их детализации: в ЛВА они базируются на укрупненной блок-схеме ППС, а в ССМ могут строиться с детальностью до отдельного ресурса.

В настоящее время цифровые двойники обычно применяются для оптимизации характеристик систем в ходе их разработки. В области рассматриваемых здесь задач исследования безопасности ППС требуются другие, специализированные для решения таких задач *ситуационные цифровые двойники* (СЦД) составных частей ППС, которые должны корректно моделировать опасные воздействия вновь разрабатываемых ППС на окружающие системы и обратно – имитировать процессы реагирования новых ППС на потенциально опасные значения характеристик уже существующих ППС. Эти требования необходимо положить в основу разработки и сертификации СЦД. Развитие ППС может привести к необходимости модификации СЦД уже существующих компонентов ППС, поскольку новые компоненты могут переходить в нештатные режимы и при таких значениях характеристик существующих частей ППС, которые не приводили к опасностям для других уже работающих компонентов ППС.

Моделирование ситуационной осведомленности в ССМ

Понятие «ситуационная осведомленность» (СО), судя по известным публикациям (например, [21]), в настоящее время связана с наиболее общим подходом к реализации ситуационного подхода к задачам исследования сложных систем во временной области. Поэтому представляется целесообразным рассмотреть возможность конкретизации этого концепта для задач исследования ППС. Исходно парадигма СО создавалась для высокодинамичных предметных областей с большими потоками информации, которые требовалось переработать человеку-оператору для принятия корректного решения в текущей ситуации (пилоты скоростных средств передвижения, боевые действия, врачи скорой помощи и т.п.). Оценка степени СО проводилась в основном качественно, путем опроса операторов при работе на тренажерах, при этом анализировались три аспекта (фазы, уровни) обеспечения СО: восприятие существенных элементов вокруг оператора, их осмысливание (comprehension) и выработка прогноза будущих состояний системы. Также отмечалась важная роль целей для принятия адекватных решений, что характерно для любых задач системного анализа [12], и необходимость применения ментальных моделей задачи. Хотя в исследованиях ППС обычно нет необходимости в системах реального времени, другие положения концепции СО представляются весьма существенными для моделирования ситуаций в ППС. Основные аналогии между рассматриваемой задачей и базовыми принципами СО состоят в следующем.

Ментальная модель ППС реализуется в ССМ концептуальной моделью исследуемой системы. Процесс восприятия окружающей среды в рамках ситуационной модели ППС происходит при задании пользователем исходной ситуации, которая его интересует; понимание моделируется расширением исходной ситуации до соответствующей полной ситуации, включающей всю информацию, необходимую для исследования данной исходной ситуации; прогнозирование осуществляется посредством имитации различных сценариев развития этой полной ситуации. В ходе имитации ЛПР может формировать и менять цель моделирования путем перевода модели ППС из одного класса ситуаций в другой.

В результате исследования перечисленных и других аналогий между ССМ и базовыми принципами СО впервые предложены численные оценки степени достижения СО в целом и на всех трех ее уровнях [10, 22]. Эти оценки применяются для координации решений нескольких ЛПР, участвующих в работе одной ППС. При переходе от иерархической модели ППС к сетцентрической модели возникает понятийная неопределенность, связанная с отсутствием конструктивного определения сетцентрических структур как системы (нет обоснования их эмерджентности [23]). Мы предлагаем преодолеть эту неопределенность путем использования СО как

метакритерия качества сетцентрической структуры. Тогда разработанные численные оценки степени СО позволяют рассматривать ее как систему и динамически формировать зоны ответственности различных ЛПР [23], что согласуется с идеологией сетцентрического подхода и составляет существенное отличие сетцентрической СКМ от иерархической.

Заключение

В настоящей работе кратко представлена интеллектуальная информационная технология ситуационного моделирования проблем превентивной аналитики с целью выявления возможностей и минимизации последствий нештатных и чрезвычайных ситуаций, могущих возникнуть при внедрении новых структурных компонентов в состав уже функционирующих промышленно-природных систем вследствие не предусмотренных заранее опасных сочетаний параметров взаимовлияющих объектов. Эта технология относится к интеллектуальным, поскольку опирается на знания экспертов как при разработке моделей исследуемых объектов, так и в ходе принятия решений о качестве их работы и возможностях модификации их структуры, опираясь на информацию от специализированных цифровых двойников таких объектов.

Распределенная реализация предложенного проекта позволяет решить вопросы правообладания и режимности применения локальных подмоделей составных частей ППС (их цифровых двойников) по аналогии с GRID-технологиями. Разработчики новых компонентов ППС смогут обосновывать безопасность включения своего объекта в структуру ППС, анализируя его взаимосвязи с другими компонентами, уже специфицированными в системе моделирования. Такой анализ представляется важной составной частью технико-экономического обоснования предполагаемых структурных решений и существенным аспектом повышения их объективности в современном стиле цифровизации экономики России.

Идея предложенного проекта становится все более и более актуальной с ростом виртуальной части нашего мира, поэтому *представляется разумным тестировать наши проекты в дополненной и/или виртуальной реальности, прежде чем внедрять их в «реальную реальность»*, которая становится все менее и менее способна защитить себя. В результате реализации проекта Интернет Вещей должен быть дополнен новым сектором – Интернетом Моделей Вещей (Internet of Models of Things – IoMoT) и превратиться в Интернет Вещей и Моделей (Internet of Things and Models – IoTaM):

$$\text{IoT} + \text{IoMoT} = \text{IoTaM!}$$

Литература

1. ГОСТ 27.002-89: Надежность в технике. Основные понятия. Термины и определения.
2. Критическая инфраструктура. – Электрон. дан. – 2022. – Режим доступа: https://mont.com/Content/files/pad_pad360%20-%20whitepaper%20-%20%D0%BA%D1%80%D0%B8%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B5%20%D0%B8%D0%BD%D1%84%D1%80%D0%B0%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D1%8B.pdf
3. Фридман, А. Я., Курбанов В. Г. Ситуационное моделирование надежности и безопасности промышленно-природных систем // Информационно-управляющие системы. – 2014. – № 4(71). – С. 1-10.
4. Фридман, А. Я., Курбанов В. Г. Формальная концептуальная модель промышленно-природного комплекса как средство управления вычислительным экспериментом // Труды СПИИРАН. – 2014. – № 6(37). – С. 424-453.
5. Фридман, А. Я. Опыт интеллектуализации методов ситуационного моделирования дискретных нестационарных пространственных объектов // Автоматика и телемеханика. – 2022. – № 6. – С. 151-168.
6. Поспелов, Д. А. Ситуационное управление: теория и практика. – М.: Наука, 1986.
7. Nam Pyo Suh. Axiomatic Design - Advances and Applications. Chapter 5. – New York: Oxford University Press, 2007. – P. 239-298.
8. Месарович, М., Мако Д., Такаха И. Теория иерархических многоуровневых систем. – М.: Мир, 1973.
9. Фридман, А. Я., Кулик Б.А. Возможности управления сетцентрическими системами на основе знаковых формализмов // Труды Кольского научного центра РАН, серия «Информационные технологии». – 2020. – Вып. 11. – С. 161–165.
10. Фридман, А. Я., Б. А. Кулик. Когнитивный подход к оценке ситуационной осведомленности в сетцентрических системах гражданского назначения // Гибридные и синергетические интеллектуальные системы: материалы V Всероссийской Поспеловской конференции с международным участием / под ред. д-ра техн. наук, проф. А. В. Колесникова [Электронный ресурс]: научное электронное издание. – Калининград: Издательство БФУ им. И. Канта, 2020. С. 489-497.
11. Ясиновский, С. И. Интеллектуальная гибридная система поддержки принятия решений РДО: вчера, сегодня, завтра // Мягкие измерения и вычисления. – 2018. – № 12 (13). – С. 33-50.
12. Альбеков, Н. Н. Эмерджентность как объект современной науки // Современные проблемы науки и образования. – 2015. – № 2. – С. 421.
13. Салуквадзе, М. Е. Задачи векторной оптимизации в теории управления. – Тбилиси: Мецниереба, 1975.

14. Фридман, А. Я. Координация и планирование управлений в локально организованных иерархических системах // Шестая Международная конференция «Системный анализ и информационные технологии» САИТ-2015 (15-20 июня 2015 г., г. Светлогорск, Россия): Труды конференции. В 2-х т. Т. 1. – М.: ИСА РАН. – С. 115-124.
15. Digital Twin: Transforming How We Make Sense of Data. – Электрон. дан. – 2022. – Режим доступа: <https://www.ptc.com/ru/industry-insights/digital-twin>
16. Некрасова, Т. И. Задача Коши для многомерного разностного уравнения в конусах целочисленной решетки // Журнал Сибирского федерального университета. Математика и физика. – 2012. – № 5(4). – С. 576-580.
17. Фридман, А.Я., Кулик Б.А. Когнитивная категоризация в многокритериальных задачах ситуационного управления // Пятнадцатая национальная конференция по искусственному интеллекту с международным участием КИИ-2016 (3-7 октября 2016 г., г. Смоленск, Россия). Труды конференции. В 3-х томах. Т. 2. – С. 225-234.
18. Tversky, A. Features of similarity // Psycholog. Rev. – 1977. – V. 84. – No. 4. – P. 327-352.
19. Фридман, А. Я. Экспертное пространство для ситуационного моделирования промышленно-природных систем // Вестник Московского университета им. С.Ю. Витте, № 1 (4), 2014, с. 233-245.
20. Рябинин, И. А. Надежность и безопасность структурно-сложных систем. – СПб., Политехника, 2000. – 248 с.
21. Endsley, M. R. Final Reflections: Situation Awareness Models and Measures // J. of Cognitive Engineering and Decision Making. – 2015. – Vol. 9. – No. 1. – P. 101-111.
22. Fridman, A., Kulik B. Situation Awareness in Modeling Industrial-Natural Complexes. Intern. Conf. Cyber-Physical Systems and Control (CPS&C'2019, St. Petersburg). In: Lecture Notes in Networks and Systems. Cyber-Physical Systems and Control. – Vol. 95, 2020, pp. 247-256.
23. Fridman, A., Oleynik A. Modelling of Situation Awareness in Net-Centric Commercial Systems // Proc. 34th Eur. Simulation and Modelling Conf. (ESM-2020) Toulouse, France, Oct. 21-23, 2020, pp. 64-67.

Фридман А.Я. Ситуационное моделирование безопасности критических инфраструктур. Представлена концепция решения задач превентивной аналитики безопасности структурных проектов развития промышленно-природных систем (ППС) различного масштаба в рамках ранее разработанной автором ситуационной системы моделирования (ССМ), которая предназначена для количественного сопоставления альтернативных вариантов реализации структуры ППС. Новизна предлагаемого подхода состоит в интеграции средств изучения слабо формализованных сложных нестационарных пространственных объектов в рамках причинно-следственных моделей и комплексном использовании экспертных знаний для формирования критериев и выбора предпочтительных структур ППС, которые при необходимости можно изучать более детально в имитационном режиме. Система моделирования обеспечивает возможность поиска и сценарного прогнозирования «узких мест» в структуре ППС с возможностью оценки ущербов от зависимых отказов и обоснования рациональных мер по их предупреждению с помощью специализированных цифровых двойников компонентов ППС.

Ключевые слова: промышленно-природная система, превентивная аналитика в задачах безопасности, ситуационный цифровой двойник.

Fridman Alexander. Situational modeling of safety of critical infrastructures. The concept of solving problems in preventive analytics of safety of structural projects for the development of industrial-natural systems (INSs) of various scales is presented within the framework of the situational modeling system (SMS) previously developed by the author, which is designed to quantitatively compare alternative options for implementing the structure of an INS. The novelty of the proposed approach lies in the integration of tools for studying poorly formalized complex non-stationary spatial objects within the framework of cause-and-effect models and the integrated use of expert knowledge to form criteria and select preferred INS structures, which, if necessary, can be studied in more detail in a simulation mode. The modeling system provides the ability to searching for and scenario-based forecasting of "bottlenecks" in the INS structure with the ability to assess losses from dependent failures and justify rational measures to prevent them by using specialized digital twins of INS components.

Key words: industrial-natural system, preventive analytics in safety problems, situational digital twin.

Структура и функционал интеллектуальной системы поддержки принятий решений в водоотведении

В.Н. Штепа

проректор по научной работе,

УО «Полесский государственный университет»

shtepa.v@polessu.by

***Штепа В.Н. Структура и функционал интеллектуальной системы поддержки принятий решений в водоотведении.** Проанализированы недостатки существующих подходов оценки состава сточных вод с использованием лабораторного оборудования и измерительных средств реального времени. Обоснованы задачи, которые необходимо решать при моделировании процессов водоотведения. На основе математической статистики установлено частичное отсутствие нормальности распределения значений показателей качества сточных вод промышленных предприятий. Разработаны структурные и функциональные параметры интеллектуальной системы поддержки принятия решений водоотведения с её интеграцией в существующие системы управления очистными сооружениями.*

Ключевые слова: интеллектуальная система, водоотведение, оперативный сбор информации, система поддержки принятия решений.

***Shtepa Vladimir. Structure and functionality intelligent decision support system for wastewater disposal.** The shortcomings of the existing approaches to assessing the composition wastewater using laboratory equipment and real-time measuring instruments are analyzed. The problems that need to be solved when modeling the processes of the water disposal are substantiated. On the basis mathematical statistics, partial lack of normality in the distribution of the values wastewater quality indicators industrial enterprises was established. Structural and functional parameters of the intelligent decision support system for wastewater disposal with its integration into existing wastewater treatment plant control systems have been developed.*

Key words: intelligent system, wastewater disposal, operational collection of information, decision support system.

***Полный текст статьи опубликован в выпуске № 3 (29), 2022 г.
научного журнала «Информатика и кибернетика»***

Семиотическая модель и ситуационное управление как форма представления и семантика концептуальной модели физических предметных областей САПР

Григорьев А.В.

Донецкий национальный технический университет, г. Донецк
grigorievalvl@gmail.com

Григорьев А.В. Семиотическая модель и ситуационное управление как форма представления и семантика концептуальной модели физических предметных областей САПР. В статье рассматривается задача построения комплекса методов и средств автоматизации построения интеллектуальных САПР. Обосновывается актуальность применения основных положений семиотической модели и ситуационного управления, предложенных Поспеловым Д.А., как инструментальных средств программного синтеза в САПР. Дается краткая характеристика методов построения интеллектуальных средств, предназначенных для извлечения, формализации и повторного использования экспертных методик проектирования сложных объектов физической природы с позиций семиотической модели и ситуационного управления. Показывается преимущества и результаты практического применения предлагаемого подхода.

Ключевые слова: семиотическая модель, ситуационное управление, концептуальная модель предметной области, САПР.

Grigoriev A.V. Semiotic model and situational control as a form of representation and semantics of the conceptual model of CAD physical subject areas. The article deals with the problem of constructing a set of methods and means of automating the construction of intelligent CAD systems. The relevance of the application of the main provisions of the semiotic model and situational management, proposed by Pospelov D.A., as tools for program synthesis in CAD, is substantiated. A brief description is given of methods for constructing intellectual tools designed to extract, formalize and reuse expert methods for designing complex objects of physical nature from the standpoint of a semiotic model and situational control. The advantages and results of the practical application of the proposed approach are shown.

Key words: semiotic model, situational management, conceptual domain model, CAD

**Полный текст статьи опубликован в выпуске № 4 (30), 2022 г.
научного журнала «Информатика и кибернетика»**

СЕКЦИЯ 1. «ПРОГРАММНЫЕ СИСТЕМЫ, ЯЗЫКИ И ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ, ТЕСТИРОВАНИЕ И КАЧЕСТВО ПО»

УДК 004.021:004.942::004.75

Экспериментальная оценка быстродействия сортировок Java Standard Library

М.Ю. Копачев^{*1}

^{*1} доцент, Рыбинский государственный авиационный технический университет
имени П. А. Соловьева,
rybkmu@gmail.com

Копачев М.Ю. Экспериментальная оценка быстродействия сортировок Java Standard Library. Для эффективного управления нагрузкой в распределенных системах необходимы механизмы прогнозирования времени обработки данных. Упорядочивание входных наборов часто является самой емкой по времени стадией обработки. Теоретические оценки времени работы тех или иных алгоритмов сортировки не учитывают множества эвристик, вводимых в современные алгоритмы. В статье описываются результаты экспериментальной оценки алгоритмов сортировки, включенных в Java Standard Library.

Ключевые слова: Java, сортировка, оценка скорости алгоритма, оценка времени исполнения, моделирование.

Введение

В настоящее время Java является одним из самых распространенных языков, используемых для создания серверных частей распределенных приложений. Этому способствуют кроссплатформенность языка, регулярные обновления, наличие большого пула готовых решений, а также мощная встроенная библиотека, реализующая большинство стандартных алгоритмов и структур данных.

Одним из важных параметров распределенных систем является время обработки пакета данных, а также прогнозируемость этого времени. Более или менее точный прогноз времени исполнения важен на всех этапах разработки программного обеспечения. Если при планировании архитектуры программы достаточно грубой асимптотической оценки скорости работы алгоритма, то при решении задач управления нагрузкой, требуются более точные оценки. Зачастую, для получения такой оценки необходимо учесть не только используемые программные средства, но и производительность, архитектуру и аппаратную конфигурацию конкретного узла вычислительной сети.

Использование распространенных методов обработки данных, вынесенных в стандартные, общедоступные библиотеки ускоряет разработку программ, но при этом переносит часть ответственности на авторов этих библиотек. В ходе совершенствования кода библиотек, происходит не только постоянное обновление используемых алгоритмов, но и внесение в них множества эмпирических улучшений. Такой процесс приводит к тому, что теоретически полученные оценки зачастую не соответствуют реальному поведению алгоритмов.

Другой причиной отклонения поведения алгоритмов от ожидаемого, является сама природа платформы Java. Чисто формально эта платформа является интерпретатором собственного байт-кода, однако достаточно давно в Java-машину встроены механизмы частичной компиляции кода «на лету», влияющие на скорость исполнения кода.

В связи с вышеперечисленным, возникает вопрос: насколько теоретические методы оценки времени выполнения алгоритмов соответствуют практическим результатам? Можно ли их применять на практике?

Для исследования были выбраны алгоритмы сортировок, реализованные в стандартной библиотеке языка Java. Основанием для выбора именно сортировок, стал анализ нестабильного поведения работающей программной системы, в которой упорядочивание входных данных занимает большую часть времени обработки пакетов.

Теоретические оценки

В стандартных библиотеках Java реализовано два различных подхода к сортировке данных: специализированный для стандартных атомарных типов, и обобщенный для пользовательских.

Первый подход заключается в применении предельно оптимизированных под сортируемые данные

алгоритмов. Сортировка атомарных типов различной длины реализована в модуле «Arrays» в виде нескольких перегруженных методов «sort()». Для каждого атомарного типа реализован отдельный метод, максимально учитывающий специфику именно этого типа. Основой для сортировок атомарных значений, согласно [1] является модификация быстрой сортировки «Dual-Pivot Quicksort». Теоретическая оценка для быстрой сортировки хорошо известна, и составляет $O(n \log_2(n))$ для среднего, и $O(n^2)$ худшего случая. Однако, алгоритм, реализованный в Java, использует специальную систему разбиения (схему В. Ярославского), что существенно изменяет его поведение. Детальный теоретический анализ схемы Ярославского приведен в статье [2]. В ней же указывается, что полученная теоретическая модель не советуется практическим результатам, так как не учитывает работу процессорного кэша.

Второй подход применяется в Java для упорядочивания коллекций, списков, векторов и других сложных структур данных, а также массивов, состоящих из неатомарных типов. Данный подход реализуется в модуле «Collections», абстрактным методом «sort()», и согласно [3] является модифицированной сортировкой слиянием (Merge Sort). Теоретическая оценка сложности $O(n \log_2(n))$ сортировки слиянием хорошо известна, и может быть взята, например, из [4]. Для некоторых классов-наследников «Collections», вместо классической разновидности используется ее более современная «Tim Sort» [5], обладающая той же теоретической оценкой.

Тестовые данные

Скорость работы алгоритмов сортировки может существенно отличаться в зависимости от природы сортируемых данных и свойств применяемого алгоритма. Оба реализованных в библиотеках Java варианта являются чувствительными к предварительному упорядочиванию – такие данные обрабатываются быстрее. С целью определения степени чувствительности, применялся набор различных синтетических тестов, а также наборы реальных данных.

Использовались следующие наборы синтетических тестов:

- равномерно перемешанные целые числа без повторений;
- равномерно перемешанные целые числа с числом повторений, не превосходящим некоторого k ;
- равномерно перемешанные, и затем частично упорядоченные целые числа, с длиной упорядоченных участков не превосходящей некоторого m .

Константы k и m подбирались исходя из подобных параметров реальных данных. Синтетических тестов, содержащих как повторения, так и частичное упорядочивание не проводилось – вместо них использовались тесты с реальными данными, полученными из вычислительной системы.

В качестве реальных тестовых данных использовались пакеты с показаниями многопараметрических датчиков. Каждый пакет содержал от 1000 до 20-ти миллионов кортежей целых чисел. Тестовое упорядочивание производилось по одному из полей кортежа, без учета остальных.

В каждой группе тестов производилось измерение времени работы, на данных постепенно возрастающего размера. Для одного и того же размера использовалось 15 наборов с разным исходным порядком ключей. Результаты 15 измерений усреднялись. Полученные данные анализировались и использовались для построения математической модели зависимости времени обработки $t(n)$ от количества сортируемых ключей n .

Отдельно были произведены тесты с различными долями повторяющихся ключей (параметр k). Целью было определение степени чувствительности исследуемых сортировок к доле повторяющихся чисел. Аналогичные тесты с варьирующей длиной упорядоченных участков (параметр m) были сделаны для определения чувствительности сортировок к предварительному упорядочиванию.

Методика измерения времени

Для измерения времени работы алгоритмов использовался встроенный в Java счетчик наносекунд «System.nanoTime()». Выбор обусловлен тем, что именно он планировался к использованию в системе управления нагрузкой. Современные Java-машины корректно обновляют «nanoTime()», используя счетчики производительности центрального процессора, и дают достаточную точность. В полученных данных не наблюдается характерных «ступеней», возникающих при использовании системных таймеров малого разрешения или аппаратных часов компьютера.

Замеры времени на каждом тесте производились 9 раз, затем 2 самых больших и 2 самых малых результата отбрасывались, а оставшиеся 5 усреднялись. Такая схема измерений продиктована тем, что тестирование производилось в виде отдельной программы, на вновь запущенной Java-машине. Виртуальная машина содержит большое количество внутренних структур, подверженных поздней инициализации, что приводит к появлению отчетливого «эффекта прогрева». Так, время работы первого прогона сортировки может на несколько порядков отличаться от времени следующего прогона при тех же самых исходных данных. Серверные виртуальные машины работают почти непрерывно, поэтому измерения для «холодного старта» не актуальны, и были отброшены.

Увеличение количества замеров более 9 не приводило к существенному уточнению результата, но увеличивало время тестирования.

Результаты тестирования

Тестирование на наборах равномерно перемешанных, не повторяющихся 32-битных целых чисел показало, что поведение алгоритмов «Dual-Pivot Quicksort» и «Merge Sort», реализованных в стандартных библиотеках языка Java, существенно отличается. На рисунке 1 показаны графики зависимости времени работы сортировки (в наносекундах) от количества сортируемых ключей.

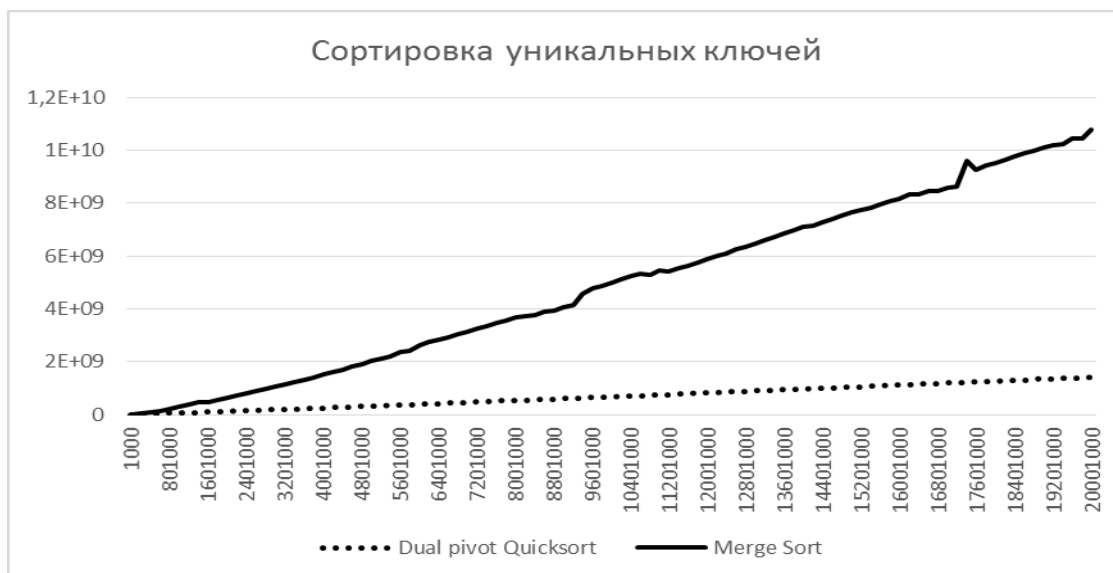


Рисунок 1 – График зависимости времени работы сортировки от количества ключей

На основании анализа данных, можно сказать, что при сортировке уникальных ключей:

- скорость работы «Dual-Pivot Quicksort» существенно выше, чем «Merge Sort»;
- математическая модель вида $t(n) = c \cdot n \cdot \log_2(n)$, достаточно точно описывает зависимость времени исполнения от обрабатываемого объема для обеих сортировок;
- вычисление коэффициента c для «Dual-Pivot Quicksort» может быть осуществлено очень быстро – частная оценка $c_0 = t(n) / n \cdot \log_2(n)$ уже оказывается достаточно точной (при $n > 100\,000$), чтобы ее можно было использовать в качестве параметра модели. Усреднение трех таких оценок дает модель с очень высоким уровнем значимости (по t-критерию уровень меньше 0,01), с отклонением реального времени работы от прогноза менее 0.1%;
- вычисление коэффициента для «Merge Sort» требует применения регрессионных методов после предварительной линеаризации данных. Полученная модель является статистически значимой, но она существенно хуже, чем у предыдущего алгоритма (значимость по t-критерию 0,05). Отклонения времени работы от прогнозного могут быть весьма существенными, и в некоторых тестах достигали 12%.

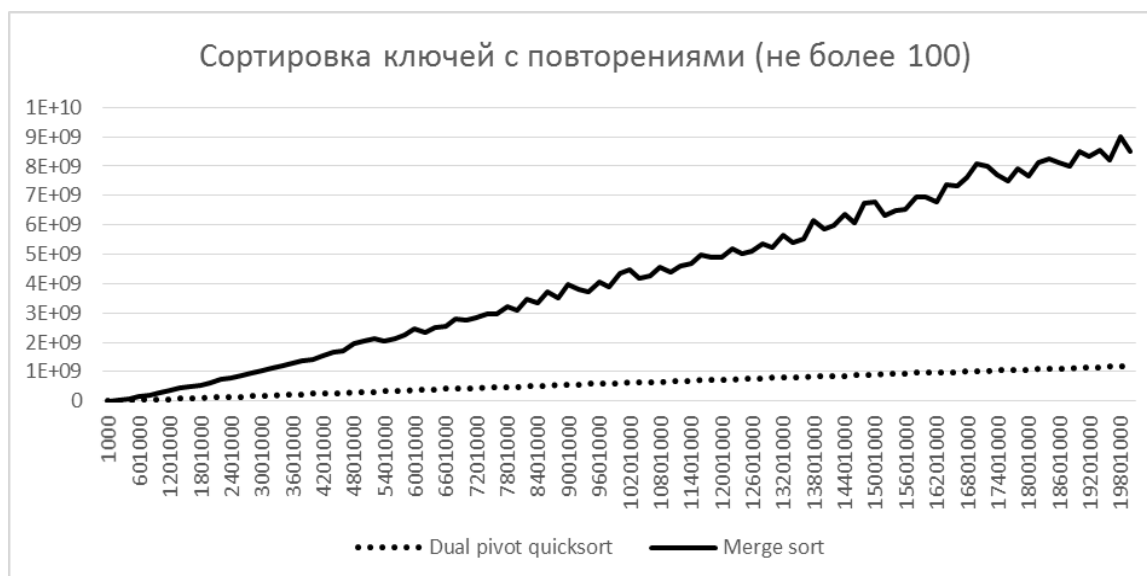


Рисунок 2 – График зависимости времени работы сортировки от количества ключей, в условиях повторяющихся значений

Тест состоящий из уникальных значений ожидаемо оказался худшим для производительности обоих алгоритмов, так как не позволял работать эвристикам, нацеленным на особые случаи входных данных. Полученную на данном этапе модель можно считать пессимистической, и использовать для предсказания верхней оценки времени обработки.

Тестирование на равномерно перемешанных числах, допускающими случайное число повторений (но не больше чем $k = 100$) дало более интересные результаты, которые представлены в виде графиков на рисунке 2.

На основании результатов тестирования с допустимыми повторяющимися значениями, можно сделать следующие заключения:

- обе реализации сортировок чувствительны к повторяющимся ключам – их появление существенно уменьшает время работы;
- появление повторяющихся ключей приводит к резкому возрастанию дисперсии данных для «Merge Sort» при $n > 4\,000\,000$;
- для «Merge Sort», модель $t(n) = c \cdot n \cdot \log_2(n)$ остается значимой, и ошибки прогнозирования остаются в пределах 10-12%, однако частота появления больших отклонений возрастает;
- для «Dual-Pivot Quicksort» модель $t(n) = c \cdot n \cdot \log_2(n)$ значимой быть перестает, а гораздо более точной (и значимой) оказывается линейная модель $t(n) = b + a \cdot n$, получаемая методом регрессии.

Появление повторяющихся ключей приводит к работе эвристик, заложенных в библиотечные алгоритмы. Влияние этих эвристик на алгоритмы существенно разное. В случае алгоритма «Dual-Pivot Quicksort», зависимость времени работы $t(n)$ от размера выборки n приближается к линейной. Однако, во всех случаях коэффициент регрессии b оказывается меньше нуля, что свидетельствует – такая модель не совсем корректна (время сортировки нулевой выборки $t(0) < 0$ не имеет смысла).

Алгоритм «Merge Sort» на появление повторяющихся ключей реагирует гораздо хуже – сильно повышается дисперсия, что ухудшает прогностические свойства математической модели. С другой стороны, применение эвристик серьезно ускоряет работу алгоритма.

Результаты тестирования сортировки частично упорядоченной последовательности оказались более близки к теоретическим ожиданиям. График с результатами тестов, при ограничении на длину упорядоченных подпоследовательностей $m = 100$, показан на рисунке 3.

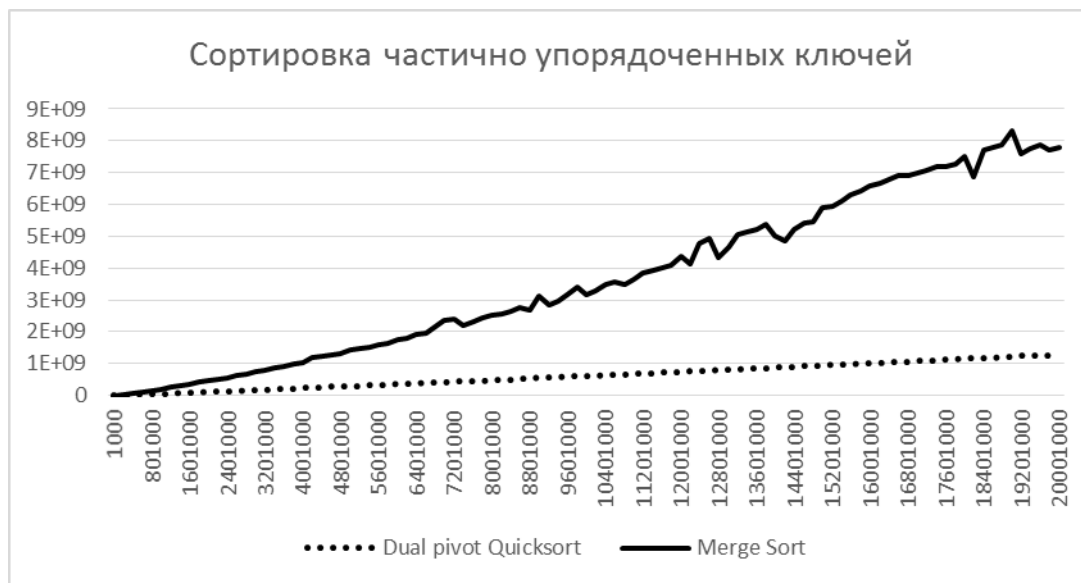


Рисунок 3 – График зависимости времени работы от размера при частично упорядоченных данных

На основании анализа результатов тестов с частично упорядоченными данными, можно сделать следующие выводы:

- обе реализации сортировок чувствительны к частичному порядку – он уменьшает время работы алгоритмов;
- для обоих алгоритмов модель $t(n) = c \cdot n \cdot \log_2(n)$ является статистически значимой;
- появление упорядоченных частей приводит к возрастанию дисперсии для «Merge Sort», но не в такой степени, как в случае повторяющихся ключей;
- для «Dual-Pivot Quicksort» оказывается значима (по t-критерию 0.05) и линейная модель $t(n) = b + a \cdot n$, причем отклонения реальных измеряемых значений от прогнозных для обеих моделей сопоставимы, и находятся в пределах 3%.

Реальные данные содержат как повторяющиеся ключи, так и интервалы с правильным порядком кортежей. В тестовом наборе, максимально повторяющийся ключ встречался 312 раз. 3/4 ключей повторялись менее 91 раза. Интервалы частичного порядка в реальных данных не превышали 17 значений, а наиболее часто

встречались упорядоченные интервалы длины 5.

Размеры пакетов реальных данных подрезались под размеры синтетических тестов, так, чтобы количество кортежей в тесте соответствовало заданному n . Сортировка производилась по одному из полей кортежа. Обработка с помощью «Dual-Pivot Quicksort» производилась путем вынесения ключа сортировки из кортежа в отдельный массив типа «long». Составной ключ получался по формуле $key = k \cdot 2^{32} + idx$, где k – ключ сортировки из кортежа, idx – номер кортежа в несортированном пакете. После применения «Dual-Pivot Quicksort», младшая часть (младшие 32 бита) содержит перестановку начальных кортежей, необходимую для упорядочивания. Время последующей перестановки ссылок на кортежи включался во время работы алгоритма.

График с результатами тестирования показан на рисунке 4.

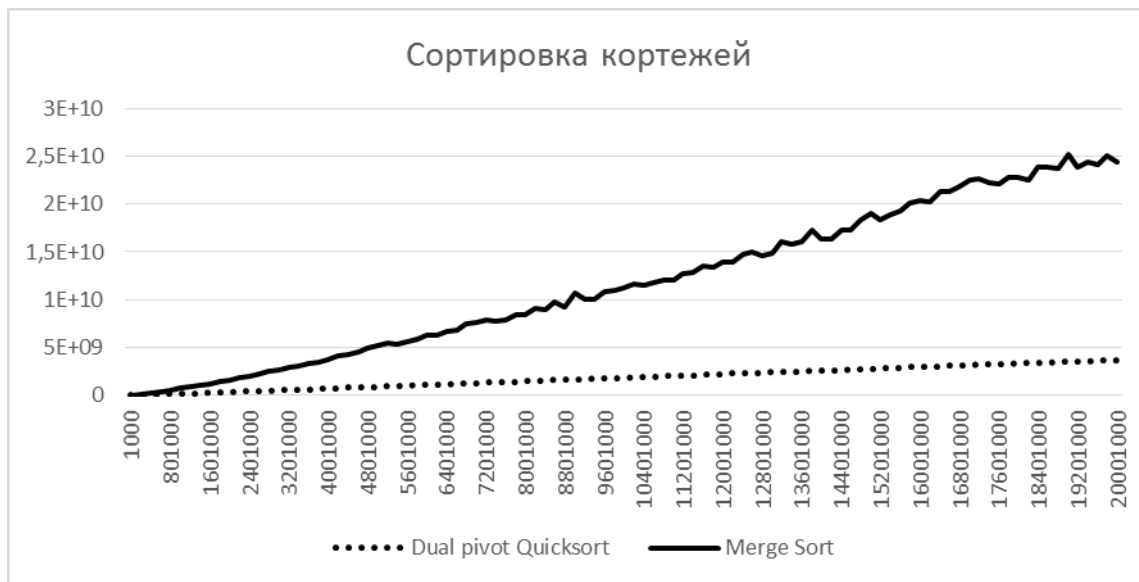


Рисунок 4 – График зависимости времени работы от количества упорядочиваемых кортежей

По результатам анализа данных, полученных на реальных кортежах, можно сделать следующие выводы:

- применение «Dual-Pivot Quicksort» значительно эффективнее чем «Merge Sort», даже учитывая, что применение первой требует дополнительных операций;

- для алгоритма «Merge Sort», модель $t(n) = c \cdot n \cdot \log_2(n)$ является статистически значимой, и наиболее точной, не смотря на высокую дисперсию, вызванную работой эвристик;

- для алгоритма «Dual-Pivot Quicksort», значимыми остаются обе модели: $t(n) = c \cdot n \cdot \log_2(n)$ и $t(n) = b + a \cdot n$.

Использование модели $t(n) = c \cdot n \cdot \log_2(n)$ для прогнозирования «Merge Sort» все также затруднено большой дисперсией, порожденной особенностями работы эвристик с повторяющимися ключами.

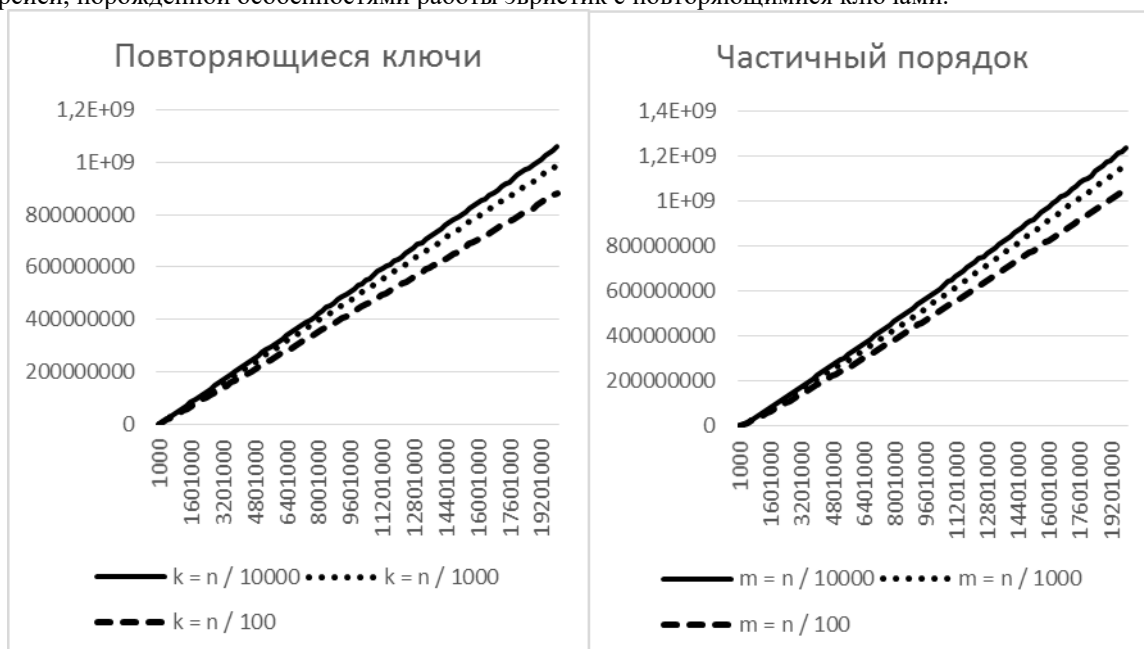


Рисунок 5 – Влияние повторяющихся ключей и частичного упорядочивания на Dual-Pivot Quicksort

На рисунке 5 показаны графики, отражающие влияние повторяющихся ключей (слева) и частичного порядка (справа) на алгоритм «Dual-Pivot Quicksort», реализованный в Java. Повторяющиеся ключи уменьшают время работы алгоритма гораздо сильнее, что хорошо согласуется с логикой работы самого алгоритма, даже без учета эвристики.

Влияние количества повторяющихся ключей на «Dual-Pivot Quicksort» настолько существенно, что при среднем количестве повторов ключей 100 и более, модель $t(n) = c \cdot n \cdot \log_2(n)$ перестает быть статистически значимой, и может давать ошибку предсказания до 10%.

Предварительное частичное упорядочивание влияет на «Dual-Pivot Quicksort» гораздо слабее – даже при достаточно больших упорядоченных отрезках в исходных данных, модель вида $t(n) = c \cdot n \cdot \log_2(n)$ остается рабочей, и может использоваться для предсказаний.

Влияние повторных ключей и частичного упорядочивания сортируемого набора на «Merge Sort» гораздо более специфична – происходит ускорение алгоритма, но при этом увеличивается дисперсия, что понижает ценность предсказаний, полученных с помощью математической модели. Наиболее подходящей, оказывается теоретически обоснованная $t(n) = c \cdot n \cdot \log_2(n)$. При некоторых особенностях входных данных оказывается значимой и линейная модель $t(n) = b + a \cdot n$, но это скорее исключение, чем правило, и ее использование на практике не оправдано.

Полученную линейную модель можно было бы использовать для формирования нижней оценки времени работы алгоритма, однако далее были обнаружены особенности, делающие такую оценку не слишком удачной.

Особенности алгоритмов, выявленные в ходе экспериментов

Анализ данных, полученных в ходе экспериментов однозначно указывает на значительно большую эффективность в Java метода «Arrays.sort()» («Dual-Pivot Quicksort») в сравнении с «Collections.sort()» («Merge sort»). Были проведены дополнительные тесты, с целью проверки, возможно ли полностью отказаться от «Collections.sort()» в пользу более быстрого алгоритма, ценой дополнительных действий. И не превысит ли цена дополнительных операций выигрыш, получаемый от замены. Были выявлены три типовых сценария, в которых такая замена возможна, и проведены тесты для каждого из них.

Первый сценарий: сортировка кортежей по одному из его целочисленных полей. В отдельном тестировании не нуждался, так как возник и был разрешен при проведении тестов на реальных данных. Метод выноса ключей сортировки в отдельный массив, и их объединения с индексами, описывался выше. Замена «Collections.sort()» в таком случае требует двух дополнительных этапов: построения массива ключей, и последующей перестановки ссылок на кортежи. Оба этапа требуют однократного пробега по исходному набору, и не ухудшают теоретическую сложность алгоритма. Практические тесты показывают, что на реальных наборах данных скорость обработки возрастает в 5-6 раз.

Второй сценарий: сортировка ключей в обратном порядке. Библиотеки Java позволяют настроить внешний компаратор для заданий порядка ключей, и передать его алгоритму сортировки, однако «Dual-Pivot Quicksort» использует Java исключительно в случае сортировки массива атомарных типов по возрастанию. Задание внешнего компаратора ведет к переключению сортировки на «Tim Sort» [5]. Несмотря на то, что «Tim Sort» работает значительно лучше, чем классический «Merge Sort», на реальных данных, скорость работы «Dual-Pivot Quicksort» в сумме с реверсом, примерно в 2-4 раза выше, чем у альтернативы.

Третий сценарий: сортировка кортежей по составному ключу из нескольких полей. Данный случай легко сводится к нескольким последовательным сортировкам по отдельным ключам, причем возможны реализации начинающие с самого младшего или самого старшего поля. Данные сортируются сначала по одной части ключа, затем по второй, и так далее. На каждом этапе формируется отдельный массив ключей, в конце каждого этапа происходит перестановка ссылок на кортежи.

При использовании сортировок с оценкой $O(n \cdot \log_2 n)$, выгоднее использовать сортировку начиная с самого значимого поля, а не с младшего. Однако, в случае реализации с помощью «Arrays.sort()» на массиве вынесенных ключей, скорость работы алгоритма, начинающего с младших полей, оказалась незначительно, но выше.

Сравнение скорости «послойной» сортировки с использованием «Dual-Pivot Quicksort» в сравнении с «Merge Sort» выходит не в пользу последней. Достигается ускорение приблизительно в 2-4 раза. Однако надо отметить, что тестирование проводилось только в случаях, когда составными частями ключа были целые числа, и таких частей было не больше 5.

Анализ причин нетипичного поведения сортировки в третьем сценарии потребовал дополнительного тестирования. Выяснилось, что в ряде случаев скорость работы алгоритма «Dual-Pivot Quicksort» не только что созданном массиве ключей оказывается лучше, чем предсказывается даже линейной моделью. Исследование показало, что такое нетипичное поведение вызвано процессорным кэшем. Формирование массива ключей и его последующая сортировка происходят друг за другом, и к моменту начала сортировки часть массива ключей все еще находится в кэше, и может быть обработана очень быстро. Более того, несмотря на то, что реализация «Dual-Pivot Quicksort» в модуле «Arrays.sort()» фактически не является cache-oblivious [6], высокая локальность обрабатываемых данных позволяет получить существенную выгоду от использования процессорного кэша.

Процессорные кэши на разных вычислительных узлах могут иметь различную архитектуру и объем, поэтому их влияние на скорость работы «Dual-Pivot Quicksort» сложно оценить. При тестировании на некоторых

узлах, функция времени работы начинала вести себя нетипично – ее график начинал изгибаться вниз, и линейная модель становилась менее точной. По результатам дополнительных тестов было принято решение о замене линейной функции кубическим сплайном. Испытания показали, что достаточно одного сплайна для описания оптимистической модели $t_{\min}(n) = an^3 + bn^2 + cn + d$ с приемлемой точностью.

Практические результаты

По результатам исследований были выданы рекомендации по полному отказу от «Merge Sort» в пользу «Dual-Pivot Quicksort», и предложены алгоритмы такого перехода.

Был сформирован набор синтетических и реальных тестов, и модуль, позволяющий спрогнозировать поведение алгоритма, основной частью которого является «Dual-Pivot Quicksort», на конкретном узле распределённой обработки. На основе синтетических тестов из неповторяющихся ключей рассчитывался коэффициент c для пессимистической модели $t_{\max}(n) = c \cdot n \cdot \log_2(n)$ оценки времени исполнения. Оптимистическая модель строилась на основании тестов на модифицированных реальных данных, в виде кубического сплайна $t_{\min}(n) = an^3 + bn^2 + cn + d$ (аппроксимацией методом наименьших квадратов). Применение моделей оценок позволило выровнять нагрузку на вычислительные узлы вычислительной системы.

Выводы

В работе описаны экспериментальные данные о реальном поведении алгоритмов сортировки, реализованных в стандартной библиотеке Java. Сформулированы рекомендации по замене менее эффективного и плохо прогнозируемого алгоритма «Merge Sort» более эффективным «Dual-Pivot Quicksort», предложены способы такой замены, и оценены выгоды от нее. Предложены методы для формирования моделей верхней и нижней оценок времени работы алгоритма на конкретном узле вычислительной сети.

Литература

1. Arrays (Java SE 11 & Java JDK 11) - Class Arrays [Электронный ресурс] – режим доступа: <https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/Arrays.html#sort>.
2. Markus E. Nebel · Sebastian Wild · Conrado Martínez Analysis of Pivot Sampling in Dual-Pivot Quicksort [Электронный ресурс] – режим доступа: <https://upcommons.upc.edu/bitstream/handle/2117/89895/pslt-full-paper-main.pdf>.
3. Collections (Java SE 11 & Java JDK 11) - Class Collections [Электронный ресурс] – режим доступа: <https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/Collections.html#sort>.
4. Скиена С. Алгоритмы. Руководство по разработке. – 2-е изд. Пер. с англ. – СПб: БХВ-Перербург, 2011, страницы 141-142.
5. Peters Tim [Python-Dev] Sorting, Python Developers Mailinglist [Электронный ресурс] – режим доступа: <https://mail.python.org/pipermail/python-dev/2002-July/026837.html>.
6. Frigo, M.; Leiserson, C. E.; Prokop, H.; Ramachandran, S. -. Cache-oblivious algorithms [Электронный ресурс] – режим доступа: <http://supertech.csail.mit.edu/papers/FrigoLePr99.pdf>.

Копачев М.Ю. *Экспериментальная оценка быстродействия сортировок Java Standard Library.* Для эффективного управления нагрузкой в распределенных системах необходимы механизмы прогнозирования времени обработки данных. Упорядочивание входных наборов часто является самой емкой по времени стадией обработки. Теоретические оценки времени работы тех или иных алгоритмов сортировки не учитывают множества эвристик, вводимых в современные алгоритмы. В статье описываются результаты экспериментальной оценки алгоритмов сортировки, включенных в Java Standard Library.

Ключевые слова: Java, сортировка, оценка скорости алгоритма, оценка времени исполнения, моделирование.

Kopachev M.Y. *Experimental evaluation of Java Standard Library build-in sorting methods..* For effective load balancing in distributed systems, algorithm to predict the data processing time are needed. Sorting data sets is often the most time-consuming part of processing. Theoretical estimates of the running time of certain sorting algorithms do not take into account the multitude of heuristics, introduced into modern algorithms. This article describes the results, obtained during the experimental evaluation of sorting algorithms build in the Java Standard Library.

Key words: Java, sorting, algorithm speed evaluation, time efficiency evaluation, modeling.

Алгоритм построения Web-API средствами платформы .NET

А.О. Истягин*¹, И.А. Коломойцева*²

*1 ст. гр. ПИ-19а, ФИСП, ДонНТУ, ist_75@mail.ru

*2 ст. преп. каф. ПИ, ФИСП, ДонНТУ, bolatiger@mail.ru

Истягин А.О., Коломойцева И.А. Алгоритм построения Web-API средствами платформы .NET. В статье рассмотрена методика построения RESTFull API средствами платформы .NET и фреймворка EntityFramework. Рассмотрены паттерны MVC, принципы Dependency Injection и SOLID, приведены рекомендации по архитектуре API и именованию объектов в соответствии с руководством Clean Code.

Ключевые слова: программная инженерия, RESTFull, API, JWT, back-end, бэкенд, ASP.NET, .NET, C#, DI, Entity Framework.

Введение

Проектирование и разработка клиент-серверных приложений требуют больших затрат времени и человеческих ресурсов. Любое правильно спроектированное приложение гораздо легче реализовать в коде, чем начинать писать без четкой архитектуры.

Существует множество подходов в реализации клиент-серверных приложений: двухуровневые и трехуровневые системы и микросервисные приложения. Двухуровневые приложения имеют клиентскую и серверную часть, в которой совмещены процессы бизнес-логики и хранение данных. Трехуровневые системы разделяют серверную часть на слои бизнес-логики и данных, что облегчает разработку и сопровождение. Микросервисные системы не имеют привычных слоев, вся бизнес-логика заключена в распределенных микросервисах. Двухуровневые системы подходят предприятиям малого бизнеса, так как они компактнее и проще в разработке, но их трудно поддерживать и расширять. Микросервисные приложения следует применять в крупных информационных системах с большими потоками данных и операций. Трехуровневые системы можно применять в системах любого размера.

Промежуточный слой любого трехуровневого приложения называется API – application programming interface – программный интерфейс приложения. Этот слой обеспечивает связь между клиентским (одним или несколькими) приложением и базой данных (также одной или несколькими). Именно в этом слое обрабатывается вся бизнес-логика приложения, т.к. клиентские интерфейсы мы изначально считаем тонкими и не содержащими функций обработки данных. Как же реализовать правильный API?

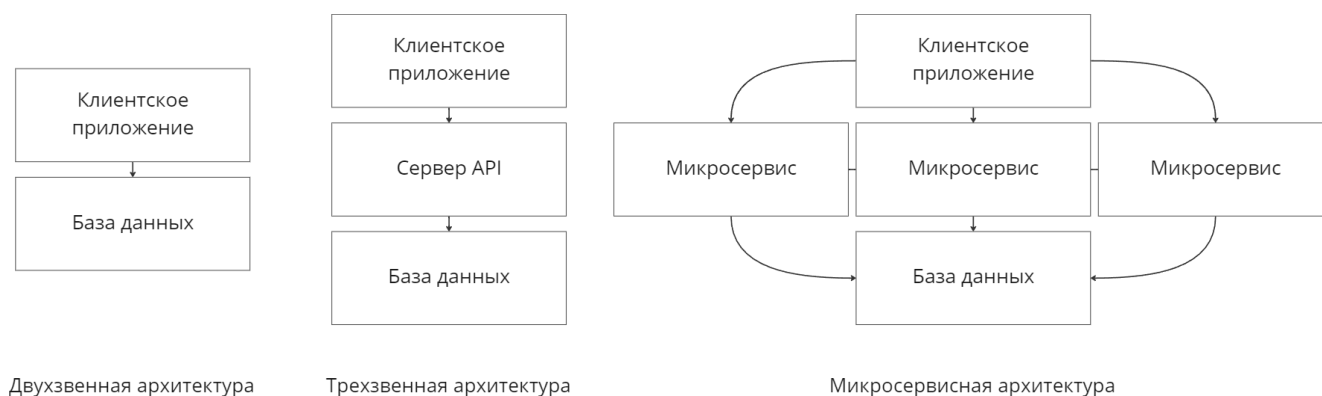


Рисунок 1 – Типы клиент-серверных архитектур

Постановка задачи

Платформа разработки .NET от Microsoft очень быстро развивается. Новые версии .NET выходят каждую осень. На данный момент актуальная версия платформы – 7, однако уже через год она таковой не будет. Предыдущий релиз .NET 6 имеет долгосрочную поддержку и соответствующую приписку LTS – long term support. Выбранная платформа имеет широкий функционал для реализации API: начиная от одностраничных

скриптов и заканчивая объемными системами обработки данных. Вариантов реализации API существует множество, каждая компания пишет по-своему, т.к. установленных стандартов нет.

Целью данной статьи является рассмотрение различных способов построения WebAPI и поиск наиболее оптимальной архитектуры промежуточного слоя.

Анализ существующих подходов

Как было сказано ранее, .NET предоставляет очень широкий спектр возможностей для разработки API, и нет никаких четких инструкций как необходимо писать код. Рассмотрим несколько способов.

Минимальный API [1] представляет собой одностраничный скрипт, который содержит все эндпоинты (конечные точки HTTP/S-запросов для связи с клиентами) и методы для их настройки. Данный метод пригоден только для очень маленьких систем, в которых мало запросов и бизнес-логики (см.рис.2.)

```
var builder = WebApplication.CreateBuilder(args);
var app = builder.Build();

app.MapGet("/", () => "Hello World!");

app.Run();
```

Рисунок 2 – Простейший пример минимального API

Рассмотренный пример возвращает строку “Hello World!”, когда к API приходит HTTP/S-GET запрос по пути “/”. Нагромождение множества запросов в один файл ведет к множественному дублированию кода, отсутствию единого стиля написания и плохой читабельности. Такой код трудно сопровождать в будущем, так как запросы никак не структурированы и могут располагаться в хаотичном порядке.

Более правильным способом является разделение API на несколько слоев. Так, например, можно выделить слой контроллеров, слой бизнес-логики и слой связи с базой данных:

- контроллеры;
- слой бизнес-логики;
- контекст.

Контроллеры – классы, которые служат для связи системы с клиентскими приложениями. Они принимают HTTP/S-запросы и вызывают методы классов из слоя бизнес-логики, после чего возвращают полученное значение. Контроллеры реализуются средствами платформы ASP.NET и не обладают никакой внутренней логикой. Сервисы бизнес-логики - представляют собой множество классов, которые читают и формируют результат, удаляют, добавляют или удаляют данные, вызывая методы класса-контекста. Для чтения данных можно применять LINQ-выражения, Entity SQL или чистый SQL с внедрением параметров. Контекст – класс-драйвер, который имеет непосредственное подключение к базе данных и читает или записывает данные в нее.

Такое разделение существенно облегчает работу программиста и позволяет писать чистый код. Также можно разделить все классы по сущностям предметной области или по функционалу

Алгоритм реализации WebAPI

Рассмотрим наиболее оптимальный способ построения API. Применим фреймворк Entity Framework [2] (далее EF) для реализации промежуточного слоя информационной системы и разберемся, чем он хорош.

Для начала необходимо спроектировать базу данных. EF предполагает подход Code First – «сначала код» в отличие от привычного Base First, в котором сначала реализовывается база, а потом функции работы с ней. После проектирование схемы данных создадим набор классов-моделей, которые в будущем будут представлять собой таблицы базы данных. Для свойств моделей можно указать аннотации и ограничения, записав их в квадратных скобках, н.: [Key], [MinLength], [MaxLength], [ForeignKey] и другие. Если не указывать первичные ключи, фреймворк сам присвоит первичный ключ первому свойству с именем Id. Также для всех полей с именем Id создастся индекс по умолчанию. Здесь же можно предусмотреть какую-либо базовую модель, которая содержит id записи, даты создания, последнего редактирования, удаления записи из таблицы и прочую информацию, которая должна быть у всех таблиц в базе. Далее необходимо создать класс контекста, пусть он называется Context и наследует методы от DbContext. В нем должны находиться шаблоны списков для всех моделей системы – DbSet<Model>.

Создав все классы и контекст, можно приступить к миграции. Миграция в EF – конвертация классов-моделей в SQL-таблицы и автоматическая установка первичных и внешних ключей, типов данных и ограничений соответствующих полей. Производятся миграции командой Add-Migration “название”, после миграции необходимо синхронизировать данные командой Update-Database. Таким образом, имеем сгенерированную базу данных и набор моделей для нее; за обработку операций связи с базой отвечает созданный класс Context.

Далее необходимо реализовать репозитории – открытые классы, из которых можно читать и в которые можно записывать данные. Если контекст отвечает только за физический доступ к данным, то в репозитории можно добавить минимальную логику (не путать с бизнес-логикой). Классы репозитория содержат списки моделей, которые ссылаются на списки из контекста. В репозиториях можно генерировать уникальные GUID, устанавливать даты создания и редактирования записей в таблицах, после чего передавать объекты в контекст для сохранения. Также в репозиториях можно обеспечить функции логгирования операций.

Базовый метод разработки предполагает создание одного репозитория для каждой модели базы данных. Если моделей много, следует применить паттерн `UnitOfWork`, чтобы не нагромождать множество одинаковых классов. В таком случае создается интерфейс базового шаблонного репозитория `BaseRepository<Model>`, а класс `UnitOfWork` хранит множество репозитория. Это уберет проект от излишнего разрастания и обеспечит удобную работу с данными из одного места.

Когда репозитории готовы, можно приступать к слою бизнес-логики. Его также стоит разделить на различные сервисы. Самый простой пример – сервисы чтения `ReadServices` и сервисы записи `WriteServices`. Первые отвечают за чтение и формирование результатов, вторые – за добавление и изменение данных. Именно в них должна быть заключена вся бизнес-логика информационной системы.

Как было сказано ранее, существует множество способов прочитать данные из базы средствами EF. Интегрированные выражения LINQ и Entity SQL сначала транслируются в чистый SQL, и лишь затем выполняются. Лишние операции могут привести к задержкам в выполнении запросов. Из этого следует необходимость самим писать SQL-запросы для получения данных с базы. Для этого создадим набор классов `Query`, в которые поместим базовые запросы `Get(id)`, `GetAll()`, специальные `GetByX(X)`, где X – параметры чтения (условия отбора, порядок сортировки, группировки и т.п.) или любые другие запросы на чтение. Однако пользователю (клиентским приложениям) не нужны сухие таблицы из базы, ему необходимы свои наборы свойств сущностей предметной области. В этом поможет библиотека `Dapper`. Она позволяет преобразовать результат SQL-запроса в экземпляр какого-либо класса. Такие классы называются DTO-модели – data transfer objects – или `WebEntity`. DTO модели служат для обмена информацией между клиентскими и серверными приложениями и содержат только нужную для клиента информацию, а не все данные из таблиц. Для удобства пользования их следует разделять на модели, приходящие с клиентского приложения и уходящие в него, добавляя приписки `Request` и `Response` в конце имени класса, н.: `GetXResponse` и `UpdateXRequest` для получения и обновления сущности X соответственно.

Не менее важной частью являются сервисы записи. В них происходит вся бизнес-логика системы и обработка всех данных. С клиентского приложения через контроллеры (о них позже) также приходят разнородные DTO-модели, свойства которых не совпадают с полями таблиц в базе данных. Для этого необходимо преобразовать данные с клиента в модели базы данных. В этом поможет библиотека `AutoMapper`, с ее помощью можно автоматически преобразовывать объекты DTO в объекты моделей БД. Для корректной работы `AutoMapper` необходимо явным образом указать, преобразования каких объектов будет происходить. Это можно сделать в одном классе `Profile` или в разделенных `XProfile`, где X – имя сущности (правила преобразования объектов в библиотеке принято называть профилями). По умолчанию `AutoMapper` переносит значения полей с одинаковым названием, однако при необходимости можно тонко настроить алгоритм трансляции моделей друг в друга. При необходимости проводить сложные операции бизнес-логики сервис может обращаться к другим сервисам и репозиториям, однако пользователь не должен ожидать ответа. После всех операций объекты можно записывать в репозитории, после чего сохранять изменения в контексте.

Верхний уровень API – контроллеры, разрабатываемые средствами веб-платформы ASP.NET. Методы этих классов вызываются, когда на адрес приложения приходит HTTP/S запрос (с указанием конечного пути – эндпоинта – и передаваемых данных). Для контроллеров также указываются аннотации перед методами, такие как `[HttpGet("{id}")]`, `[HttpPost]`, `[Route("api/[controller]")]` и другие. Контроллеры также разделяют по ключевым сущностям предметной области, и каждый контроллер хранит в себе соответствующие сервисы. При вызове методов он передает данные в методы сервисов и возвращает ответ из них, оборачивая ответ в `Ok(message)` при успешном завершении и `Conflict(message)` в случае ошибки. Эти методы присвоят HTTP/S ответу соответствующие заголовки и статус ответа, после чего вернут ответ клиентской программе. В общем случае передаются объекты в JSON-нотации. Их удобно распознавать на стороне клиента, но они могут иметь избыточное описание при передачах массивов. Важно: в контроллерах не стоит описывать какую-либо логику приложения, это необходимо делать в других классах.

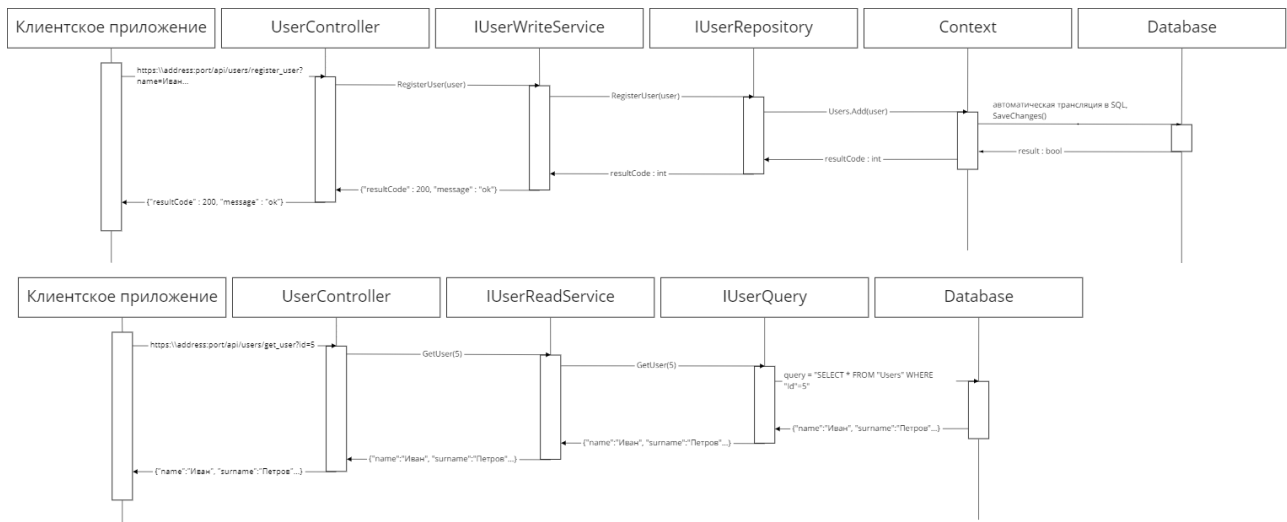


Рисунок 3 – Пример диаграммы последовательностей жизненного цикла запроса

Для функционирования API разработать вышеперечисленные классы недостаточно. Применяя принцип DI [4] – Dependency Injection – внедрение зависимостей – все разработанные классы и их интерфейсы необходимо явно указать в скрипте настройки приложения. Чтобы понять, зачем это, разберем базовый жизненный цикл запроса API. HTTP/S-запрос приходит с клиента, обрабатывается API, проходя через различные сервисы, и возвращается результат обработки. Запросы и процессы их обработки никак не связаны между собой, все операции проходят независимо друг от друга. Существует три способа указать явное внедрение зависимостей. *AddTransient* подразумевает, что сервис создается каждый раз, когда его запрашивают. Этот жизненный цикл лучше всего подходит для легковесных, не фиксирующих состояние, сервисов. *AddScoped* – сервис создается единожды для каждого запроса. *AddSingleton* – сервис создается при первом запросе (или при запуске приложения), а затем каждый последующий запрос будет использовать этот же объект сервиса.

Применяя *AddScoped*, для каждого отдельного запроса экземпляры соответствующих классов (контроллеров, сервисов, репозитория, профилей) создаются заново. Экземпляры всех нижележащих классов необходимо передавать вышележащим экземплярам. Указанные в скрипте настройки классы будут автоматически подставляться в те конструкторы, которые их требуют. Например, в конструкторе контроллера должны быть сервисы, в сервисах – запросы или репозитории. Это существенно сократит время разработки и упростит понимание кода в будущем.

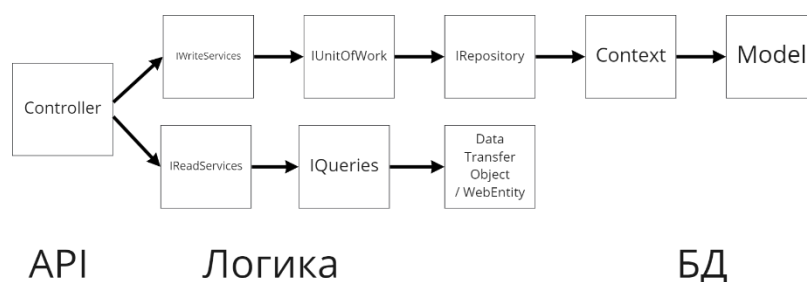
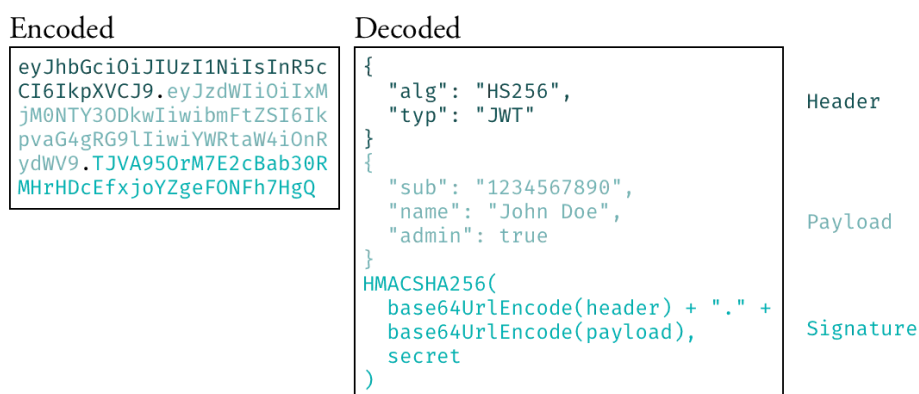


Рисунок 4 – Структура ASP.NET Web API

Немаловажной частью любой системы является механизм безопасной регистрации и авторизации пользователей. Рассмотрим такой механизм, предложенный платформой ASP.NET, который называется ASP.NET Identity. Помимо непосредственной регистрации и авторизации в системе он также позволяет проводить интеграцию со многими сторонними сервисами. Для начала нужно указать контексту, что в базе будет храниться информация о пользователях и их ролях: унаследуем созданный контекст от *ApplicationDbContext*, также создадим модели пользователей и ролей, унаследованные от библиотечных *IdentityUser* и *IdentityRole* соответственно. В скрипте настройки API можно указать параметры регистрации пользователей: длина пароля, необходимость ввода цифр, букв верхнего регистра, блокировка после неуспешных попыток ввода и др. Теперь необходимо заново провести миграцию и обновить базу данных, чтобы таблицы с полями и пользователями были занесены в БД. Чтобы создать пользователя-администратора и базовые роли по умолчанию, создадим класс *DBSeed*, в котором инициализируем необходимые данные. Методы класса *DBSeed* должны вызываться в методе *OnModelCreating()* контекста. Далее необходимо реализовать функционал JWT – JSON web token. Токены JWT

представляют собой зашифрованные строки, которые передаются пользователю и используются как идентификатор сессии. В классе `JWNMiddleware` (от `middleware` – промежуточный слой) реализуем функционал генерации и валидации токенов. Генерация происходит в методе `Login()` контроллера пользователей, валидация – при каждом обращении к какому-либо запросу. Осталось указать, какие роли могут выполнять те или иные запросы. Для этого укажем в аннотациях соответствующих методов список ролей, которые могут выполнять данные методы (или взаимодействовать со всеми методами контроллера сразу).



Для комфортного отображения, полученного API следует применить технологию Swagger [5]. Данное средство позволяет визуализировать все контроллеры и их методы, передаваемые и возвращаемые данные и сохранять токены авторизации. Сгенерированная Swagger-документация представляет собой JSON-файл, в котором отображена вся информация об эндпоинтах API. Такой файл можно передать разработчикам клиентского приложения, что благотворно повлияет на процесс разработки.

Выводы

управляющие методы контроллеров и сервисы чтения и записи. Помимо MVC данный подход API удовлетворяет принципу SOLID. Схожие методы, с учетом специфики требований, используют многие крупные предприятия, в том числе Microsoft, IBM, Intel, Cisco, Siemens, Starbucks, StackOverflow и другие.

Список литературы

1. Общие сведения о минимальных API [Электронный ресурс] / Режим доступа: <https://learn.microsoft.com/ru-ru/aspnet/core/fundamentals/minimal-apis?view=aspnetcore-6.0&viewFallbackFrom=aspnetcore-2.0>
2. Документация по Entity Framework [Электронный ресурс] / Режим доступа: <https://learn.microsoft.com/ru-ru/ef/>
3. Документация по AutoMapper [Электронный ресурс] / Режим доступа: <https://docs.automapper.org/en/stable/>
4. Симан, М. Внедрение зависимостей в .NET / М. Симан. — Санкт-Петербург: Питер, 2013. — 464 с.
5. API Development for Everyone [Электронный ресурс] / Режим доступа: <https://swagger.io>
6. Фримен, А. ASP.NET MVC 4 с примерами на C# 5.0 для профессионалов, 4-е издание / А. Фримен — М.: «Вильямс», 2013. — 688 с.
7. Мартин, Р.С. Быстрая разработка программ. Принципы, примеры, практика / Р. С. Мартин, Дж. В. Ньюкирк, Р. С. Косс. — Вильямс, 2004, ISBN 5-8459-0558-3, ISBN 0-13-597444-5
8. Мартин, Р.С. Чистый код: создание, анализ и рефакторинг. Библиотека программиста / Р.С. Мартин. — СПб.: Питер, 2013. — 464 с.: ил

Истягин А.О., Коломойцева И.А. Алгоритм построения Web-API средствами платформы .NET. В статье рассмотрена методика построения RESTFull API средствами платформы .NET и фреймворка EntityFramework. Рассмотрены паттерны MVC, принципы Dependency Injection и SOLID, приведены рекомендации по архитектуре API и именованию объектов в соответствии с руководством Clean Code.

Ключевые слова: программная инженерия, RESTFull, API, JWT, back-end, бэкенд, ASP.NET, .NET, C#, DI, Entity Framework.

Istyagin A.O., Kolomoitseva I.A. Algorithm for building a Web-API using the .NET platform. The article discusses the methodology for building a RESTFull API using the .NET platform and the EntityFramework framework. The MVC patterns, the principles of Dependency Injection and SOLID are considered, recommendations on the API architecture and object naming are given in accordance with the Clean Code guidelines.

Keywords: software engineering, RESTFull, API, JWT, back-end, бэкенд, ASP.NET, .NET, C#, DI, Entity Framework.

Разработка шаблона мобильного лендинг-приложения для Android

С.В. Щедрин^{*1}, Д.Г. Кривошея^{*2}, Д.В. Харламов^{*3},

^{*1} старший преподаватель кафедры программной инженерии, Донецкий национальный технический университет,
do010575ssv@mail.ru

^{*2} старший преподаватель кафедры тактики и общевоинских дисциплин,
Академия Министерства внутренних дел Донецкой Народной Республики имени Ф.Э. Дзержинского,
tsp-kafedra@mail.ru

^{*3} студент, Донецкий национальный технический университет,
jjharlcontact@mail.ru

Щедрин С.В., Кривошея Д.Г., Харламов Д.В. Разработка шаблона мобильного лендинг-приложения для Android. Статья рассматривает практический подход к конвейерной разработке одностраничных мобильных приложений по схожему типу. Предложена рабочая бизнес-модель для выпуска и контроля производства приложений. Разработан нативный Android-шаблон приложения для непосредственного использования и тонкой настройки под требуемую предметную область.

Ключевые слова: мобильная разработка, Android, Java, лендинг, шаблон, реклама, метрика.

Введение

Мобильные приложения сегодня являются крупнейшим сегментом программного обеспечения наряду с десктопными и веб-приложениями. Сегодня они используются не только для общения, развлечений, доступа к информации [1, 2]; но и распространения различных видов услуг [3].

Существует три основных вида заработка на мобильных приложениях:

- встроенная реклама;
- продажа приложений;
- платная подписка на услуги.

Наиболее оптимальным вариантом для небольших приложений, ориентированных на массового пользователя, является реклама, встроенная в приложения [4].

При разработке приложений такого типа на первый план выходит скорость написания и количество этих приложений.

Постановка задачи

Рассмотрим следующую бизнес-модель: заработок на рекламе, вшиваемой во множество (от сотен до нескольких тысяч) однотипных приложений, написанных по схожему шаблону. Эти приложения должны содержать какую-либо полезную для пользователя информацию, файлы, руководства и т.д., в чём будет заключаться реальная польза, побуждающая потенциального пользователя к установке подобных приложений. Интерфейс каждого нового приложения должен быть уникализирован посредством изменения цветов, стилей, введению новых элементов интерфейса. С определенной периодичностью выпуска приложений (~ 10 штук) следует менять используемый шаблон на новый во избежание нарушения правил Google Play.

Также стоит рассматривать и простые одностраничные приложения практически любой предметной области.

В приложения могут быть встроены следующие типы рекламы:

- баннеры;
- межстраничная;
- нативная;

Приложение должно содержать одну базовую страницу (активность) и одну побочную информативную активность, содержащую информацию о приложении, его версии, и политику конфиденциальности.

Существующие варианты реализации

В бесплатном доступе представлено немного вариантов.

Существует репозитории с готовыми дизайнами для некоторых практических случаев (см. рис. 1) [5]. Отличие этих шаблонов в том, что это фактически готовый дизайн приложения, в котором нет широких возможностей для кастомизации. Но по факту многие репозитории являются платным продуктом.

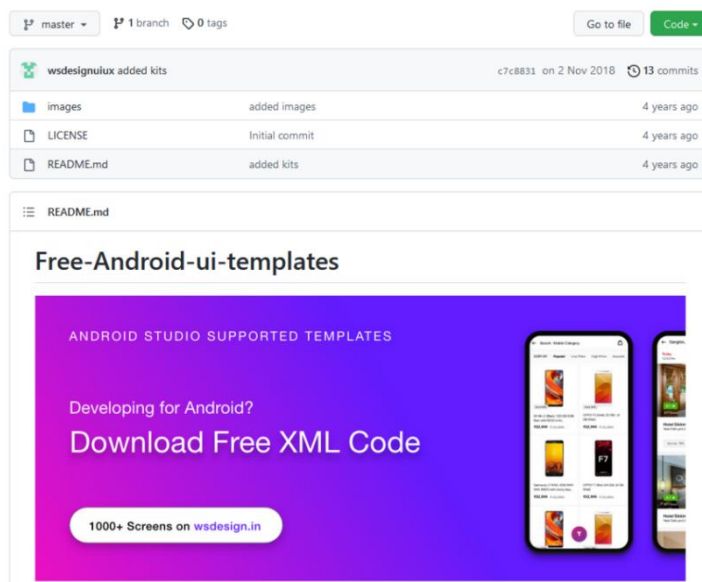


Рисунок 1 – Предлагаемые XML-шаблоны на GitHub

Также существует вариант с использованием компилирующего обработчика шаблонов FreeMaker. Для этого нужно создавать ftl-файлы, содержащие шаблонный текст генерируемого кода. Под шаблоном в этом контексте подразумевается набор метаданных, необходимый при построении диалога для ввода пользовательских параметров. На данный момент считается устаревшим компанией Google [6].

Большинство других вариантов являются платными практическими реализациями интерфейса конкретной предметной области.

Проектирование

Построим диаграмму классов системы (см. рис. 2).

Рассчитываем базовый набор экранов:

- главный экран (основное содержание и максимальная информативность);
- информационный экран (о приложении, ссылка на политику конфиденциальности, лицензия);
- политика конфиденциальности.

Так как пишем нативное Android-приложение, то все классы активностей (экранов) должны наследоваться от класса AppCompatActivity библиотек совместимости AndroidX. Сами активности располагаются в пакете проекта и переопределяют методы наследуемого класса в соответствии с нуждами прикладного программиста.

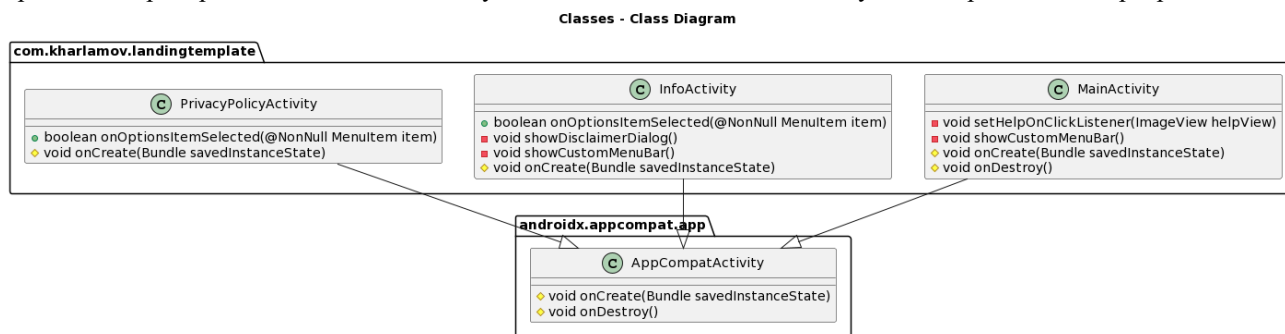


Рисунок 2 – Диаграмма классов

Компоненты проекта включают различные ресурсы: строковые (локализации), изображения (значки), XML-макеты интерфейса, шрифты, константы для дизайна, стили оформления (см. рис. 3). На диаграмме представлены также набор инструментов разработчика Android SDK и пакет классов активностей (экранов), изображенных на диаграмме классов.

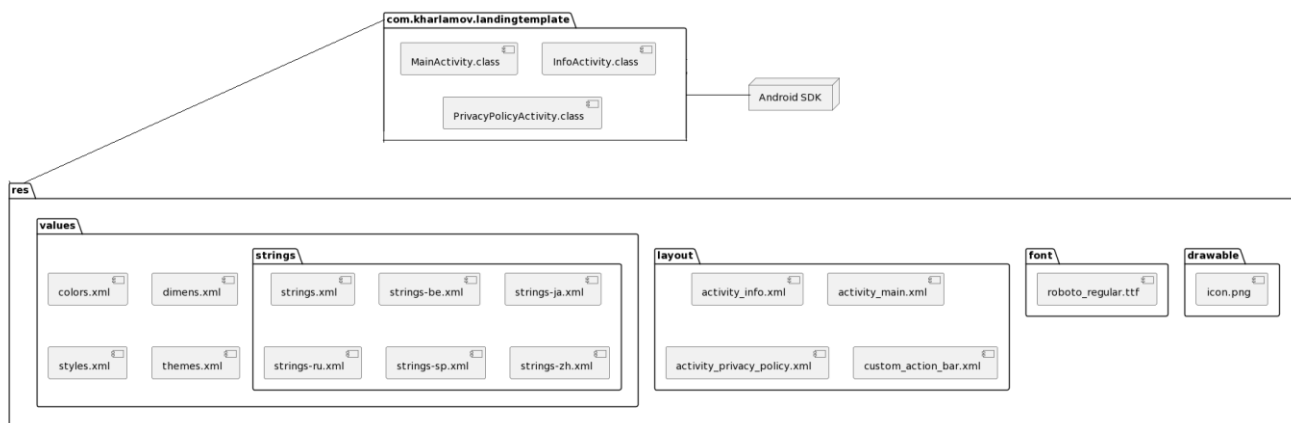


Рисунок 3 – Диаграмма компонентов

Разработка

Для разработки шаблона воспользуемся язык программирования Java и официальной средой разработки Android Studio.

Создадим три активности, которые ассоциируются с экранами приложения. Классы активностей отвечают за логику обработки информации в приложении. С ними связаны XML-макеты графического интерфейса.

Реализуем требуемые методы и значок перехода к информационной активности с главного экрана (см. рис. 4).

```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        showCustomMenuBar();

        ImageView helpView = findViewById(R.id.helpView);
        setHelpOnClickListener(helpView);
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
    }

    private void setHelpOnClickListener(ImageView helpView) {
        helpView.setOnClickListener(view ->
            startActivity(new Intent(packageContext, this, InfoActivity.class)));
    }

    private void showCustomMenuBar() {
        this.getSupportActionBar().setDisplayOptions(ActionBar.DISPLAY_SHOW_CUSTOM);
        getSupportActionBar().setDisplayShowCustomEnabled(true);
        getSupportActionBar().setCustomView(R.layout.custom_action_bar);
        getSupportActionBar().setElevation(0);
    }
}
```

Рисунок 4 – Главная активность

В информационной активности выведем информацию о приложении, кнопки для дисклеймера и политики конфиденциальности, информацию о текущей версии (см. рис. 5). В дисклеймере может указываться какой-либо отказ от ответственности, но в примере находится информация о лицензировании.

```

public class InfoActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_info);

        AppCompatActivity disclaimerButton = findViewById(R.id.disclaimerButton);
        disclaimerButton.setOnClickListener(view1 -> showDisclaimerDialog());
        AppCompatActivity privPolButton = findViewById(R.id.privPolButton);
        privPolButton.setOnClickListener(
            view1 -> startActivity(new Intent( packageContext this, PrivacyPolicyActivity.class))
        );

        TextView appDescrView = findViewById(R.id.appDescrView);
        appDescrView.setMovementMethod(LinkMovementMethod.getInstance());

        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    }

    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {

        if (item.getItemId() == android.R.id.home) {
            finish();
            return true;
        }
        return super.onOptionsItemSelected(item);
    }

    private void showDisclaimerDialog() {

        AlertDialog.Builder builder = new AlertDialog.Builder( context this);

        builder.setMessage("BSU License")
            .setTitle("Disclaimer")
            .setPositiveButton( text "OK", (dialog, id) -> {
                dialog.cancel();
            });

        Dialog dialog = builder.create();
        dialog.show();
        ((TextView) dialog.findViewById(android.R.id.message)).setMovementMethod(LinkMovementMethod.getInstance());
    }
}

```

Рисунок 5 – Информационная активность

Активность политики конфиденциальности демонстрирует её полное содержание со всеми ссылками на связанные политики используемых сервисов. Также включается код для навигации пользователя по приложению.

```

public class PrivacyPolicyActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_privacy_policy);

        TextView info = findViewById(R.id.infoText);
        info.setMovementMethod(LinkMovementMethod.getInstance());

        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.Q) {
                info.setJustificationMode(LineBreaker.JUSTIFICATION_MODE_INTER_WORD);
            }
        }

        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    }

    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {

        if (item.getItemId() == android.R.id.home) {
            finish();
            return true;
        }
        return super.onOptionsItemSelected(item);
    }
}

```

Рисунок 6 – Активность политики конфиденциальности

Исходной локализацией приложения считаем английский. Займемся локализацией шаблона для следующих языков:

- русский;

- Файлы ресурсов представим на нескольких языках (см. рис. 4).

Рисунок 7 – Строковые ресурсы приложения на нескольких языках

AndroidManifest.xml | activity_policy.xml | values strings.xml | betatranslator.xml | translations Editor | PrivacyPolicyActivity.java

Edit translations for all locales in the translations edition. Open editor Hide notification

resources

```
1 <string name="app_name">Squid Game для Minecraft</string>
2
3 <string name="install">Установить Mod</string>
4
5 <string name="about">О приложении</string>
6 <string name="share">Поделиться</string>
7
8
9
10 <string name="share_text">Крутое приложение с модами для Майнкрафт: "</string>
11 <string name="app_desc">" Это приложение создано для установки карт, текстур, аддонов, скинов в игру.\n
12 <p>Применение модов в игре:</p>\n\n
13 <p>648220: Запустите Minecraft и перейдите в редактирование мира.</p>\n\n
14 <p>648220: Выберите Наборы текстур.</p>\n\n
15 <p>648220: Найдите текстур пак и активируйте его.</p>\n\n
16 <p>648220: Выберите Наборы параметров.</p>\n\n
17 <p>648220: Найдите дополнение и активируйте его.</p>\n\n
18 <p>648220: Мод установлен, приятной игры!</p>\n"</string>
19
20 <string name="priv_policy">Политика конфиденциальности</string>
21
22 <string name="privacy_policy_text">
23 "«Политика конфиденциальности»</p>\n\n
24 Компания Minecraft Apps создала приложение Squid Game for Minecraft как бесплатное приложение. Эта услуга предоставляется Minecraft Apps Соврпну бесплатно и предназначена
25 Эта страница используется для информирования посетителей о нашей политике в отношении сбора, использования и раскрытия личной информации, если кто-либо решил использовать наше при
26 приложение, используя наше приложение, вы соглашаетесь на сбор и использование информации в отношении этой политики. Личная информация, которую мы собираем, используется для
27 Термины, используемые в настоящей Политике конфиденциальности, имеют то же значение, что и в наших Полных условиях, которые доступны в Squid Game for Minecraft, если иное не
28 «Сбор и использование информации»</p>\n\n
29 Для удобства использования нашего приложения мы можем потребовать от вас предоставить нам определенную личную информацию. Информация, которую мы запрашиваем, будет хранит
30 Приложение использует сторонние сервисы, которые могут собирать информацию, используемую для вашей идентификации.
31 Ссылка на политику конфиденциальности сторонних поставщиков услуг, используемых приложением\n
32 648220: <a href="https://www.google.com/policies/privacy/">Google Play Services</a>\n
33 648220: <a href="https://support.google.com/admob/answer/6128543?hl=en">AdMob</a>\n
34 648220: <a href="https://firebase.google.com/policies/analytics/">Google Analytics for Firebase</a>\n
35 648220: <a href="https://firebase.google.com/support/privacy/">Firebase Crashlytics</a>\n
36 648220: <a href="https://www.facebook.com/about/privacy/update/printable">Facebook</a>\n
37
38 <p>«Данные»</p>\n\n
39 «Личные данные»</p>\n\n
40 Мы собираем данные, как всякий раз, когда вы используете наше приложение, в случае ошибки в приложении мы собираем данные и информации (через продукты сторонних производи
41 Cookies</p>\n\n
```

Рисунок 8 – Пример заполненного файла текстовых ресурсов шаблона для реального приложения

The image displays a sequence of five mobile app screens, likely representing the development or testing of a privacy policy application. Each screen shows a red status bar at the top with the time (19:00, 21:00, 21:00, 21:00, 21:00) and battery level (40%, 40%, 56%, 56%, 56%). The app's title bar is red with the text 'MyAppName' and a back arrow. The main content area is white.

- Screen 1:** Features a large, colorful icon of a laptop with a code editor, a document, and a server. Below the icon, the text reads: 'Этот проект является шаблоном односторонних приложений для любой предметной области.' (This project is a template for single-sided applications for any subject area). Below this, under the heading 'Применение:' (Application:), is a list: '• Коммерческое.', '• Образовательное.', '• Любое.'
- Screen 2:** Features a large, colorful icon of a laptop with a code editor, a document, and a server. Below the icon, the text reads: 'Этот проект является шаблоном односторонних приложений для любой предметной области.' (This project is a template for single-sided applications for any subject area). Below this, under the heading 'Применение:' (Application:), is a list: '• Коммерческое.', '• Образовательное.', '• Любое.'
- Screen 3:** Features a large, colorful icon of a laptop with a code editor, a document, and a server. Below the icon, the text reads: 'Этот проект является шаблоном односторонних приложений для любой предметной области.' (This project is a template for single-sided applications for any subject area). Below this, under the heading 'Применение:' (Application:), is a list: '• Коммерческое.', '• Образовательное.', '• Любое.'
- Screen 4:** Features a large, colorful icon of a laptop with a code editor, a document, and a server. Below the icon, the text reads: 'Этот проект является шаблоном односторонних приложений для любой предметной области.' (This project is a template for single-sided applications for any subject area). Below this, under the heading 'Применение:' (Application:), is a list: '• Коммерческое.', '• Образовательное.', '• Любое.'
- Screen 5:** Features a large, colorful icon of a laptop with a code editor, a document, and a server. Below the icon, the text reads: 'Этот проект является шаблоном односторонних приложений для любой предметной области.' (This project is a template for single-sided applications for any subject area). Below this, under the heading 'Применение:' (Application:), is a list: '• Коммерческое.', '• Образовательное.', '• Любое.'

Рисунок 9 – Результат, пустой шаблон приложения

Политика конфиденциальности

Политика конфиденциальности (англ. privacy policy) является документом, который регулирует внутренние процессы компании по обработке персональных данных. По закону должна быть свободно доступна пользователю для ознакомления. В шаблоне политика конфиденциальности доступна из информационного экрана. В текущем случае политика конфиденциальности требуется для оповещения пользователя о том, что различные метрики применяются в приложении для сбора и обработки информации о пользовании им приложением (как Google – в случае подключения рекламы, так и разработчик приложения – в случае использования сторонних метрик).

В соответствии с разделом 4.3 лицензионного соглашения Android SDK [8] и разделом 4.8 соглашения о распространении для разработчиков Google Play [9] личные и конфиденциальные данные пользователя включают, помимо прочего:

- имена, пароли, информацию для входа, личную информацию;
- всю личную информацию, предоставляемую пользователем на хранение;
- информацию аккаунта Google, доступ к которому осуществляется только для целей, регламентированных пользователем;

Политика конфиденциальности

Документ вступает в юридическую силу с момента его опубликования/размещения в мобильном приложении, если только иной срок не указан непосредственно в тексте самой Политики конфиденциальности. Если пользователь не согласен с некоторыми положениями Политики конфиденциальности, он должен немедленно прекратить пользоваться мобильным приложением.

Сбор статистики и аналитика

Определена политика конфиденциальности, поэтому можно организовать сбор и обработку выше указанных данные пользователя.

Так как бизнес-модель предполагает выпуск большого количества приложений, то появляется потребность распознавать наиболее популярные в плане использования и эффективности рекламы приложения.

Для оценки посещаемости и анализа действий пользователя в отношении приложения включим в него средства бесплатного сервиса «Яндекс.Метрика». Наглядная сводка информации позволит оценить разработчику, веб-мастеру или заказчику текущую ситуацию использования приложения (см. рис. 10).

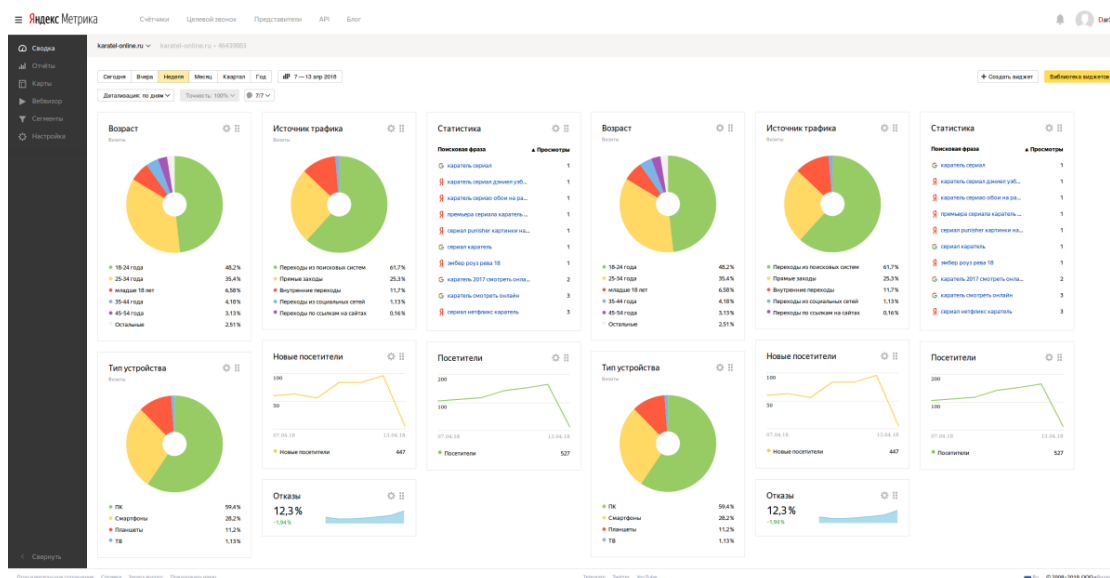


Рисунок 10 – «Яндекс.Метрика»

Для использования API метрики в приложении программисту потребуется указать ключ API, предоставляемый сервисом.

Варианты использования

Шаблон является универсальным решением и подходит под любую простую задачу: лендинг, информационная страница, политика конфиденциальности.

На момент написания статьи в целях развития выбранной бизнес-модели на основе шаблона были созданы различные сборники модификаций для детской игры (см. рис. 11). Приложения позволяют загрузить и установить

требуемые модификации, ознакомившись перед этим со статистикой по каждому моду (размер файла, количество загрузок, количество оценок «Мне нравится»).

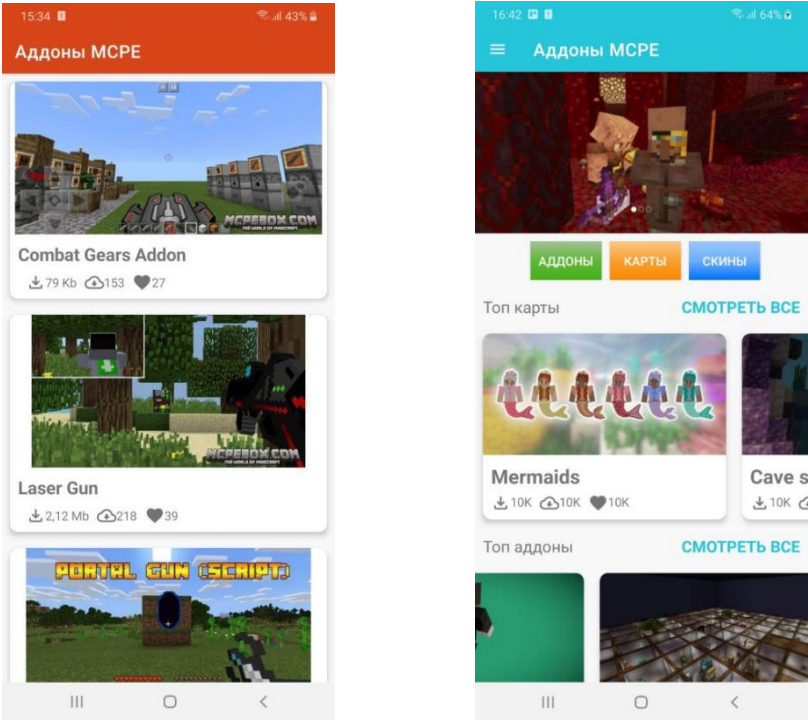


Рисунок 11 – Модификации для детской игры

Во втором приложении (см. рис. 12) были использованы новые элементы интерфейса: карусель и встроенный непосредственно в интерфейс активности HTML-проигрыватель видеохостинга YouTube. Приложение получило обновленные цвета и стили.

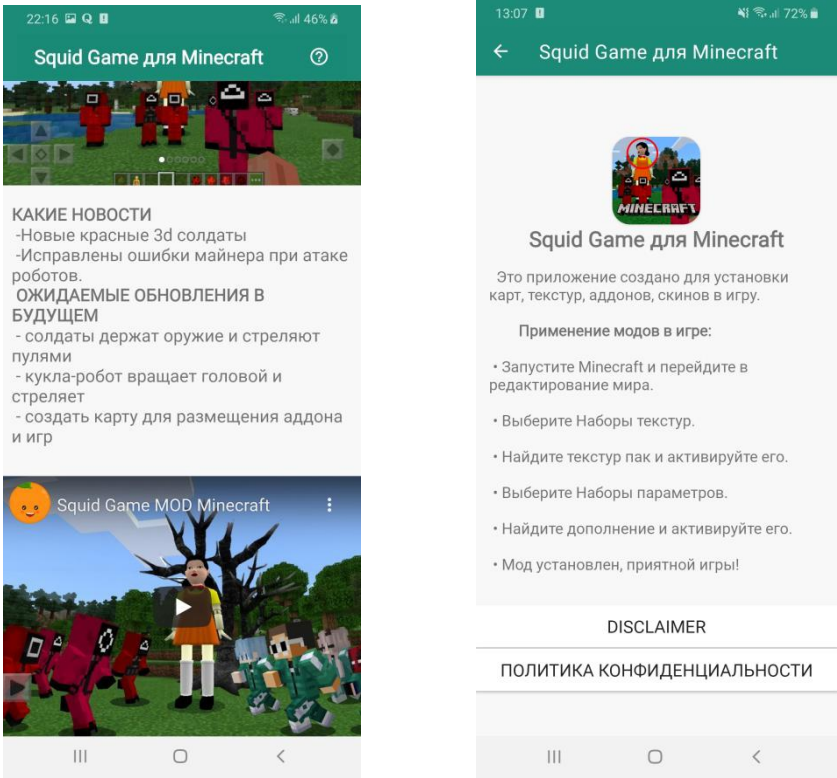


Рисунок 12 – Вариант отредактированного шаблона со встроенным проигрывателем YouTube

Выводы

Получили шаблон приложения в виде проекта для официальной среды разработки Android – Android Studio. Результаты проекта доступны на хостинге GitHub [10]. Можно просмотреть или загрузить исходный код, присоединиться к разработке. Шаблон бесплатен, готов к изменению и использованию «из коробки».

Предложена бизнес-модель для заработка на конвейерной разработке простых приложений

Проект имеет ряд преимуществ перед имеющимися аналогами:

- некоммерческий продукт, распространяемый по лицензии BSD;
- использование актуальных технологий;
- универсальность, применимость к любой предметной области;
- высокий потенциал для повторного использования и тонкой настройки;
- локализация на наиболее распространенные языки мира.

Литература

1. Mobile App Download and Usage Statistics, Facts, Types [Электронный ресурс] / . – Режим доступа: <https://market.us/statistics/internet/mobile-app-download-and-usage/>. - Загл. с экрана.
2. Global: mobile app downloads by segment 2017-2025 | Statista [Электронный ресурс] / . – Режим доступа: <https://www.statista.com/forecasts/1262881/mobile-app-download-worldwide-by-segment>. - Загл. с экрана.
3. Wei, J. Mobile Electronic Commerce / J. Wei. – : CRC Press, 2014. – 315 с.
4. Appel, G. On the monetization of mobile apps / G. Appel, B. Libai, E. Muller // International Journal of Research in Marketing. – 2020. – Т. 37, № 1. – С. 93-107
5. GitHub - wsdesignuiux/Android-ui-templates: Download free android app templates free and paid. [Электронный ресурс] / . – Режим доступа: <https://github.com/wsdesignuiux/Android-ui-templates>. - Загл. с экрана.
6. Магическая шаблонизация для Android-проектов / Хабр [Электронный ресурс] / . – Режим доступа: <https://habr.com/ru/company/hh/blog/529948/>. - Загл. с экрана.
7. Yandex.Metrica — free All-Round Web Analytics: session replay, heat maps and form analytics combined with advanced reporting tools [Электронный ресурс] / . – Режим доступа: <https://metrica.yandex.com>. - Загл. с экрана.
8. Terms and conditions | Android Developers [Электронный ресурс] / . – Режим доступа: <https://developer.android.com/studio/terms>
9. Google Play [Электронный ресурс] / . – Режим доступа: <https://play.google.com/about/developer-distribution-agreement.html>. - Загл. с экрана.
10. kharlamovd/android-landing-template: This is a simple free-to-use android landing (or just single-page) template to build any type of useful applications. [Электронный ресурс] / . – Режим доступа: <https://github.com/kharlamovd/android-landing-template>. - Загл. с экрана.

Щедрин С.В., Кривошея Д.Г., Харламов Д.В. Разработка шаблона мобильного лендинг-приложения для Android . Статья рассматривает практический подход к конвейерной разработке одностраничных мобильных приложений по схожему типу. Предложена рабочая бизнес-модель для выпуска и контроля производства приложений. Разработан нативный Android-шаблон приложения для непосредственного использования и тонкой настройки под требуемую предметную область.

Ключевые слова: мобильная разработка, Android, Java, лендинг, шаблон, реклама, метрика.

Shchedrin S.V., Krivosheya D.G., Kharlamov D.V. Development of a template for an Android landing application. The article considers a practical approach to the pipelined development of single-page mobile applications of a similar type. A working business model for the release and production control of applications is proposed. A native Android application template has been developed for direct use and fine-tuning for the required subject area.

Keywords: mobile development, Android, Java, landing page, template, advertising, metrics.

Модель выполнения кода в среде CLR

В.В. Данилевич^{*1}, О.В. Рычка^{*2}

^{*1} студент, Донецкий национальный технический университет,
v.danilevitch2013@yandex.ru

^{*2} к.т.н, доцент, Донецкий национальный технический университет,
olga_rychka@mail.ru, OrcID: 0000-0001-6822-7306

Данилевич В.В., Рычка О.В. Модель выполнения кода в CLR. В данной статье будет рассмотрена модель выполнения кода в среде Common Language Runtime. Изложен процесс компиляции IL-кода в машинные команды, декомпиляции IL-кода, а также будут описаны средства декомпиляции IL-кода, предоставляемые компанией Microsoft.

Ключевые слова: среда выполнения, CLR, IL-код, управляемый модуль, метаданные, JIT компиляция, ILDasm.

Введение

Важной частью таких платформ как .NET и .NET Framework является общезыковая среда выполнения (Common Language Runtime) [1]. Цель этой статьи — дать краткий обзор среды CLR и определить термины, с которыми разработчик может столкнуться при работе с ней. Также в этой главе изложен процесс построения приложения или набора распространяемых компонентов (файлов), которые содержат типы (классы, структуры и т. п.), и затем объяснено, как выполняется приложение.

Компиляция исходного кода в управляемые модули

Общезыковая среда выполнения (Common Language Runtime, CLR) — это среда выполнения, которая подходит для разных языков программирования [2].

Среда предоставляет следующие возможности: управление памятью; загрузка сборок; безопасность; обработка исключений; синхронизация.

Данные возможности доступны всем языкам, которые используют данную среду. Поскольку CLR не знает на каком языке написан исходный код, то разработчик может выбрать любой язык программирования для решения узконаправленных задач. Однако компилятор выбранного языка обязан поддерживать общезыковую среду. Рисунок 1 иллюстрирует процесс компиляции файлов с исходным кодом. Следуя схеме на рисунке исходный код программы анализируется компилятором, после чего формируется управляемый модуль — стандартный переносимый исполняемый (PE) файл 32-разрядной (PE32) или 64-разрядной Windows (PE32+).



Рисунок 1 – Компиляция исходного кода в управляемые модули

Управляемый модуль состоит из следующих частей: заголовок PE32 или PE32+, он обозначает тип файла: GUI, CUI или DLL, а также содержит временную метку; заголовок CLR, включает в себя версию CLR, флаги, метку метаданных MethodDef точки входа в управляемый модуль, а размер и расположение метаданных модуля; метаданные — это таблицы, описывающие типы данных и их члены; IL-код — код, создаваемый компилятором.

Объединение управляемых модулей в сборку

Общезыковая среда работает только со сборками. Сборка (assembly) — это абстрактное понятие, понять смысл которого на первых порах бывает нелегко [2]. Функции сборки: объединяет один или несколько управляемых модулей или файлов ресурсов; элементарная единица многократного использования, безопасности и управления версиями. Рисунок 2 представляет структуру сборки. На этом рисунке изображены управляемые модули и файлы ресурсов, с которыми работает программа. Данной программой создается PE32+ файл, включающий в себя и блок данных, называемый манифестом (manifest). Манифест представляет собой обычный набор таблиц метаданных, в которых заключены файлы входящие в сборку общедоступные экспортируемые типы [1]. Сборка является управляемым модулем в тех случаях, когда сборка состоит только из одного файла.



Рисунок 2 - Объединение управляемых модулей в сборку

Сборка решает проблему разделения логического и физического представления компонента, который поддерживает много кратное использование, безопасность и управление версиями. Программист волен сам определиться с решением о разбиении программного кода и ресурсов.

Модули сборки также содержат сведения о других сборках, на которые они ссылаются (в том числе номера их версий). Эти данные делают сборку самоописываемой (self-describing). Другими словами, среда CLR может определить все прямые зависимости данной сборки, необходимые для ее выполнения.

Исполнение кода сборки

Управляемые модули содержат метаданные и IL-код. IL-код — это независящий от процессора машинный язык. Он позволяет работать с объектами и имеет команды для создания и инициализации объектов, вызова виртуальных методов и непосредственного манипулирования элементами массивов. IL можно рассматривать как объектно-ориентированный машинный язык. IL является стековым языком. Инструкции IL также являются не типизованными. IL-код обеспечивает безопасность приложения и его устойчивость перед ошибками. В процессе компиляции IL в машинные инструкции CLR выполняется процедура, называемая верификацией — анализ высокоуровневого кода IL и проверка безопасности всех операций.

Для преобразования IL-кода в машинные команды используется JIT-компилятор (Just-In-Time).

При первом вызове метода, среда CLR находит все типы данных, на которые ссылается данный метод и выделяет внутренние структуры данных, для управления доступом к типам. В каждой записи есть адрес реализации метода. При инициализации структуры общезыковая среда выполнения заносит в каждую запись адрес недокументированной функции, которая содержится в самой среде CLR. Пример выполнения первого обращения к методу представлен на рисунке 3.

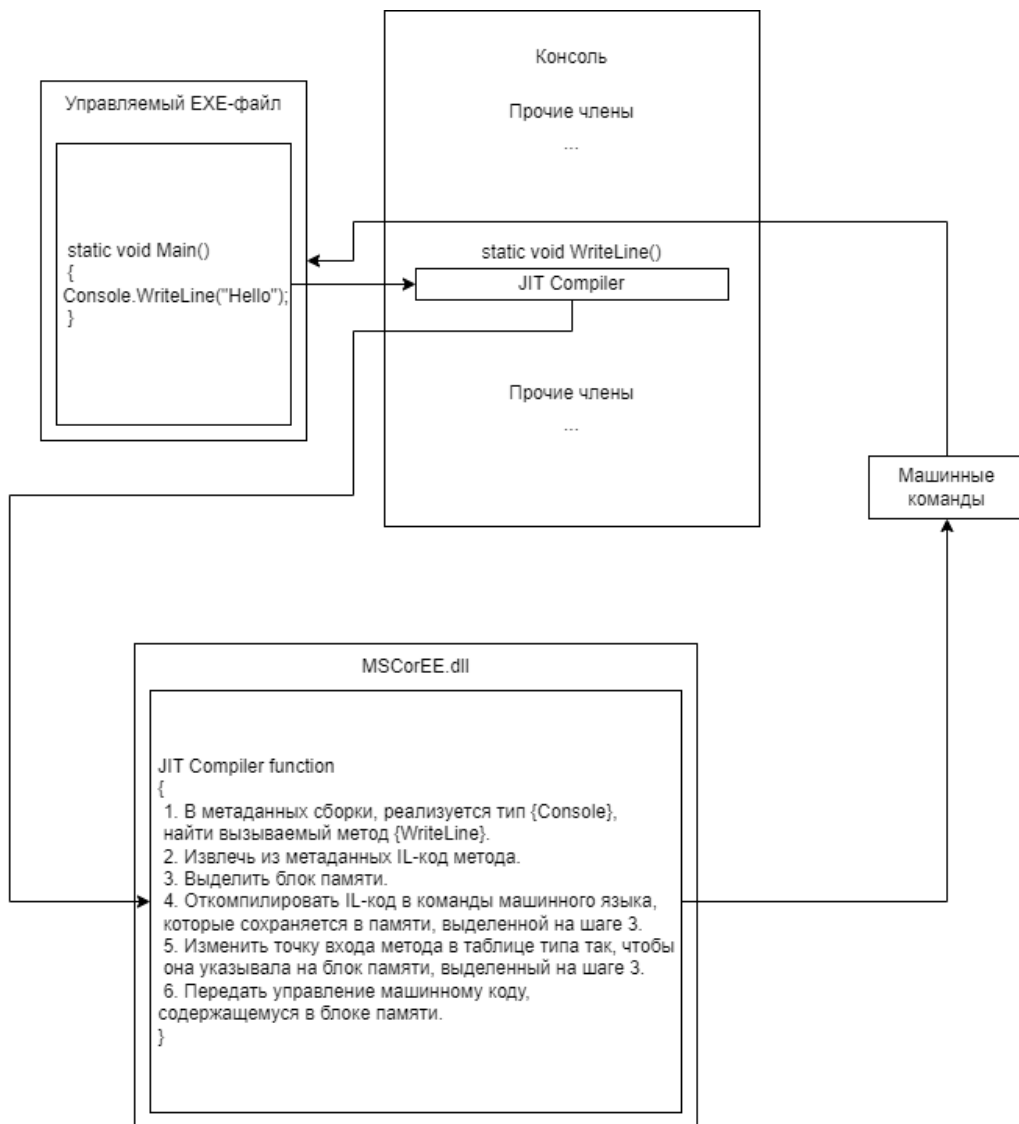


Рисунок 3 – Первый вызов метода

При повторном вызове метода, так как код метода был уже скомпилирован, обращение к блоку памяти будет производиться напрямую. На рисунке 4 представлено повторное обращение к методу.

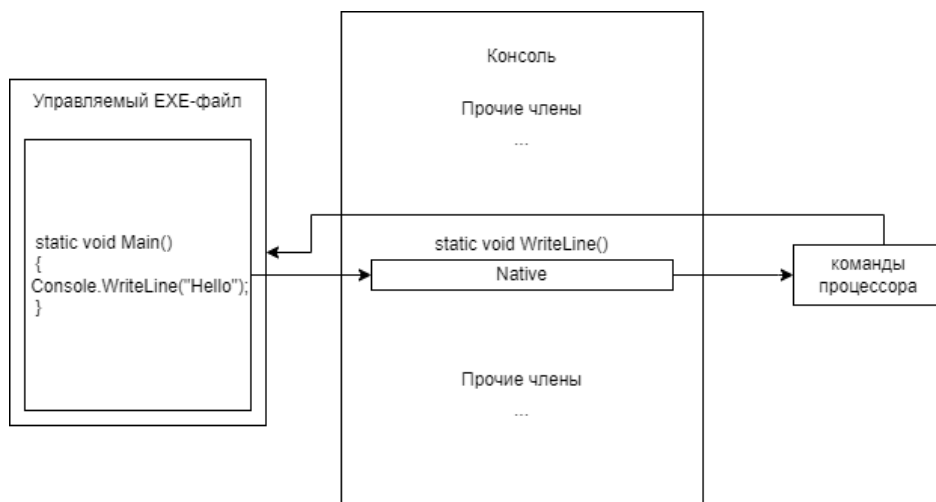


Рисунок 4 – Повторный вызов метода

Таким образом, снижение производительности снижается только при первом вызове метода, при последующие вызовы выполняются без задержек. Однако, машинные команды хранятся в динамической памяти, поэтому при закрытии приложения скомпилированный код уничтожается. JIT-компилятору придётся заново компилировать IL-код при повторном запуске приложения.

Однако если скорости JIT-компиляции Вам покажется мало, Вы можете воспользоваться генератором образов в машинном коде (Ngen.exe)[3]. Данная программа компилирует IL-код сборки в машинные команды и сохраняет его в файл. При загрузке сборки среда автоматически загружает заранее откомпилированную версию сборки.

Дизассемблирование IL-кода

Для дизассемблирования IL-кода используется программа ILDasm.exe, разработанная Microsoft [4]. При установке .Net Framework 4.X.X она устанавливается вместе с пакетом по пути C:\Program Files (x86)\Microsoft SDKs\Windows\v10.0A\bin\NETFX 4.X.X Tools. Также можно установить утилиту ILDasm для .Net через диспетчер пакетов Nuget, для этого в строку поиска нужно ввести nuget-пакет Microsoft.NETCore.ILDasm рисунок 5.

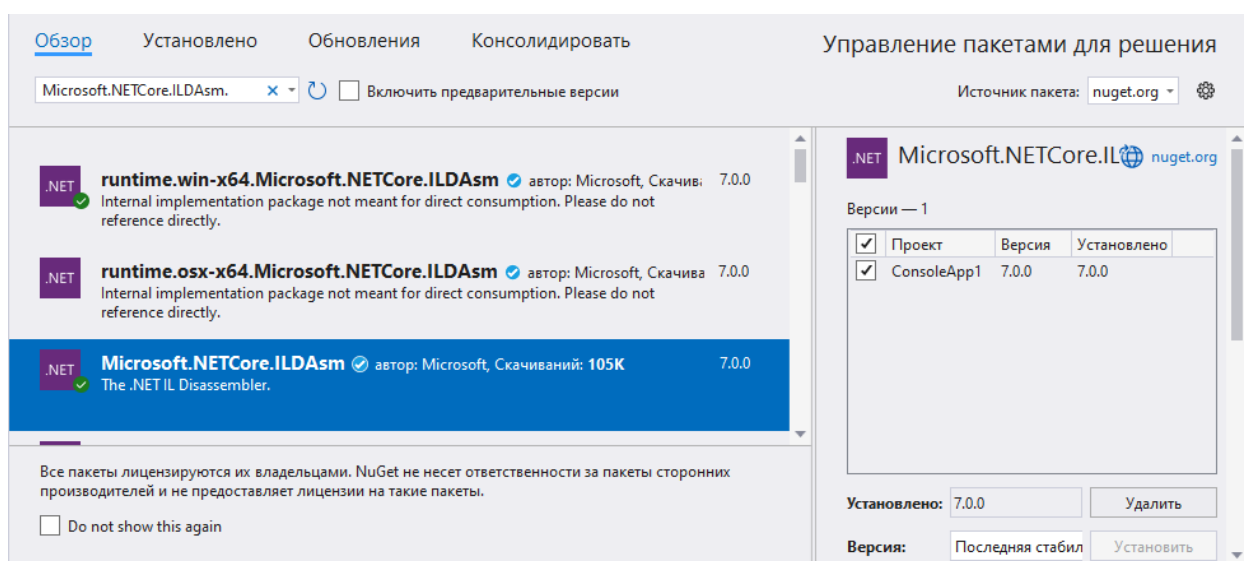


Рисунок 5 – Установка утилиты ILDasm

На рисунке 7 представлены результаты декодирования IL-кода, который приведен на рисунке 6, с помощью графического приложения ILDasm.exe.

```

Using static System.Net.Mime.MediaTypeNames;
namespace ClassLibrary1
{
    public class Class1
    {
        public Class1() {}
        public static Boolean operator ==(Class1 t1, Class1 t2)
        {
            return true;
        }
        public static Boolean operator !=(Class1 t1, Class1 t2)
        {
            return false;
        }
        public static Class1 operator +(Class1 t1, Class1 t2) { return null; }
        public String Anroperty
        {
            get { return null; }
            set {}
        }
        public String this[Int32 x]
        {
            get { return null; }
            set {}
        }
        public event EventHandler AnEvent;
    }
}

```

Рисунок 6 – Пример кода на C#

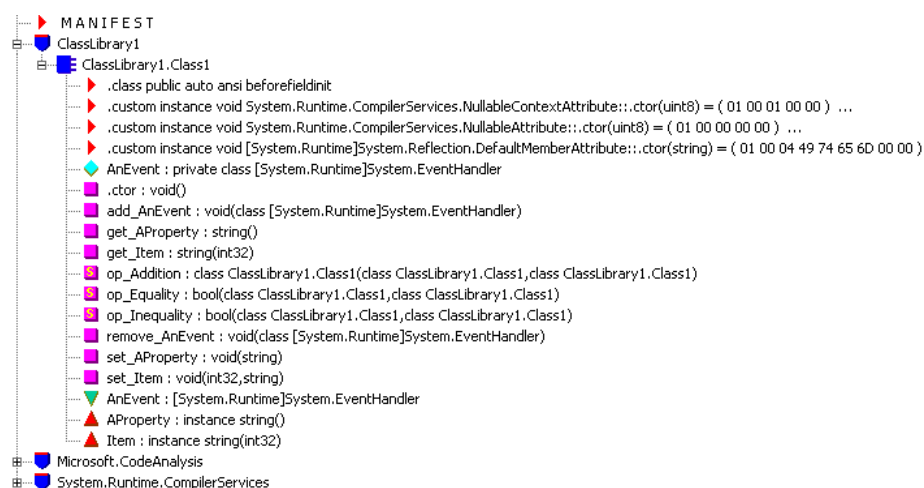


Рисунок 7 – Использование графического приложения ILDasm.exe

Для того чтобы выполнить декодирования IL-кода через утилиту ILDasm требуется в командной строке перейти в папку с файлом сборки и передать его утилите с помощью команды: `ildasm (путь к сборке) /out=(имя файла с il-кодом)` (см. рис. 8). Команда `out` записывает в указанный пользователем файл результаты декодирования.

```
Microsoft Windows [Version 10.0.19044.1415]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\Vlad>cd C:\Users\Vlad\Desktop\ALLabs\ПБид\ConsoleApp1\ClassLibrary1\bin\Release\net6.0

C:\Users\Vlad\Desktop\ALLabs\ПБид\ConsoleApp1\ClassLibrary1\bin\Release\net6.0>C:\Users\Vlad\Desktop\ALLabs\ПБид\ConsoleApp1\ClassLibrary1\bin\Release\net6.0\runtimes\win-x64\native\ildasm ClassLibrary1.dll /out=ClassLibrary1.il

C:\Users\Vlad\Desktop\ALLabs\ПБид\ConsoleApp1\ClassLibrary1\bin\Release\net6.0>
```

Рисунок 8 – Декомпиляция код с помощью утилиты ILDasm

На рисунке 9 представлен результат выполнения декомпиляции.

```
.method public hidebysig specialname rtspecialname
    instance void .ctor() cil managed
{
    // Code size          7 (0x7)
    .maxstack 8
    IL_0000: ldarg.0
    IL_0001: call        instance void [System.Runtime]System.Object::.ctor()
    IL_0006: ret
} // end of method Class1::.ctor

.method public hidebysig specialname static
    bool op_Equality(class ClassLibrary1.Class1 t1,
                    class ClassLibrary1.Class1 t2) cil managed
{
    // Code size          2 (0x2)
    .maxstack 8
    IL_0000: ldc.i4.1
    IL_0001: ret
} // end of method Class1::op_Equality
```

Рисунок 9 – Результаты использования утилиты ILDasm

В таблице 1 продемонстрированы соответствия между конструкциями языка программирования и эквивалентными полями/методами CLR.

Таблица 1 Поля и методы типа ClassLibrary1

Член типа	Разновидность	Эквивалентная конструкция языка программирования
AnEvent	Поле	Событие; имя поля – AnEvent, тип – System.EventHandler
.ctor	Метод	Конструктор
Finalize	Метод	Финализатор
add_AnEvent	Метод	Метод добавления обработчика события
get_AProperty	Метод	Get-метод доступа свойства
get_Item	Метод	Get-метод индексатора
op_Addition	Метод	Оператор +
op_Equality	Метод	Оператор ==
op_Inequality	Метод	Оператор !=
remove_AnEvent	Метод	Метод удаления обработчика события
set_AProperty	Метод	Set-метод доступа свойства
set_Item	Метод	Set-метод индексатора

Для просмотра IL кода в графическом приложении щёлкните два раза по выбранному методу. На рисунке 10 изображен участок с IL кодом конструктора класса ClassLibrary1.Class1.

```

ClassLibrary1.Class1::ctor: void()
Найти  Найти далее
.method public hidebysig specialname rtspecialname instance void .ctor() cil managed
{
    // Размер кода: 9 (0x9)
    .maxstack 8
    IL_0000: ldarg.0
    IL_0001: call instance void [System.Runtime]System.Object::.ctor()
    IL_0006: nop
    IL_0007: nop
    IL_0008: ret
} // end of method Class1::.ctor

```

Рисунок 10 – IL-код конструктора ClassLibrary1.Class1

.method public hidebysig specialname rtspecialname instance void .ctor() cil managed является элементом метаданных Method Definition. Такие служебные слова как public и instance определяют флаги элемента Method Definition. Public – показывает, что конструктор доступен всем членам, для которых видим родительский класс. Instance – указывает на то, что метод связан с объектом, а не с классом.

Служебное слово hidebysig указывает на сокрытие базового класса с такой же сигатурой.

Specialname указывает, что имя данного элемента может иметь особое значение для инструментов, отличных от CLI.

Rtspecialname указывает, что имя данного элемента имеет особое значение для командной строки. Служебное слово void определяет, что данный метод ничего не возвращает. Служебные слова cil и managed определяют флаги элемента Method Definition и означают, что тело метода представлено IL-кодом.

.maxstack 8 – директива обозначающая максимальное число элементов метода, которое может находиться в стеке во время вычислений.

Метка IL_0000 используется для определения смещений внутри IL-кода при компиляции программы.

Ldarg.0 – флаг указывает на загрузку в стек вычислений неявного параметра this.

call instance void [System.Runtime]System.Object::.ctor() – это вызов родительского конструктора, в данном случае происходит вызов конструктора экземпляра класса Object, так как в среде CLR все классы наследуются от него.

Ret – является пустой командой необходимой для поддержания функции «Изменить и продолжить» в среде Visual Studio во время процесса отладки. Также они отвечают за точки останова на управляющих командах (for, while-do, if, else, try, catch, finally). Во время оптимизации компилятор C# удаляет данные команды, оптимизируя поток управления программой рисунок 11.

```
.method public hidebysig specialname rtspecialname
    instance void .ctor() cil managed
{
    // Code size          7 (0x7)
    .maxstack 8
    IL_0000: ldarg.0
    IL_0001: call        instance void [System.Runtime]System.Object::.ctor()
    IL_0006: ret
} // end of method Class1::.ctor
```

Рисунок 11 – Пример оптимизации кода

Ret – выполняет передачу управления элементу, вызвавшему данный метод. При чтении процессором данной инструкции, он передает управление следующей функции, получая из стека её адрес.

Заключение

Среда выполнения CLR предоставляет удобные инструменты программисту для разработки сложных приложений. Управления памятью, загрузка сборок, безопасность, обработка исключений, синхронизация, данные возможности доступны на всех языках программирования, поддерживающие данную среду выполнения. Благодаря Just-In-Time компилятору и оптимизации кода предоставляемых компанией Microsoft библиотек, производительность выполнения приложения не уступает аналогичному, написанному на неуправляемом коде.

Литература

1. C# 7.0 Справочник. Полное описание языка / Албахари Б., Албахари Дж. – СПб.:ООО «Альфа-книга», 2018. – 1024 с.
2. CLR via C# 4-е изд. / Рихтер Дж. – СПб.: Питер, 2013. – 896 с.
3. Ngen.exe (генератор образов в машинном коде) [Электронный ресурс]. – Режим доступа: <https://learn.microsoft.com/ru-ru/dotnet/framework/tools/ngen-exe-native-image-generator>
4. ildasm.exe (дизассемблер IL) [Электронный ресурс]. – Режим доступа: <https://learn.microsoft.com/ru-ru/dotnet/framework/tools/ildasm-exe-il-disassembler>

Данилевич В.В., Рычка О.В. Модель выполнения кода в CLR. В данной статье будет рассмотрена модель выполнения кода в среде Common Language Runtime. Изложен процесс компиляции IL-кода в машинные команды, декомпиляции IL-кода, а также будут описаны средства декомпиляции IL-кода, предоставляемые компанией Microsoft.

Ключевые слова: среда выполнения, CLR, IL-код, управляемый модуль, метаданные, JIT компиляция, ILDasm.

Danilevich Vladislav, Rychka Olga Code execution model in CLR. This article will examine the code execution model in the Common Language Runtime environment. The process of compiling IL-code into machine commands, decompiling IL-code is described, and the means of decompiling IL-code provided by Microsoft will also be described.

Keywords: runtime environment, CLR, IL code, managed module, metadata, JIT compilation, ILDasm.

Выбор архитектуры программного обеспечения для создания образовательного портала

А. Д. Суров^{*1}, Д. Г. Раннев^{*2}, А. В. Чернышова^{*3}

^{*1} студент, Донецкий национальный технический университет,
john_doe_844@mail.ru

^{*2} студент, Донецкий национальный технический университет,
ivannnn.nnnavi@mail.ru

^{*3} старший преподаватель, Донецкий национальный технический университет, chernyshova.alla@rambler.ru,
OrcID: 0000-0003-2546-2167, SPIN-код: 3318-2066

Суров А. Д., Раннев Д. Г., Чернышова А. В. Выбор архитектуры программного обеспечения для создания образовательного портала. В статье представлен краткий обзор существующих архитектурных подходов в разработке web-приложений. Спроектирована начальная архитектура образовательного портала, основанная на микросервисной архитектуре. Кратко описана ее структура и приведен обоснованный выбор архитектурного стиля.

Ключевые слова: архитектура ПО, проектирование, микросервисная архитектура, образовательный портал, микросервисы, API

Введение

Разработка веб-приложений в 2022 году – это уже не инновация. За относительно небольшое время сформировалось много различных архитектур, выполняющих различные функции. Тем не менее, с каждым годом требования к качеству и функциональным возможностям web-приложений растут, из-за чего появляется необходимость в изучении новых архитектурных шаблонов и поиске новых инструментов.

В данной статье рассмотрим основные архитектурные подходы в web-разработке, а также спроектируем архитектуру образовательного портала с целью дальнейшей реализации приложения с учётом выбранной архитектуры.

Анализ существующих архитектурных подходов. Определение архитектуры программного обеспечения

У понятия архитектуры программного обеспечения существует очень много определений, но единственно верного, с которым согласились бы абсолютно все, нет. Одно из самых популярных было сформулировано Леном Бассом: «Программная архитектура вычислительной системы – это набор структур, необходимых для ее обсуждения и состоящих из программных элементов, связей между ними и свойств, присущих этим элементам и связям». Говоря простыми словами, архитектура ПО – это совокупность элементов и связей, с помощью которых можно сформировать структуру и представить архитектуру программного продукта [1].

Популярные архитектурные решения в разработке web-приложений

На данный момент существует большое количество различных архитектур, на основе которых можно создавать web-приложения: монолитная, сервис-ориентированная, микросервисная и бессерверная[2].

Монолитный подход

Монолитная архитектура ПО делится на однопроцессный и модульный монолит.

Однопроцессный монолит – самый распространённый и простой подход, который представляет из себя систему, работающую как один процесс. При объединении модулей монолита в одно целое появляется модульный монолит. Одной из его особенностей является не только возможность параллельной разработки над разными частями, но также работа с различными базами данных, соответствующими конкретному модулю. Однако даже при таком подходе изменение одной из частей может зависеть от остальных. Сравнение монолитов изображено на рисунке 1.

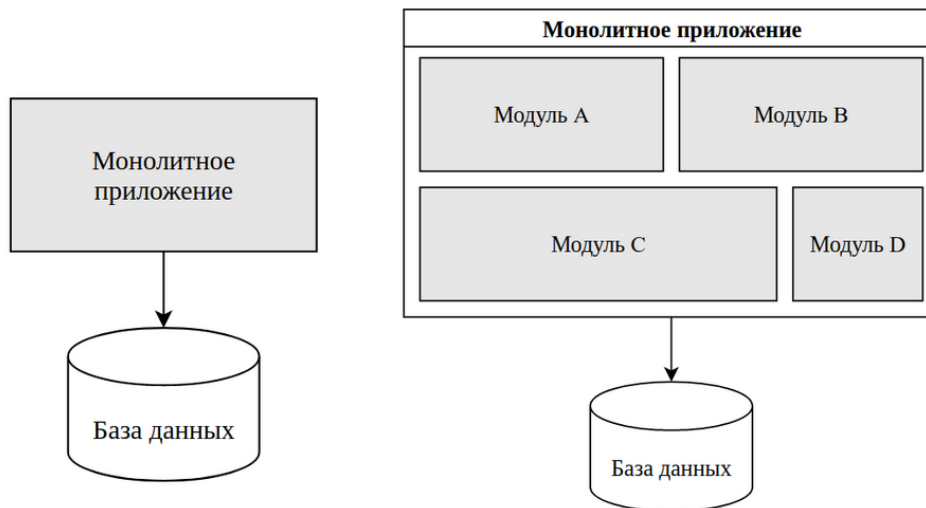


Рисунок 1 – Однопроцессный и модульный монолиты

Достоинством монолита является простота развёртывания, упрощённый мониторинг, отсутствие проблем повторного использования кода и т. д. Однако эта простота в развёртывании, разработке, тестировании и сопровождении становится недостатком при росте приложения, особенно когда над ним работает команда из нескольких человек. Последнее, помимо возможных трудностей тестирования, создаёт проблему привязки к конкретному списку технологий: все модули должны быть написаны на одном языке программирования, работа системы строго привязана к определённой базе данных и т. д.

Помимо этого, главным недостатком монолитного подхода является то, что при аварийном завершении программы прекращается работа всей системы. Особенно критичным этот недостаток считается тогда, когда сбой произошёл из-за ошибок, связанных с невнимательностью программиста, поскольку в короткие сроки потребуется не только внести изменение в исходный код программы, но и развернуть само приложение. В крупных же проектах, состоящего из миллионов строк кода, такое развёртывание может занимать не один час, поэтому чаще всего на практике оно происходит согласно какому-либо расписанию.

Бессерверный подход

Бессерверная архитектура – архитектурная модель, в основе которой лежит идея об управлении системой посредством функций без наличия полного контроля над циклом запросов и ответов.

Приложения на основе бессерверной архитектуры базируются на использовании BaaS и FaaS решений, делегируя задачи хранения и обработки данных другим сервисам. Вся бизнес-логика бессерверных приложений реализуется с помощью отдельных экземпляров управляющего кода, тем самым снимая с разработчиков необходимость заботиться о серверной инфраструктуре приложений [3].

Для большего понимания можно привести примеры сервисов, реализующих бессерверный подхода: AWS, Azure, Amazon. Все эти сервисы предоставляют различные функции – вычислительные мощности, дисковое пространство и т. д.

Делегирование функции управления инфраструктурой является как преимуществом, так и недостатком подхода. С одной стороны, это лишает необходимости создавать сервера, конфигурировать их и сопровождать. С другой - сильно повышает зависимость системы от поставщика услуг.

Сервис-ориентированный подход

Сервис-ориентированная архитектура (COA) – архитектурный, модульный подход в разработке программного обеспечения, базирующийся на использовании обособленных, слабо связанных или несвязанных, легко заменяемых компонентов, взаимодействие между которыми реализуется на основе общепринятых протоколов [4].

В такой архитектуре каждый компонент реализует определенный сервис и предоставляет открытые интерфейсы для взаимодействия с ним (API). Протоколы взаимодействия, как и технологии, с помощью которых реализуются компоненты, могут быть разными и отличаться друг от друга в рамках одной системы (разные компоненты могут быть реализованы с применением различных технологий и взаимодействовать по различным протоколам). Примеры используемых протоколов: SOAP, XML-RPC.

Также для взаимодействия нескольких компонентов в рамках одной системы реализуется корпоративная сервисная шина (ESB) [5].

ESB – программное обеспечение для обмена данными между разными сервисами системы. Примерная схема сервис-ориентированной архитектуры представлена на рисунке 2.

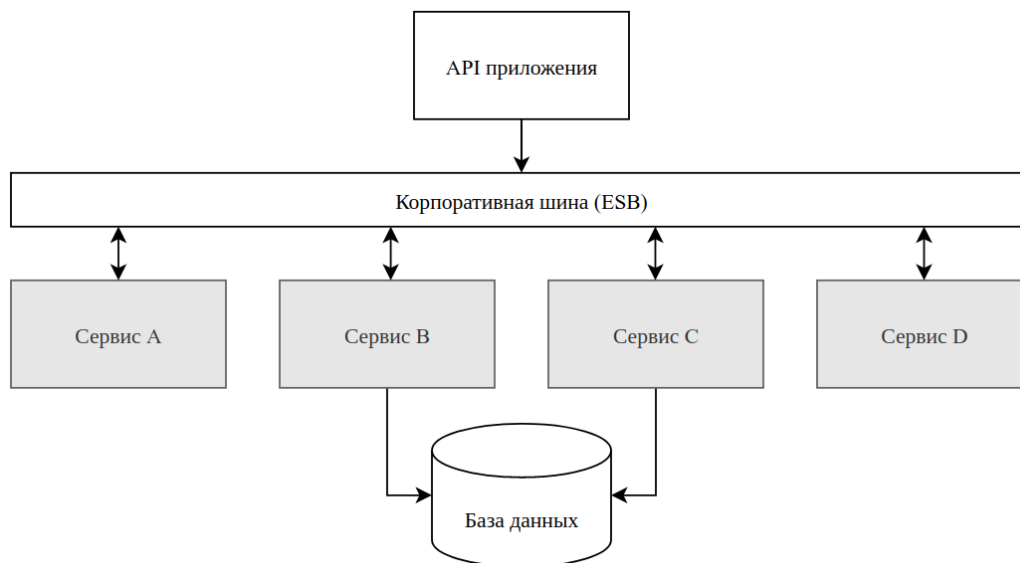


Рисунок 2 – Сервис-ориентированная архитектура

СОА обладает целым рядом преимуществ:

- простота сопровождения: сопровождение отдельных модулей, реализующих конкретный набор функций как сервис, намного проще, чем сопровождение объемного монолита;
- более высокая надежность, чем у монолитной архитектуры, которая достигается за счет более простого процесса тестирования и отладки отдельных модулей;
- масштабируемость: возможность запускать несколько экземпляров одной службы и масштабировать отдельные модули независимо друг от друга позволяет значительно экономить экономические, человеческие и другие ресурсы;
- независимая разработка: при наличии спроектированных открытых интерфейсов различных служб появляется возможность вести их разработку параллельно и независимо друг от друга [6].

К минусам данного подхода можно отнести увеличение сложности управления, так как каждая служба должна обрабатывать большое количество запросов, обеспечивая при этом достаточно высокую скорость ответа. При достаточно малом объеме разрабатываемого продукта, архитектурный подход перестает себя оправдывать: увеличиваются затраты, ресурсы на разработку, а достоинства перестают быть существенными.

Микросервисный подход

Микросервисная архитектура (МСА) по своей сути является одним из вариантов сервис-ориентированной архитектуры. Разница заключается в том, что микросервисная архитектура предполагает разбиение программного комплекса на максимально малые части (микросервисы), а также минимизацию совместно используемых компонентов. В отличие от СОА, в МСА компоненты должны обладать полной автономией, иметь свои базы данных и т. д. Философия МСА заключается в том, что все микросервисы должны выполнять лишь одну, достаточно простую функцию [7], но делать ее качественно. Пример МСА приведен на рисунке 3.

Микросервисы, наследуют все преимущества СОА, могут быть реализованы с использованием разных технологий и использовать различные протоколы для взаимодействия между собой. Такая архитектура максимально упрощает процесс обслуживания и развертывания отдельных модулей в силу их небольшого размера, упрощает процесс масштабирования, позволяя масштабировать каждый сервис индивидуально, увеличивает автономность команд разработчиков и позволяет максимально быстро проводить ротации в командах программистов, так как для понимания сути работы микросервиса и изучения кода уходит минимальное количество времени. Микросервисная архитектура позволяет быстро внедрять новые решения и технологии.

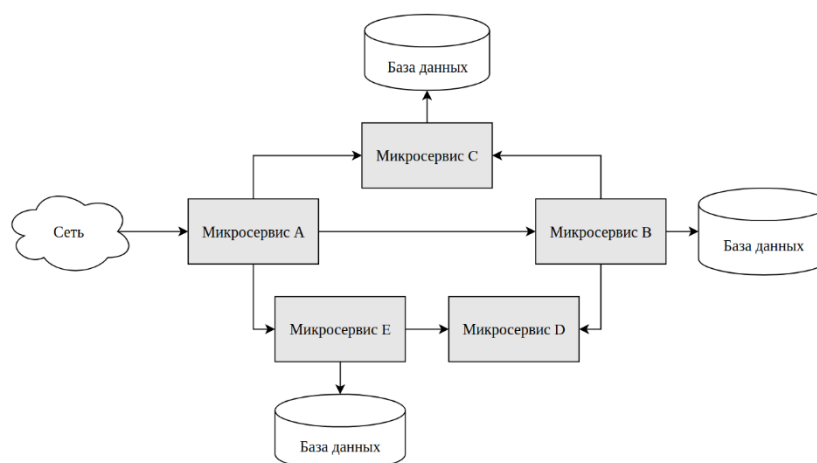


Рисунок 3 – Микросервисная архитектура

Вербальное описание учебного портала

Сервис позволяет регистрировать учебные заведения, преподавателей и студентов, а также наполнять информацией справочники. Преподаватели получают возможность организовывать учебные занятия, распространять методические пособия и другие файлы, выдавать задания и проводить тестирование. Студенты имеют возможность выполнять задания, обеспечиваются круглосуточным доступом ко всем необходимым для обучения материалам и т. д.

Выбор архитектуры

В связи с большим объемом и высокой сложностью разрабатываемого продукта, было принято решение не рассматривать бессерверную архитектуру для реализации образовательного портала. Наличие ярко выраженных обособленных частей, позволяющих разбить продукт на несколько небольших автономных модулей, заставляет задуматься об использовании СОА либо МСА. Так как СОА больше подходит для больших, корпоративных приложений, которые сложно разбить на небольшие микросервисы, было принято решение остановиться на микросервисной архитектуре.

Функциональность системы тесно связана со стойкостью всей системы в моменты пиковой нагрузки, когда в учебное время резко активизируются все студенты и преподаватели. Благодаря особенностям МСА, появляется возможность решать проблемы производительности, масштабируя лишь отдельные микросервисы системы. Для этого можно расположить балансировщик нагрузки перед несколькими экземплярами каждого из микросервисов, что позволит распределять запросы между ними. В случае расширения проекта и увеличения количества посетителей, его масштабирование также сводится к увеличению количества экземпляров отдельных микросервисов. Подбор серверов с учетом особенностей функционирования отдельных микросервисов позволит существенно сэкономить на сопровождении проекта.

Структура системы

Таблица 1 – Описание микросервисов

Микросервис	Описание
Расписание	Оформление расписания диспетчером, уведомление студентов о начале занятий, мониторинг расписания
Курсы	Организация образовательных курсов и занятий
Задания	Создание, выполнение и проверка заданий
Тестирование	Создание тестов и проведение тестирования обучающихся
Авторизация	Регистрация и авторизация. Микросервис не имеет своей собственной базы данных, лишь генерирует JWT-токен для авторизации и аутентификации
ВУЗы	Инкапсуляция данных о ВУЗах, кафедрах, дисциплинах и т. д.
Успеваемость	Формирование отчетов об успеваемости студентов

Проектирование БД и сопутствующие проблемы

Как уже было сказано ранее, каждый микросервис должен быть абстрагирован от других и иметь свою базу данных, если выполняет функцию хранения информации. В таком случае появляется вопрос: как из одного сервиса использовать статические или динамические данные другого сервиса?

Для решения этого вопроса существует несколько распространённых подходов: дублирование таблиц и данных, хранение информации в конфигурационных файлах и использование сервиса как хранилище с возможностью чтения и записи через внешний API [8]. Первые два подхода применяются только к справочным данным, изменение которых происходит редко, но сами данные являются важными и часто используемыми. Последний же подход наоборот универсален и применим как к статическим данным, так и к динамическим. Пример такого подхода изображён на рисунке 4.

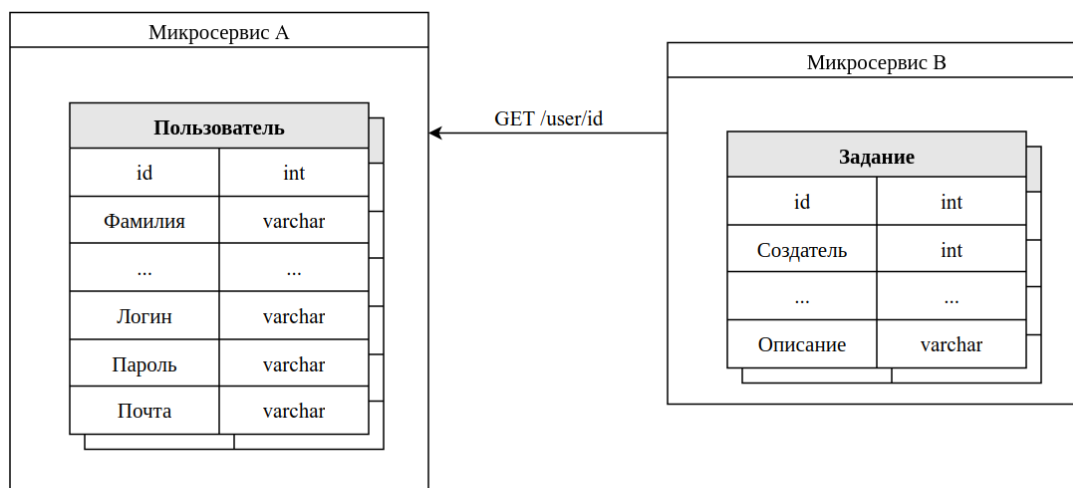


Рисунок 4 – Пример подхода «Сервиса как хранилища с API»

Проблема проектирования API

Самый очевидный подход к проектированию API состоит в том, чтобы позволить всем клиентам напрямую взаимодействовать с микросервисами. Такой подход используется в монолитной архитектуре, однако, несмотря на кажущуюся простоту, он имеет несколько недостатков, которые заставляют отказаться от него. При таком подходе, клиенту часто приходится обращаться к нескольким микросервисам за один раз, чтобы собрать всю нужную информацию. Это заметно снижает скорость ответа и уровень инкапсуляции, из-за чего разработчики вынуждены менять все клиенты параллельно с API, что может вызывать трудности. Микросервисы могут использовать разные протоколы для передачи данных, которые могут быть несовместимы при работе с клиентом, поэтому, предоставляя клиентам прямой доступ к микросервисам, разработчики должны либо отказаться от использования «неудобных» для клиентов протоколов, либо реализовать их поддержку. Решением этих проблем может быть шаблон «API шлюз».

API-шлюз – это сервис, который служит точкой входа в приложение из внешнего мира[9]. Он отвечает за маршрутизацию запросов, объединение API и другие возможности. Принимая запросы к системе извне, шлюз маршрутизирует запросы по всем микросервисам и одновременно выполняет функцию преобразования протоколов. Для внешних клиентов предоставляется один протокол, а ко внутренним сервисам шлюз обращается по другому протоколу. Также шлюз может выполнять функции ограничения частоты запросов, кэширования, сбора каких-либо показателей и ведения журнала логов.

Для взаимодействия микросервисов в системе образовательного портала была выбрана пара протоколов REST API и gRPC.

Результат проектирования архитектуры образовательного портала

Результат проектирования архитектуры образовательного портала представлен на рисунке 6.

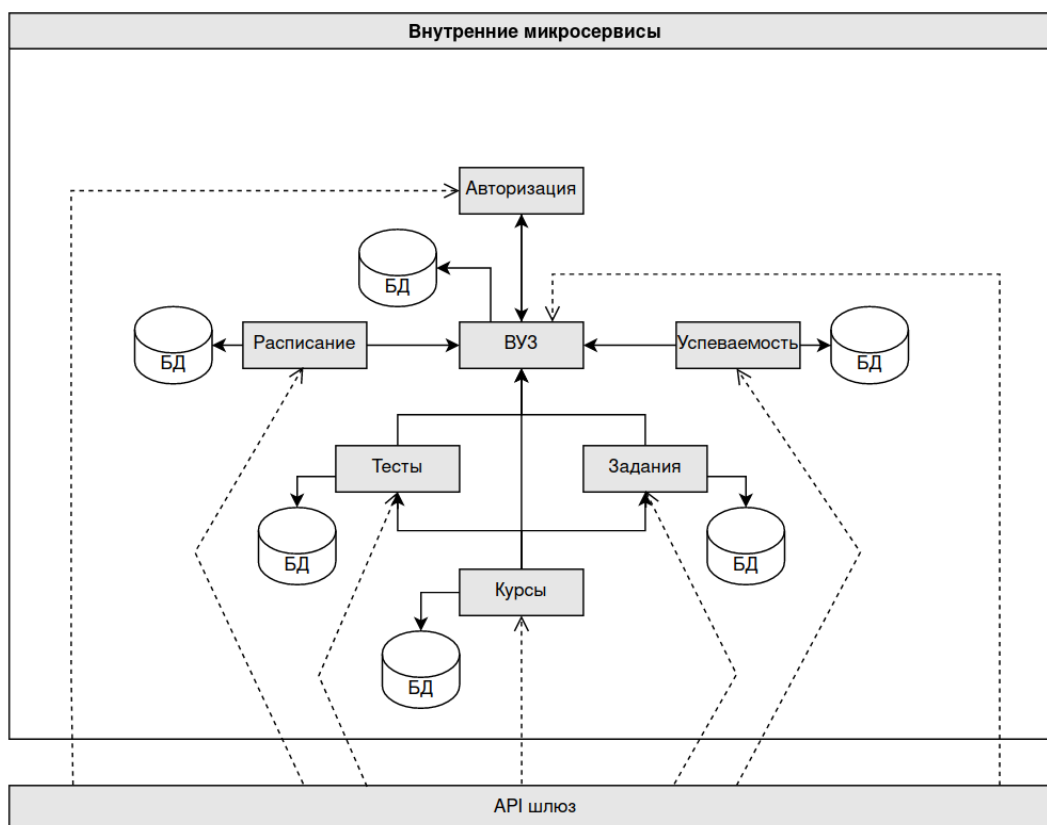


Рисунок 6 – Архитектура образовательного портала

Как было описано выше, архитектура образовательного портала базируется на МСА и состоит из семи микросервисов. Каждый из них имеет свою, обособленную базу данных, кроме микросервиса «Авторизация». Этот микросервис не хранит какие-либо данные, его задача генерировать токены для авторизации и аутентификации пользователей, заменять их и валидировать.

Взаимодействовать с микросервисами клиенты будут через API шлюз, который будет служить маршрутизатором запросов к микросервисам, проверять валидность токенов авторизации (используя функционал микросервиса «Авторизация»), а также выполнять функцию преобразователя протоколов с REST API на gRPC.

Выводы

Исследование существующих архитектур и их возможностей – одно из самых интересных, важных и ключевых направлений не только в web-разработке, но и в программной инженерии в целом. Именно от грамотного проектирования и правильно выстроенной архитектуры зависит будущий успех разработки. В рамках данной статьи были рассмотрены основные архитектурные подходы в сфере web-разработки, а также описаны ключевые преимущества и свойства МСА, затронуты некоторые ее проблемы и их возможные решения в ходе проектирования архитектуры образовательного портала.

МСА является одной из самых популярных архитектур, которая может помочь решить целый ряд проблем: повысить масштабируемость продукта, увеличить автономность рабочих групп, упростить процесс внедрения новых технологий и т.д. Однако следует помнить о том, что архитектура – это лишь инструмент, который следует уметь применять правильно. МСА будет плохим решением в случаях, когда: разрабатываемое ПО будет устанавливаться пользователем, нет четкого понимания как провести декомпозицию системы, нет понимания цели применения данного подхода.

Перспективами развития архитектурной модели образовательного портала является ее модернизация путем проработки существующих компонентов системы (их декомпозиция), добавление новых микросервисов, а также проработки механизмов взаимодействия модулей.

Перспективами развития проекта образовательного портала является готовый программный комплекс, разработанный на основе созданной архитектуры.

Литература

1. Documenting Software Architectures [Электронный ресурс] - Режим доступа: http://sunner.cn/courses/SA/Documenting_Software_Architecture,_2nd_edition.pdf

2. Monolith vs SOA vs Microservices vs Serverless Architecture [Электронный ресурс] - Режим доступа: <https://rubygarage.org/blog/monolith-soa-microservices-serverless>
3. Peter Sbarski, Yan Cui, Ajay Nair, Serverless Architectures on AWS, 2nd Edition - Manning Publications Co. - 257 с.
4. А.М. Дергачев, Ю.Д. Кореньков, И.П. Логинов, А.Г. Сафронов, ТЕХНОЛОГИИ ВЕБ-СЕРВИСОВ - СПб: Университет ИТМО, 2021. - 102 с.
5. Ильин И. В., Анисифоров А. Б., Левина А. Н. Модели обмена данными в интегрированной информационной системе эффективного управления инновационно-промышленным кластером // Научно-технические ведомости СПб ГПУ. Сер.: Экономические науки. 2011. № 6 (137). С. 240-247.
6. Сервис-ориентированная архитектура [Электронный ресурс] - Режим доступа: <http://citforum.ru/internet/webservice/soa/>
7. Ньюмен С., От монолита к микросервисам: Пер. с англ. — СПб.: БХВ-Петербург, 2021. — 272 с.
8. Ньюмен С., Создание микросервисов. — СПб.: Питер, 2016. — 304 с.
9. Ричардсон К., Микросервисы. Паттерны разработки и рефакторинга. — СПб.: Питер, 2019. — 544 с.

Суров А. Д., Раннев Д. Г., Чернышова А. В. Выбор архитектуры программного обеспечения для создания образовательного портала. В статье представлен краткий обзор существующих архитектурных подходов в разработке web-приложений. Спроектирована начальная архитектура образовательного портала, основанная на микросервисной архитектуре. Кратко описана ее структура и приведен обоснованный выбор архитектурного стиля.

Ключевые слова: архитектура ПО, проектирование, микросервисная архитектура, образовательный портал, микросервисы, API

Surov A. D., Rannev D. G., Chernyshova A. V. The choice of software architecture for creating an educational portal. The article provides a brief overview of existing architectural approaches in the development of web applications. The initial architecture of the educational portal based on the microservice architecture was designed. Its structure is briefly described and a reasonable choice of architectural style is given.

Keywords: software architecture, design, microservice architecture, educational portal, microservices, API

Разработка и развертывание Telegram бота для оповещения о новостях ДонНТУ

В.Ю. Олейник^{*1}, А.В. Чернышова^{*2}

^{*1} студент, Донецкий национальный технический университет,
ultimathul3@ya.ru

^{*2} старший преподаватель, Донецкий национальный технический университет,
chernyshova.alla@rambler.ru, OrcID: 0000-0003-2546-2167, SPIN-код: 3318-2066

Олейник В.Ю., Чернышова А.В. Разработка и развертывание Telegram бота для оповещения о новостях ДонНТУ. В статье представлен краткий обзор мессенджера Telegram и его возможностей для разработчиков. Спроектирован, разработан и размещен на сервер Telegram бот для оповещения о новостях ДонНТУ.

Ключевые слова: Telegram, Bot API, язык программирования Go, автоматизированный сбор информации, развертывание.

Введение

Telegram – это облачное приложение для обмена сообщениями (мессенджер), которое делает акцент на скорость и безопасность. В марте 2022 года Telegram стал самым популярным мессенджером в России. За первые две недели марта его доля в общем объеме трафика в мессенджерах увеличилась с 48% до 63% по сравнению с аналогичным периодом февраля месяца.

Telegram предоставляет два вида API (программных интерфейсов сервиса) для разработчиков:

1. Telegram API – позволяет создавать собственные клиенты Telegram.
2. Bot API – позволяет создавать программы (боты), которые используют сообщения Telegram в качестве интерфейса.

Название “бот” произошло от сокращения слова “робот”. Telegram бот представляет собой специальный аккаунт, который получает команды от пользователя, обрабатывает их и генерирует ответ. Также бот может самостоятельно отправлять сообщения пользователю для каких-либо оповещений. Физически команды обрабатываются на серверах разработчиков ботов обычными программами.

Платформа Telegram содержит более десяти миллионов ботов, которые используются для самых различных целей: аналитика данных, модерация групп и каналов, создание опросов, интеграция с внешними сервисами, размещение HTML5 игр и многое другое.

Боты Telegram используются многими пользователями мессенджера и являются важным его элементом.

Постановка задачи

Факультеты ДонНТУ имеют собственные каналы для оперативной публикации информации. Целью данной статьи является создание Telegram бота для оповещения о новостях ДонНТУ, которого можно встроить в Telegram каналы факультетов. На момент написания статьи подобных ботов не существует, следовательно, данная задача является актуальной и имеет практическую ценность.

Анализ требований

Исходя из постановки задачи, были определены следующие требования для Telegram бота:

- Telegram бот должен работать с пользователями, группами и каналами;
- Telegram бот должен иметь команду для подписки на рассылку новостей;
- Telegram бот должен иметь команду для отмены подписки на рассылку новостей;
- Telegram бот должен иметь команду для ручной отправки необходимой новости ДонНТУ по ссылке;
- Telegram бот должен с заданной периодичностью проверять новости с сайта ДонНТУ;
- Telegram бот должен отправлять новости всем подписавшимся на рассылку пользователям, группам и каналам.

Регистрация Telegram бота

Перед началом разработки бота его нужно зарегистрировать в самом мессенджере Telegram. Для этого

необходимо обратиться к боту BotFather и указать название и логин создаваемого бота (см. рис. 1). Там же есть возможность задать описание боту, поменять фото его профиля, создать список доступных команд.

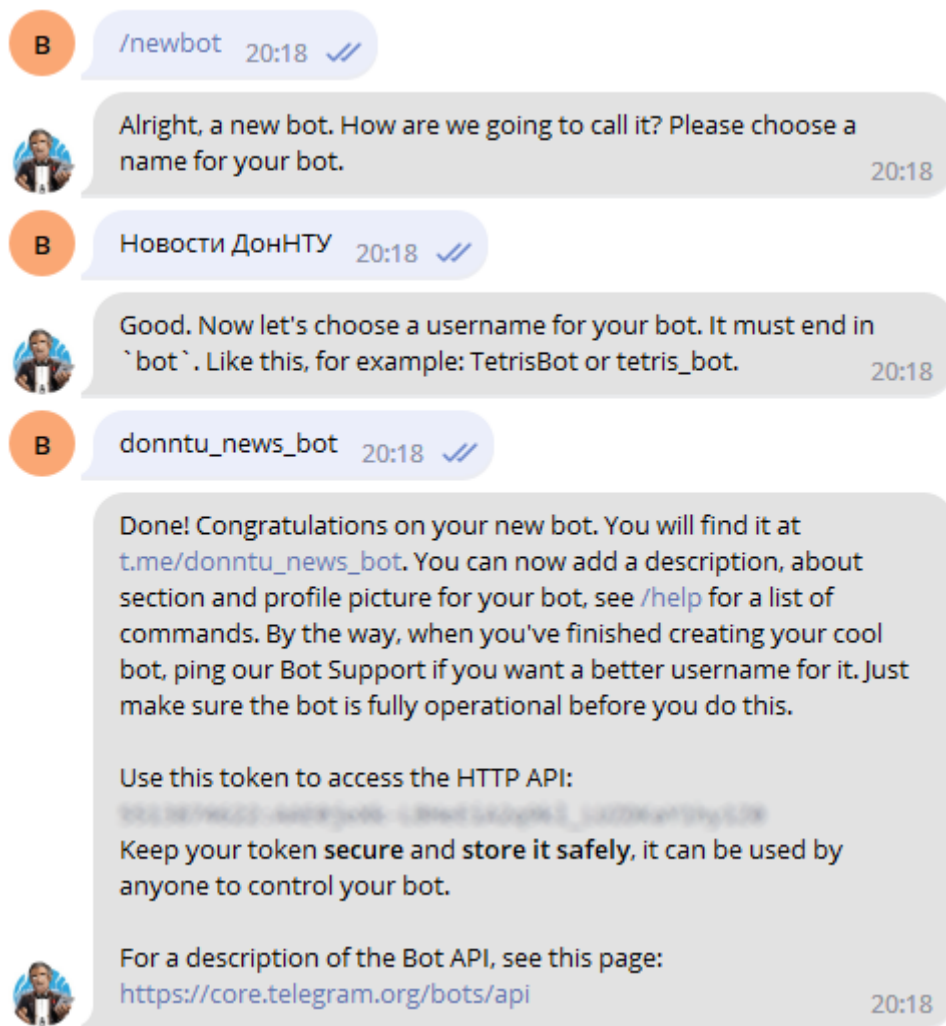


Рисунок 1 – Взаимодействие с ботом BotFather

После регистрации BotFather отправит токен доступа регистрируемого бота.

Токен доступа – это секретный ключ, состоящий из набора цифр, букв латинского алфавита и символов “:”, “-”, “_”. Токен доступа используется для взаимодействия с Bot API Telegram.

Взаимодействие с Bot API

Взаимодействие с Bot API происходит с помощью GET или POST HTTP запросов на сервер Telegram. В URL запросов необходимо указать токен доступа бота, имя метода и параметры метода при необходимости. Шаблон и примеры запросов изображены на рисунке 2.

```
https://api.telegram.org/bot<ТОКЕН_ДОСТУПА>/<ИМЯ_МЕТОДА>
```

```
https://api.telegram.org/bot123456:ABC-DEF1234ghIk1-zyx57W2v1u123ew11/getMe
```

```
https://api.telegram.org/bot123456:ABC-DEF1234ghIk1-zyx57W2v1u123ew11/sendMessage?chat_id=12345678&text=Привет
```

Рисунок 2 – Шаблон и примеры запросов

В Bot API имеются методы для отправки сообщений (фото, видео, аудио, документы, геолокация), редактирования информации чатов (редактирование фото, названия, описания), взаимодействия с пользователями чатов (блокировка, разблокировка пользователей, назначение привелегий) и многие другие [1].

Ответ на запрос приходит в формате JSON, который имеет обязательное логическое поле “ok” и необязательное поле “description” с описанием результата. Также в случае неудачного запроса возвращается поле “error_code” с кодом ошибки.

Получение обновлений для Telegram бота

Для того чтобы бот знал о произошедших событиях (пользователь отправил сообщение, пригласил в чат и др.), он должен получить обновления от сервера Telegram. Существует 2 способа получить обновления для бота:

- метод Bot API “getUpdates”;
- вебхуки (webhooks).

Схематично различия между этими двумя способами изображены на рисунке 3.

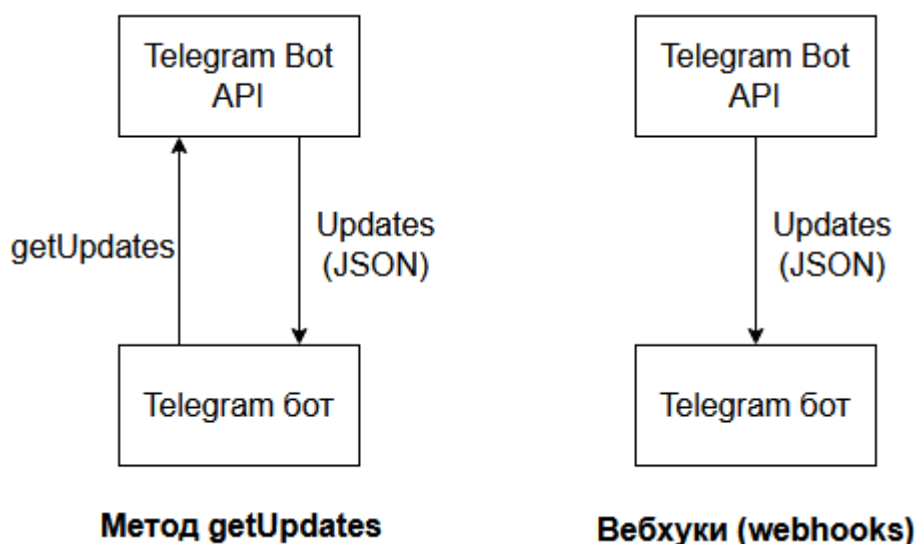


Рисунок 3 – Различия между способами получения обновлений

Способ с методом “getUpdates” предполагает, что бот с некоторой периодичностью обращается к серверу Telegram для получения обновлений. Данный способ имеет несколько недостатков:

- пользователь может ощущать значительную задержку получения ответа от бота (в зависимости от периода обращения бота к серверу);
- процессорное время сервера бота тратится без пользы в случае длительного отсутствия обновлений.

Способ с вебхуками лишен недостатков способа с методом “getUpdates”. В этом случае сервер Telegram самостоятельно посылает обновления на сервер бота. Однако для этого сервер бота должен соответствовать следующим требованиям:

- белый IP-адрес сервера (все устройства в сети Интернет могут напрямую обращаться к устройству с белым IP-адресом);
- сервер должен работать по протоколу HTTPS.

Для разрабатываемого бота был выбран способ получения обновлений с помощью вебхуков. Активация данного способа была произведена с помощью метода setWebhook со следующими параметрами:

- secret_token (секретный токен);
- url (унифицированный адрес ресурса).

В параметре “secret_token” был указан произвольный ключ, который Telegram будет отправлять боту в каждом запросе с обновлениями в заголовке “X-Telegram-Bot-Api-Secret-Token”. Это необходимо для того, чтобы убедиться, что входящий запрос был отправлен именно сервером Telegram.

В параметре “url” был указан домен donntu-news.ga, полученный через сервис поставщика бесплатных доменов Freenom. Настройки домена изображены на рисунке 4.

Modify Records

Name	Type	TTL	Target	
	A	3600	194.87.69.188	Delete
WWW	A	3600	194.87.69.188	Delete

Рисунок 4 – Настройки домена donntu-news.ga

В настройках должны быть как минимум две доменные записи, содержащие IP-адреса серверов, на которые будет ссылаться домен. Второй IP-адрес может быть отличен от первого, тогда, если сервер на первом IP-адресе недоступен, запрос будет отправлен на сервер второго IP-адреса. Колонка TTL (time to live, время жизни) указывает на максимальное количество маршрутизаторов, через которые может пройти IP-пакет до своей цели, перед тем как будет уничтожен. Благодаря этому IP-пакет не будет бесконечно блуждать по сети Интернет.

Физически бот будет расположен на виртуальном сервере (VPS) российского хостинга JustHost. IP-адрес этого сервера и был указан в настройках домена. Характеристики выбранного VPS следующие:

- ОС Ubuntu Server 20.04 64bit;
- ЦП Intel Xeon E5-2697 (доступно 1 ядро);
- 512 Мб ОЗУ;
- 10 Гб ПЗУ.

Данных характеристик достаточно для работы разрабатываемого бота. Подключение к VPS и работа с ним происходит по протоколу SSH.

Разработка Telegram бота

Для разработки Telegram бота выбран язык программирования Go. Язык был официально представлен компанией Google в 2009 году. Он имеет лаконичный синтаксис, является компилируемым и в своём составе содержит множество библиотек для разработки серверных приложений, следовательно, хорошо подходит для решения поставленной задачи.

Проект приложения бота состоит из следующих компонентов:

- пакет “api”;
- пакет “db”;
- пакет “handler”;
- пакет “logger”;
- пакет “observer”;
- пакет “parser”;
- пакет “types”;
- файл “.env”;
- файл “main.go”.

В файле “main.go” содержится точка входа в программу, где запускается сервер. Перед этим инициализируются модуль логирования и база данных, считывается “.env” файл, в котором находятся настройки бота: имя домена, токен доступа, секретный токен и период проверки новостей сайта ДонНТУ в минутах. Для работы сервера по протоколу HTTPS необходим SSL сертификат, который позволяет использовать безопасное зашифрованное соединение. Стандартный пакет “autocert” языка Go позволяет получить SSL сертификат от центра сертификации Let’s Encrypt (см. рис. 5).

```
// autocert provides automatic access to certificates from Let's Encrypt
certManager := autocert.Manager{
    Prompt:      autocert.AcceptTOS,
    HostPolicy:  autocert.HostWhitelist(os.Getenv("DOMAIN")),
    Cache:       autocert.DirCache("certs"),
}
```

Рисунок 5 – Менеджер пакета “autocert” для получения SSL сертификата от Let’s Encrypt

В менеджере пакета “autocert” передается имя домена (считанное из “.env” файла) и директория, куда будет

сохранен SSL сертификат.

Получение сертификата происходит по протоколу ACME (Automated Certificate Management Environment, протокол автоматического управления сертификатами), разработанный Исследовательской группой по интернет-безопасности для их сервиса Let's Encrypt [2].

Telegram позволяет использовать и самоподписанные SSL сертификаты, т.е. сертификаты, подписанные самим сервером. В этом случае в методе “setWebhook” необходимо передать параметр “certificate” для загрузки публичного ключа сертификата. Т.к. для самоподписанного сертификата не обязательно иметь домен, вместо параметра “url” можно передать параметр “ip_address” (IP-адрес сервера).

Пакет “types” содержит необходимые типы данных (структуры) для работы бота:

- Update (обновление от Telegram);
- Response (ответ от Telegram);
- InputMedia (для отправки медиа файлов).

Пакет “api” содержит необходимые методы для взаимодействия с Bot API:

- отправка сообщения;
- отправка группы фотографий.

Пакет “db” содержит необходимые методы для взаимодействия с базой данных. В качестве системы управления базами данных была выбрана SQLite3, её особенностью является то, что вся база данных состоит всего из одного файла. Это позволяет с легкостью переносить базу данных с одного сервера на другой. Для выполняемой задачи не требуется высокая производительность от СУБД, следовательно, выбор SQLite3 является оптимальным. База данных состоит из двух таблиц:

- subscribers (пользователи, каналы или группы, подписанные на рассылку новостей);
- last_news (таблица, которая хранит дату последней новости на сайте ДонНТУ).

Пакет “logger” содержит модуль логирования, с помощью которого во время работы бота в файл записываются следующие события с временной меткой: запуск бота, получение обновлений от Telegram, отправка сообщений пользователям, проверка новостей, возникновение исключительных состояний. Использование данного модуля поможет быстро установить и исправить ошибки в случае их возникновения.

Пакет “handler” содержит функции для обработки запросов:

- функция отправки новости вручную (ссылка на новость задаётся пользователем);
- функция подписки на рассылку;
- функция отмены подписки на рассылку.

Перед тем, как обработать запрос, проверяется заголовок “X-Telegram-Bot-Api-Secret-Token” на соответствие секретному токenu, считанному из “.env” файла.

Каждый запрос обрабатывается в отдельной горутине (goroutine, go-подпрограмма) [3]. По принципу работы горутин схожи с потоками в операционных системах, однако имеют значительные отличия по реализации:

- горутинami управляет встроенный в язык Go планировщик, а потоками – планировщик ОС;
- горутина занимает гораздо меньше оперативной памяти (от 2 КБ, потоки – от 1-2 МБ);
- у горутин время запуска быстрее, чем у потоков.

Таким образом, обработка запросов на многоядерных серверах будет осуществляться параллельно и эффективно благодаря горутинам.

Пакет “parser” содержит функции для парсинга веб-страницы новостей сайта ДонНТУ. Парсинг (parse, разбор) HTML – это обработка HTML документа страницы и извлечение из неё необходимой информации. Для этих целей была использована библиотека “goquery”, которая преобразовывает HTML документ в узлы (nodes), с которыми можно взаимодействовать средствами языка. Поиск нужных узлов осуществляется с помощью CSS-селекторов. CSS-селектор – это шаблон, соответствующий элементам на веб-странице. На рисунке 6 изображен фрагмент кода поиска всех изображений новости (тег “img”), которые расположены в элементе с классом “l-content” (в данном случае div блок, контент самой новости на странице).

```
doc.Find(".l-content").Find("img").Each(func(i int, s *goquery.Selection) {
    src, _ := s.Attr("src")

    if i == 0 {
        titleImg = fmt.Sprintf("<a href=\"%s\">Фото</a>", src)
    } else {
        otherImgs = append(otherImgs, src)
    }
})
```

Рисунок 6 – Фрагмент кода поиска изображений новости

У каждого тега “img” извлекается атрибут “src” – ссылка на изображение. Главное изображение новости в

виде тега “a” со ссылкой отправляется вместе с содержанием новости. Остальные изображения ввиду особенностей Bot API добавляются в список и отправляются отдельным сообщением. Чтобы Telegram правильно интерпретировал сообщения с разметкой HTML необходимо в метод отправки сообщения (“sendMessage”) передать параметр “parse_mode” (режим разбора) со значением “HTML”. Также, необходимо учитывать, что Telegram поддерживает не все теги HTML, а значит, требуется после парсинга содержимого новости удалить лишние теги, либо заменить их нужными символами для приближения к оригинальной разметке.

Пакет “observer” представляет собой функцию “наблюдатель”, которая запускается в отдельной горутине. Данная функция с заданной в “.env” файле периодичностью парсит дату и время последней опубликованной новости на сайте и сравнивает с датой и временем последней рассылаемой новости (хранится в БД). В случае если первая дата позже второй, новость парсится полностью и отправляется всем пользователям, каналам и группам, подписавшимся на рассылку. Дата и время этой новости записываются в БД.

Выводы

Разработанный в соответствии с требованиями Telegram бот автоматизирует процесс рассылки новостей ДонНТУ. Бот был размещён на виртуальный сервер и протестирован. Исходный код доступен в github репозитории “donntu-news-tg-bot” [4].

К недостатком данной реализации можно отнести то, что в случае изменения разметки страницы новостей на сайте ДонНТУ может наблюдаться некорректная работа бота. Возможные улучшения реализации бота могут быть в пакете “parser” и направлены на увеличение гибкости парсинга новостей для уменьшения вероятности возникновения нештатных ситуаций в будущем.

Литература

1. Telegram Bot API Reference [Электронный ресурс]. – Режим доступа: <https://core.telegram.org/bots/api>
2. Automatic Certificate Management Environment (ACME) RFC [Электронный ресурс]. – Режим доступа: <https://datatracker.ietf.org/doc/rfc8555/>
3. Go на практике / Батчер М., Фарина М. – М.: ДМК Пресс, 2017. – 374 с.
4. GitHub репозиторий Telegram бота для оповещения о новостях ДонНТУ [Электронный ресурс]. – Режим доступа: <https://github.com/ultimathul3/donntu-news-tg-bot>

Олейник В.Ю., Чернышова А.В. Разработка и развертывание Telegram бота для оповещения о новостях ДонНТУ. В статье представлен краткий обзор мессенджера Telegram и его возможностей для разработчиков. Спроектирован, разработан и размещен на сервер Telegram бот для оповещения о новостях ДонНТУ.

Ключевые слова: Telegram, Bot API, язык программирования Go, автоматизированный сбор информации, развертывание.

Oleinik Vadim, Chernyshova Alla. Development and deployment of Telegram bot for notification of DonNTU news. The article provides a brief overview of the Telegram messenger and its capabilities for developers. Designed, developed and deployed on the server Telegram bot for notification of DonNTU news.

Key words: Telegram, Bot API, Go programming language, automated information collection, deployment.

Разработка сервиса определения местоположения членов группы с доверительными отношениями

О.В. Морозова

ст. преподаватель кафедры программной инженерии им. Л.П. Фельдмана, Донецкий национальный технический университет, malyaw0n@mail.ru

Морозова О.В. Разработка сервиса определения местоположения членов группы с доверительными отношениями. В статье рассмотрены основные возможности сервиса, общая схема взаимодействия устройств и структура приложения. Рассмотрены основные технологии применяемые при разработке приложения. Описаны основные особенности работы приложения.

Ключевые слова: Android-приложение, SQLite, Google Mobile Services, Android SDK, CRUD, GPS.

Введение

В современном обществе невозможно представить человека без мобильного устройства – это случилось после того как современные телекоммуникации поднялись на новый уровень и мобильный интернет стал широко распространён, пользователи стали проводить перед экраном смартфона еще больше времени, чем за компьютером. В наше время дети идут в начальную школу с собственными смартфонами, а пенсионеры проводят время во «ВКонтакте» и знают о Google Play. Поэтому существует большой пласт сервисов отслеживания местоположения координат членов группы с доверительными отношениями, установив небольшое приложение, пользователь получает детальную информацию о нахождении и передвижений члена группы. Членом доверительной группы могут выступать, как члены одной семьи для отслеживания передвижения детей и пожилых людей, так и группы людей по интересам, которые готовы и дают свое согласие на передачу информации.

Рассмотрим несколько популярных приложений, представленных в среде интернет:

– Life360 [1] позволяют просматривать координаты участников круга на карте, доступной по приглашению; выбирать, когда пользователь хочет открыть свое местоположение членам круга; общаться в чате с пользователями любого круга; получать сведения о координатах участника программы; определять местоположение потерянного мобильного телефона с точностью до 5 метров; работает независимо от наличия и состояния связи;

– Family Locator [2] позволяет определять координаты членов группы, но имеет несколько недостатков и платный контент;

– My Kids Safety [3] позволяет определять местонахождение всех членов группы; расставлять контрольные точки перемещения; создавать группы пользователей и эксплуатироваться несколькими пользователями, также является платным трекером;

– Glympse [4] позволяет видеть всех участников группы на одной карте; определение местонахождения участника группы в режиме реального времени; требования безопасности обязуют произвести верификацию номеров мобильных телефонов, работает не на всех версиях Андроид и не во всех странах.

При рассмотрении и анализе этих приложений можно сделать выводы, что большинство из предлагаемых сервисов являются платными, невозможность хранить данные на своих серверах, требует специальных устройств для включения GPS-навигатора, требует админ доступ к устройству и работают не на всех версиях Андроид. Исходя из данных недостатков, был разработан собственный сервис для определения местоположения, и его цель - информирование членов группы о местоположении в любой момент времени.

Постановка задачи

При изучении данной проблемы были рассмотрены все возможные средства реализации. Большинство из них требует или персонального компьютера, или мобильного устройства, остальное имеет меньший охват населения. Специализированные устройства GPS-трекеры являются платными и не всегда имеют полный функционал настройки. Персональные компьютеры вследствие своей стационарности не удобны для решения данной задачи. Поэтому для разработки сервиса были выбраны мобильные устройства, а как следствие и операционная система Андроид. Сервис представляет с собой связку между Android-приложением Family

Location, которая работает как клиентская часть, и серверной частью Family Location Server. Серверная часть полностью автоматизирована и выполняет функция хранения и извлечение данных о местоположении групп людей (см. рис. 1).



Рисунок 1 - Общая схема взаимодействия устройств системы

Основные средства разработки приложения.

SQLite — все таблицы, индексы, триггеры, представления — хранится в одном файле. Хотя SQLite не использует архитектуру клиент-сервер, приложения, выполняющие чтение/запись в базы данных SQLite, обычно называют «клиентами SQLite». Движок SQLite — не отдельно работающий процесс, с которым взаимодействует программа, а библиотека. Чтобы встроить SQLite в код, достаточно подключить нужную библиотеку. В качестве протокола обмена применяются вызовы функций библиотеки SQLite [5].

Android SDK — это среда программирования, в которую входят библиотеки, исполняемые файлы, скрипты, документация и т.д. Android SDK компилирует код вместе с любыми данными и ресурсами в файл с расширением *.apk*. Он содержит всё необходимое для установки приложения на Android-устройство [6].

Google Mobile Services (GMS) — это набор проприетарных приложений и интерфейсов прикладного программирования (API) от Google, которые обычно предварительно устанавливаются на устройства Android, такие как смартфоны, планшеты и смарт-телевизоры [7].

Основные возможности сервиса

Регистрация участника группы [8,9]. Регистрация производится по email и паролю. Данные записываются в SQLite таблицу «accounts» и синхронизируется с сервером. При любом последующем входе используется безопасное хранение данных ОС Андроид и используются данные крайней учетной записи.

Параметры (возможность настройки личной серверной части). Можно выбрать один из двух протоколов подключения - HTTPS или HTTP. URL-адрес сервера с серверной частью Family Location Server. Порт для подключения к серверной части Family Location Server и время обновления. Во время своей работы приложение отправляет новые данные об Аккаунте и текущее местоположение Аккаунта на сервер. Так же приложение с сервера получает информацию о местоположении Членов группы. Можно указать частоту синхронизации в миллисекундах. Производится проверка подключения к серверу Family Location Server после изменения настроек и их «Сохранения» или сброс настроек «По умолчанию».

Создание групп. Группа - набор (список) членов группы, объединенных одним названием. Для добавления людей в группу нужно выбрать их из списка доступных. Эти данные хранятся в SQLite таблице «families». Член группы — это существующий аккаунт, которого возможно отследить. Для каждого члена группы необходимо получить разрешение на отслеживание местоположения. После создания члена группы ему отправляется запрос с разрешением на получение геоданных. После получения разрешения можно получать информацию о местонахождении. Когда член группы создан его можно добавить в группу или несколько групп. Таким образом все участники данного сервиса образуют взаимосвязанную общность. В идеале все участники своих групп создают доверительный круг общения.

Локации. Локация - область на географической карте в виде многоугольника имеющая свое название. Если координаты месторасположения члена группы попадают в область Локации, то считается что член группы находится в этой Локации.

Разрешение на отслеживание. Разрешение представляет собой структуру (см.рис.2) хранящую данные о том, кто сделал запрос на отслеживание и к кому это запрос.

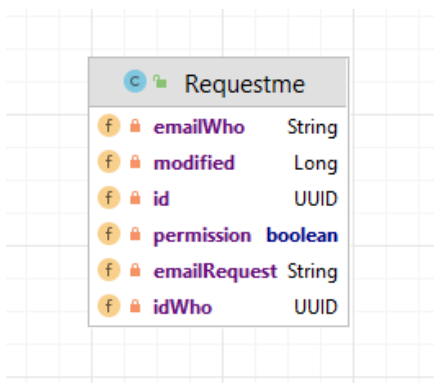


Рисунок 2 – Структура сущности запросов на разрешение

Когда один пользователь добавляет участника группы себе в аккаунт, то этому участнику приходит запрос с разрешением на отслеживание. По умолчанию отслеживание запрещено (УК РФ Статья 137. Нарушение неприкосновенности частной жизни). У каждого пользователя в аккаунте есть свой список запросов с разрешением на отслеживание, где он в любое время может изменить свое разрешение (разрешить/запретить). После получения разрешения аккаунт, сделавший запрос получает доступ к местоположению этого члена группы (см. рис. 3).

```
List<Relative> relatives = new ArrayList<>();
List<Requestme> requestmes = requestmeRepository.getAllByEmailWho(account.getEmail());
family.getRelatives().forEach(v -> {
    if (requestmes.stream().anyMatch(r -> r.getEmailRequest().equals(v.getEmail()) && r.isPermission()))
        relatives.add(v);
});
```

Рисунок 3 - Выбор членов группы, которые разрешили отслеживать свое местоположение

Просмотр местоположения членов группы. Представляет собой отображения членов группы по локациям. Тем самым мы можем наглядно просмотреть последнее местоположение всех участников группы в удобном виде с отображением названия локации, где находится каждый член группы и временем, когда он там находился. Отображаются только созданные локации и одна системная локация «Вне локаций». Так же есть возможность просмотра трека члена группы за период, где на карте отображается путь члена группы за промежуток времени. Отображаются только те члены группы, которые дали согласие на отслеживание.

Обработка запросов к серверу

Взаимодействие между клиентом и сервером во время авторизации - выполняется отправка данных на сервер (1), а от сервера возвращаются данные (2), которые получает клиент (см. рис. 4). На сервер отправляются такие данные как логин и пароль. В данном случае очень важно обезопасить передачу информации на сервер, так как пароль могут перехватить. Поэтому рекомендуется использовать SSL сертификат на удаленном сервере, а в приложении переключиться на HTTPS протокол передачи данных. Это позволит шифровать данные во время передачи. В данной системе имеются функции передачи данных по протоколу HTTP, но это не безопасно. После проверки учетных данных на сервере клиенту возвращаются данные о текущем аккаунте.

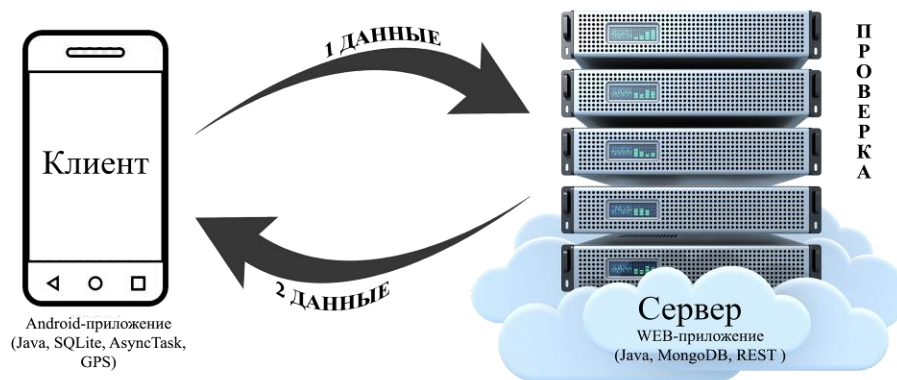


Рисунок 4 - Схема взаимодействия при авторизации пользователя в мобильном приложении

Взаимодействие между клиентом и сервером при операциях CRUD (создание/чтение/изменение/удаление). Система отправляет следующие данные на сервер: CRUD групп, CRUD членов групп, CRUD локаций, изменение запросов на разрешение и отправка GPS-координат (см. рис.5). Особенностью данной архитектуры, является, то что при отсутствии Интернета данные хранятся на мобильном устройстве и потом синхронизируются с сервером при появлении сети. Во время некоторых операций может возникнуть частный случай, когда одни и те же данные редактируются с двух или более мобильных устройств одновременно. Решение такой ситуации – верными считаются последние занесенные данные. Такая коллизия может возникнуть если использовался один тот же аккаунт на нескольких устройствах.

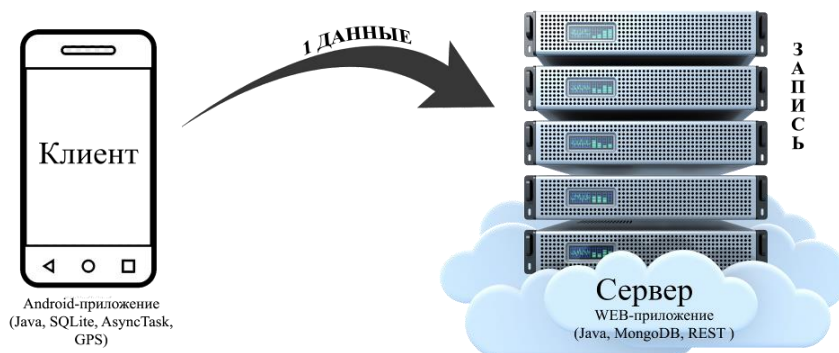


Рисунок 5 - Схема взаимодействия при записи данных на сервер

Получение данных о местоположении членов групп. Клиент может запросить данные о местоположении членов группы, а также данные о пути следования члена группы. Эта информация отображается в удобном для пользователя виде в мобильном приложении (см. рис. 6).

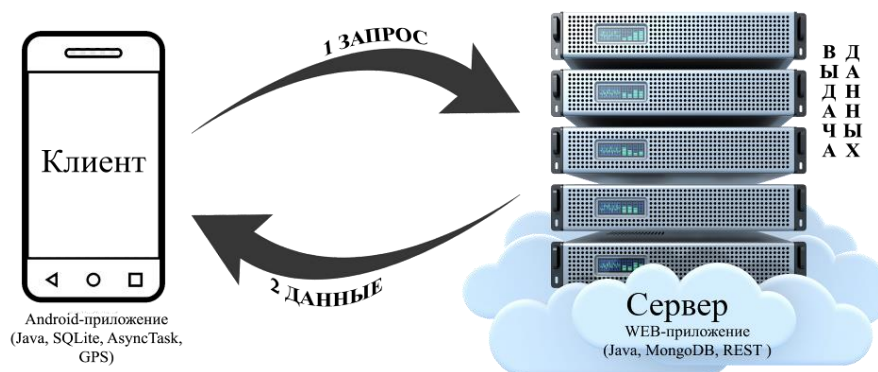


Рисунок 6 - Схема взаимодействия при выборке данных с сервера

Реализация данного функционала заключается в выборке соответствующих членов группы, с условием что они дали разрешение на отслеживание своего местоположения, и проверки нахождения члена группы в

соответствующей локации (см. рис. 7). Результаты этой выборки заполняются в соответствующую структуру данных `List<RelativeInLocation>` и передаются на мобильное устройство клиента в виде json-файла. Android-приложение принимает эти данные и отображает их в удобном для пользователя виде.

```

    relatives.forEach(r -> {
        Position position = getLastPositionByEmail(r.getEmail());
        if (position != null && position.getId() != null) {
            account.getLocations().forEach(v -> {
                if (isInside(v.getPoints(), new Point(position.getLat(), position.getLon()))) {
                    RelativeInLocation inLocation = new RelativeInLocation();
                    inLocation.setLocationId(v.getId());
                    inLocation.setLocationName(v.getName());
                    inLocation.setRelativeId(r.getId());
                    inLocation.setRelativeName(r.getName());
                    inLocation.setLat(position.getLat());
                    inLocation.setLon(position.getLon());
                    inLocation.setModified(position.getModified());
                    list.add(inLocation);
                }
            });
        }
    });

    public Position getLastPositionByEmail(String email) {
        Query query = new Query();
        query.addCriteria(Criteria.where("accountEmail").is(email));
        query.with(Sort.by(Sort.Direction.DESC, ...properties: "modified"));
        return mongoTemplate.findOne(query, Position.class);
    }
}

```

Рисунок 7 – Код выборки местоположения членов группы

Рассмотрим схему взаимодействия сущностей в сервисе (см. рис. 8). Основой всего является аккаунт пользователя (Account). На мобильном устройстве хранится информация только об одном текущем пользователе, а на сервере данные о всех аккаунтах, зарегистрированных в системе. Аккаунт включает в себя такие данные о пользователе как логин, пароль, группы, всех членов группы, локация и дату последнего изменения текущего аккаунта. Эта дата может быть использована для синхронизации между разными мобильными устройствами, подключенными к одному и тому же аккаунту.

Любая группа (Family), входящая в аккаунт, представляет собой список членов группы (Relative), объединённых общим названием. В свою очередь член группы – это краткие данные об аккаунте, зарегистрированном в системе, только email и название. Таким образом, один и тот же email может быть добавлен как член группы в любое множество разных аккаунтов. При этом списки членов группы в аккаунте и группах аккаунта отличаются, где списки членов групп (`List<Relative>` в Family) в группах аккаунта, являются подмножеством списка членов групп аккаунта (`List<Relative>` в Account).

Локация (Location) представляет собой многоугольник, представленный в виде точек его вершин (Point). В мобильном приложении этот многоугольник создается и отображается на карте, тем самым формализуя конкретную географическую область. Вершина многоугольника — это GPS-координаты представленные в виде широты и долготы в градусах в виде десятичной дроби.

Запросы на разрешение отслеживания членов групп (Requestme) хранятся как совокупные данные об аккаунте запросившим разрешении (idWho, emailWho), аккаунте к которому пришел запрос (emailRequest) и непосредственно ответе на это запрос (boolean permission).

GPS данные (Position) о местоположении мобильного устройства хранится в структуре определенного вида и включает в себя ссылку на аккаунт, широту, долготу, скорость передвижения и дату определения данных. Все эти данные создают полную картину о местонахождении мобильного устройства в момент времени.

На основании данных о группах аккаунта, локациях аккаунта и GPS данных других аккаунтов формируются данные о местоположении членов группы (RelativeInLocation). Именно эти выбранные данные используются в мобильном приложении для отображения расположения членов групп. Для определения вхождения члена группы в локацию на серверной части сервиса была решена задача определения вхождения точки в полигон.

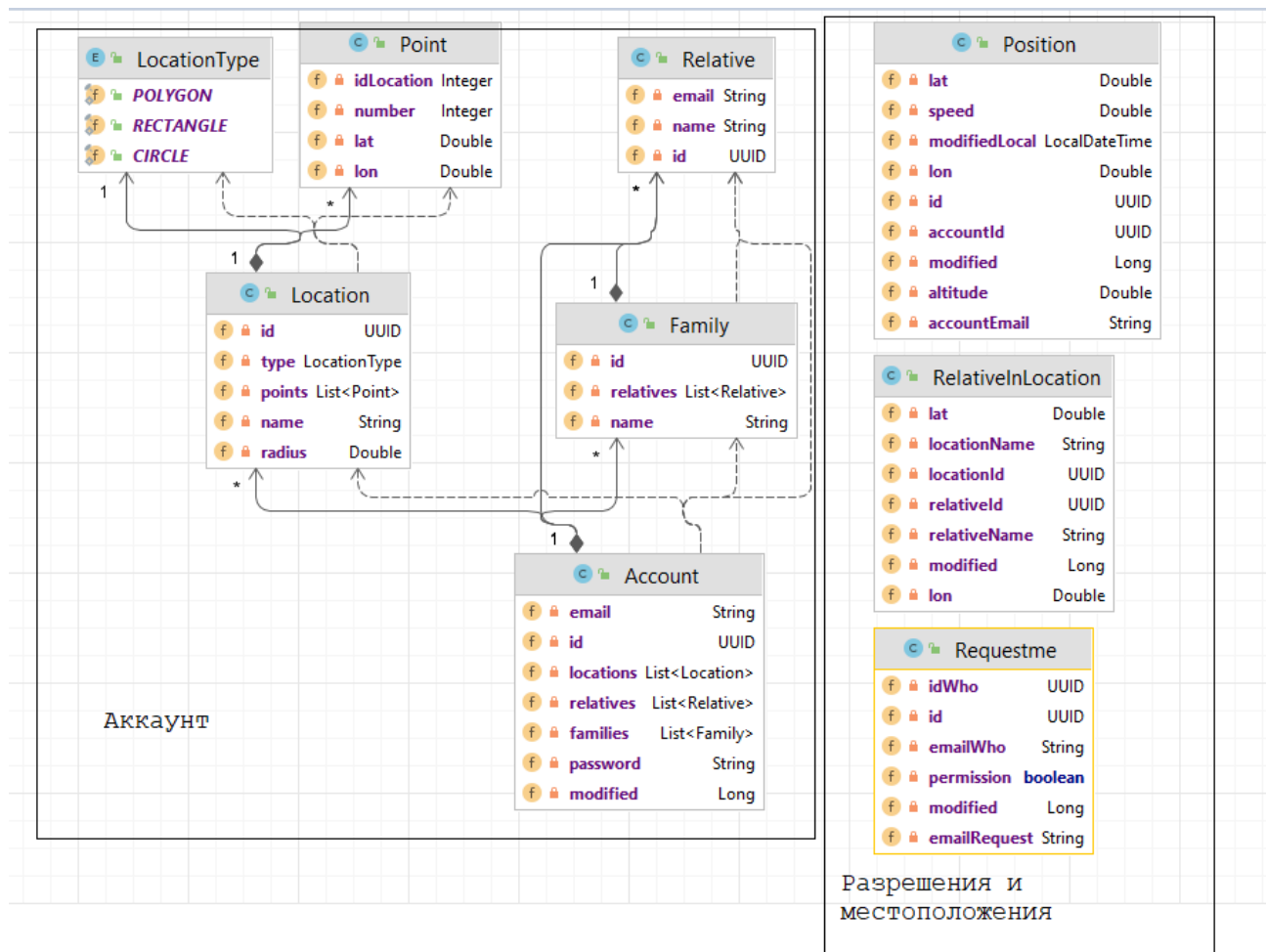


Рисунок 8 – Схема взаимодействия сущностей

Особенности реализации

Рассмотрим технологии используемые для создания сервиса:

- использование Android SDK с minSdk 24 (Android 7);
- для реализации интерфейса используется 'com.google.android.material';
- функционал работы с сущностями данного сервиса реализован на Java;
- реализована обработка асинхронных запросов между клиентской и серверной частью сервиса (для общения с БД на сервере и БД на мобильном устройстве используется запросы по протоколам HTTPS или HTTP; БД на сервере - MongoDB БД, а на мобильном устройстве – SQLite).
- для отображения карт в мобильном приложении используется 'org.osmdroid' [10];
- реализован сервис для обнаружения текущего местоположения мобильного устройства (координаты GPS) используя 'com.google.android.gms';
- реализованы функционал для работы с БД SQLite – создание, удаление, редактирование таблиц и записей;
- реализован алгоритм безопасной авторизации с использованием SSL сертификата.

Выводы

Достоинством данного сервиса является:

- открытость;
- возможность его установки на любую версию Андроида выше 7;
- возможность как подключиться к общедоступному Web-серверу, так и установить серверную часть на свой частный сервер;

- возможность работы приложения в фоновом режиме, что убирает необходимость держать приложение постоянно активным;
- возможность настройки частоты синхронизации данных между мобильным устройством и сервером, что влияет на расход заряда мобильного устройства;
- возможность использовать один аккаунт на нескольких устройствах;
- сохранность всех веденных данных даже при потере интернет соединения.

Дальнейшая перспектива развития данного сервиса состоит в возможности создания более устойчивого соединения между приложением и сервером (переход на WebSocket) и увеличение интеграции с другими сервисами типа социальных сетей и общедоступных сервисов карт (Google Maps, Яндекс Карта и др.).

Список литературы

1. Программа Life360: обзор семейного локатора [Электронный ресурс] / Интернет-ресурс. - Режим доступа: [www/URL: https://gpscool.ru/otslezhivanie-mestopolozheniya/programma-life360-obzor-semejnego-lokatora](http://www.URL: https://gpscool.ru/otslezhivanie-mestopolozheniya/programma-life360-obzor-semejnego-lokatora) - Загл. с экрана.
2. Family Locator – GPS-трекер и бесплатные сообщения для всей семьи [Электронный ресурс] / Интернет-ресурс. - Режим доступа: [www/URL: https://appleinsider.ru/obzory-prilozhenij/app-store-family-locator-gps-treker-i-besplatnye-soobshheniya-dlya-vsej-semi.html](http://www.URL: https://appleinsider.ru/obzory-prilozhenij/app-store-family-locator-gps-treker-i-besplatnye-soobshheniya-dlya-vsej-semi.html) - Загл. с экрана.
3. Обзор приложений для контроля местонахождения членов семьи [Электронный ресурс] / Интернет-ресурс. - Режим доступа: [www/URL: https://gpscool.ru/sistemy-gps-slezheniya/obzor-prilozhenij-dlya-kontrolya-mestonahozhdeniya-chlenov-semi](http://www.URL: https://gpscool.ru/sistemy-gps-slezheniya/obzor-prilozhenij-dlya-kontrolya-mestonahozhdeniya-chlenov-semi) - Загл. с экрана.
4. Glympse — безопасный и простой способ поделиться своим местоположением с помощью мобильного устройства [Электронный ресурс] / Интернет-ресурс. - Режим доступа: [https://gadgetshelp.com/internet/glympse-bezopasnyi-i-prostoi-sposob-podelitsia-svoim-mestopolozheniem-s-pomoshchiu-mobilnogo-ustroistva/](http://www.URL: https://gadgetshelp.com/internet/glympse-bezopasnyi-i-prostoi-sposob-podelitsia-svoim-mestopolozheniem-s-pomoshchiu-mobilnogo-ustroistva/) - Загл. с экрана.
5. What Is SQLite? [Electronic resource] / Интернет-ресурс. - Режим доступа: [www/URL: https://www.sqlite.org/index.html](http://www.URL: https://www.sqlite.org/index.html) - Загл. с экрана.
6. Android Mobile App Developer Tools – Android Developers [Electronic resource] / Интернет-ресурс. - Режим доступа: [www/URL: https://developer.android.com/](http://www.URL: https://developer.android.com/) - Загл. с экрана.
7. Google Play services | Android Developers [Electronic resource] / Интернет-ресурс. - Режим доступа: [www/URL: https://www.android.com/gms/?authuser=0&hl=ru_RU](http://www.URL: https://www.android.com/gms/?authuser=0&hl=ru_RU) - Загл. с экрана.
8. Family Location [Электронный ресурс] / Интернет-ресурс. - Режим доступа: [www/URL: https://bitbucket.org/Rolog/family-location/src/master/](http://www.URL: https://bitbucket.org/Rolog/family-location/src/master/) - Загл. с экрана.
9. Family Location Server [Электронный ресурс] / Интернет-ресурс. - Режим доступа: [www/URL: https://bitbucket.org/Rolog/family-location-server/src/master/](http://www.URL: https://bitbucket.org/Rolog/family-location-server/src/master/) - Загл. с экрана.
10. Osmdroid [Electronic resource] / Интернет-ресурс. - Режим доступа: [www/URL: https://osmdroid.github.io/osmdroid/Home.html](http://www.URL: https://osmdroid.github.io/osmdroid/Home.html) - Загл. с экрана.

Морозова О.В. Разработка сервиса определения местоположения членов группы с доверительными отношениями. В статье рассмотрены основные возможности сервиса, общая схема взаимодействия устройств и структура приложения. Рассмотрены основные технологии применяемые при разработке приложения. Описаны основные особенности работы приложения.

Ключевые слова: Android-приложение, SQLite, Google Mobile Services, Android SDK, CRUD, GPS.

Morozova O.V. Development of a service for determining the location of group members with trust relationships. The article discusses the main features of the service, the general scheme of device interaction and the structure of the application. The main technologies used in the development of the application are considered. The main features of the application are described.

Keywords: Android application, SQLite, Google Mobile Services, Android SDK, CRUD, GPS.

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

УДК 004.42

Повышение эффективности протоколов передачи данных, использующих криптографическую защиту

В.В. Бондаренко^{*1}, А.В. Чернышова^{*2}

^{*1} магистрант ст. преподаватель кафедры программной инженерии им. Л.П. Фельдмана,
Донецкий национальный технический университет,
vadimbond.2000@gmail.com

^{*2} ст. преподаватель ст. преподаватель кафедры программной инженерии им. Л.П. Фельдмана,
Донецкий национальный технический университет,
chernyshova.alla@rambler.ru, SPIN-код: 3318-2066

Бондаренко В.В., Чернышова А.В. Повышение эффективности протоколов передачи данных, использующих криптографическую защиту. Статья посвящена актуальной проблеме — современным тенденциям в области протоколов передачи данных с криптографической защитой. В рамках статьи выявлены достоинства и недостатки современных протоколов передачи данных с криптографической защитой, произведен поиск способов их устранения при проектировании авторского протокола передачи данных с криптографической защитой для приложений определенного класса. Сформулированы требования к протоколу передачи данных с криптографической защитой.

Ключевые слова: VPN, протокол, атака, шифрование, аутентификация, PPTP, L2TP, IPSec, SSL.

Введение

В течение длительного периода времени люди нуждались в засекречивании какой-либо информации. Из-за такой необходимости в защите информации выросла целая наука - криптография. Ранее криптография имела огромную значимость для правительства государств, но с возникновением Интернета алгоритмами информационной безопасности стали интересоваться и другие категории граждан с целью защиты своей информации, коммерческой или персональной [1].

Сегодня пользователи, работающие в интернете, задают вопросы «Насколько надежна их переписка при использовании почтовых сервисов?», «Насколько безопасны социальные сети?», «Можно ли безопасно хранить данные в облачном хранилище?». Работая удаленно, сотрудники компаний нуждаются в средствах защиты передаваемой информации при использовании ресурсов фирмы. Используя электронный документооборот, предприятие, учреждение, коммерческая фирма также использует различные способы защиты передаваемой и хранимой информации на основе существующих классических решений - безопасная аутентификация, использование защищенных протоколов передачи данных, работающих на различных уровнях относительно стека протоколов TCP/IP, применение сервисов защищенного хранилища данных. Но тем не менее, при предлагаемых разнообразных решениях по обеспечению сохранности информации, коммерческих и свободно распространяемых продуктов, направленных на защиту передаваемых по сети данных, задачи по повышению эффективности протоколов передачи данных, использующих криптографическую защиту, являются, безусловно, актуальными. Широко используемые протоколы с криптографической защитой имеют ряд достоинств. Но присутствуют и недостатки. Выполнив детальный анализ существующих на текущий момент решений на рынке протоколов с криптографической защитой, можно прийти к выводу о том, что можно изменить в архитектуре или алгоритмах реализации существующих протоколов, чтобы повысить эффективность протокола и адаптировать его, например, к определенному классу приложений, выбрав при этом утвержденные стандарты шифрования.

Анализ современных тенденций в области протоколов передачи данных с криптографической защитой

В последнее время множеству сотрудников пришлось перейти на формат удалённой деятельности, и многие наниматели рекомендовали своим работникам задействовать VPN для безопасного доступа к внутрикорпоративным сервисам.

VPN - это набор технологий, которые позволяют создавать логическую структуру на поверхности физической. Он используется для обеспечения безопасности трафика от перехвата преступником и защиты действий в сети интернет обычных пользователей.

VPN необходим для организации удаленной работы. Работая из дома, используя VPN-сервис, есть возможность получить онлайн-доступ к документации и сервисам фирмы или организации, при этом соединение будет гораздо защищённым, данные будут нелегко перехватить и расшифровать.

VPN также необходим для объединения разных офисов одной и той же компании. Офисы могут быть отдалены друг от друга на большое расстояние.

При подключении к Wi-Fi в кафе, на станциях метро и т.д., чтобы злоумышленники не могли украсть личные данные пользователя, также можно воспользоваться сервисом VPN. При использовании общедоступной сети использование VPN-сервиса делает работу с веб-сайтами относительно безопасной.

Использование социальных сетей в публичных зонах Wi-fi без сервиса VPN может привести не только к перехвату конфиденциальной информации, но и в дальнейшем есть угроза использования данных аккаунтов в своих целях, пройдя процедуру аутентификации от имени пользователя. Утеря информации для доступа к почтовым сервисам, при незащищенной аутентификации на почтовых серверах, приведет к еще более серьёзным последствиям, так как к почтовым аккаунтам могут быть привязаны другие аккаунты web-приложений, в том числе и сервисов, используемых на предприятиях, в учреждениях, на фирмах, а также профили социальных сетей и другие почтовые аккаунты. Тогда все приложения, связанные с этим почтовым ящиком, могут быть атакованы. Но самым неприятным может быть утечка данных банковских карт, если пользователь решил оформить перевод денег кому-нибудь, подключившись к общедоступному Wi-Fi.

VPN может использоваться также для обеспечения анонимности пользователя при работе в сети Интернет. При использовании VPN затруднительно определить, какие сайты посещает пользователь, каким браузером пользуется, где находится территориально и т.д. Необходимость скрыть своё местоположение может потребоваться в путешествиях при использовании различных интернет-приложений.

Рассмотрим принцип работы VPN.

VPN-соединение - это можно сказать «туннель» между пользователем и сервером. Каждый узел кодирует данные перед тем, как они попадут в «туннель».

Подключившись к VPN, система идентифицирует сеть и начинает распознавание (сопоставляет введенный пароль с паролем в своей базы данных).

Далее сервер авторизует пользователя, то есть обеспечивает право на осуществление поставленных действий: чтение интернет-почты, серфинг в Интернете и т.д. После формирования соединения весь трафик передается между компьютером пользователя и сервером в зашифрованном виде. Компьютер пользователя располагает IP-адресом, выданным интернет-провайдером. VPN-сервер изменяет IP-адрес компьютера пользователя на свой собственный. И уже с VPN-сервера вся информация передаётся на внешние ресурсы, запрашиваемые пользователем. Теперь можно получить доступ к ресурсам относительно надежным способом.

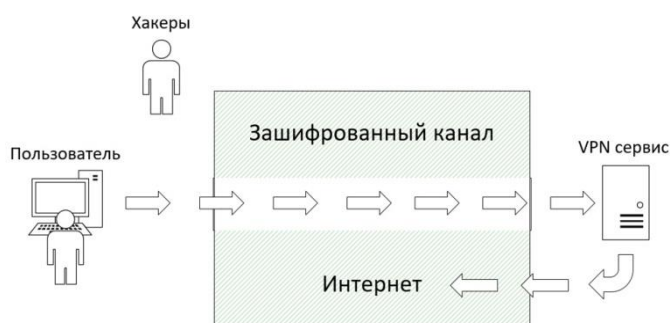


Рисунок 1 – Как работает VPN

Тем не менее, следует учитывать, что не вся информация зашифрована. Разные провайдеры VPN могут иметь разные характеристики, к примеру степень шифрования, хранение логов (журнал, в котором хранится информация о посещенных сайтах, сокрытие факта подключения к серверу, реальный IP-адрес и т.д.) и сотрудничество в выдаче информации третьим лицам.

Если VPN-провайдер вообще не записывает логи, то передавать третьим лицам просто нечего. А сокрытие факта подключения к серверу - это уже редкая услуга. Если соединение установлено неправильно или соединение

внезапно отключено, с частью данных может произойти утечка. Технология Multihop VPN поможет решить проблему, которая предполагает подключение к сайту сразу через несколько серверов.

Рассмотрим популярные протоколы VPN.

PPTP – Point-to-Point Tunneling Protocol [2], использует два соединения – одно для управления, другое – для инкапсуляции данных, поддерживается всеми операционными системами, не требует много вычислительных мощностей, но обладает плохой защищенностью. Используемые методы шифрования этого протокола устарели, есть ошибки в реализации протокола от Microsoft. Нет шифрования по умолчанию, на взлом требуется менее суток. Протокол считается менее защищенным, чем IPSec. Используется, когда защита данных не очень важна или когда нет других вариантов.

PPTP анализировали с точки зрения безопасности. В протоколе были обнаружены уязвимости, связанные с используемым протоколом аутентификации PPP, протоколом MPPE и интеграции между аутентификациями MPPE и PPP для установки сессионного ключа. Уязвимости PPTP:

- MSCHAP-v1 ненадежен;
- MSCHAP-v2 уязвим для словарной атаки;
- при использовании MSCHAP-v1, MPPE использует одинаковый RC4 сессионный ключ для шифрования информационного потока в обоих направлениях;
- MPPE использует RC4 поток для шифрования. Отсутствует метод для аутентификации цифрового потока и поэтому этот поток уязвим для атаки, делающей фальсификацию битов. [3]

L2TP – Layer 2 Tunneling Protocol - Протокол L2TP расширяет модель PPP, позволяя размещение терминальных точек L2 и PPP в различных физических устройствах, подключенных к сети с коммутацией пакетов [4]. Рассматриваемый протокол более эффективен для построения виртуальных сетей, но более требователен к вычислительным ресурсам, стандартно не включает шифрование. Он работает вместе с другими протоколами, чаще всего с IPSec. Используется интернет-провайдерами и пользователями в корпоративных целях.

IPSec – Internet Protocol Security [5] - ряд протоколов и стандартов для безопасных подключений. Достоинствами IPSec являются хорошая архитектура и надежность алгоритмов. Недостатками IPSec являются сложность в настройке, из чего следует понижение защищенности при неверной настройке; для этого требуется большое количество вычислительных возможностей, но данный недостаток компенсируется аппаратным ускорением алгоритма шифрования AES. Он часто используется в сочетании с другими методами.

SSL – Secure Sockets Layer & TLS – Безопасность транспортного уровня - это группа методов, которая включает протоколы TLS, SSL и другие средства защиты. Преимущество протокола заключается в том, что он свободно передается большинством общедоступных сетей. Недостатком протокола является довольно низкая производительность, сложность в настройке и необходимость установки дополнительного программного обеспечения. Протокол SSL используется на веб-сайтах, URL-адреса которых начинаются с https.

Некоторые реализации VPN: OpenVPN [6], Microsoft SSTP [7]. OpenVPN имеет открытый исходный код, реализован практически для каждой платформы и является достаточно безопасным. Он выполняет функцию безопасного туннеля для трансляции информации через один TCP/UDP порт. Главное преимущество OpenVPN состоит в легкости инсталляции и настройки, что является редким случаем для таких инструментов. OpenVPN соединяет клиент/серверную архитектуру. Он должен быть поставлен на все узлы VPN сети, там, где один узел должен быть сервером, а другие - клиентами. VPN шлюзы объединяют один и тот же ключ для дешифрования и шифрования информации при размещении статических ключей. В этом случае изменение настройки будет простым, но при этом появится затрудненность передачи и безопасности ключа. Если кто-то завладеет этим ключом - он сможет декодировать данные.

Для того чтобы это избежать, следует применить инфраструктуру открытых ключей (PKI). При этом каждый узел управляет двумя ключами: закрытый ключ доступный только его хозяину и открытый ключ, известный всем. Данную подсистему использует OpenSSL, соединенный в OpenVPN, для аутентификации VPN узлов до того, как начать передачу закодированных данных.

Microsoft SSTP – протокол туннелирования защищенных сокетов SCTP по определению является протоколом прикладного уровня. Он предназначен для синхронного обмена данными между двумя программами и позволяет использовать несколько конечных точек приложения через одно сетевое соединение между одноранговыми узлами. Это позволяет эффективно использовать коммуникационные ресурсы, доступные в сети.

Протокол SSTP основан на SSL вместо PPTP или IPSec и использует TCP-порт 443 (или другие, как реализовано в Icof UTM) для ретрансляции трафика SSTP. Хотя он тесно связан с SSL, прямое сравнение между SSL и SMTP невозможно, поскольку, в отличие от SSL, SSTP - это просто протокол туннелирования. Есть много причин выбрать SSL, а не IPSec в качестве основы для SSTP. IPSec нацелен на поддержку VPN-соединения, поэтому SSL стал лучшей основой для разработки STP, поскольку он поддерживает роуминг.

Более подробно о существующих протоколах передачи данных с криптографической защитой можно ознакомиться в статье [8].

Выявление недостатков и уязвимостей существующих алгоритмов защиты в сетях передачи данных

Защищенные свойства протоколов могут быть объявлены, когда они объявлены самими авторами протокола (и, как правило, влекут всевозможные доводы в пользу осуществления этих функций), и подразумевается, когда авторы какой-либо структуры рассчитывают на разработку защищенных свойств каким-либо протоколом.

Атака на защищенный протокол - это попытка проанализировать протокольные сообщения и/или выполнить действия, не предусмотренные протоколом, с целью разрушения предъявленных или обуславливаемых свойств протокола.

Атака считается удачной, если нарушено хотя бы одно из предъявленных или подразумеваемых свойств протокола.

В случае удачной атаки на обуславливаемые свойства мы уточним, что это успешная атака на применение протокола в какой-либо подсистеме. Это будет объяснять, конечно, не о недостатках самого протокола, а о неправильном выборе протокола (настроек) составителями системы.

Есть огромное число атак на протоколы. Многие атаки имеют общие критерии, что даёт возможность использовать классы атак для облегчения мониторинга и разработки протоколов, более устойчивых к большому количеству классов атак.

MitM «Атака посередине» [9] (man-in-the-middle attack). Класс атак, при которых преступник повторяет и подменяет текст сообщений, проходящих между двумя или более пользователями протокола, а пользователи не догадываются о том, что переписка «прослушивается» считая, что они общаются лично друг с другом без участия третьего лица. Протоколы, которые не осуществляют обоюдную аутентификацию сторон, беззащитны для этой атаки (цель G1). Традиционный пример атаки такого класса – атака на протокол Диффи—Хеллмана.

Повторная атака с повторной передачей [9]. Класс атак, при которых преступник фиксирует все письма, проходящие в одном сеансе протокола, а затем дублирует их в новом, под видом одного из участников первого сеанса. Наглядными примерами протоколов, к которым применима эта атака, являются протоколы Wu-Lan и протокол Shamir keyless protocol.

Атака подмены типа TF [9]. Класс атак, при которых злоумышленник, используя сообщение, переданное в сеансе локального протокола, создает уникальное, передавая его на другом раунде протокола под образом сообщения другого типа (с другой целью). Например, протоколы Wide-Mouth Frog, Denning—Sako, Yahalom и Otway—Riis уязвимы для таких атак.

PS Атака параллельного сеанса [10] - это атака, при которой преступник инициирует некоторое количество параллельных сеансов протокола, чтобы использовать сообщения из одного сеанса в другом. Наглядным примером протокола, восприимчивого для этого класса атак, считается симметричный вариант Нидхэма—Шредера.

Атака STS с известным одноразовым ключом [10] – это атака на протокол повторной передачи, организованная на знании одноразового секретного ключа.

Атака KN с использованием известного сеансового ключа. [10]

Классы атак, при которых злоумышленник получает доступ к временным секретным ключам, используемым в протоколах (например, к актуальным сеансовым ключам), после чего он может гарантировать, например, аутентификацию или, по крайней мере, установление сеанса от имени одной из сторон протокола.

Атака UKS с неизвестным сеансовым ключом [10] – это атака, состоящая в попытке получить информацию о долгосрочном ключе или любую другую ключевую информацию, которая позволяет создавать вновь сеансовые ключи для других сеансов протокола.

Класс атак на протоколы аутентификации ключей, при которых злоумышленник получает возможность доказать одной из сторон факт владение ключом (например, путем повторения сообщения из законного сеанса), хотя сам ключ ему был не известен. Например, симметричный протокол Нидхэма—Шредера уязвим для этого класса атак.

Следует подчеркнуть, что, если не указано иное, при анализе криптографического протокола (не конкретных систем) используется предположение о долговечности всех используемых криптографических примитивов. Например, предполагается, что, хотя и существует безопасный обмен информацией с использованием сессионного ключа, разработанного в сеансе некоторого криптографического протокола, у злоумышленника не будет достаточно ресурсов и времени для получения этого сессионного ключа путем атаки на используемые шифры или криптографически стабильные хэш-функции.

С другой стороны, следует предположить, что сеансовые ключи, полученные в рамках сеансов протокола, будут получены преступником через некоторое время (однако, намного дольше, чем время самого сеанса связи) (классы атак STS и KN). И по прошествии гораздо более длительного времени преступник сможет получить доступ к «мастер-ключам» — ключам долгосрочного использования, так что следует разработать протоколы с генерацией сеансовых ключей, в том числе со свойством G9/PFS.

Поиск возможных способов устранения найденных недостатков в алгоритмах протоколов передачи данных (прикладной уровень)

Одним из первых алгоритмов назначения открытого ключа был предложен Хеллманом М. Э. и Диффи У. в 1976 году. [11] Для его реализации участники должны договориться о большом значении простого числа p и обуславливаемого компонента α мультипликативной группы $Z * p = \{1, 2, \dots, p-1\}$. Чтобы сгенерировать общий ключ k , они должны сформировать некоторые числа x , $1 \leq x \leq p-2$ и y , $1 \leq y \leq p-2$ соответственно. После этого они могут начать общение посредством сообщений. Протокол ДН:

$$\begin{aligned} A \rightarrow B &: \alpha^x \bmod p, \\ A \leftarrow B &: \alpha^y \bmod p. \end{aligned}$$

Рисунок 2 – Пример обмена сообщениями

Искомый общий ключ теперь вычисляется по формуле

$$k = (\alpha^y)^x = (\alpha^x)^y \bmod p.$$

Рисунок 3 – Искомый общий ключ

Слабость: отсутствие аутентификации сторон.

Протокол МТИ. Интересный подход к защите протокола ДН от атаки «противник в середине» был предложен Т. Мацумото, И. Такашима и Х. Имаи (Matsumoto T., Takashima Y., Imai H.) в 1986 г. Они предложили серию протоколов, предполагающих наличие у абонентов открытых ключей и использующих различные модификации процедуры выработки общего ключа.

Рассмотрим протокол МТИ/A0. Предположим, что участники A и B имеют секретные ключи a , $1 \leq a \leq p-2$, и b , $1 \leq b \leq p-2$, соответственно и публикуют свои открытые ключи $\beta_A = \alpha^a \bmod p$ и $\beta_B = \alpha^b \bmod p$. Для выработки общего секретного ключа k они должны сгенерировать случайные числа x , $1 \leq x \leq p-2$, и y , $1 \leq y \leq p-2$, соответственно, а затем обменяться следующими сообщениями:

$$\begin{aligned} A \rightarrow B &: \alpha^x \bmod p, \\ A \leftarrow B &: \alpha^y \bmod p. \end{aligned}$$

Рисунок 4 – Пример обмена сообщениями

Теперь участники A и B вычисляют общий ключ $k = \alpha^{(xb+ya)} \bmod p$ по формулам.

$$\begin{aligned} k_A &= (\alpha^y)^a \beta_B \bmod p, \\ k_B &= (\alpha^x)^b \beta_A \bmod p \end{aligned}$$

Рисунок 5 – Вычисление общего ключа

Любая подмена сообщений приводит к тому, что все стороны получают различные значения ключа, в результате чего становится невозможным чтение всей передаваемой информации. Тем самым свойство аутентификации ключа протокола при атаке «противник в середине» не нарушено. Вместе с тем этот протокол не обеспечивает аутентификации сторон и подтверждения правильности получения ключа.

Протокол STS. Попытка построения аутентифицированного протокола на базе ДН была предпринята в протоколе STS (station-to-station), созданном У. Диффи (W. Diffie), П. Ван Ооршотом (P. Van Oorschot) и М. Вейнером (M. Wiener) в 1992 г.

$$\begin{aligned} A \rightarrow B &: A, B, m_A = \alpha^x \bmod p, \\ A \leftarrow B &: B, A, m_B = \alpha^y \bmod p, E_k(\text{Sig}_B(m_B, m_A)), \\ A \rightarrow B &: A, B, E_k(\text{Sig}_A(m_A, m_B)). \end{aligned}$$

Рисунок 6 – Пример обмена сообщениями

Здесь Sig_A и Sig_B — цифровые подписи пользователей A и B соответственно, $k = \alpha^{xy} \bmod p$ — искомый общий ключ.

Вставка во второе и третье сообщения протокола значений цифровых подписей позволяет гарантировать достоверность получения сообщения именно от того пользователя, от которого это сообщение получено.

Шифрование значений подписей пользователей с помощью симметричного алгоритма E введено для того, чтобы обеспечить взаимное подтверждение правильности вычисления значения ключа, так как при неверно вычисленном ключе невозможно получить верные значения цифровых подписей.

Модифицированный протокол STS. В 2004 г. К. Бойд и А. Матура предложили следующую модификацию протокола STS:

$$\begin{aligned}
 A \rightarrow C(B) : & \quad A, B, m_A, \\
 C \rightarrow B : & \quad C, B, m_A, \\
 C \leftarrow B : & \quad B, C, m_B, E_k(\text{Sig}_B(m_B, m_A)), \\
 A \leftarrow C(B) : & \quad B, A, m_B, E_k(\text{Sig}_B(m_B, m_A)), \\
 A \rightarrow C(B) : & \quad A, B, E_k(\text{Sig}_A(m_A, m_B)).
 \end{aligned}$$

Рисунок 7 – Пример обмена сообщениями по протоколу STS

где $k_0 = f(k)$ — ключевой параметр хеш-функции, вычисляемый как значение некоторой функции от результирующего сеансового ключа $k = \alpha x \bmod p$.

Атака: для данной версии протокола можно применить двустороннюю атаку с неизвестным общим ключом (bilateral unknown key-share attack, BUKS attack):

$$\begin{aligned}
 A \rightarrow C : & \quad A, B, m_A, \\
 C \rightarrow D : & \quad A, B, m_A, \\
 D \rightarrow B : & \quad D, B, m_A, \\
 D \leftarrow B : & \quad B, D, m_B, \text{Sig}_B(m_B, m_A), h_{k_0}(m_B, m_A), \\
 C \leftarrow D : & \quad B, A, m_B, h_{k_0}(m_B, m_A) \\
 A \leftarrow C : & \quad C, A, m_B, \text{Sig}_C(m_B, m_A), h_{k_0}(m_B, m_A), \\
 A \rightarrow C : & \quad A, C, \text{Sig}_A(m_A, m_B), h_{k_0}(m_A, m_B), \\
 C \rightarrow D : & \quad A, B, h_{k_0}(m_A, m_B), \\
 D \rightarrow B : & \quad D, B, \text{Sig}_D(m_A, m_B), h_{k_0}(m_A, m_B).
 \end{aligned}$$

Рисунок 8 – Алгоритм атаки

Нарушитель С, используя свой законный обмен с участником В, может применить против участника А атаку, в результате которой он может убедить А в том, что он выступает от имени В. Нарушитель С использует свой законный обмен с участником В и убеждает А в том, что он выступает от имени В. Сеанс с участником В остается незавершенным, так как С, не зная секретного ключа участника А, не сможет подобрать правильный ответ для В. Поэтому любое его сообщение на третьем шаге будет участником В отвергнуто.

Данная атака не представляет реальной опасности, так как при этом нарушитель не будет знать секретного ключа $k = \alpha x \bmod p$ и поэтому не сможет читать передаваемые сообщения, передаваемые от А к В. Однако в результате участник А не будет ничего подозревать и примет участника С за В.

Для повышения защищенности можно расширить протокол за счет использования двухфакторной аутентификации при вводе пароля пользователя и при вводе ключа разблокировки личности, например, через отправку кода подтверждения на e-mail адрес. А также за счет введения в реализацию клиента правил, не позволяющих генерировать простой пароль. При этом e-mail адреса для аутентификации на ресурсе могут храниться на сервере ресурса и задаваться при регистрации на конкретном ресурсе. А e-mail, используемый для разблокировки личности, может храниться в зашифрованном виде вместе с кодом разблокировки, чтобы исключить возможность подмены e-mail адреса на адрес злоумышленника. В таком случае e-mail, необходимый для активации кода разблокировки личности, может задаваться при первом запуске приложения. Оба типа e-mail адресов могут быть изменены при активации кода разблокировки личности. [12]

Также можно генерировать QR-код, хеш которого будет зашифрован и передаваться пользователю, у которого на другом устройстве будет ключ для расшифровки картинку, а из этой картинку будет формироваться ключ для шифрования информации между двумя пользователями. [13]

Формирование требований к протоколу передачи данных с криптографической защитой

Существует достаточно большое количество различных типов атак, которые зависят от конкретной реализации протокола.

Например, для криптографического протокола, основанного на системах симметричного шифрования, можно использовать особенности самих систем шифрования и, в частности, реализованные методы и режимы шифрования, синхронизации и т.д.

Для защиты от таких атак необходимо проанализировать архитектуру протокола и структуру передаваемых сообщений для выявления возможных уязвимостей, которые позволяют навязывать сообщения с известными или идентичными значениями определенных полей, или путем подмены типа различных полей. Для криптографических протоколов, организованных на основании систем асимметричного шифрования, недостатком является замена открытого ключа одного из партнёров другим открытым ключом, при этом секретная половина этого ключа известна противнику. В частности, это позволяет противнику узнать содержание зашифрованных сообщений, отправленных этому участнику. В этом случае свойство связности открытого ключа и идентификатора участника нарушается. Поэтому атака такого типа называется атакой привязки. Для защиты от таких атак их сертификаты используются вместо открытых ключей, совершая индивидуальный механизм (PKI) для выдачи, отзыва и проверки их корректности.

Выполнив анализ средств защиты в существующих протоколах с криптографической защитой, обнаружив достоинства и недостатки рассмотренных протоколов и их уязвимость к атакам определенного типа, можно сформировать некоторые общие требования к протоколам передачи данных с криптографической защитой, работающих на прикладном уровне:

- использование безопасной аутентификации пользователей на основе двухфакторной аутентификации с использованием криптографических алгоритмов;
- использование симметричных криптографических алгоритмов с большим размером ключа для каждой сессии взаимодействия клиента и сервера, клиента и клиента (при непосредственном взаимодействии двух клиентов без участия сервера, если протокол предусматривает такое взаимодействие);
- использование разовых ключей симметричного алгоритма для шифрования сессий между участниками обмена;
- использование алгоритмов с открытым ключом для защиты ключа симметричного алгоритма шифрования;
- генерация разных пар ключей асимметричного алгоритма шифрования для участников взаимодействия, при этом ключи должны быть не минимального размера;
- обеспечение контроля целостности передаваемых данных с помощью электронно-цифровой подписи;
- использование дополнительных средств защиты, обеспечивающих более надежный обмен ключами (токены, QR-коды);
- возможность использования «арбитра» при генерации ключей для проверки целостности данных;
- возможность использования разрабатываемого протокола с криптографической защитой поверх работы стандартных протоколов обмена информацией и использование такого протокола в клиент-серверных приложениях.

Описав общие требования к протоколу, в дальнейшем можно переходить к разработке архитектуры протокола прикладного уровня с криптографической защитой. При этом при проектировании архитектуры протокола следует предусмотреть возможность повышения эффективности защиты передаваемой информации, как на этапе аутентификации, так и на этапе передачи данных, а также предусмотреть контроль целостности передаваемых защищенных данных.

Выводы

В статье приведен анализ современных тенденций в области протоколов передачи данных с криптографической защитой. Рассмотрена технология VPN – как один из вариантов безопасной передачи данных в сети с использованием криптографической защиты, описаны её популярные протоколы, выделены достоинства и недостатки популярных протоколов. Более подробный обзор протоколов представлен в статье [8].

Также был проведен анализ протоколов с точки зрения их уязвимости к атакам определённого типа.

Описан поиск возможных способов устранения найденных недостатков в алгоритмах протоколов передачи данных на прикладном уровне.

В статье представлены возможные требования к авторскому протоколу передачи данных с криптографической защитой.

В дальнейшем при проектировании архитектуры протокола с криптографической защитой будут проработаны возможные варианты повышения эффективности протокола передачи данных с криптографической защитой для приложений определенного класса таким образом, чтобы исключить возможность проведения атак некоторых типов. Основным требованием для такого протокола будет надежность.

Литература

1. Дошина, А. Д. Криптография. Основные методы и проблемы. Современные тенденции криптографии / А. Д. Дошина, А. Е. Михайлова, В. В. Карлова. — Текст : непосредственный // Современные тенденции технических наук : материалы IV Междунар. науч. конф. (г. Казань, октябрь 2015 г.). — Казань : Бук, 2015. — С. 10-13. — URL: <https://moluch.ru/conf/tech/archive/163/8782/> (дата обращения: 23.10.2022).
2. RFC 2637 Point-to-Point Tunneling Protocol (PPTP) [Энциклопедия сетевых протоколов [Электронный ресурс] — Режим доступа: <https://www.protokols.ru/WP/rfc2637/> - Загл. с экрана.

3. Bruce Schneier. Cryptanalysis of Microsoft's Point-to-Point Tunneling Protocol (PPTP) [Электронный ресурс] — Режим доступа: <https://docviewer.yandex.ru/?tm=1666977162&tld=ru&lang=en&name=paper-pptp.pdf&text=Bruce+Schneier%2C+Cryptanalysis+of+Microsoft%27s+Point+to+Point+Tunneling+Protocol+%28PPTP%29&url=https%3A//www.schneier.com/wp-content/uploads/2016/02/paper-pptp.pdf&lr=142&mime=pdf&10n=ru&sign=d91206e1f3ea382ccc3e21ecf004ef00&keyno=0> - Загл. с экрана.
4. Семёнов Ю.А. Протокол туннелей на сетевом уровне L2 (L2TP) [Электронный ресурс] — Режим доступа: <http://book.itep.ru/4/44/l2pr.htm> - Загл. с экрана.
5. Internet Protocol Security [Электронный ресурс] — Режим доступа: https://ru.on1.click/tiki-index.php?page=info_Internet_Protocol_Security - Загл. с экрана.
6. OpenVPN. Что такое OpenVPN [Электронный ресурс] — Режим доступа: <https://openmaniak.com/ru/openvpn.php> - Загл. с экрана.
7. Обзор Secure Socket Tunneling Protocol [Электронный ресурс] — Режим доступа: <https://vc.ru/dev/119780-obzor-secure-socket-tunneling-protocol> - Загл. с экрана.
8. Бондаренко, В. В. Обзор протоколов передачи данных, использующих криптографическую защиту / В. В. Бондаренко, А. В. Чернышова // Современные информационные технологии в образовании и научных исследованиях (СИТОНИ-2021) : Материалы VII Международной научно-технической конференции, Донецк, 23 ноября 2021 года / Под общей редакцией В.Н. Павлыша. – Донецк: Донецкий национальный технический университет, 2021. – С. 282-288. – EDN ILKYYL.
9. Clark J., Jacob J. A Survey of Authentication Protocol Literature: Version 1.0. 17 Nov. 1997 [Электронный ресурс] — Режим доступа: <http://www.cs.york.ac.uk/jac/papers/drareview.ps.gz> - Загл. с экрана.
10. Криптографические протоколы – Основные определения [Электронный ресурс] — Режим доступа: https://cryptoprotocols.ucoz.com/index/main_definitions/0-2# - Загл. с экрана.
11. Черёмушкин А. В. Криптографические протоколы: основные свойства и уязвимости // Прикладная дискретная математика. — 2009. — нояб. — вып. 2. — с. 115—150. — URL: cyberleninka.ru/article/n/kriptograficheskie-protokoly-osnovnye-svoystva-i-uyazvimosti.pdf.
12. Грищенко, О. С. Анализ алгоритмов аутентификации в распределенных программных системах / О. С. Грищенко, А. В. Чернышова // Программная инженерия: методы и технологии разработки информационно-вычислительных систем (ПИИВС-2018) : Сборник научных трудов II Международной научно-практической конференции, Донецк, 14–15 ноября 2018 года. Том 1. – Донецк: Донецкий национальный технический университет, 2018. – С. 102-108. – EDN XBCKSN.
13. Грищенко, О. С. Защита информации в приложениях для мгновенной передачи сообщений / О. С. Грищенко, А. В. Чернышова // Информатика, управляющие системы, математическое и компьютерное моделирование (ИУСМКМ-2017) : Сборник материалов VIII Международной научно-технической конференции в рамках III Международного Научного форума Донецкой Народной Республики, Донецк, 25 мая 2017 года / Редколлегия: Ю.К. Орлов [и др.]. – Донецк: Донецкий национальный технический университет, 2017. – С. 115-119. – EDN KKRGSJS.

Бондаренко В.В., Чернышова А.В. Повышение эффективности протоколов передачи данных, использующих криптографическую защиту. Статья посвящена актуальной проблеме — современным тенденциям в области протоколов передачи данных с криптографической защитой. В рамках статьи выявлены достоинства и недостатки современных протоколов передачи данных с криптографической защитой, произведен поиск способов их устранения при проектировании авторского протокола передачи данных с криптографической защитой для приложений определенного класса. Сформированы требования к протоколу передачи данных с криптографической защитой.

Ключевые слова: VPN, протокол, атака, шифрование, аутентификация, PPTP, L2TP, IPSec, SSL.

Bondarenko V.V., Chernyshova A.V. Improving the efficiency of data transmission protocols using cryptographic protection. The article is devoted to an urgent problem — modern trends in the field of data transmission protocols with cryptographic protection. The article identifies the advantages and disadvantages of modern data transmission protocols with cryptographic protection, searches for ways to eliminate them when designing an author's data transmission protocol with cryptographic protection for applications of a certain class. The requirements for the data transfer protocol with cryptographic protection have been formed.

Keywords: VPN, protocols, attack, encryption, authentication, PPTP, L2TP, IPSec, SSL.

Методы безопасной аутентификации в клиент-серверных приложениях

А.В. Чернышова^{*1}, А.А. Афанасьева^{*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,
anne.afa@mail.ru

Чернышова А.В., Афанасьева А.А. Методы безопасной аутентификации в клиент-серверных приложениях. В статье представлен обзор методов и подходов безопасной аутентификации пользователей, указаны их особенности и возможности применения. Определены основные задачи протоколов аутентификации. Перспективной задачей является разработка собственного протокола безопасной аутентификации для клиент-серверных приложений с учетом достоинств и недостатков современных протоколов и стандартов.

Ключевые слова: методы аутентификации, клиент-серверная архитектура, протоколы аутентификации, защита информации, криптографический протокол.

Введение

Одной из самых важных задач для любой компьютерной системы является достоверная и безопасная аутентификация её субъектов. Для решения данной задачи в протоколах аутентификации применяют особые правила передачи информации и различные криптографические алгоритмы.

Безопасная аутентификация пользователя происходит в операционной системе, при работе компьютеров в доменных структурах, при работе пользователей в сети при использовании протоколов стека TCP/IP. Не являются исключением и клиент-серверные приложения. Разработчики современных клиент-серверных приложений обязательно должны решить вопрос безопасной аутентификации пользователей в своём приложении.

Сегодня сложно представить, что программисты при разработке клиент-серверного приложения не используют безопасные средства аутентификации пользователей на основе криптографических алгоритмов. Любое современное приложение использует средства защиты для передачи конфиденциальной информации. Для этого используются защищенные протоколы аутентификации, либо свои предложения безопасной аутентификации на базе существующих стандартов и решений.

Каждый пользователь современных компьютерных систем сталкивается с процедурами аутентификации неоднократно, работая в системе каждый день [1]. Работа в операционной системе, использование сервисов социальных сетей, работа в корпоративной сети организации, использование электронной почты, интернет-банкинг – вот малый перечень, где используется безопасная аутентификация. Отдельно следует упомянуть биометрические средства аутентификации пользователей по отпечатку пальцев, ладони, радужки глаза, а также аутентификация по голосу.

Аутентификация пользователя является первой ступенью обеспечения защиты данных. На сегодняшний день существует достаточно методов и подходов аутентификации субъектов различных систем. Но при всем недостатке средств и протоколов аутентификации с криптографической защитой, взломы программных систем всё равно случаются, происходит утечка информации при получении несанкционированного доступа к системе.

Почему же так происходит? При казалось бы отличных средствах аутентификации, до сих пор актуальны вопросы несанкционированного доступа к программным системам, реализующим в том числе и модель клиент-сервер.

Вероятно, проблема комплексная, связанная с проектированием и разработкой приложения, а также используемых средствах защиты на разных этапах создания и функционирования приложения.

Отдельного внимания заслуживают приложения, использующие облачное хранилище данных. Утечка информации с облачных хранилищ происходит не так редко и является проблемой в современном мире.

Так на каком этапе при проектировании и разработке приложений необходимо усилить средства защиты, чтобы повысить безопасность программных систем.

Для начала проанализируем некоторые средства аутентификации, чтобы сделать вывод об их безопасном использовании при разработке приложений с клиент-серверной архитектурой.

Клиент-серверная архитектура

Архитектура «Клиент-Сервер» (также используются термины «сеть Клиент-Сервер» или «модель Клиент-Сервер») предусматривает разделение процессов предоставления услуг и отправки запросов на них на разных компьютерах в сети, каждый из которых выполняют свои задачи независимо от других [2].

Данная архитектура является основой интернет-сервисов и веб-приложений, социальных сетей, а также других приложений, распространяющих информацию по сети интернет. Под клиентом понимается объект, который запрашивает необходимые от сервера данные на уровне взаимодействия двух приложений – клиентской части приложения и серверной части приложения.

Как правило, клиент-серверная модель отличается своей эффективностью. Она достигается в основном за счет разделения выполнения отдельных задач приложения между различными платформами. Сервер предоставляет услуги, а клиент – потребляет. В то же время, приложение-клиент не управляет всеми данными, а лишь предоставляет пользователю результаты посылаемых на сервер запросов.

Множество клиентов могут одновременно иметь доступ к услугам сервера и каким-либо общим ресурсам. Целостность данных в клиент-серверной архитектуре обусловлена их централизованным управлением. При этом предусмотрена независимость индивидуальных данных каждого клиента.

При рассмотрении процесса предоставления доступа в клиент-серверных приложениях можно встретить такие термины, как идентификация, аутентификация и авторизация. Данные понятия не являются тождественными, но все они относятся к элементам защиты информации в приложениях типа клиент-сервер.

Под идентификацией пользователя или аппаратно-программного компонента понимают процедуру его распознавания, как легального субъекта системы, на основе некоторого уникального идентификатора. Обычно данный идентификатор хранится в базе данных системы и присваивается субъекту при его регистрации. Часто на этапе идентификации распознавание информации о пользователе происходит на основе логина. Некоторому пользователю соответствует некоторый идентификатор и логин, который присваивается при регистрации в системе.

Аутентификация (англ. authentication) — это основа безопасности любой системы, которая заключается в проверке подлинности данных о пользователе сервером [3]. Аутентификация сама по себе не подразумевает проверку прав пользователя. Возможности доступа к конкретным функциям и информации определяются на следующем этапе, а именно на этапе авторизации.

Документы, определяющие стандарты аутентификации

На сегодняшний момент выделяют два основных документа, определяющих стандарты аутентификации:

- ГОСТ Р ИСО/МЭК 9594-8-98 — Основы аутентификации;
- FIPS 113 — Computer Data Authentication.

В стандарте - ГОСТ Р ИСО/МЭК 9594-8-98 — Основы аутентификации определён формат информации аутентификации, хранимой справочником. Стандарт описывает способ получения из справочника информации аутентификации, устанавливает предпосылки о способах формирования и размещения в справочнике информации аутентификации, определяет способы, с помощью которых прикладные программы могут использовать информацию аутентификации для выполнения аутентификации. Также стандарт описывает, каким образом с помощью аутентификации могут быть обеспечены другие услуги защиты.

В настоящем стандарте изложены два вида аутентификации: простая, использующая пароль как проверку заявленной идентичности, и строгая, использующая удостоверения личности, созданные с использованием криптографических методов. Простая аутентификация предлагает некоторые ограниченные возможности защиты от несанкционированного доступа. В качестве основы услуг, обеспечивающих защиту, должна использоваться только строгая аутентификация.

Аутентификация (и другие услуги защиты) могут быть предложены только в контексте определенной стратегии защиты. Пользователи прикладной программы должны сами определить свою собственную стратегию защиты, которая может быть ограничена услугами, предусмотренными стандартом.

Задача стандартов, определяющих прикладные программы, которые используют основы аутентификации, состоит в том, чтобы установить правила протокольных обменов, которые необходимо осуществить для обеспечения аутентификации, основываясь на информации аутентификации, полученной из справочника. Протоколом, используемым прикладными программами для получения удостоверения личности из справочника, является протокол доступа к справочнику, определенный в ИСО/МЭК 9594-5. [4]

FIPS 113 — Computer Data Authentication. Настоящий стандарт устанавливает Data Authentication Algorithm (DAA), который может быть использован для обнаружения несанкционированных изменений данных, как преднамеренных, так и случайных. Стандарт основан на алгоритме, указанном в Data Encryption Standard (DES) Federal Information Processing Standards Publication (FIPS PUB) 46, и совместим как с Department of the Treasury's Electronic Funds and Security Transfer Policy and the American National Standards Institute (ANSI) так и с Standard for Financial Institution Message Authentication. Данный стандарт используется для контроля над целостностью передаваемой информации средствами криптографической аутентификации. [5]

Методы и подходы к аутентификации

На практике существуют различные способы аутентификации. Выбор наиболее подходящего метода для конкретной программной системы является одним из важнейших решений при проектировании её подсистемы безопасности.

Выделяют такие подходы к аутентификации, как «субъект знает», «субъект обладает» и «субъект существует».

В первом случае подразумевается наличие у пользователя компьютерной системы информации, известной только ему. Это могут быть пароль, секретный ключ или цифровой код, передаваемый в протоколе для аутентификации. Под обладанием субъекта имеется в виду наличие у него, к примеру, токена или генератора паролей. Последний подход предполагает присутствие в протоколе проверки индивидуальных биометрических признаков пользователя, таких как голос или отпечатки пальцев.

В зависимости от используемых технических средств возможна комбинация перечисленных методов. В криптографических протоколах чаще реализован именно первый подход, основанный на опознании пользователя по паролю.

Аутентификация по паролю является, пожалуй, самым широко используемым методом. В данном случае каждый субъект системы обладает секретным паролем, состоящим из комбинаций буквенных и цифровых последовательностей. Различия протоколов, основанных на проверке пароля, состоят в средствах хранения пароля в системе и методах его проверки.

В рамках простой аутентификации передача идентификатора и пароля пользователя может выполняться как в открытом, так и в зашифрованном виде. На рисунке 1 представлена схема аутентификации пользователя с использованием отображения его пароля посредством односторонней функции.

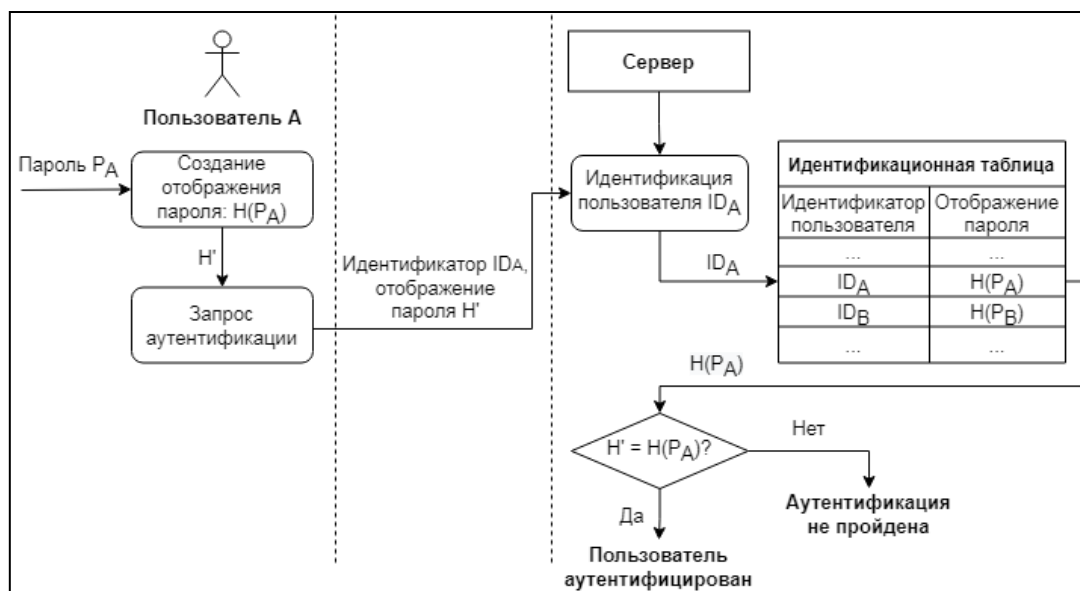


Рисунок 1 – Схема простой аутентификации с использованием отображения пароля

При использовании парольной информации для аутентификации пользователя существуют угрозы разглашения, прослушивания и подбора пароля. Для их предотвращения используют как криптографические методы, так и средства, затрудняющие или замедляющие возможности перебора или угадывания пароля.

В рамках парольной аутентификации существуют варианты с фиксированным и одноразовым паролями. В случае, если пароль может оставаться неизменным от сеанса к сеансу, он является фиксированным. Рекомендации по длительности использования такого пароля предусмотрены регламентом программной системы. Предполагается, что пароли должны периодически меняться самим пользователем. Однако, в таком случае всё так же необходим надежный механизм передачи пароля по незащищенным каналам связи. Парольная аутентификация потенциально уязвима ввиду использования социотехники, внедрения клавиатурных шпионов, возможности перехвата [6].

Одноразовые пароли, в отличие от фиксированных, не могут быть использованы для повторной аутентификации. Как понятно из названия, каждый из таких паролей может быть использован только один раз и прекращает свое действие после очередной аутентификации субъекта.

Аутентификация по одноразовому паролю часто применима в совокупности с фиксированным паролем для дополнительного подтверждения личности субъекта. При этом используется так называемая двухфакторная аутентификация, значительно повышающая безопасность системы. Данный метод предполагает аутентификацию пользователя не только с помощью его пароля, но также посредством наличия у него некоторого

устройства для генерации одноразовых паролей.

Существует несколько методов на основе одноразовых паролей, используемых в протоколах аутентификации. Метод одноразовых паролей с разделяемыми списками подразумевает наличие заранее определенного списка или карты паролей, которые должны храниться как клиентом, так и сервером. Очередной пароль может быть выбран клиентом для использования в протоколе аутентификации и затем удален из общего списка. Очевидно, что данный подход требует предварительного формирования достаточно длинного списка паролей, а также поиска пароля, с которым клиент в очередной раз пытается аутентифицировать себя.

Одноразовые пароли могут быть переданы пользователю в электронном виде через SMS или другой канал связи. Этот метод также популярен для дополнительной аутентификации, к примеру, во время изменения фиксированного пароля или других данных субъекта, выполнения финансовых операций, подтверждения операций с банком.

В системах с надежным каналом связи, предусматривающем дополнительные действия при возможном обрыве связи, может быть использован метод одноразовой парольной аутентификации на основе последовательно обновляемых паролей. При таком подходе клиент и сервер имеют первоначально только один пароль. Каждый последующий пароль шифруется на ключе, полученном из предыдущего пароля.

Ещё одним методом аутентификации, основанном на одноразовых паролях, является использование последовательностей паролей, построенных на хеш-функциях. Данный метод, предложенный Лесли Лампортом, предполагает модификацию пароля с помощью однонаправленной функции. Для его использования клиент и сервер должны согласовать первоначальный пароль и количество применения хеш-функции.

Одноразовые пароли могут быть получены различными методами. К наиболее популярным на данный момент относят программные (RSA SecurID) или аппаратные реализации токенов (Google Authenticator), генерирующих одноразовые пароли с помощью некоторого секретного ключа и временной метки. При этом, принимающая сторона также должна обладать секретным ключом. Аппаратные токены более применимы в системах с разграничением доступа для предоставления функции аутентификации субъекта одного приложения другому. Данный токен генерируется на основе достоверных сведений о пользователе, может быть использован другими приложениями для аутентификации или даже авторизации субъекта.

Помимо протоколов аутентификации на основе пароля существуют также протоколы на основе «запроса-ответа». В подобных протоколах субъект может быть аутентифицирован посредством проверки знания им некоторой секретной информации без предъявления самого секрета (см.рис.2). Часть таких протоколов подразумевают знание секрета сервером, в то время как для некоторых протоколов в принципе нет такой необходимости.

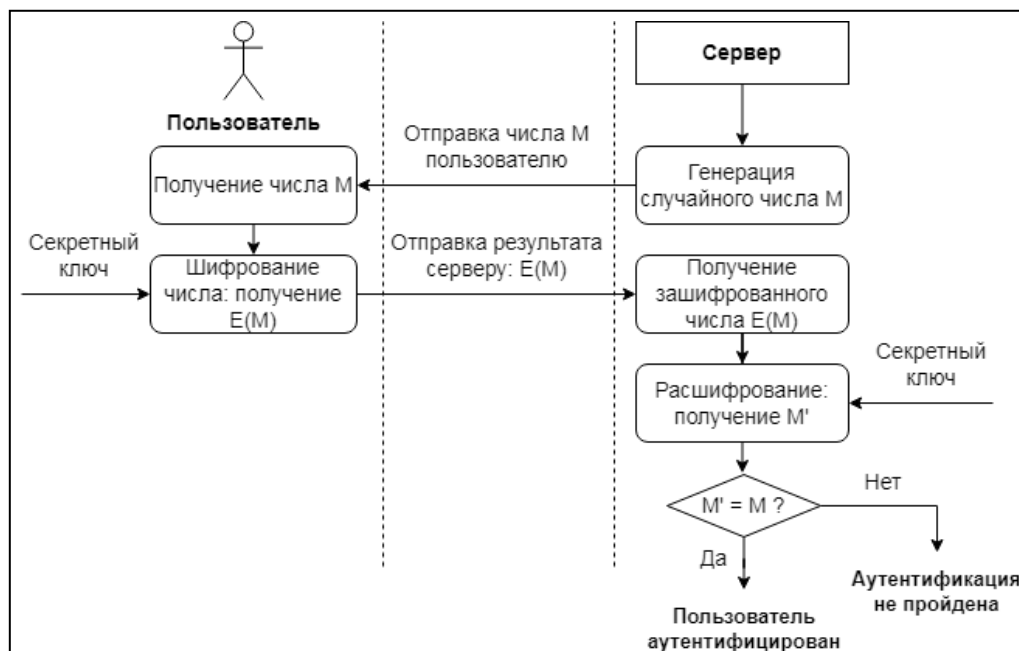


Рисунок 2 – Схема строгой аутентификации на основе алгоритма шифрования

Такой подход основан на ответе пользователя на изменяемые с каждым сеансом запросы сервера. Более того, ответ аутентифицируемого субъекта должен зависеть как от вопроса, так и от некоторого секрета. Все запросы принимающей стороны (сервера) должны отличаться друг от друга, чтобы избежать перехвата полезной информации, которая может быть обнаружена злоумышленником при прослушивании незащищенных каналов связи. В случае игнорирования данного требования, секретная информация может быть также использована нелегальным субъектом при следующем сеансе аутентификации.

К тому же, в протоколах на основе метода «запроса-ответа» необходимо обеспечивать свежесть получаемой секретной информации. В процессе сеанса может производиться проверка и подтверждение актуальности той или иной сгенерированной величины. Это позволяет предоставлять гарантии уникальности сеанса даже в случае его прерывания и повторной попытки аутентификации.

Параметрами величин, уникальных для каждого сеанса, служат, к примеру, случайные числа или числовые последовательности. Также могут быть использованы метки времени. Не менее важно гарантировать целостность этих параметров с помощью их криптографических преобразований на основе других данных, используемых в протоколе.

Существует также возможность аутентификации пользователя по сгенерированным заранее ключам доступа. Ключи доступа генерируются сервером, как уникальный и достаточно длинный набор символов, и сохраняются в клиентских приложениях, выполняющих аутентификацию посредством полученных ключей. Данный метод весьма популярен в различных веб-сервисах.

Примечательно, что к данному подходу применимы дополнительные настройки ключа пользователем: срок действия каждого из конкретных ключей (обычно ограничен, но может быть и постоянным), а также уровень доступа субъекта (делегирование прав при аутентификации по заданному ключу). Часть систем используют данный подход в качестве дополнительного метода аутентификации в то время, как некоторые из них позволяют только такой метод аутентификации для сторонних клиентских приложений. В большей степени это связано с попыткой дополнительно обезопасить данные пользователя (логин и пароль) от перехвата по каналам связи.

Анализ протоколов аутентификации

Протокол аутентификации представляет собой тип компьютерного протокола связи или криптографического протокола, специально предназначенного для передачи аутентификационных данных между двумя объектами [7]. Протоколы аутентификации помогают в контроле доступа клиентов системы к её ресурсам, а также способствуют учету использования ресурсов системы пользователями.

Главная задача протокола заключается в указании четкой последовательности действий для аутентификации пользователя. Для выполнения данной задачи определяется синтаксис протокола и тип необходимой информации. Это позволяет аутентифицировать как клиента, подключающегося к серверу, так и сам сервер, предоставляемый услуги для подключаемого субъекта.

Аутентификация по двухточечному протоколу PPP (Point-to-Point) используется для установления связи между несколькими узлами сети. Данный протокол считается одним из ранних протоколов, обеспечивающих аутентификацию соединения. В данном механизме аутентификации используются такие специально разработанные протоколы, как PAP, CHAP и EAP.

PAP (Password Authentication Protocol) протокол подразумевает, что аутентификация производится по заранее известному взаимодействующим объектам паролю. При этом клиент многократно отправляет принимающей стороне запрос аутентификации, а именно пакеты с именем и паролем, до тех пор, пока не получит его подтверждение. Сами по себе данные пересылаются в открытом виде, что уже не является безопасным способом, даже при учете мгновенного отклика на посылаемый запрос.

Процесс аутентификации в CHAP (Challenge Handshake Protocol) протоколе инициируется хостом в любое время сеанса и может быть многократно повторён с каким-либо промежутком времени. CHAP не предполагает повторную передачу пароля субъекта. В данном случае, для подтверждения текущего состояния аутентификации, клиент получает от сервера случайную строку, выполняет криптографические преобразования (хеш-функцию MD5) на основе собственного пароля и полученной строки, и отправляет результат вместе с собственным именем серверу. Сервер производит подобные преобразования. По результату сравнения хэш-значений можно судить об успешности аутентификации.

Протокол Extensible Authentication Protocol (EAP) изначально был создан специально для PPP.

На данный момент он используется в различных версиях IEEE 802.1x и продолжает дорабатываться для поддержания все более новых механизмов аутентификации. Более 40 методов данного протокола реализуют особые расширенные способы проверки подлинности на основе общей стандартизированной структуры EAP.

В более крупных сетях обычно используются более сложные протоколы (к примеру, архитектуры AAA), предназначенные как для аутентификации пользователя, так и для его авторизации и учета используемых пользователем ресурсов (см. табл. 1).

Простой протокол управления доступом TACACS основан на архитектуре AAA. Сам по себе протокол не предполагает шифрование данных при их передаче. Однако, его более поздние версии поддерживают не только шифрование передаваемых пакетов, но и обработку аутентификации, авторизации и учета на отдельных серверах.

Еще один протокол RADIUS (Remote Authentication Dial-in User Service) предоставляет механизм аутентификации субъектов распределенной сети. Протокол сам по себе основан на клиент-серверной архитектуре, где в качестве клиента обычно выступает сервер для хранения данных. В дополнение к этому пользовательские пароли между клиентами и серверами RADIUS передаются в зашифрованном виде во избежание перехвата паролей при их передаче через незащищенные сети [8].

Созданный на базе протокола RADIUS протокол DIAMETER является улучшенной версией своего

предшественника. DIAMETER изначально подразумевался как базовый протокол с возможностью его расширения. Протокол обладает более высоким уровнем надежности (использует транспортный протокол TCP) и безопасности благодаря защиты данных на сетевом и транспортном уровнях.

Kerberos – ещё один специально разработанный для аутентификации субъектов системы протокол. Чаще всего он используется для реализации механизма одноразовой аутентификации в нескольких приложениях.

Процесс аутентификации по протоколу Kerberos отличается своей сложностью по сравнению с предыдущими протоколами. В протоколе используется понятие Ticket (билет). Билет Kerberos — это сертификат, выданный сервером аутентификации (доверенной стороной) и зашифрованный с помощью его ключа [9]. Одним из преимуществ протокола Kerberos, обеспечивающих очень высокий уровень сетевой безопасности, является то, что во всех сетевых взаимодействиях в открытом виде не передаются ни пароли, ни хэши паролей.

Таблица 1 – Анализ протоколов аутентификации

Протокол	Схемы аутентификации	Транспортный уровень	Защита данных
TACACS	PAP, CHAP	UDP	данные не зашифрованы
TACACS+	Clear text (ASCII, PAP), CHAP, ARAP	TCP	поддержка зашифрованного DES пароля и отображения пароля MD5
RADIUS	PAP, CHAP, EAP, MD5	UDP	шифрование пароля, поддержка EAP-TLS
DIAMETER	PAP, CHAP, EAP	TCP, SCTP	данные не зашифрованы, поддержка шифрования канала TLS, IPSec
Kerberos	на основе ticket (служба TGS, билет TGT)	TCP/UDP	поддержка шифрования канала (TLS), DES (устарело), поддержка токенов AES 128/256

Выводы

Клиент-серверная архитектура уже является основой практически любой корпоративной сети, интернет-сайтов, социальной сети, веб-приложений и не только. Не исключено, что и дальнейшее развитие информационных технологий будет происходить с непосредственным использованием данной архитектуры. А значит вопросы безопасной аутентификации разработанных клиент-серверных приложений, безусловно, являются актуальными и сегодня, несмотря на то, что существуют на сегодняшний день безопасные протоколы аутентификации.

Помимо обеспечения целостности данных, важной задачей любого клиент-серверного приложения является аутентификация клиентов и дальнейшее предоставление им доступа к необходимой информации (авторизация). Данный этап неотъемлем при предоставлении доступа множеству субъектов к различным ресурсам сервера в клиент-серверной архитектуре.

Одним из самых важных механизмов защиты на уровне клиент-серверного приложения является аутентификация.

В статье приведены современные методы и подходы к аутентификации пользователей, указаны их особенности и возможности применения. Также описаны основные документы, определяющие стандарты аутентификации. Также рассмотрены основные понятия и задачи протоколов аутентификации.

В дальнейшем планируется провести более углубленное изучение современных протоколов аутентификации, что позволит выявить их достоинства и недостатки, выполнив сравнительный анализ протоколов аутентификации, детально изучить стандарты в области аутентификации, а также определить требования к повышению эффективности существующих протоколов аутентификации при использовании в клиент-серверных приложениях.

Литература

1. Афанасьев, А. А. Аутентификация. Теория и практика обеспечения безопасного доступа к информационным ресурсам. Учебное пособие / А. А. Афанасьев, Л. Т. Веденев, А. А. Воронцов. – Москва : Горячая линия - Телеком, 2012. – 550 с. – ISBN 978-5-9912-0257-2. – EDN RBAWOP.
2. Архитектура «Клиент-сервер» [Электронный ресурс] – Режим доступа: <https://itelon.ru/blog/arkhitektura-klient-server/> - Загл. с экрана.
3. Что такое Аутентификация: Методы и Элементы [Электронный ресурс] – Режим доступа: <https://molotokrus.ru/sposoby-autentifikatsii-informatsionnaya-bezopasnost/> - Загл. с экрана.
4. Государственный стандарт Российской Федерации ГОСТ Р ИСО/МЭК 9594-8-98 — Основы аутентификации [Электронный ресурс] - Режим доступа: ГОСТ Р ИСО/МЭК 9594-8-98 Информационная технология. Взаимосвязь открытых систем. Справочник. Часть 8. Основы аутентификации (gostrf.com) – Загл. с экрана.

5. FEDERAL INFORMATION PROCESSING STANDARD PUBLICATION The Keyed-Hash Message Authentication Code (HMAC)

6. Шапиро, Л. Аутентификация и одноразовые пароли. Теоретические основы. Часть 1 / Л. Шапиро // Системный администратор. – 2012. – № 9(118). – С. 88-91. – EDN RFVKTR.

7. Протокол аутентификации гост [Электронный ресурс] – Режим доступа: <https://sert-service.ru/protokoly-autentifikatsii-gost/> - Загл. с экрана.

8. Протокол RADIUS. Перевод RFC 2865 [Электронный ресурс] – Режим доступа: <https://rfc.com.ru/rfc2865.htm>

9. Kerberos: служба аутентификации для компьютерных сетей [Электронный ресурс] – Режим доступа: <https://gost.isi.edu/publications/kerberos-neuman-tso.html> - Загл. с экрана.

Чернышова А.В., Афанасьева А.А. Методы безопасной аутентификации в клиент-серверных приложениях. В статье представлен обзор методов и подходов безопасной аутентификации пользователей, указаны их особенности и возможности применения. Определены основные задачи протоколов аутентификации. Перспективной задачей является разработка собственного протокола безопасной аутентификации для клиент-серверных приложений с учетом достоинств и недостатков современных протоколов и стандартов.

Ключевые слова: методы аутентификации, клиент-серверная архитектура, протоколы аутентификации, защита информации, криптографический протокол.

Chernyshova A.V., Afanaseva A.A. Secure authentication methods in client-server applications. The article provides an overview of the methods and approaches for secure user authentication, their features and possibilities of application are indicated. The main tasks of authentication protocols are also defined. A promising task is to develop our own secure authentication protocol for client-server applications, taking into account the advantages and disadvantages of modern protocols and standards.

Keywords: authentication methods, client-server architecture, authentication protocols, information security, cryptographic protocol.

Повышение эффективности и производительности баз данных путём оптимизации SQL-запросов

О.Г. Артеменко

ассистент, Донецкий национальный технический университет,
olga_artyomenko@bk.ru, SPIN-код: 3643-6760

***Артеменко О.Г. Повышение эффективности и производительности баз данных путём оптимизации SQL-запросов.** Статья посвящена вопросам, рассматривающим значимость оптимизации SQL-запросов для повышения эффективности работы современных баз данных. Приведены примеры методик оптимизации запросов, представлены результаты практических примеров, наглядно демонстрирующих действенность описанных методик. На основании проведенного исследования, приведены зависимости времени выполнения оптимизированных и неоптимизированных запросов от количества элементов в выборке.*

***Ключевые слова:** оптимизация SQL-запросов, база данных, повышение эффективности, производительность, методы оптимизации*

Введение

Информационная эра диктует тенденции развития современного мира, направленные на повсеместную компьютеризацию человечества, что, в свою очередь, выступает основным движущим механизмом автоматизирования промышленной, управленческой и научной работы, где обязательны сохранение, обработка, получение, передача и сбор в единое целое всей информации.

Автоматизирование на персональных компьютерах изменяет стандарты переработки данных, придавая слаженную работу промышленности и организаций на базе более новейшей информационной технологии.

Использование персонального компьютера в виде механизма обработки информации в разных областях человеческой деятельности повышает информационную культуру общества, способствуя без осложнений перейти к информационному обществу, где информация является самым ценным материалом наравне с финансовыми, энергетическими и другими ресурсами.

Широко известна история успеха Натана Ротшильда, когда реальная информация, полученная раньше других, позволила банкиру вести беспроигрышную игру на бирже. Тогда же, Ротшильд произносит свой легендарный афоризм: «Кто владеет информацией — тот владеет миром». Наши современные реалии уже не раз подтверждали этот факт. В нужное время полученная, правильно обработанная и четко представленная информация зачастую увеличивает эффективность принимаемых решений и, следовательно, их результат.

Первым базам данных нашли применение в химии, ядерной физике, космонавтике и остальных науках, которые требуют систематического подхода к работе с информацией. Последующая эволюция компьютеризации и компьютерных характеристик привело человечество к тому, что базы данных оказались в разработке фактически во всех областях занятости человека, и стали повседневно использоваться в разных экономических объектах: от сельского хозяйства до финансовых систем [1].

Однако, самым прорывным достижением применения баз данных стала всемирная сеть интернет, которая по всей сути является самой крупной и обширной базой данных, известной на весь мир.

Из всего вышеизложенного можно прийти к выводу, что базы данных в современном мире это скорее неизбежность и необходимость, так как всё, с чем мы каждый день взаимодействуем в жизни, по всей видимости, зафиксировано в какой-нибудь базе.

Основные принципы оптимизации БД

Для обеспечения высокой производительности БД, что в свою очередь, способствует развитию бизнеса в той или иной отрасли просто необходимо использование наиболее производительного оборудования и оптимизированной БД.

Большинство компаний склоняется в сторону первого варианта, ведь при оптимизации базы данных нужны сразу несколько квалифицированных специалистов с узкой специализацией. Понятно, что в непрофильной компании содержать таких специалистов экономически нецелесообразно, и гораздо выгоднее приобрести новое более производительное и более дорогое оборудование. Однако, в случае высоконагруженной базы данных, новое оборудование обеспечит незначительный прирост производительности, при довольно высоких затратах. В таких ситуациях оптимизация СУБД является наиболее выгодным решением в сравнении с приобретением нового оборудования.

В общей сложности существует несколько рекомендуемых принципов оптимизации для сложной работы БД. В первую очередь их можно рассматривать как 2 категории:

- аппаратная оптимизация
- программная оптимизация

И если с первым пунктом все, в принципе, довольно понятно, в плане – чем лучше железо, тем оно эффективнее и выше его работоспособность, то со вторым – уже не все так просто, учитывая, что программная оптимизация тоже в большой степени напрямую зависима от аппаратной, начиная с момента настройки самого SQL-сервера. Сюда относится и эффективное использование ресурсов процессора, и оптимальное распределение дискового пространства, и оптимизированное размещение файлов баз данных, и грамотное распределение памяти, и непосредственная оптимизация самой операционной системы.

Если же рассматривать программную оптимизацию уже более в отрыве от аппаратной, то тут можно выделить три основных блока рекомендаций:

- непосредственно сама оптимизация БД и СУБД в целом;
- оптимизация взаимодействия приложения и MS SQL Server (далее будем рассматривать взаимодействие приложения .NET и MS SQL Server);
- оптимизация самих запросов.

Для первого блока рекомендаций существует следующие способы оптимизации работы БД и СУБД в целом:

- устаревание процедурного кэша;
- неоптимальные индексы;
- неоптимальные статистики.

Касаясь второго блока оптимизации взаимодействия приложения и MS SQL Server, то для достижения эффективности в данном случае применимы следующие рекомендательные принципы:

1. Не принуждать пользователя ждать отклика приложения, а асинхронно отправлять запросы к СУБД.
2. Реализовать механизм отложенного запуска и системы накопления запросов (отправка запросов так называемыми «пачками»), а не отправлять запросы к СУБД единично.
3. Для снижения количества обращений к самой СУБД следует использовать хэширование для всех компонентов программного обеспечения.
4. Верно назначать уровень изоляции при работе с БД.
5. На серверной стороне ПО при необходимости реализовать очередь обращений к СУБД, которая будет учитывать не только специфику работы информационной системы, но и также аппаратные возможности сервера СУБД или группы серверов СУБД.
6. По возможности, не запрашивать сразу все от СУБД и потом применять фильтр, а заранее фильтровать как можно больше данных.
7. Стараться избегать фильтрации по большому массиву данных не на стороне СУБД.
8. Не писать логи компонентов и логи БД в одну таблицу [2].

Что же касается третьего блока, то в нем можно рассматривать следующие составляющие:

- поиск «тяжелых» SQL запросов и их оптимизация;
- анализ причин почему одни запросы более оптимальны, нежели другие и что именно делает их такими;
- оценка ситуаций, когда запросы выполняются слишком долго;
- исследование индексов, их производительности, плюсов и минусов [3].

Общие рекомендации по оптимизации запросов SQL

Общие рекомендации по оптимизации самих запросов, в нескольких случаях могут пересекаться со вторым блоком программной оптимизации БД.

Среди основных выделяют:

- максимально возможную фильтрацию данных еще до момента соединения их с другими таблицами;
- как можно меньше по объему данных сортировать результирующий набор;
- избегать конструкции DISTINCT, LIKE '%...', OUTER JOIN, особенно на больших объемах данных [4];
- использовать подзапросы в самой выборке для случаев, когда в выборке требуется лишь одно поле от соединяемой таблицы, а не соединять такую таблицу;
- учитывать имеющиеся индексы при фильтрации, агрегации и выборке, чтобы оптимизатор мог ими воспользоваться;
- возвращать только те поля, которые действительно необходимы, а не все поля из всех соединенных таблиц;
- при обновлении и удалении данных (если фильтр или агрегация построены не по кластерному индексу или помимо кластерного индекса есть и другие условия или агрегация), то не производить операцию сразу по таблице, а сначала выбрать удаляемые/изменяемые данные во временную таблицу (которая будет состоять из столбцов, которые входят в кластерный индекс, а также всех прочих необходимых полей для обновления) и уже после применять непосредственно удаление/обновление.

- вынести часть условия в фильтр, а не перегружать условия для соединения таблиц.

Примеры практик оптимизации SQL запросов

Для достижения конечной цели исследование проводилось на группе из 1 400 000 объектов.

1. Использование операторов GROUP BY и DISTINCT

Рассмотрим варианты группировок по полям таблицы.

Запрос с использованием GROUP BY будет выполняться дольше, в среднем 1.3с.

```
select ii.ASSET_NUMBER
from inv_inventory_item_os ii
      join INV_BILL b on b.id = ii.BILL_ID
      join INV_COMMISSION_INFO ci on ci.id = b.COMMISSION_INFO_ID
where ci.INVENTORY_ID = 3541
group by ii.ASSET_NUMBER;
```

Рисунок 1 – Листинг запроса с конструкцией GROUP BY

Запрос с использованием DISTINCT имеет выигрыш в скорости, по сравнению с предыдущим, в среднем он обрабатывает за 1с 150мс.

```
select distinct ii.ASSET_NUMBER
from inv_inventory_item_os ii
      join INV_BILL b on b.id = ii.BILL_ID
      join INV_COMMISSION_INFO ci on ci.id = b.COMMISSION_INFO_ID
where ci.INVENTORY_ID = 3541;
```

Рисунок 2 – Листинг запроса с конструкцией DISTINCT

При этом план выполнения запроса будет выглядеть одинаково в обоих случаях (см. рис. 3).

Operation	Params	Rows	Total Cost
▼ Select		1	25397.0
▼ [G] Group By (HASH UNIQUE)		1	25397.0
▼ Hash Join (HASH JOIN SEMI)		1	25396.0
▼ Nested Loops (NESTED LOOPS SEMI)		1	25396.0
▼ Unknown (STATISTICS COLLECTOR)			
▼ Nested Loops		1	25395.0
▼ Access (VIEW)		1307	25395.0
▼ [G] Group By (HASH UNIQUE)		1307	25395.0
Full Scan (TABLE ACCESS FULL)	table: INV_INVENTORY_ITEM_OS;	1401232	25350.0
▼ Index Scan (TABLE ACCESS BY INDEX ROWID)	table: INV_BILL;	1	0.0
Unique Index Scan (INDEX UNIQUE SCAN)	index: SYS_C008379;	1	0.0
▼ Index Scan (TABLE ACCESS BY INDEX ROWID)	table: INV_COMMISSION_INFO;	1	1.0
Unique Index Scan (INDEX UNIQUE SCAN)	index: SYS_C008404;	1	0.0
Full Scan (TABLE ACCESS FULL)	table: INV_COMMISSION_INFO;	1	1.0

Рисунок 3 – План выполнения запросов с конструкциями DISTINCT и GROUP BY

2. Использование Select вместо Select *

Одним из лучших способов повышения эффективности является снижение нагрузки на базу данных. Это можно сделать, ограничив объем информации, извлекаемой из каждого запроса. Выполнение запросов с помощью Select * извлечет всю соответствующую информацию, доступную в таблице базы данных. Он извлечет всю ненужную информацию из базы данных, что займет много времени и увеличит нагрузку на базу данных.

В качестве примера, рассмотрим имя таблицы inv_inventory_item_os.

- Select * from inv_inventory_item_os;
- Select asset_number from inv_inventory_item_os;

Таким образом, лучший подход — использовать оператор Select с определенными параметрами для получения только необходимой информации. Использование Select снизит нагрузку на базу данных и повысит производительность.

3. Использование Exist() вместо Count()

И Exist(), и Count() используются для поиска того, есть ли в таблице определенная запись или нет. Но в большинстве случаев Exist() намного эффективнее, чем Count(). Поскольку Exist() будет работать до тех пор, пока не найдет первую совпадающую запись, тогда как Count() продолжит работу и предоставит все совпадающие записи. Следовательно, эта практика оптимизации SQL-запросов экономит много времени и вычислительной мощности. EXISTS останавливается, когда логический тест оказывается верным, тогда как COUNT(*) должен подсчитывать каждую строку, даже после того, как она прошла тест.

4. По возможно необходимо использовать оператор CASE

В качестве примера рассмотрим запрос с использованием UNION, который на заданной выборке будет выполняться 3.3с.

```
select ii.id, 'Актив'
from inv_inventory_item_os ii
where BASE_ASSET_NUMBER is null
union
select ii.id, 'Субактив'
from inv_inventory_item_os ii
where BASE_ASSET_NUMBER is not null;
```

Рисунок 4 – Листинг запроса с конструкцией UNION

Operation	Params	Rows	Total Cost
▼ Select		1401232	30464.0
▼ Sort Unique		1401232	30464.0
▼ Union All (UNION-ALL)			
Full Scan (TABLE ACCESS FULL)	table: INV_INVENTORY_ITEM_OS;	1401192	25352.0
▼ Access (TABLE ACCESS BY INDEX ROWID BATCHED)	table: INV_INVENTORY_ITEM_OS;	40	35.0
Full Index Scan (INDEX FULL SCAN)	index: OS_BASE_ASSET_NUMBER_IDX;	40	1.0

Рисунок 5 – План выполнения запроса с конструкцией UNION

В тех же условиях, запрос с использованием CASE будет выполняться в четыре раза быстрее, в среднем 800мс. Это непосредственно связано с выполнением гораздо меньшего количества операций.

```
select ii.id, case when ii.BASE_ASSET_NUMBER is null then 'Актив' else 'Субактив' end
from inv_inventory_item_os ii;
```

Рисунок 6 – Листинг запроса с конструкцией CASE

Данное различие четко наблюдается на плане выполнения запроса (см. рис 7).

Operation	Params	Rows	Total Cost
▼ Select		1401232	25351.0
Full Scan (TABLE ACCESS FULL)	table: INV_INVENTORY_ITEM_OS;	1401232	25351.0

Рисунок 7 – План выполнения запроса с конструкцией UNION

5. Использование оператора WHERE вместо HAVING

При использовании оператора LIKE, лучшим вариантом будет вынести его в HAVING, что позволит сократить время выполнения запроса и, в свою очередь, увеличить производительность. Например, запрос без оператора LIKE будет выполняться 700-800мс для указанного объема данных.

```

select ii.ASSET_NUMBER, count(ii.id)
from inv_inventory_item_os ii
      join inv_bill b on b.id = ii.BILL_ID
where lower(ii.ASSET_NUMBER) like lower('%2283%')
group by ii.ASSET_NUMBER
HAVING count(ii.id) > 1;

```

Рисунок 8 – Листинг запроса с использованием оператора LIKE в конструкции WHERE

А запрос с LIKE вынесенным в HAVING 500-600мс, что приносит выигрыш в производительности.

```

select ii.ASSET_NUMBER, count(ii.id)
from inv_inventory_item_os ii
      join inv_bill b on b.id = ii.BILL_ID
group by ii.ASSET_NUMBER
HAVING count(ii.id) > 5
      and lower(ii.ASSET_NUMBER) like lower('%2283%');

```

Рисунок 9 – Листинг запроса с использованием оператора LIKE в конструкции HAVING

6. Использование оператора ORDER BY

Необходимо использовать оператор ORDER BY только в случае необходимости. SQL создает рабочие таблицы и помещает туда данные. Затем он организует данные в рабочей таблице на основе запроса и возвращает результаты.

Запрос с использованием сортировки будет выполняться примерно 2.6с, в свою очередь, запрос без участия сортировки выполнится в нашем случае в пять раз быстрее, всего за 600мс.

```

select ii.NAME
from inv_inventory_item ii
order by NAME desc;

```

Рисунок 10 – Листинг запроса с конструкцией LIKE

```

select ii.NAME
from inv_inventory_item ii;

```

Рисунок 11 – Листинг запроса с конструкцией LIKE

7. Использование соединений вместо подзапросов

В первом примере используется подзапрос. Выполнение данного запроса на выборке 1 миллион записей займет 7.5с. В то же время использование соединений может сократить этот результат. В рассмотренном случае примерно до 6.9с.

```

select ii.id,
       (select BILL_NUMBER
        from INV_BILL b
        where b.id = ii.BILL_ID) as billNumber
from inv_inventory_item_os ii;

select ii.id, b.BILL_NUMBER
from inv_inventory_item_os ii
      join INV_BILL b on b.id = ii.BILL_ID;

```

Рисунок 12 - Листинг кода с использованием подзапроса и соединения

Если сравнивать соединения и подзапросы, соединения обычно выполняются быстрее. Время поиска запроса с использованием объединений почти всегда будет лучше, чем при использовании подзапроса.

Причина в том, что соединения уменьшают нагрузку на базу данных, заменяя несколько запросов одним запросом соединения. Это, в свою очередь, позволяет лучше использовать возможности базы данных для поиска, фильтрации и сортировки записей. При этом по мере того, как вы добавляете в запрос больше соединений, серверу базы данных приходится выполнять больше работы, что приводит к замедлению времени извлечения данных.

Хотя соединения являются необходимой частью извлечения данных из нормализованной базы данных, важно, чтобы соединения были написаны правильно, поскольку неправильные соединения могут привести к серьезному снижению производительности и неточным результатам запроса. Также в некоторых случаях подзапрос может заменить сложные объединения и объединения с минимальным снижением производительности, если таковые имеются.

Анализ зависимости времени выполнения от количества элементов в выборке

Для наглядности были проведены эксперименты на трёх выборках с разным количеством объектов. Были рассмотрены выборки размером 1400, 14000 и 1400000 записей, для описанных ранее конструкций запросов. Полученные результаты подтверждают эффективность описанных выше примеров.

Таблица 1 - Показатели скорости работы сортировки для конструкций Union и Case

Кол-во объектов выборки	Union	Case
1400	600мс	500мс
14000	750мс	700мс
140000	1с 400мс	1с 200мс

Таблица 2 - Показатели скорости работы сортировки для конструкций Having и Where

Кол-во объектов выборки	Having	Where
1400	330мс	260мс
14000	360мс	300мс
140000	400мс	360мс

Таблица 3 - Показатели скорости работы сортировки для конструкций Having и Where с использованием оператора LIKE

Кол-во объектов выборки	Where	Having
1400	1с 400мс	250мс
14000	1с 500мс	300мс
140000	1с 700мс	400мс

На первый взгляд представленные результаты времени выполнения SQL-запросов создают впечатление, что разница несущественна, однако, т.к. в современных БД объем информации превышает размер тестовых данных во много раз и запросы могут содержать в себе множество различных операторов, хорошая оптимизация запросов покажет существенный результат, что естественно отразится на производительности и эффективности работы.

Заключение

Рассмотрены основные принципы оптимизации баз данных. Исследованы методы оптимизации SQL-запросов, способствующие повышению производительности и эффективности работы СУБД.

В качестве наглядного примера рассмотрены некоторые примеры практик оптимизации SQL-запросов, в результате чего определено временное соотношение обработки оптимизированных и неоптимизированных запросов на выборках разной величины.

На основании представленных результатов можно сделать вывод о том, что вопрос оптимизации SQL-запросов является одним из ключевых в решении проблем, связанных с недостаточной производительностью баз данных. Это связано с сумасшедшими объемами данных, явно превышающих размеры обычных текстовых данных в разы и разы, с которыми работают современные СУБД.

Литература

1. Применение баз данных в современном мире [Электронный ресурс] – Режим доступа: <https://idaten.ru/technology/primenenie-baz-dannih-v-sovremennom-mire>
2. Три аспекта оптимизации (БД и ПО) [Электронный ресурс] – Режим доступа: <https://habr.com/ru/post/349910/>
3. Шварц Б., Зайцев П., Ткаченко В.. MySQL по максимуму. Оптимизация, репликация, резервное копирование. 3-е изд. – СПб.: Питер, 2018. – 864с.
4. Эрик Редмонд, Джим. Р. Уилсон. Семь баз данных за семь недель. Введение в современные базы данных и идеологию NoSQL. Под редакцией Жаклин Картер / Пер. с англ. Слинкин А. А. – М.: ДМК Пресс, 2013. – 384с.

Артеменко О.Г. Повышение эффективности и производительности баз данных путём оптимизации SQL-запросов. Статья посвящена вопросам, рассматривающим значимость оптимизации SQL-запросов для повышения эффективности работы современных баз данных. Приведены примеры методик оптимизации запросов, представлены результаты практических примеров, наглядно демонстрирующих действенность описанных методик. На основании проведённого исследования, приведены зависимости времени выполнения оптимизированных и неоптимизированных запросов от количества элементов в выборке.

Ключевые слова: оптимизация SQL-запросов, база данных, повышение эффективности, производительность, методы оптимизации

Artemenko O.G. Improving the efficiency and performance of databases by optimizing SQL queries. The article is devoted to issues that consider the importance of optimizing SQL queries to improve the efficiency of modern databases. Examples of query optimization techniques are given, the results of practical examples are presented that clearly demonstrate the effectiveness of the described techniques. Based on the study, the dependences of the execution time of optimized and non-optimized queries on the number of elements in the sample are given.

Keywords: SQL query optimization, database, efficiency improvement, performance, optimization methods

Объектный анализ и моделирование автоматизированной системы учета деятельности предприятия

А.В. Боднар^{*1}, Коломойцев В.А.^{*2}

^{*1} к.э.н., доцент, Донецкий национальный технический университет,
linabykova13@ya.ru, Orcid: 0000-0002-1800-0178, SPIN-код: 5068-7409

^{*2} магистрант, Донецкий национальный технический университет
lion-6v@mail.ru

Коломойцев В.А., Боднар А.В. Объектный анализ и моделирование автоматизированной системы учета деятельности предприятия. В работе рассматривается возможность применения технологий программного модуля для подсистемы «Дисконтные карты» в конфигурации «1С.Предприятие. ИБЦКАФЕ». Реализовано предоставление удобного функционала, выявление преимуществ и создание базы данных клиентов, а также сравнение результатов работы программы лояльности в структуре сети заведений общественного питания «Путеводная звезда».

Ключевые слова: дисконтные карты, информация, OLE Automation сервер, Oracle, Borland.

Kolomoitsev V.A., Bodnar A.V. Object analysis and modeling of an automated system for accounting for the activities of an enterprise. The paper considers the possibility of using the technologies of the software module for the subsystem "Discount cards" in the configuration "1C. Enterprise. IBTSKAFE". Providing convenient functionality, identifying advantages and creating a database of customers, as well as comparing the results of the loyalty program in the structure of the network of catering establishments "Guiding Star".

Keywords: discount cards, information, OLE Automation server, Oracle, Borland.

**Полный текст статьи опубликован в выпуске № 3 (29), 2022 г.
научного журнала «Информатика и кибернетика»**

СЕКЦИЯ 3. «ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ, АНАЛИЗ ДАННЫХ И РАСПОЗНАВАНИЕ ОБРАЗОВ»

УДК 004.58

Разработка рекомендательной системы на основе контентно-ориентированной фильтрации для книжного интернет-магазина

О.С. Леонова^{*1}, И.А. Коломойцева^{*2}

^{*1} студент, Донецкий национальный технический университет, loksana864@gmail.com

^{*2} старший преподаватель, Донецкий национальный технический университет, bolatiger@mail.ru

Леонова О.С., Коломойцева И.А. Разработка рекомендательной системы на основе контентно-ориентированной фильтрации для книжного интернет-магазина. Данная работа посвящена разработке рекомендательной системы для книжного интернет-магазина. В работе рассмотрены виды рекомендательных систем, метод построения рекомендаций для данной предметной области. Описана методология и особенности процесса формирования исходных массивов данных. Реализован подход, основанный на методе контентной фильтрации.

Ключевые слова: рекомендательная система, контентно-ориентированная фильтрация, косинусное сходство, книжный интернет-магазин, фильтрация информации.

Введение

Рекомендательная система – комплекс алгоритмов, программ и сервисов, задача которого предсказать, что может заинтересовать того или иного пользователя на основе информации о профиле человека, его более ранних действиях и выборе, предоставляя персонализированные предложения, основанные на истории симпатий пользователя [1]. Рекомендательные системы на данный момент распространены на многих интернет-площадках: новостных лентах, музыкальных сервисах, сайтах с фильмами, – но наибольшее значение рекомендательные системы имеют для интернет-магазинов. По оценкам McKinse в интернет-сервисе Amazon.com за год совершается около 5 млрд. транзакций, при этом 30% совершается на основе рекомендаций системы [2]. Таким образом, приблизительно 125 млн. покупок в месяц совершается, благодаря рекомендательной системе. Кроме того, рекомендательные сервисы могут положительно влиять на конверсию (из посетителей в покупателей), частоту покупок и средний чек (через количество товаров в заказе), что в совокупности ведет к увеличению прибыли компании.

Но, несмотря на такие высокие показатели, по данным DataInsight среди 250 крупнейших российских интернет-магазинов системы рекомендации используют меньше половины [3]. Это можно объяснить программной сложностью реализации рекомендательных систем, дополнительными расходами на разработку и отсутствием универсального решения.

Книжные рекомендательные сервисы менее популярны, чем рекомендательные системы для музыки или фильмов. Особенности разработки рекомендательных систем для книжных сервисов посвящены статьи Прокофьевой И.А., Волковой Л.Л. и Mohd Sabri N. [4,5,6]. Примером успешного книжного интернет-сервиса с рекомендательной системой является компания «ЛитРес». Построение рекомендации основывается на информации из профиля пользователя, анализе его поведения и его ленты новостей, а также предпочтениях, указанных пользователем в самом приложении. Исходя из полученных данных, рекомендательный сервис «ЛитРес» формирует список книг, потенциально интересных пользователю.

Следует отметить, что при разработке рекомендательной системы невозможно использовать уже готовые решения, поскольку рекомендательная система является дополнительным программным модулем, который интегрируется в основное приложение и формирует рекомендации на основе данных, собираемых основными модулями. В зависимости от набора данных, который ему предоставляется модулем базы данных, и обратной связи пользователей, а также от бизнес-процессов и целей компании, алгоритмы рекомендательных систем будут отличаться. Например, невозможно построить рейтинговую систему рекомендаций, если в основном модуле отсутствует функция оценки книг пользователем.

Целью данной статьи является разработка рекомендательной системы для веб-приложения сети книжных магазинов «Буклет». «Буклет» - сеть книжных магазинов, которые работают на локальном рынке Донецкой и

Луганской Народных республик. Предполагаемое количество пользователей веб-сайта составляет 13 тысяч пользователей. Объем книжной продукции в среднем 43 тысячи уникальных книжных наименований.

Описание задачи

Следует отметить, что компанию «Буклет» нельзя сравнивать с «ЛитРес» и другими крупными интернет-сервисами с рекомендательными системами. Данная книжная сеть имеет отличные от других книжных интернет-магазинов бизнес-процессы и ограничения функциональности в некоторых моментах.

Приведем некоторые особенности веб-приложения и бизнес-процессов «Буклета», которые могут выступать ограничивающими факторами при выборе алгоритма рекомендательной системы. Так, например, в веб-приложении на данный момент отсутствует функция оценки книги, отзывы и опрос выбора предпочтений. То есть, в веб-приложении фактически отсутствует явная обратная связь от пользователей.

Обратной связью (*feedback*) пользователя на некоторый объект в рекомендательных системах называют событие, по которому можно судить о предпочтении пользователя к объекту. Обратная связь может быть явной (*explicit feedback*) и неявной (*implicit feedback*). В приложении на данный момент реализована только неявная связь в виде:

- посещения страницы с описанием объекта более одного раза;
- добавления товара в корзину;
- добавления товара в «Избранное».

К этим данным мы будем возвращаться при выборе рекомендательной системы. Также следует уточнить, что данное веб-приложение находится на этапе разработки и еще не наполнено активными пользователями.

Формализуем задачу. Заданы:

- множество пользователей U и множество товаров I ;
- транзакции пользователей в интернет-магазине.

Каждый товар i имеет категорию c (жанр), которая объединяет товары в группы по некоторым характеристикам. Транзакцией в интернет-магазине является добавление в корзину или в «избранное» пользователем u товара i .

Требуется для каждого пользователя $u \in U$ построить список из k наиболее релевантных товаров, каждый из которых отсортирован по убыванию релевантности.

Виды рекомендательных систем

Можно выделить три основных подхода к построению рекомендательных систем:

- 1) контентная фильтрация (*content-based filtering*);
- 2) коллаборативная фильтрация (*collaborative filtering*);
- 3) гибридный подход.

Суть контентной фильтрации заключается в том, что мы сопоставляем пользователей с тем контентом или товарами, которые им нравились в прошлом [7]. То есть, контентно-ориентированные системы выдвигают гипотезу, что, если пользователь интересовался элементом в прошлом, он снова будет заинтересован в нем в будущем.

Методы коллаборативной фильтрации основаны на гипотезе о том, что те, кто одинаково оценивал какие-либо объекты в прошлом, склонны давать похожие оценки в будущем [7]. Такие методы устойчивы и способны выявлять скрытые свойства объектов, что улучшает релевантность рекомендаций. В данном подходе действует схема накопления знаний: чем больше пользователь просматривает и дает оценок, тем более точные и персонализированные рекомендации он получает.

Поскольку веб-приложение на данный момент находится в разработке и не имеет активных пользователей невозможно использовать коллаборативную фильтрацию из-за проблемы «холодного старта». То есть, чтобы построить рекомендательную систему на основании вкусов пользователей, у нас должно быть достаточно информации о взаимодействии пользователь-элемент. Мы не можем давать рекомендации, пока пользователи не взаимодействуют со значительным количеством товаров. Кроме того, даже после сбора необходимой информации, все равно останется проблема составления рекомендаций для новых пользователей в системе. Эти проблемы можно решить, запрашивая у пользователей другой тип данных во время регистрации (пол, возраст, интересы и т. д.) и используя метаинформацию из элементов, связать их с другими существующими элементами в базе данных. Однако данные процессы не интегрированы в существующую функциональность приложения, их добавление повлечет за собой растущие объемы времени на разработку и внесение изменений в уже существующие структуры.

Основанные на содержании методы, в отличие от коллаборативных, гораздо меньше страдают от проблемы «холодного старта». Происходит так, потому что новые пользователи или элементы (продукты, ресурсы и т.д.) могут быть описаны по их характеристикам (содержанию), и, следовательно, могут быть сделаны соответствующие предложения для этих новых элементов [8]. Поэтому было принято решение разработать рекомендательную систему на основе контентно-ориентированной фильтрации.

Набор данных

При формировании собственного набора данных к характеристикам книг были отнесены следующие: название, аннотация, автор, жанр.

В ходе первичной обработки все таблицы были преобразованы в DataFrame-объекты, которые были объединены в один DataFrame, хранящий полную информацию обо всех имеющихся книгах. DataFrame – структура данных, используемая в библиотеке Pandas, представляет собой проиндексированный многомерный массив значений.

После завершения предварительной обработки данных программы был получен массив сведений о 12981 книгах со следующими столбцами: «id», «Название книги», «Аннотация», «Автор», «Список жанров».

Реализация рекомендательного алгоритма фильтрации

Имеются данные о книгах. Задача – сравнить на основе этих данных, насколько две книги похожи между собой. Оценивается похожесть книг на основе аннотации и метаданных (автор, жанры книги). Предварительно была проведена очистка аннотаций от знаков пунктуации и «стоп-слов» (предлоги, суффиксы, междометия, цифры, частицы), поскольку они используются почти в каждой аннотации и не несут смысловой нагрузки.

Поскольку данные имеют текстовый вид, их необходимо преобразовать к векторному виду. Преобразование выполняется с помощью статистической меры TF-IDF, которая используется для определения важности слова в контексте документа [4]. Значимость слова пропорциональна используемости слова в документе и обратно пропорциональна частоте его употребления в других текстах коллекции. Мера TF-IDF определяется следующим выражением (см. формулу 1):

$$feature_i = Count(word_i \in x) \times \log \frac{N}{Count(word_i \in D) + \alpha}, \quad (1)$$

где x – конкретный текст, D – набор текстов, N – количество текстов в наборе, α – сглаживающий гиперпараметр ($=1$), который предотвращает возможное деление на 0. $Count(word_i \in D)$ – число раз, которое i -ое слово встречается во всех текстах.

Таким образом, основная идея данного метода состоит в том, что термин является более важным, если он имеет большое число вхождений в один документ и редко входит в остальные документы коллекции, то есть не является распространённым.

Аннотации разбиваются на токены – слова – после чего вычисляется матрица TF-IDF. Каждая аннотация и метаданные представляется в виде вектора, длина которого равна количеству различных слов (размеру словаря для набора текстов), и содержащего значимость i -го слова словаря для данной аннотации. Для подсчета матрицы TF-IDF использовался класс `TfidfVectorizer` библиотеки `Sklearn`.

Каждое описание книги представляется в виде вектора, схожесть двух книг на основе описаний может быть определена как пространственная близость двух векторов. Мерой близости было выбрано косинусное сходство [9]. Косинусное сходство – значение косинуса угла между двумя векторами в многомерном пространстве, задается следующей формулой (2):

$$similarity(A, B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n A_i^2} \times \sqrt{\sum_{i=1}^n B_i^2}}, \quad (2)$$

Для подсчета косинусного сходства использовался метод `cosine_similarity` библиотеки `Sklearn`. Полученная матрица схожести между векторами позволяет легко ранжировать книги по похожести на исходную – для этого необходимо взять вектор из матрицы схожести, соответствующий заданной книге и отсортировать книги по убыванию значений косинусного расстояния.

На рисунке 1 представлены 10 книг, отобранных таким способом как наиболее похожих на книгу «Основы локальных компьютерных сетей. Учебник для СПО».

```
get_rec('Основы локальных компьютерных сетей. Учебное пособие для СПО', uni_sim)
```

	Название
6356	Инфокоммуникационные системы. Аппаратно...
3453	Инфокоммуникационные системы. Учебник
3945	Создание сайтов на основе WordPress. Уч...
6243	Трубопроводные сети. Учебное пособие
6371	Инфокоммуникационные системы. Программ...
7876	Английский язык для электротехнических ...
6775	Информационная безопасность и защита ин...
10369	Защита электрических сетей напряжением ...
3454	Инфокоммуникационные системы. Учебник

Length: 10, dtype: object [Open in new tab](#)

Рисунок 1 – Рекомендации, полученные на основе схожести аннотаций

Из рисунка 1 видно, что рекомендации в целом соответствуют ожиданиям. Исключение составляет рекомендация книги «Трубопроводные сети. Учебное пособие». Здесь проявляется недостаток данного подхода: рекомендации основаны на похожести употребляемых в аннотации слов, что не всегда говорит о похожести книг.

В примере рекомендации строятся на основе одной книги, но в интернет-магазине необходимо давать рекомендации на основе книг, понравившихся пользователю. Алгоритм построения рекомендаций будет следующим: на вход подается список, сформированный из книг, просмотренных пользователем более двух раз, содержание «Корзины» и содержание «Избранное». Если книга находится одновременно в двух или трех категориях, ей присваивается больший вес. Отбираются только последние 50 элементов, так как вкусы со временем могут меняться. Далее список ранжируется по весу и подается на вход в рекомендательную систему. Для каждой книги вычисляется отдельная матрица схожести, результаты объединяются в общую матрицу, находится вес «схожести» книг, ранжируется по убыванию. Книги с наибольшим весом «схожести» с интересами пользователя попадают в начало рекомендаций.

На рисунке 2 представлены рекомендации для заданного списка книг. В исходном списке была одна книга по основам локальных сетей, 2 книги писателя Р.Толкина и 2 книги по философии. В полученных рекомендациях преимущественно в начале находятся похожие книги Р.Толкина или с упоминанием его имени, а далее книги с философским уклоном и далее книги, схожие по тематике с первой.

```
In 94 1 books = ['Основы локальных компьютерных сетей. Учебное пособие для СПО'],
      2 ['Властелин колец. Братство кольца'], ['Полная история Средиземья'],
      3 ['Занимательная философия: учебное пособие'], ['Несвоевременные размышления.
      4 Эссе']
      5
      6 2 get_rec(books, uni_sim)

Out 94 1 'Хоббит',
      2 'Миры Дж. Р. Р. Толкина. Реальный мир легендарного Средиземья',
      3 'Герои Толкина',
      4 'Конец главы: Роман',
      5 'Воля к власти. Опыт переоценки всех ценностей',
      6 'Падение кумиров. Избранное',
      7 'Утренняя заря',
      8 'Генеалогия морали',
      9 'По ту сторону добра и зла',
     10 'Правые мыслители',
     11 'Инфокоммуникационные системы. Аппаратное обеспечение. Учебник',
     12 'Создание сайтов на основе WordPress. Учебное пособие. СПО',
     13
```

Рисунок 2 – Рекомендации, на основе списка книг пользователя

На основании полученных результатов можно сделать вывод, что выбранный подход дает достаточно удовлетворительные результаты, и его можно внедрять в веб-приложение.

Заключение

В работе рассмотрены виды рекомендательных систем, метод построения рекомендаций для книжного интернет-магазина. Описана методология и особенности процесса формирования исходных массивов данных. Реализован подход, основанный на методе контентной фильтрации.

В продолжении работы над рекомендательной системой предполагается улучшение рекомендательной системы путем объединения методов коллаборативной фильтрации и алгоритмов фильтрации по содержанию. Таким образом, в случае появления нового пользователя в системе, у которого нет истории покупок, будет использоваться рекомендация на основе алгоритмов фильтрации по содержанию, а в случае большого объема статистических данных – строить более точный прогноз, используя методы коллаборативной фильтрации.

Литература

1. Melville P., Mooney R., Nagarajan R. Content-Boosted Collaborative Filtering for Improved Recommendations (англ.) // University of Texas, USA : Материалы конф. / AAAI-02, Austin, TX, USA, 2002. – 2002. – P. 187-192.
2. Mackenzie I., Meyer C., Noble S. How retailers can keep up with consumers. [Электронный ресурс] – Режим доступа: <https://www.mckinsey.com/industries/retail/our-insights/how-retailers-can-keep-up-with-consumers> (дата обращения: 03.11.2022)
3. Интернет-магазины и товарные рекомендации [Электронный ресурс] / Ф.Вирин [и др.] – Режим доступа: <https://oborot.ru/articles/internet-magazin-i-tovarnye-rekomendacii-i35409.html> (дата обращения: 03.11.2022)

4. Прокофьева, И. А. Реализация рекомендательных алгоритмов для книг и литературных произведений / И. А. Прокофьева, О. А. Гордеева, М. А. Кудрина // Труды международного симпозиума "Надежность и качество". – 2022. – Т. 1. – С. 211-216.
5. Волкова, Л. Л. О разработке рекомендательной системы, предлагающей книги по предпочтениям пользователя / Л. Л. Волкова, М. М. Токарева, А. А. Ланко // Новые информационные технологии в автоматизированных системах. – 2017. – № 20. – С. 239-244.
6. Mohd Sabri N., Jaffar N. A. Book recommendation based on collaborative filtering technique //ESTEEM Academic Journal. – 2022. – Т. 18. – С. 92-103.
7. Кутянин А. Р. Рекомендательные системы: обзор основных постановок и результатов //Интеллектуальные системы. Теория и приложения. – 2017. – Т. 21. – №. 4. – С. 18-30.
8. Коврик, К. А. Об особенностях разработки рекомендательной системы для пользователей мессенджера Telegram / К. А. Коврик, И. А. Коломойцева // Программная инженерия: методы и технологии разработки информационно-вычислительных систем (ПИИВС-2020) : сборник научных трудов III Международной научно-практической конференции (студенческая секция), Донецк, 25–26 ноября 2020 года. Том 2. – Донецк: Донецкий национальный технический университет, 2020. – С. 166-170.
9. Similarity and Distance Metrics for Data Science and Machine Learning [Электронный ресурс] – Режим доступа: <https://medium.com/dataseries/similarity-and-distance-metrics-for-data-science-and-machine-learning-e5121b3956f8> (дата обращения: 08.11.2022).

Леонова О.С., Коломойцева И.А. Разработка рекомендательной системы на основе контентно-ориентированной фильтрации для книжного интернет-магазина. Данная работа посвящена разработке рекомендательной системы для книжного интернет-магазина. В работе рассмотрены виды рекомендательных систем, метод построения рекомендаций для данной предметной области. Описана методология и особенности процесса формирования исходных массивов данных. Реализован подход, основанный на методе контентной фильтрации.

Ключевые слова: рекомендательная система, контентно-ориентированная фильтрация, косинусное сходство, книжный интернет-магазин, фильтрация информации.

Leonova Oksana, Kolomoitseva Irina. Development of a recommendation system based on content-oriented filtering for an online bookstore. This work is devoted to the development of a recommendation system for an online bookstore. The paper considers the types of recommendation systems, the method of building recommendations for this subject area. The methodology and features of the process of forming the initial data arrays are described. An approach based on the content filtering method is implemented.

Key words: recommendation system, content-oriented filtering, cosine similarity, online bookstore, information filtering.

Прогнозирование стоимости недвижимости на основе модели линейной регрессии

И.В. Савицкая*1, Р.В. Лазарский*2

*1 ассистент, Донецкий национальный технический университет
savitskaya.irina.v@gmail.com, SPIN-код: 1584-6161

*2 бакалавр, Донецкий национальный технический университет
rus.lazarskiy@mail.ru

Савицкая И.В., Лазарский Р.В. Прогнозирование стоимости недвижимости на основе модели линейной регрессии. В статье определена важность значения прогнозирования в сфере недвижимости. Представлено краткое описание основных моделей прогнозирования и подробное описание метода линейной регрессии. Рассмотрен метод градиентного спуска для минимизации функции потерь. Определена проблема переобучения и описаны основные методы регуляризации в линейной регрессии.

Ключевые слова: недвижимость, прогнозирование, машинное обучение, линейная регрессия, регуляризация.

Savitskaya I.V., Lazarskiy R.V. Forecasting the price of real estate based on a linear regression model. The article defines the importance of the value of forecasting in the field of real estate. A brief description of the main forecasting models and a detailed description of the linear regression method are presented. The gradient descent method for minimizing the cost function is considered. The overfitting problem is defined and the main methods of regularization in linear regression are described.

Keywords: real estate, forecasting, machine learning, linear regression, regularization.

Введение

В современном мире задача прогнозирования является одним из наиболее значимых факторов в управлении, принятии решения и планировании в любой сфере человеческой деятельности. Прогнозирование является научно обоснованным суждением о будущих положениях объекта и(или) о других возможных способах достижения этого положения. Необходимость прогнозирования обусловлена тем обстоятельством, что положения объекта в будущем являются важными для решений, принимаемых в настоящий момент [1].

Рынок недвижимости – один из сегментов рынка, который специализируется на сделках с недвижимостью и сегодня занимает важное место в жизни людей. Эта отрасль экономики в последние годы отметилась стремительным развитием, что эксперты объясняют увеличением и распространением ипотечных кредитов среди населения, оживлением поиска инвестиционной недвижимости инвесторами, увеличением темпов строительства и рядом других факторов. Следует также отметить то, что происходит рост цен на недвижимость, что повышает ее привлекательность как объекта инвестиций и источника коммерческой прибыли. Соответственно, целью приобретения квартир, помимо жилья, является получение дохода от их пользования и последующей продажи [2].

Таким образом, выбор темы данной статьи определен ролью, которую жилая недвижимость играет в экономической и социальной жизни человеческого общества, а также наличием задач, которые появляются в ходе оценки стоимости объектов жилой недвижимости с учётом множества признаков.

Цель статьи – представление метода линейной регрессии для анализа и прогнозирования стоимости недвижимости.

Выбор метода

Прогнозирование на рынке недвижимости может осуществляться эвристическим и статистическим методами.

Эвристический – метод прогнозирования цен на объекты недвижимости состоит в экспертном анализе количественных и качественных данных, влияющих на изменения цен (арендных ставок), и выявлении результирующей тенденции.

Этапы прогнозирования:

1) анализ макроэкономических прогнозов изменения и выявление тенденции изменения цен и арендных ставок;

2) анализ социально-экономических показателей в регионе исследования;

3) анализ рынка недвижимости, определение уровня его развития;

4) общее заключение о результирующей тенденции на рынке недвижимости.

Статистический метод состоит в использовании динамического ряда значений уровня цен для построения статистической модели процесса и получения на ее основе экстраполяционного прогноза.

Для реализации данного метода требуется накопленная база статистических данных о динамике цен за длительный период времени. Аналитик должен построить модель прогнозирования, подобрав подходящее уравнение тренда. Все разнообразие тенденций развития на рынке недвижимости может быть сведено к шести основным формам тренда [3].

1. Линейная форма тренда.

$$\hat{y} = a_0 + a_1 t, \quad (1)$$

где a_0 – начальный уровень тренда в момент или период, принятый за начало отсчёта времени t , а – константа прямолинейного тренда.

Линейный тренд хорошо отражает тенденцию пропорционального изменения цен на рынке недвижимости. Равнодействующая этих факторов часто выражается в прямолинейном тренде.

2. Параболическая форма тренда.

$$\tilde{o} = \dot{a}_0 + \dot{a}_1 t + a_2 t^2 \quad (2)$$

где a_i – постоянная параболическая тренда.

Параболическая форма выражает ускоренное или замедленное изменение стоимости недвижимости с постоянным ускорением.

3. Экспоненциальная форма тренда.

$$\tilde{o} = \dot{a} k^t \quad (3)$$

где k – темпы изменения в рядах.

Если $k > 1$, экспоненциальный тренд выражает тенденцию ускоренного и все более ускоряющегося возрастания стоимости недвижимости.

Если $k < 1$, экспоненциальный тренд отражает тенденцию постоянно и все более замедляющегося снижения стоимости недвижимости.

4. Тренд в форме степенной кривой.

$$\tilde{o} = a_0 t^{a_1} \quad (4)$$

Степенная форма используется для изображения изменений с разной мерой пропорциональности изменений во времени.

5. Гиперболическая форма тренда.

$$\hat{y} = a_0 + \frac{a_1}{t} \quad (5)$$

Если $a > 0$, гиперболический тренд выражает тенденцию замедляющегося снижения уровней, если $a < 0$ – тенденцию замедляющегося роста уровней.

6. Логистическая форма тренда.

$$\hat{y} = \frac{y_{\max} - y_{\min}}{e^{a_0 + a_1 t} + 1} + y_{\min} \quad (6)$$

где e – основание натуральных логарифмов, $y_{\max}$, $y_{\min}$ – максимальное и минимальное значения уровней ряд.

Машинное обучение – это использование математических моделей данных, которые помогают компьютеру обучаться без непосредственных инструкций. Оно считается одной из форм искусственного интеллекта (ИИ). При машинном обучении с помощью алгоритмов выявляются закономерности в данных. На основе этих закономерностей создается модель данных для прогнозирования. Аналитические системы могут учиться выявлять закономерности и принимать решения с минимальным участием человека. Чем больше данных обрабатывает такая модель и чем дольше она используется, тем точнее становятся результаты. Это очень похоже на то, как человек оттачивает навыки на практике.

Основным методом прогнозирования и анализа рынка недвижимости является анализ данных о недвижимости с использованием модели линейной регрессии. Модель множественной линейной регрессии является распространенным многомерным статистическим методом. Это алгоритм контролируемого машинного обучения, в котором прогнозируемый результат непрерывен с постоянным наклоном. Он используется для прогнозирования значений в непрерывном диапазоне вместо классификации значений по категориям [4]. Она

обладает преимуществами удобного приложения и простой модели. Поэтому широко используется в повседневном производстве и различных научных исследованиях.

В частности, рынок недвижимости анализируется в основном с точки зрения стоимости объектов недвижимости. В этом случае цена действует как индикатор результатов. Такие параметры, как общая площадь объекта, количество комнат в нем, этаж, удаленность от центра города, год постройки, количество квартир (т.е. соседей) в доме, основной материал дома, средний размер платы за ЖКУ, средний рейтинг пользователей сайтов с объявлениями об объекте и многое другое являются факторными показателями. Необходимость анализа рынка недвижимости на основе модели множественной регрессии объясняется тем, что с ее помощью можно определить факторы, влияющие на стоимость приобретения недвижимости. Такая модель способна представить процесс ценообразования на рынке недвижимости в пределах определенной территории (региона, города или городской местности). И это очень важно для простого гражданина, который, выбирая жилье, учитывает многие качественные и количественные характеристики своей будущей квартиры.

Линейная регрессия

Линейная регрессия – это подход к моделированию взаимосвязи между одной зависимой переменной и одной или несколькими объясняющими переменными, непрерывными или дискретными. Этот метод помогает в прогнозировании будущих значений одной переменной с помощью другой переменной. Это позволяет предсказать, какой вес одна переменная имеет для другой переменной, используя прошлые данные обеих переменных [5]. На рисунке 1 показано, как линейная регрессия аппроксимирует взаимосвязь между объектами на оси x и y:

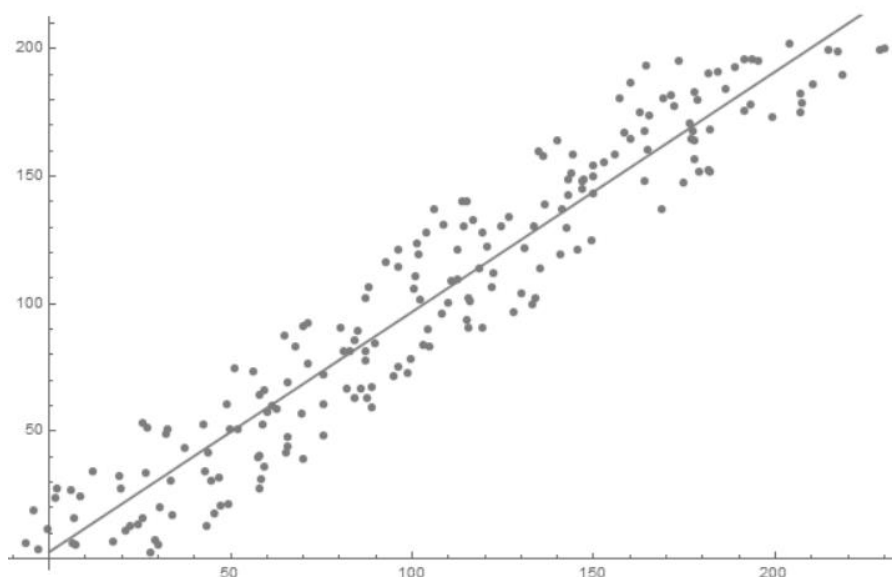


Рисунок 1 – Пример линейной регрессии

Определим линейную регрессию с помощью формулы:

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n, \quad (7)$$

где $\hat{y}$ – прогнозируемое значение, n – количество признаков, x_i – значение i -го признака, θ_j – параметр j -й модели или вес. Также θ_0 известен как термин смещения.

Чтобы обучить модель, вначале необходимо измерить, насколько хорошо модель соответствует данным обучения. Для этой цели обычно используют функцию затрат среднеквадратичной ошибки (MSE), которая определяется с помощью формулы:

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m (\theta^T (x^{(i)}) - y^{(i)})^2, \quad (8)$$

где m – указывается количество выборок, $y^{(i)}$ является реальным значением i -й выборки. Будем использовать MSE для измерения среднеквадратичной разницы между оценочными значениями и реальными значениями.

Для нахождения значения θ , которое минимизирует функцию затрат, представим метод градиентного спуска.

Градиентный спуск – это метод машинного обучения, который обычно используется в контексте оптимизации. Будем использовать этот алгоритм для нахождения минимума функции путем итеративного

нахождения направления с самым крутым наклоном относительно текущего местоположения. Вычислим самый крутой наклон к текущему местоположению, используя градиентный спуск.

В частности, если функция затрат MSE имеет $J(\theta)$ вид формулы 8, тогда градиент равен:

$$\frac{\partial J(\theta)}{\partial \theta} = \frac{2}{m} \sum_{i=1}^m [(\theta^T(x^{(i)}) - y^{(i)})x^{(i)}]. \quad (9)$$

После этого обновим веса с помощью градиента, умноженного на скорость обучения α :

$$\theta := \theta - \alpha * \frac{\partial J(\theta)}{\partial \theta}. \quad (10)$$

На рисунке 2 показаны шаги спуска с холма для нахождения локального минимума.

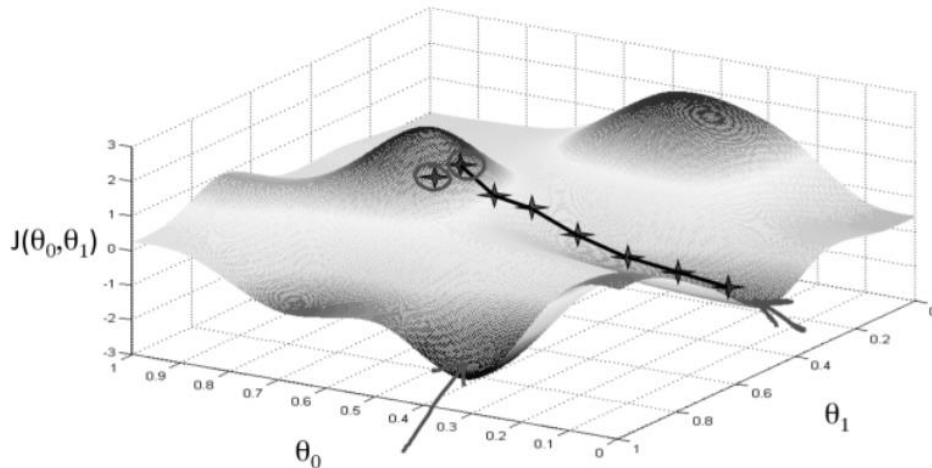


Рисунок 2 – Пример градиентного спуска

В рассматриваемом случае этот подход приводит к глобальному минимуму, потому что функция потерь является выпуклой функцией и имеет только один локальный минимум, который также является и глобальным. Градиентный спуск распространен и широко применяется во многих методах машинного обучения, включая нейронные сети. Несмотря на это, у него есть некоторые недостатки.

Поскольку градиентный спуск является итеративным подходом, для достижения оптимального решения может потребоваться время. Кроме того, сходимость зависит от скорости обучения α .

Проблема переобучения и её решение

Переобучение – негативное явление, возникающее, когда алгоритм обучения вырабатывает предсказания, которые слишком близко или точно соответствуют конкретному набору данных и поэтому не подходят для применения алгоритма к дополнительным данным или будущим наблюдениям [6].

Регуляризация – это метод машинного обучения, который пытается добиться обобщения модели. Это означает, что модель хорошо работает не только с обучающими или тестовыми данными, но и с данными, которые она получит в будущем. Таким образом, для достижения этой цели регуляризация уменьшает веса до нуля, чтобы препятствовать сложным моделям. Соответственно, это позволяет избежать переобучения и уменьшает дисперсию модели.

Существует три основных метода регуляризации в линейной регрессии:

1. Регрессия лассо
2. Гребневая регрессия
3. Эластичная сеть

Регрессия лассо, или регуляризация L1, представляет собой метод, который увеличивает функцию затрат на штраф, равный сумме абсолютных значений весов без перехвата из линейной регрессии. Формально определим регуляризацию L1 как:

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m (\theta^T(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^n |\theta_j|, \quad (11)$$

где λ – параметр регуляризации.

В основном, λ контролирует степень регуляризации. В частности, чем выше значение λ , тем меньше становятся веса. Чтобы найти лучшее λ , можно начать с $\lambda = 0$ и измерить ошибку перекрестной проверки на каждой итерации, увеличивая λ с фиксированным значением.

Аналогично регрессии лассо, *регрессия гребня* или регуляризация L2, добавляет штраф к функции затрат. Единственное отличие состоит в том, что штраф рассчитывается с использованием квадратов значений весов, не связанных с перехватом, из линейной регрессии. Таким образом, определим регуляризацию L2 как:

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m (\theta^T (x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^n \theta_j^2 \quad (12)$$

Как и в случае с регуляризацией L1, можно оценить параметр таким же образом. Единственное различие между лассо и гребнем заключается в том, что гребень сходится быстрее, в то время как лассо чаще используется для выбора объектов.

Эластичная сетевая регрессия представляет собой комбинацию двух предыдущих подходов; он сочетает в себе штрафы как от регрессии гребня, так и от регрессии лассо. Поэтому определим эластичную сетевую регуляризацию по формуле:

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m (\theta^T (x^{(i)}) - y^{(i)})^2 + \lambda_1 \sum_{j=1}^n |\theta_j| + \lambda_2 \sum_{j=1}^n \theta_j^2 \quad (13)$$

В отличие от предыдущих двух методов, в данной формуле есть два параметра регуляризации, λ_1 и λ_2 . Следовательно, необходимо найти оба параметра.

Заключение

В данной статье рассмотрены основные модели прогнозирования, шесть основных форм трендов тенденций развития на рынке недвижимости, дано понятие машинному обучению, подробно разобран термин регрессия, перечислены ее типы и особое внимание уделено линейной регрессии. В том числе, рассмотрены три наиболее часто используемых метода регуляризации и способ нахождения параметра регуляризации. Приведены формулы, лежащие в основе методов линейной регрессии и регуляризации.

Литература

1. Озеров, Е.С. Экономический анализ и оценка недвижимости / Е.С. Озеров. – Санкт-Петербург: Питер, 2007. – 536 с.
2. Анализ рынка недвижимости на основе модели множественной регрессии – общее представление, использование модели и применение [Электронный ресурс] / Анна Евкова // Evkova – Электрон. дан. – 2016. – Режим доступа: <https://www.evkova.org/analiz-ryinka-nedvizhimosti-na-osnove-modeli-mnozhestvennoj-regressii-obschee-predstavlenie-ispolzovanie-modeli-i-primenenie> – Загл. с экрана.
3. Модели прогнозирования на рынке недвижимости [Электронный ресурс] // Studme – Электрон. дан. – 2022. – Режим доступа: https://studme.org/211024/ekonomika/modeli_prognozirovaniya_rynke_nedvizhimosti – Загл. с экрана.
4. House price prediction using linear regression [Электронный ресурс] / Simran Kaur // linuxhint – Электрон. дан. – 2021. – Режим доступа: <https://linuxhint.com/house-price-prediction-linear-regression/> – Загл. с экрана.
5. How to calculate the regularization parameter in linear regression [Электронный ресурс] / Enes Zvornichanin // Baeldung – Электрон. дан. – 2021. – Режим доступа: <https://www.baeldung.com/cs/regularization-parameter-linear-regression#2-gradient-descent> – Загл. с экрана.
6. Переобучение [Электронный ресурс] // Ifmo – Электрон. дан. – 2022. – Режим доступа: <https://neerc.ifmo.ru/wiki/index.php?title=Переобучение> – Загл. с экрана.

Савицкая И.В., Лазарский Р.В. Прогнозирование стоимости недвижимости на основе модели линейной регрессии. В статье определена важность значения прогнозирования в сфере недвижимости. Представлено краткое описание основных моделей прогнозирования и подробное описание метода линейной регрессии. Рассмотрен метод градиентного спуска для минимизации функции потерь. Определена проблема переобучения и описаны основные методы регуляризации в линейной регрессии.

Ключевые слова: недвижимость, прогнозирование, машинное обучение, линейная регрессия, регуляризация.

Savitskaya I.V., Lazarskiy R.V. Forecasting the price of real estate based on a linear regression model. The article defines the importance of the value of forecasting in the field of real estate. A brief description of the main forecasting models and a detailed description of the linear regression method are presented. The gradient descent method for minimizing the cost function is considered. The overfitting problem is defined and the main methods of regularization in linear regression are described.

Keywords: real estate, forecasting, machine learning, linear regression, regularization.

Алгоритм формирования тезауруса для расширения запроса информационно-поисковой системы на примере предметной области «Программирование»

И.А. Коломойцева

ст. преподаватель кафедры программной инженерии им. Л.П. Фельдмана, Донецкий национальный технический университет, bolatiger@mail.ru,

Коломойцева И.А. Алгоритм формирования тезауруса для расширения запроса информационно-поисковой системы на примере предметной области «Программирование». В статье приведена схема ИПС с использованием тезауруса. Приведено определение информационно-поискового тезауруса и указаны основные особенности его структуры. Описана структура тезауруса для предметной области «Программирование». Представлен алгоритм автоматизированного формирования такого тезауруса на основе корпуса текстов по программированию. Сделан вывод о том, что использование тезауруса приведет к увеличению точности поиска.

Ключевые слова: тезаурус, информационно-поисковая система, корпус текстов, терм, вес термина, программирование.

Kolomoitseva I.A. The algorithm for the formation of the thesis to expand the request of the information retrieval system using the example of the subject area "Programming". The article shows the IPS scheme using the thesis. The definition of information-search Thesaurus is given and the main features of its structure are indicated. The structure of Thesaurus for the subject area "Programming" is described. The algorithm of the automated formation of such a thesis on the basis of the programming body is presented. It is concluded that the use of Thesaurus will lead to an increase in the accuracy of the search.

Keywords: thesaurus, information-search system, text of the text, term, thermal weight, programming.

Введение

Объём электронных документов, с которыми сталкивается человек, постоянно растёт. И при этом всё большую актуальность приобретает задача поиска информации.

Информационный поиск (Information Retrieval, IR) – это процесс поиска в большой коллекции некоего неструктурированного материала (обычно – документа), удовлетворяющего информационные потребности [4]. При этом информационная потребность – это тема, которой реально интересуется пользователь (следует её отличать от информационного запроса).

На полноту информационного поиска влияет синонимия – явление, при котором одно и то же понятие выражается разными словами. Обычно пользователь, увидев, что запрос не удовлетворил его информационную потребность, переформулирует (расширяет) запрос самостоятельно словами-синонимами или словами, относящимися к интересующей его предметной области. Процесс расширения (переформулирования) запроса можно автоматизировать с использованием тезаурусов. Существующие тезаурусы, такие как тезаурусы типа WordNet, являются словарями общей лексики и не учитывают особенности различных предметных областей, например, программирования.

Составление тезауруса является очень трудоёмкой задачей. Для узкой предметной области процесс составления тезауруса можно автоматизировать.

Целью данной статьи является описание алгоритма автоматизированного формирования тезауруса для предметной области «Программирование».

Понятие информационно-поисковой системы

Важным источником информации для информационного поиска являются документы. Документы (documents) – это любые объекты, на основе которых решено построить систему информационного поиска [4]. В рамках статьи рассматриваются текстовые неструктурированные документы.

Группа документов, по которой осуществляется поиск, называется коллекцией (collection), или корпусом (corpus), или массивом текстов (body of texts) [4].

Самая распространённая задача информационного поиска – это поиск по произвольному запросу (ad hoc retrieval). Цель задачи – найти в коллекции документы, которые являются наиболее релевантными по отношению к произвольным информационным потребностям, выражаемым при помощи однократных запросов пользователей [4]

Информационная потребность (Information need) – это тема, о которой пользователь желает знать больше (следует отличать её от запроса) [4]

С точки зрения пользователя, документ является релевантным, если он удовлетворяет его информационную потребность.

Такая релевантность называется содержательной.

Но большинство существующих информационно-поисковых систем реализуют формальную релевантность – соответствие, получаемое путем сравнения модели поискового запроса с моделью документа по определенному алгоритму.

Общая схема информационно-поисковой системы (ИПС) представлена на рисунке 1.

В классической информационно-поисковой системе, если найденные документы не удовлетворяют информационную потребность пользователя, он переформулирует запрос и опять запускает процедуру поиска. Пользователь ИПС повторяет эту процедуру до тех пор, пока не найдёт нужные документы или пока не откажется от поиска.

Существуют автоматические или полуавтоматические (с участием пользователя) способы уточнения (расширения) запросов. Методы уточнения запросов делятся на две категории: локальные и глобальные [2, 4]. Методу уточнения запроса с помощью тезауруса относятся к глобальным. Расширение запроса улучшает такие характеристики ИПС как полноту, а иногда и точность.

Точность (precision) определяет, какая доля документов из найденного набора является релевантной по отношению к информационной потребности [1, 4].

Полнота (recall) определяет, какая доля релевантных документов из коллекции возвращена системой [1, 4].

Схема информационно-поисковой системы, использующей для расширения (уточнения) запроса тезаурус приведена на рисунке 2.



Рисунок 1 – Общая схема информационно-поисковой системы

Понятие информационно-поискового тезауруса

Тезаурус – это словарь, в котором слова и словосочетания с близкими значениями сгруппированы в единицы, называемые понятиями, концептами, или дескрипторами, и в котором явно (в виде отношений, иерархии) указываются семантические отношения между этими понятиями (концептами, дескрипторами) [3].

Среди тезаурусов можно выделить информационно-поисковые тезаурусы. По [3] информационно-поисковый тезаурус – это нормативный словарь, явно указывающий отношения между терминами и предназначенный для описания содержания документов и поисковых запросов.

Самой важной задачей при разработке информационно-поискового тезауруса является отбор терминов [3].

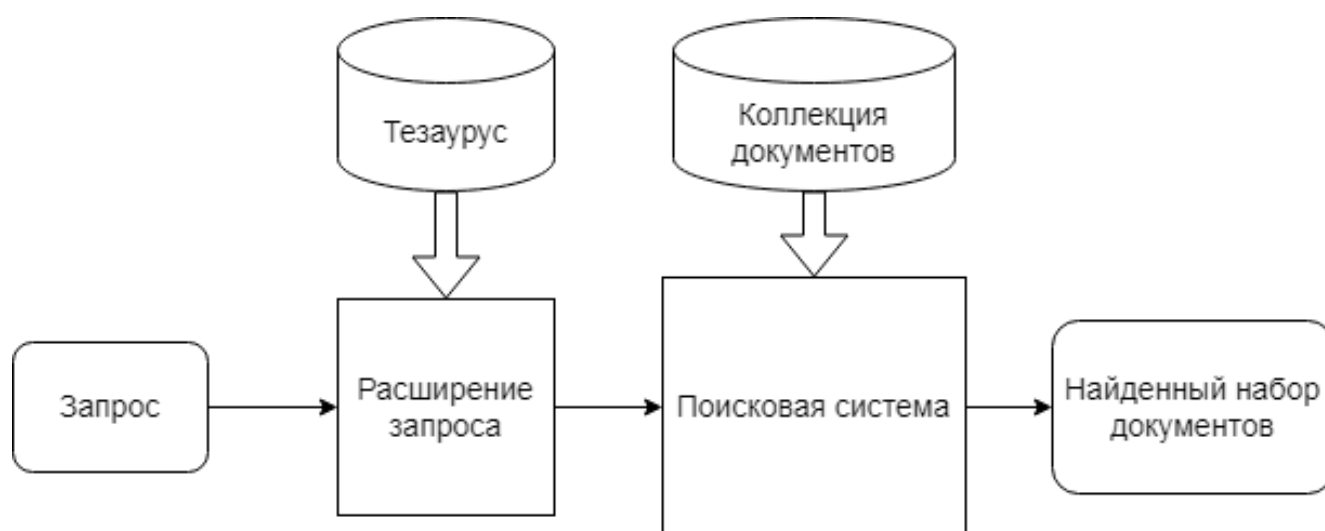


Рисунок 2 – Общая схема информационно-поисковой системы с тезаурусом

Пути отбора терминов [3]:

- 1) найти тезаурусы из близких предметных областей и отобрать подходящие термины из них;
- 2) получить термины в результате опроса экспертов предметной области;
- 3) получить термины из текстов предметной области с помощью автоматизированных методов или ручной обработки документов.

При этом из тезаурусов следует исключить слишком частотные термины как малоинформативные для целей поиска.

Тезаурус можно рассматривать как вид онтологии.

Информационно-поисковые тезаурусы относятся к онтологиям с преимущественно таксономическими отношениями [3]. Таксономические отношения - это отношения вида класс-подкласс.

Алгоритм построения словаря термов для тезауруса

Наполнение тезауруса по предметной области «Программирование» выполняется в полуавтоматическом режиме.

Для этого используется набор документов по предметной области «Программирование». В настоящий момент этот корпус текстов содержит около 1500 документов. На основе этого корпуса текстов строится словарь термов, которые помещаются в структуру тезауруса, используемого для расширения запросов.

При построении словаря термов выполняются следующие этапы:

- 1) выделение термов в документах;
- 2) удаление стоп-слов;
- 3) нормализация термов;
- 4) лемматизация термов;
- 5) формирование словаря термов;
- 6) расчёт весов для термов;
- 7) разметка словаря термов вручную.

Рассмотрим эти этапы подробнее.

Под термом будем понимать единицу словаря. Это может одно слово или устойчивое словосочетание, так называемая n-грамма.

В описываемом подходе на 1-м этапе происходит разбиение документа на отдельные слова, при этом в качестве разделителя выступают пробелы. Также строятся по каждому документу списки с разбиениями по 2, 3 и 4 слова, чтобы можно было выделить термы «Операционная система», «Язык программирования», «Объектно-ориентированное программирование» и т.п.

Все рассмотренные ниже этапы применяются ко всем построенным спискам термов.

На втором этапе выполняется удаление стоп-слов. К стоп-словам относятся слова, которые сами по себе не несут смысла. Например, предлоги, союзы, частицы такие, как «и», «а», «но», «в» «на» и т.п. В последнее время в информационно-поисковых системах стоп-слова не удаляют, так как они уточняют смысл запроса [5]. Но учитывая, что словарь термов используется для построения тезауруса, участвующего в процессе расширения запросов, стоп-слова можно безболезненно удалить, так как добавление их в запрос не приведёт к увеличению ни полноты, ни точности поиска. В рассматриваемом алгоритме стоп-слова собраны в отдельный словарь-список. В него входят, кроме предлогов, союзов и частиц ещё и местоимения. Первоначально этот список составлен

вручную. В настоящий момент разрабатывается алгоритм пополнения этого списка общеупотребимыми словами, добавление которых к запросу не приводит к увеличению точности или полноты поиска.

К нормализации термов относится удаление знаков препинания, приведение термов к нижнему регистру, а также аббревиатур к одному виду.

Существует два способа при построении словарей на основе корпуса текстов сократить их объём – стемминг и лемматизация. Стемминг – это поиск общей основы слова за счёт удаления изменяющей части [5]. Лемматизация – это способ построения канонической формы слов с использованием грамматических правил [5]. Стемминг работает быстрее, чем лемматизация, но при использовании стемминга некоторые слова могут быть потеряны. Например, после применения стемминга к словам «день» и «дня» останется одна буква «д». Поэтому в данном алгоритме для сокращения объёма словаря используется лемматизация.

После того, как каждый документ корпуса представлен в виде списка нормализованных и прошедших лемматизацию термов строится общий словарь термов корпуса путём объединения списков термов документов и удаления из них дубликатов.

Для того, чтобы определить важность термов, необходимо рассчитать их веса. После того, как будет установлено, какие из термов являются синонимами, при распределении термов в рамках структуры тезауруса на основе этого веса будет определён главный синоним.

Существует несколько способов расчёта весов термов.

Самый простой способ задания веса:

$$tf_i^{(j)}$$

(tf – term frequency – частота термина) – количество (частота) вхождений термина t_i в документе d_j [4].
Используется также нормализованная частота [7]:

$$w_i^{(j)} = tf_i^{(j)} / \max_{1 \leq k \leq n} tf_k^{(j)}$$

Использование tf не учитывает, в каком количестве документов из коллекции текстов встретился терм. Если какое-то слово встречается во всех или почти всех документах, его важность в запросе желательно снизить до нуля. В этом случае для определения веса термина в документе используется обратная документная частота (inverse document frequency) [4]:

$$w_i^{(j)} = idf_i = \log \frac{N}{n_i},$$

где N – количество документов в коллекции, (или – document frequency – документная частота) – количество документов в коллекции, содержащих терм

Также при определении веса учитывается одновременно и tf, и idf [4, 6]:

$$w_i^{(j)} = tf_idf_i^{(j)} = tf_i^{(j)} \cdot df_i.$$

В рассматриваемой системе веса термов рассчитываются с помощью метрики tf_idf.

Из списков с n-граммами все словосочетания со значением метрики tf_idf меньше 0,1 отбрасываются.

После построения словарей термов с весами они вручную размечаются. Указывается какой терм к какому понятию тезауруса относится. После этого термы автоматически заносятся в структуру тезауруса.

Структура тезауруса для предметной области «Программирование»

Структуру тезауруса для предметной области «Программирование» можно представить в виде дерева [7].
Фрагмент этого дерева приведен на рисунке 3.

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.

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) (одноточные, многоточные).

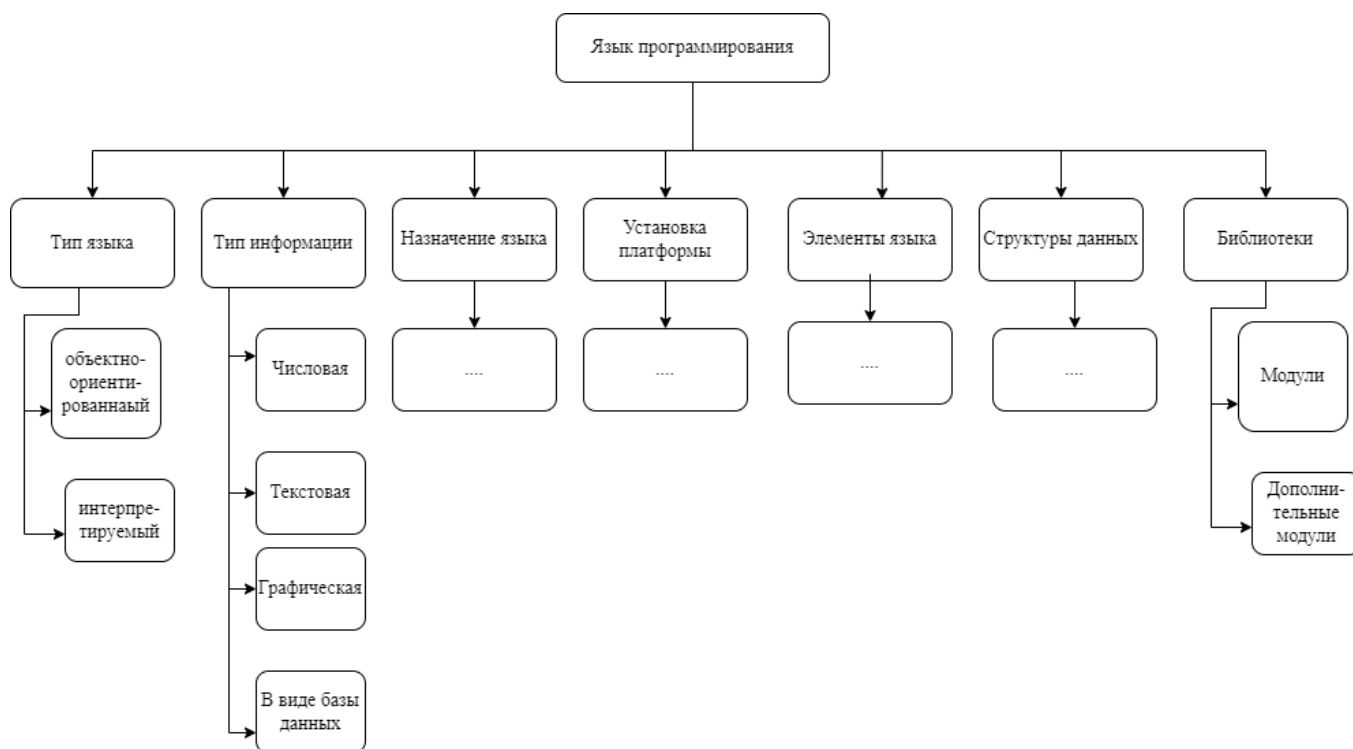


Рисунок 3 – Фрагмент структуры тезауруса для предметной области «Программирование»

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 соответствии 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

В статье приведено определение основных элементов информационно-поисковой системы. Дана схема ИПС с использованием тезауруса. Приведено определение информационно-поискового тезауруса и указаны основные особенности его структуры. Описан алгоритм автоматизированного построения тезауруса на основе корпуса естественно-языковых текстов по программированию. Приведена структура тезауруса для предметной области «Программирование». Сделан вывод о том, что использование тезауруса приведет к увеличению точности поиска.

Литература

1. Додонов, А.Г. Компьютерные сети и аналитические исследования /А.Г. Додонов, Д.В. Ландэ, В.Г. Путятин. – К.: ИПРИ НАН Украины, 2014. – 486 с.
2. Коломойцева, И.А. Анализ способов расширения запроса информационно-поисковой системы / И.А. Коломойцева // Материалы VII Международной научно-технической конференции «Современные

информационные технологии в образовании и научных исследованиях (СИТОНИ-2021)». – Донецк: ДонНТУ. – С. 350-357.

3. Лукашевич, Н.В. Тезаурусы в задачах информационного поиска. – М.: Издательство Московского университета, 2011. – 512 с.

4. Маннинг, К.Д. Введение в информационный поиск / К.Д. Маннинг, П. Рагхаван, Х. Шютце. – М.: ООО «И.Д. Вильмс», 2011. – 528 с.

5. Лейн, Х. Обработка естественного языка в действии / Х. Лейн, Х. Хапке, К. Ховард. — СПб.: Питер, 2020. — 576 с.

6. Ландэ Д.В. Интернетика. Навигация в сложных сетях: модели и алгоритмы/ Д. В. Ландэ, А. А. Снарский, И. В. Безсуднов. – М.: Книжный дом «ЛИБРОКОМ», 2009. – 264 с.

7. 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.. – Донецк: ДонНТУ. – С. 155-159.

Коломойцева И.А. Алгоритм формирования тезауруса для расширения запроса информационно-поисковой системы на примере предметной области «Программирование». В статье приведена схема ИПС с использованием тезауруса. Приведено определение информационно-поискового тезауруса и указаны основные особенности его структуры. Описана структура тезауруса для предметной области «Программирование». Представлен алгоритм автоматизированного формирования такого тезауруса на основе корпуса текстов по программированию. Сделан вывод о том, что использование тезауруса приведет к увеличению точности поиска.

Ключевые слова: тезаурус, информационно-поисковая система, корпус текстов, терм, вес терма, программирование.

Kolomoitseva I.A. The algorithm for the formation of the thesis to expand the request of the information retrieval system using the example of the subject area "Programming". The article shows the IPS scheme using the thesis. The definition of information-search Thesaurus is given and the main features of its structure are indicated. The structure of Thesaurus for the subject area "Programming" is described. The algorithm of the automated formation of such a thesis on the basis of the programming body is presented. It is concluded that the use of Thesaurus will lead to an increase in the accuracy of the search.

Keywords: thesaurus, information-search system, text of the text, term, thermal weight, programming.

Распознавание лиц человека статистическими методами из пакета Matlab

Д.В. Дручевский^{*1}, О.И. Федяев^{*2}

^{*1} магистрант, Донецкий национальный технический университет,
eldercs312@yandex.com

^{*2} к.т.н, доцент, Донецкий национальный технический университет,
fedyayev@donntu.ru, OrcID: 0000-0001-6822-7306

Дручевский Д.В., Федяев О.И. Распознавание лиц человека статистическими методами из пакета Matlab. Статья посвящена разработке программной модели распознавания людей по изображению лиц с использованием алгоритма Eigenface и инструментов пакета MatLab. Алгоритм распознавания основан на статистическом методе главных компонент. Этапы подготовки и распознавания в модели реализованы на программном уровне и описаны математически. Анализ качества распознавания проведен на разных объемах обучающего множества.

Ключевые слова: распознавание образов, лицо человека, метод главных компонент, пакет Matlab, алгоритм Eigenface.

Введение

Обработка изображений всегда была активной областью исследований. В области обработки изображений существует множество усовершенствований, инноваций, разработок и модификаций. В настоящее время научно-практическая деятельность в области компьютерного зрения постоянно расширяется, наполняясь новыми идеями и разработками. По оценкам специалистов среднестатистический человек может идентифицировать знакомое лицо в толпе с точностью 97,53%. Но это меньше по сравнению с современными алгоритмами, которые добились точности 99,8%. А в последние несколько лет они достигли практически совершенства [1].

Существует много способов распознавания лица: метод гибкого сравнения на графах, нейронные сети, скрытые марковские модели, статистические методы и другие [2]. Все они по-разному решают проблемы, влияющие на качество систем распознавания – это изменение освещённости, положения головы, размера лица и др. В последние годы эффективные системы распознавания были построены на основе свёрточных нейронных сетей [2]. Для упрощения создания систем распознавания создано несколько специализированных библиотек (TensorFlow, Keras), предоставляющие разработчикам хорошие инструменты [2]. В последнюю версию пакета Matlab также включены программные средства для разработки систем распознавания [2].

Цель выполненной работы состоит в исследовании эффективности методов и инструментальных средств, предлагаемых пакетом Matlab, для разработки программных моделей автоматического распознавания лиц, позволяющих строить на их основе реальные системы видеоконтроля за людьми. В частности, авторы проанализировали эффективность применения метода главных компонент (РСА) для решения задачи распознавания лиц [3]. Этот метод успешно применяется в статистике при интеллектуальном анализе данных для уменьшения пространства признаков без существенной потери информации. В задачах распознавания, где большие объёмы данных и важна скорость их обработки, вполне обосновано применить его для представления изображения лица вектором малой размерности (главных компонент), который сравнивается затем с эталонными векторами, занесёнными в базу данных.

Разработанная программная модель распознавания лиц построена на основе алгоритма *Eigenface*, использующего статистический метод главных компонент. Такой подход к автоматическому распознаванию не требует машинного обучения или свёрточной нейронной сети.

Схема распознавания лиц на основе программных средств пакета Matlab

Классическое решение задачи распознавания лиц человека включает несколько типовых этапов. Видеопоток изображений, на которых расположены лица людей, формируется видеокамерой в реальном масштабе времени. Далее выполняется детектирование, локализация и нормализация лиц на изображении. На третьем этапе формируются характерные признаки распознаваемого лица, по которым путём сравнения с базой эталонных лиц осуществляется распознавание человека. На рис.1 показана функциональная схема описанного классического подхода к распознаванию людей по лицам.

Такая система компьютерного зрения должна предусматривать этап обучения на распознавание заданного контингента людей. Для этого предварительно проводятся следующие этапы: подготовка данных о распознаваемых людях и их оцифровка, формирование признаков лиц методом обучения и загрузка в базу данных необходимой информации о лицах.

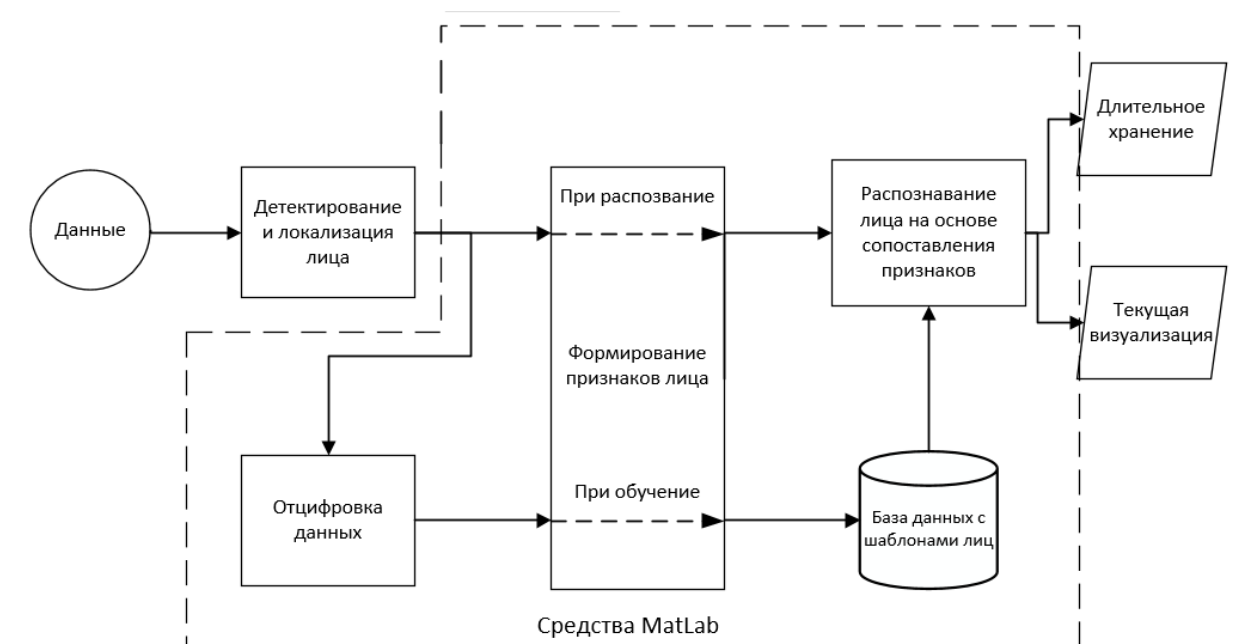


Рисунок 1 - Структура системы распознавания лиц на основе MatLab

В данной статье рассматривается только программная реализация средствами пакета Matlab отдельных блоков структуры на рис.1, которые обведены пунктирной линией. Таким образом, изображения лиц поступали не из видеопотока, а брались из готовой базы, а именно, из стандартной датасет лаборатории ORL (AT&T), в которой изображения лиц представлены на однородном фоне с разрешением 92x112 пикселей [3]. Эта датасет содержит изображения лиц 40 людей в 10 различных ракурсах (всего 400 изображений). Изображения из датасет получены с различной освещённостью, положением головы, с очками и без, открытыми и закрытыми глазами.

Рассмотрим технологию подготовки набора данных ORL для распознавания лиц средствами пакета Matlab. Для отдельного человека создана своя папка с именами (s1, s2, s3, ..., s40), каждая из них содержит 10 файлов с числовыми именами (1, 2, 3, ..., 10) изображений этого человека с фронтальным лицом [4]. Изображения представлены в оттенках серого цвета.

Для оцифровки и загрузки изображений лиц из базы данных в модуль распознавания была создана на языке Matlab специальная функция load_database, текст которой изображён на рис. 2. Функция возвращает числовую форму изображений.

```
function output_value = load_database() % объявляется функция, которая возвращает числовую форму изображений
persistent loaded; %постоянная переменная
persistent numeric_Image;
if(isempty(loaded)) %проверка на пустоту
all_Images = zeros(10304,400); % 10304 пикселя в изображение. По 40 раз заменяют нулями
for i=1:40 %объединение папок
cd(strcat('s',num2str(i)));
for j=1:10 %объединение изображений внутри них
image_Container = imread(strcat(num2str(j),'.pgm'));
all_Images(:,(i-1)*10+j)=reshape(image_Container,size(image_Container,1)*size(image_Container,2),1);
%преобразование изображений в матрицу с однии столбцом
end
disp('Loading Database');
cd ..
end
numeric_Image = uint8(all_Images); %преобразование данных в 8-ми битное целое число без знака
end
loaded = 1;
output_value = numeric_Image;
```

Рисунок 2 - Фрагмент кода программы для пакета Matlab загрузки видеоданных о лицах

Функция load_database весь обучающий набор лиц преобразует функцией reshape в одну общую матрицу данных all_Images (10304x400), где каждый столбец представляет собой один экземпляр плоского изображения

лица ($92 \times 112 = 10304$ пикселей), развёрнутого в столбец. Все лица обучающего набора приведены к одному размеру. Внутри двух циклов с помощью функции `strcat` происходит объединение папок, а потом с помощью функции `imread` выполняется загрузка всех изображений в матрицу `all_Images` с последующим преобразованием данных в 8-битные целые числа без знака. Схема преобразования показана на рис.3. На этом рисунке $n=400$, $m=92$, $p=112$, $m \times p = 10304$.

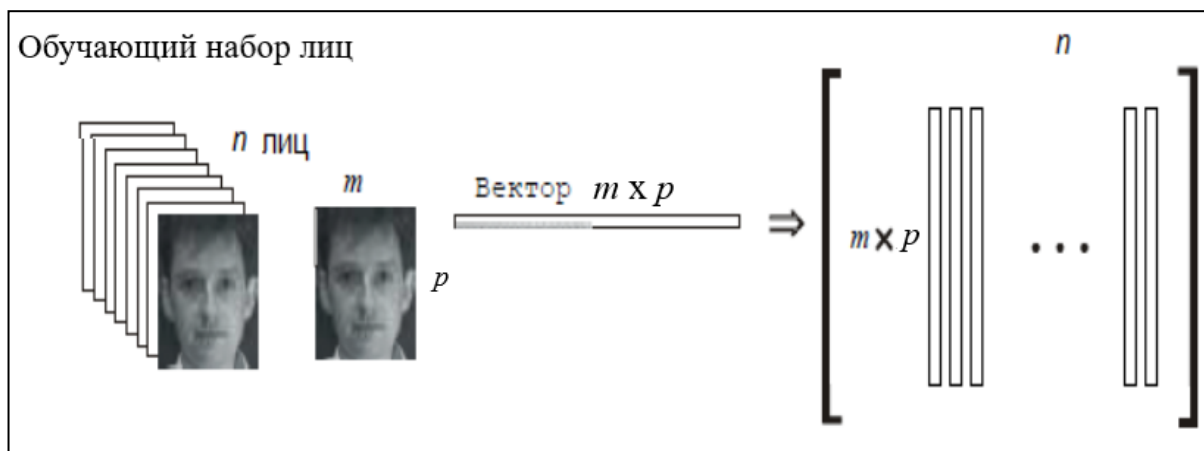


Рисунок 3 - Преобразование обучающего набора лиц в одну общую матрицу

Факт загрузки базы лиц в модуль распознавания отмечается присвоением единицы постоянной переменной `loaded`, чтобы не происходила повторная загрузка во время работы модуля распознавания. Примеры изображений лиц из датасет ORL показаны на рис.4.

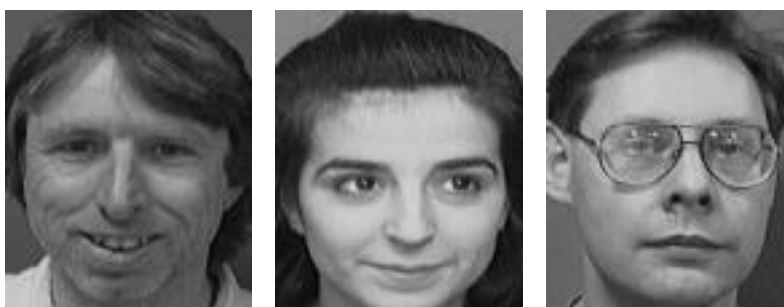


Рисунок 4 - Примеры лиц из базы данных ORL, используемой для обучения модуля распознавания

Распознавание лиц на основе статистического подхода

В данной работе задача распознавания лиц решалась методом главных компонент (Principal component analysis, PCA) [5]. Этот метод хорошо проработан математически, достаточно простой в реализации по сравнению с методами машинного обучения или моделями искусственных нейронных сетей. Кроме того, в пакете Matlab есть все средства для его программной реализации.

Основное достоинство метода – значительное уменьшение размерности пространства признаков, сохраняющих индивидуальность описания; способность выявлять различные «изменчивости» в обучающей выборке изображений лиц и описывать эту «изменчивость» в базисе нескольких ортогональных собственных векторов.

Для работы модуля распознавания на этапе подготовки данных была сформирована специальным образом (см. рис.3) матрица всех изображения `all_Images` (обозначим её X), размерность которой определяется по одной из координат количеством изображений n , а по второй - количеством пикселей в каждом изображении лица $d = m \times p$. Для большей объективности проведена нормализация матрицы X с учётом 0-го среднего и 1-й дисперсии:

$$x_{ij} = x_{ij} - \mu_i, \quad \mu_i = \frac{1}{n} \sum_{j=1}^n x_{ij}$$

$$x_{ij} = \frac{x_{ij}}{\sigma_i}, \quad \sigma_i = \frac{1}{n-1} \sum_{j=1}^n (x_{ij} - \mu_i)^2$$

Для нормализованной матрицы $X(d \times n)$ находим ковариационную матрицу $S(d \times d)$, элементы которой определяют корреляции между всеми парами переменных (пикселей). Ковариационная матрица S набора X разлагается на матрицу собственных векторов $W(d \times d)$ и диагональную матрицу собственных значений $\Lambda(d \times d)$, которые связаны каноническим уравнением:

$$S \cdot W = \Lambda \cdot W.$$

Ранжируя собственные вектора W в порядке убывания их собственных значений λ , оставляем только первые k векторов по правилу:

$$\frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^n \lambda_i} > R, \quad \text{где порог } R = 0,9.$$

В результате отбрасывания менее значимых векторов мы из матрицы W оставляем главные компоненты (признаки) в виде подматрицы $V(d \times k)$, в столбцах которой располагаются наиболее значимые собственные вектора. Это первый шаг к уменьшению размерности исходных данных, т.к. $k < d$. Учитывая, что исходная матрица изображений лиц X осталась неизменной (не считая операцию нормализации), можно трансформировать X , т.е. переориентировать данные X с исходных осей на оси, представленные главными компонентами. Это выполняется путём перемножения транспонированной матрицы главных компонент $V(d \times k)$ на исходный набор $X(d \times n)$, в результате получим новое более компактное описание эталонной базы лиц X (all_Images) в виде другой матрицы $Y(k \times n)$

$$Y = V^T \cdot X.$$

Данный метод оставляет в Y самые значительные особенности описания лиц в форме X . Таким образом, используя ограниченное количество собственных векторов, получаем сжатую аппроксимацию входных изображений лиц, которые затем храним в базе данных в виде векторов коэффициентов, служащих одновременно ключами поиска в базе данных лиц.

Полученный один раз на обучающем множестве лиц набор собственных векторов V используется для кодирования всех остальных изображений лиц, которые представляются взвешенной комбинацией этих собственных векторов. Поэтому процесс распознавания нового лица $Z(d \times 1)$ заключается в сравнении его двойника $F(k \times 1)$, получаемого путём трансформации Z по осям главных компонент V

с векторами компактной базы лиц Y на основе евклидовой метрики [6]. На рис. 4 изображён результат выполнения модуля распознавания [6].

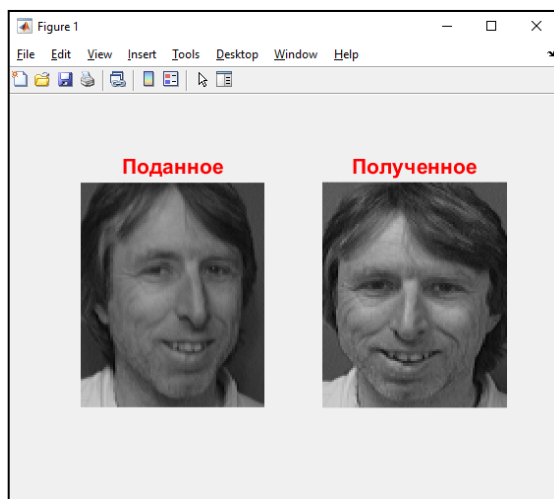


Рисунок 5 – Результат обработки изображения приложением

Оценка качества модели распознавания

Анализ разработанной программной модели распознавания проведен на стандартной базе данных ORL, включающей 400 лиц. В качестве тестовых изображений выбирались случайным образом лица, входящие в используемую базу. Проведено исследование влияния количества лиц одного человека в обучающем множестве на качество распознавания (см. рис 6). Из экспериментов было выявлено, что при подаче на обучение 10 изображений 1 человека вероятность правильного распознавания составляет около 94%.

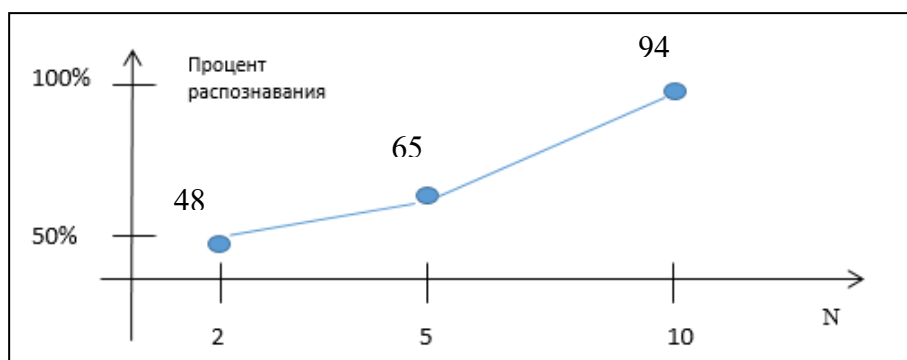


Рисунок 6 – Зависимость качества распознавания от размера обучающего множества: N - количество лиц одного человека в обучающем множестве

Заключение

Разработана программная модель распознавания людей по изображению лиц с использованием алгоритма Eigenface и инструментов пакета MatLab. Алгоритм распознавания основан на статистическом методе главных компонент. Этапы подготовки и распознавания в модели реализованы на программном уровне и описаны математически. Анализ качества распознавания проведен на разных объемах обучающего множества.

Выполненная работа показала, что используемый алгоритм распознавания, основанный на методе главных компонент, является крайне простым и может быть использован для распознавания лиц наряду с нейронными сетями. У алгоритма есть недостатки, такие как: чувствительность к уровню освещенности видеоизображений, низкая скорость распознавания и слабая дискриминация между классами лиц.

Литература

1. Хабр [Электронный ресурс] - Режим доступа: <https://habr.com/ru/company/droider/blog/568764/> - Загл. с экрана.
2. Брилюк Д.В., Старовойтов В.В. Распознавание человека по изображению лица нейросетевыми методами - 2002. - вып. 1. - с. 4-11. - URL: http://uiip.bas-net.by/structure/1_ori/starovoitov/Starovoitov_Publication_section/11_Starovoitov02prep.pdf.
3. ML | Face Recognition Using Eigenfaces (PCA Algorithm) [Электронный ресурс] - Режим доступа: <https://www.geeksforgeeks.org/ml-face-recognition-using-eigenfaces-pca-algorithm/> - Загл. с экрана.
4. Face Recognition using Matlab [Электронный ресурс] - Режим доступа: <https://www.nzfaruqui.com/face-recognition-using-matlab/> - Загл. с экрана.
5. Face Recognition using Eigenfaces and Distance Classifiers [Электронный ресурс] - Режим доступа: <https://onionesquereality.wordpress.com/2009/02/11/face-recognition-using-eigenfaces-and-distance-classifiers-a-tutorial/> - Загл. с экрана.
6. Распознавание изображений. Алгоритм Eigenface [Электронный ресурс] - Режим доступа: <https://habr.com/ru/post/68870/> - Загл. с экрана.

Дручевский Д.В., Федяев О.И. Распознавание лиц человека статистическими методами из пакета Matlab. Статья посвящена разработке программной модели распознавания людей по изображению лиц с использованием алгоритма Eigenface и инструментов пакета MatLab. Алгоритм распознавания основан на статистическом методе главных компонент. Этапы подготовки и распознавания в модели реализованы на программном уровне и описаны математически. Анализ качества распознавания проведен на разных объемах обучающего множества.

Ключевые слова: распознавание образов, лицо человека, метод главных компонент, пакет Matlab, алгоритм Eigenface.

Druchevsky D.V., Fedyaev O.I. Recognition of human faces by statistical methods from the Matlab package. The article is devoted to the development of a software model for recognizing people by face images using the Eigenface algorithm and the tools of the MatLab package. The recognition algorithm is based on the statistical method of principal components. The stages of preparation and recognition in the model are implemented at the software level and are described mathematically. The analysis of recognition quality was carried out on different volumes of the training set.

Key words: pattern recognition, human face, principal component method, Matlab package, Eigenface algorithm.

Алгоритмическое обеспечение СППР оптимального планирования производственно-логистической деятельности предприятия

Т.Г. Дмитриук^{*1}

^{*1} ассистент, Донецкий национальный технический университет,
tnauka@lenta.ru, OrcID: 0000-0003-4803-5555, SPIN-код: 9249-2928

Дмитриук Т.Г. Алгоритмическое обеспечение СППР оптимального планирования производственно-логистической деятельности предприятия. В статье осуществлена разработка функциональной структуры (как части информационного обеспечения) СППР оптимального планирования производственно-логистической деятельности предприятия на примере ООО «Донецкий пивоваренный завод». Внедрение разработки на практике возможно при создании системы управления процессами на предприятиях аналогичной структуры третьего уровня производственной иерархии.

Ключевые слова: алгоритмическое обеспечение, информационное обеспечение, СППР, функциональная структура, планирование.

Dmitriuk T.G. Algorithmic support of the DSS for optimal planning of production and logistics activities of the enterprise. The article develops the functional structure (as a part of the information support) of the DSS for the enterprise production and logistics activities optimal planning on the example of LLC "Donetsk Brewery". Implementation of the development in practice is possible when creating a process management system at the third level of the production hierarchy enterprises of a similar structure.

Key words: algorithmic support, information support, DSS, functional structure, planning.

**Полный текст статьи опубликован в выпуске № 3 (29), 2022 г.
научного журнала «Информатика и кибернетика»**

Визуальный контроль присутствия студентов в аудитории на основе глубокой нейронной сети

А.А. Суханов^{*1}, Н.М. Ткачёв^{*1}, О.И. Федяев^{*2}

^{*1} магистрант, Донецкий национальный технический университет,
studysukhanov@mail.ru

^{*2} к.т.н, доцент, Донецкий национальный технический университет,
fedyaev@donntu.ru, OrcID: 0000-0001-6822-7306

Суханов А.А., Ткачёв Н.М., Федяев О.И. Визуальный контроль присутствия студентов в аудитории на основе глубокой нейронной сети. В статье рассмотрена задача автоматизации учёта присутствия студентов на занятии в аудитории. Свёрточная нейронная сеть модели VGGFace формирует признаки лица человека. Идентификация личности происходит по сходству признаков лиц. Программная версия системы реализована с использованием библиотек Keras и OpenCV.

Ключевые слова: распознавание лиц человека, глубокая нейронная сеть, видеоконтроль.

Введение

Несмотря на достигнутые в последние годы успехи в реализации компьютерного зрения, по-прежнему есть ряд нерешённых проблем в этой области. Основные трудности компьютерного распознавания лиц, которые необходимо преодолеть, состоят в том, чтобы распознавать человека по изображению лица независимо от изменения ракурса и условий освещённости при съёмке [2]. Кроме того, с каждым годом технологии CV стали находить всё больше применений в различных отраслях индустрии и повседневной жизни.

В настоящее время большие перспективы в решении данных проблем связывают с применением глубоких нейронных сетей. К этому классу относится многослойная свёрточная нейронная сеть (ConvolutionalNeuralNetwork) [1,6].

На данный момент свёрточная нейронная сеть и её модификации считаются лучшими по точности и скорости распознавания объектов на изображении. Более того, распознавание лиц из видеопотока с камеры показывает, что нейронные сети с такой архитектурой способны работать в режиме реального времени даже на устройствах с ограниченными ресурсами. Важным фактором в распространении нейросетей стали возможности аппаратного ускорения нейровычислений при помощи таких библиотек, как Nvidia CUDA и OpenCL.

В последнее время известными фирмами предложены мощные библиотеки, в которых реализованы различные модели глубоких нейронных сетей, позволяющие решать сложные задачи распознавания. Поэтому целью данной работы является оценка возможности реализации нейросетевого распознавания лиц на базе инструментальных средств библиотеки Keras и OpenCV [3,4] и создание системы оперативного визуального контроля присутствия студентов на учебных занятиях.

Схема распознавания лиц

Задача обнаружения лица на изображении является «первым шагом», т.е. предобработкой в процессе решения поставленной задачи [5]. Однако и сама информация о присутствии и, возможно, количестве лиц на изображении или в видеопотоке может быть полезна для таких приложений, как охранные системы и содержательная индексация базы данных изображений или видеофрагментов.

Блок обнаружения лиц должен принимать изображения с веб-камеры, выделяет и локализует на них лица (см.рис.1). Эту функцию выполняет алгоритм детектирования лиц на принятом изображении. В результате формируется изображение локализованного лица и после нормализации подаётся на следующий блок системы. В работе локализация лиц выполняется методом Виола-Джонса. К достоинствам данного метода можно отнести высокую степень правильной локализации лица, малое число ложных срабатываний, высокую скорость работы. Он в меньшей степени чувствителен к освещённости. Точность выделения лиц на снимке по алгоритму Виола-Джонса в идеальных условиях достигает 90-95%, что вполне приемлемо для решения практических задач. Все изображения выделенных лиц нормируются к стандартному размеру 224x224 пикселей, каждый из которых представлен в виде трёх составляющих в формате RGB.

Далее изображение лица передаётся в блок формирования вектора признаков, который реализует распознавательную функцию $f: X \rightarrow Y$, где X - множество входных изображений лиц; Y - множество векторов

признаков для лиц из X . Таким образом, нейросетевая функция f каждому выделенному лицу $x \in X$ ставит в соответствие вектор признаков $y \in Y$ ($y = f(x)$), которыми характеризуется данное лицо.

Для выработки признаков лица применялась свёрточная нейронная сеть, которая предварительно была обучена её создателями на примерах фотографий 2622-х человек (по 1000 фотографий на человека) [1,3]. 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.



Рисунок 1 - Основные процессы компьютерного распознавания лиц человека

При настройке системы необходимо предварительно сформировать базу данных лиц для всех распознаваемых людей, представленных конечным множеством соответствующих фамилий L . С этой целью для $\forall x$ с помощью нейросетевой распознавательной функции f определяется множество правильных пар

$$\{(y, l) \mid y = f(x), x \in \bar{X}, l \in L\},$$

где $\bar{X}$ - множество подготовленных фотографий распознаваемых лиц, т.е. эталоны изображений распознаваемых лиц; y - вектор признаков изображения лица $x \in \bar{X}$; l - фамилия человека, фотография которого изображена на снимке x . Всё множество пар (y, l) заносится в базу данных векторов признаков лиц.

В штатном режиме работы системы, т.е. при распознавании, в блоке сравнения вектор признаков распознаваемого лица, полученный с выхода свёрточной нейронной сети, сравнивается со всеми векторами базы данных. Процедура сравнения основывается на методе вычисления косинусного сходства вектора распознаваемого лица с каждым вектором-эталон 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; $n = 2622$.

Расознаваемое лицо считается соответствующим эталону, если полученный коэффициент сходства выше определённого значения (в работе использовалось значение 0,7).

Архитектура свёрточной нейронной сети

Свёрточная нейронная сеть имеет многослойную архитектуру и состоит из свёрточных слоёв (convolution layers) и субдискретизирующих слоёв (subsampling layers или pooling layers, слоёв подвыборки), которые чередуются друг с другом [6].

В каждом слое имеется набор из нескольких плоскостей признаков. Причём нейроны одной плоскости имеют одинаковые веса, ведущие к соответствующим локальным участкам предыдущего слоя. Изображение предыдущего слоя как бы сканируется небольшим окном, т.е. пропускается сквозь набор весов (ядро свёртки), и результат сканирования отображается на соответствующем нейроне текущего слоя. Ядро свёртки интерпретируют как графическое кодирование какого-либо признака, например, наличие горизонтальной или вертикальной линии. Таким образом, набор плоскостей представляет собой карты признаков (feature map), что позволяет каждой плоскости находить «свои» участки изображения в любом месте предыдущего слоя.

Операция подвыборки (объединения), выполняет уменьшение размерности сформированных карт признаков. В данной архитектуре сети считается, что информация о факте наличия искомого признака важнее точного знания его координат, поэтому из нескольких соседних нейронов карты признаков выбирается максимальный и принимается за один нейрон уплотнённой карты признаков меньшей размерности. За счёт данной операции, помимо ускорения дальнейших вычислений, сеть становится более инвариантной к масштабу входного изображения.

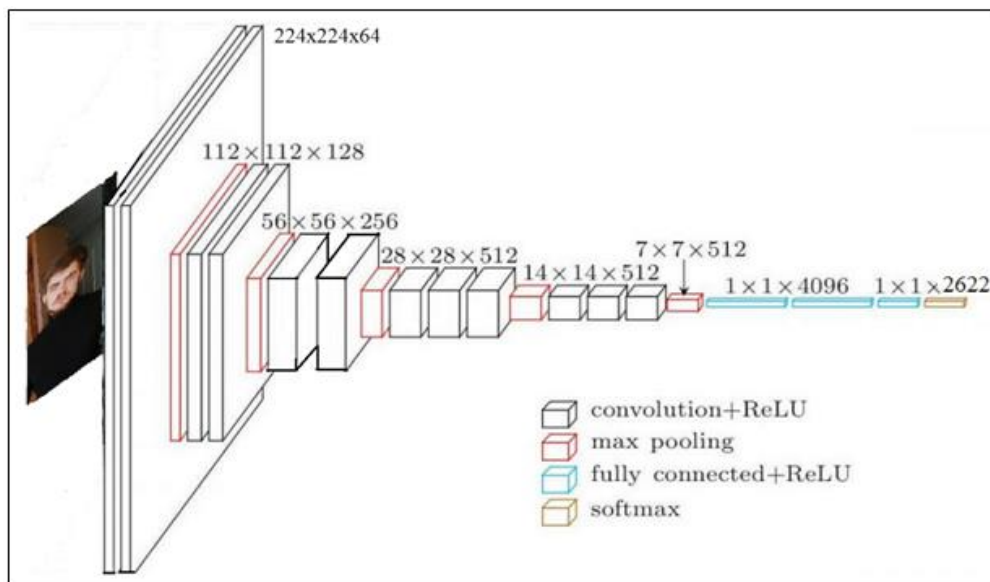


Рисунок 2 – Архитектура свёрточной нейронной сети модели VGGFace

Чередование слоёв позволяет из предыдущих «карт признаков» составлять следующие «карты признаков», содержащие более общие характеристики, меньше зависящие от искажений изображения. На каждом следующем слое карта уменьшается в размере, но увеличивается их количество. На практике это означает способность распознавания сложных иерархий признаков. Обычно после прохождения нескольких слоёв карта признаков вырождается в вектор или даже скаляр, но таких карт признаков становится сотни. На выходе свёрточных слоёв сети дополнительно устанавливается многослойный персептрон. Обучается сеть стандартным методом обратного распространения ошибки.

В 2014 году была продемонстрирована модель глубокой свёрточной сети VGG-16, которая достигла хороших результатов в точности распознавания изображений. Эта модель показала, что точность работы глубокой сети значительно вырастает при увеличении количества слоёв до 16-19 по сравнению со старыми моделями, использовавшими до 10 слоёв [3].

В системе видеоконтроля студентов в качестве модели свёрточной нейронной сети использовалась нейросеть VGGFace (рис. 4). Эта модель разработана и обучена разработчиками VisualGeometryGroup Оксфордского университета. 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. Шаг фильтра при сканировании свёрточного слоя составляет 1 пиксель. Пространственное дополнение (padding) входа свёрточного слоя выбрано с учётом сохранности пространственного разрешения после операции свертки, т. е. дополнение равно 1 для 3x3 свёрточных слоёв. Пространственный пулинг осуществляется при помощи пяти max-pooling слоёв, которые следуют за последним в группе свёрточных слоев. Операция max-pooling выполняется на окне размера 2x2 пикселей с шагом 2. Все нейроны скрытых слоёв имеют функцию активации ReLU. После стека свёрточных слоёв идут три полносвязных слоя: первые два имеют по 4096 каналов, третий — 2622 каналов (по числу распознаваемых классов). Последним в архитектуре расположен soft-max слой.

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. Несмотря на то, что сеть не обучается на базе данных, с которой она используется, столь большое количество признаков позволяет каждому лицу создать уникальный отпечаток. При этом предполагалось, что лица, в равной мере схожие с соответствующими лицами из обучающего множества, также схожи между собой, потому о схожести двух образов лиц можно судить путём сравнения их векторов признаков. Этот принцип называется «Transfer learning».

Поскольку данная сеть уже предварительно обучена, весовые коэффициенты предварительно загружаются из файла. На выходе сеть формирует 2622-мерный вектор признаков лица.

Видеоконтроль присутствия студентов в аудитории

При создании прототипа системы видеоконтроля были использованы язык программирования Python и библиотеки OpenCV, Keras. Библиотека Keras максимально упрощает моделирование нейронных сетей, автоматизируя расчёт размеров данных и способов связи между слоями.

Задача автоматического учёта присутствия студентов на учебных занятиях включает следующие подзадачи:

- съёмка входящих в аудиторию студентов на видеокамеру;
- анализ кадров видеопотока на наличие лиц;
- сравнение выделенных лиц с базой данных учащихся;
- отметка о присутствии на занятии (или опоздании) в случае успешной идентификации;
- сохранение данных в журнале посещений.

Во время работы системы могут возникнуть различные проблемы, которые могут препятствовать точности распознавания: угол поворота головы студента был недостаточно прямым по отношению к камере, слабое освещение, студент прошёл мимо камеры очень быстро, слишком большое количество лиц на изображении, из-за чего некоторые кадры не были захвачены и т.д. Преподаватель может допустить ошибку при использовании системы, например, включив режим регистрации опозданий слишком рано. Поэтому в системе предусмотрен режим ручного редактирования электронного журнала.

На рисунке 3 представлена форма подсистемы управления студентами, с помощью которой можно осуществлять:

- добавление, изменение имён и удаление групп;
- отображение студентов по группам, добавление, редактирование и удаление студентов;
- изменение группы студентов;
- добавление и изменение фотографии (фотография может быть выбрана из файла либо с камеры; при этом на фотографии может быть обнаружено несколько лиц, в этом случае верное лицо необходимо выбрать).

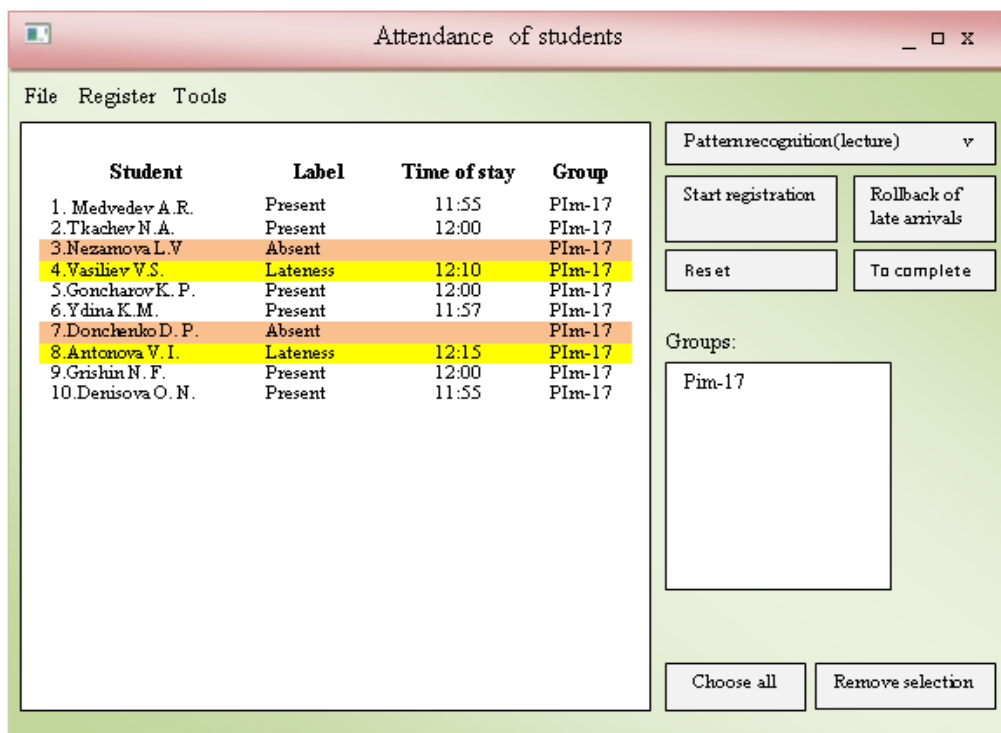


Рисунок 3 – Главное окно системы регистрации студентов в аудитории

Преподаватель может визуально наблюдать и управлять информацией о студентах с помощью главного окна системы видеорегистрации. В левой части окна высвечивается таблица со списком студентов группы, в которой преподаватель проводит занятие. В таблице во время входа студентов в аудиторию фиксируется время прибытия и его состояние (не отмечен, прибыл, опоздал, отсутствует). Для удобства видеоконтроля цвет строки студента меняется в зависимости от его состояния. На данном рисунке показан факт регистрации студента в таблице текущего занятия.

В окне справа находятся элемент выбора дисциплины, кнопки остановить/продолжить распознавание, перехода в режим опоздания, сброса занятия и его завершения. По умолчанию все студенты в списке имеют статус «Не отмечен» и при распознавании обозначаются как «Присутствует». При переходе в режим опоздания (кнопка «Опоздание») статус всех неотмеченных студентов изменяется на «Отсутствует», а те, кто будет

распознан в этом режиме, помечаются статусом «Опоздание». В случае отката режима опоздания все отсутствующие студенты вновь становятся неотмеченными, а опоздавшие – присутствующими.

Кнопка «Завершить» прекращает визуальную регистрацию студентов и записывает данные из таблицы текущего занятия в общий журнал группы (потока).

Таким образом, цифровой журнал представляет собой реляционную таблицу, содержащую следующие данные: студент, дата проведения занятия, предмет, группа, отметка о присутствии. В системе предусмотрено визуализация журнала в двух формах: по определённому предмету в определённой группе, по конкретному студенту.

Заключение

В статье предложен подход к решению задачи распознавания лица человека на основе свёрточной нейронной сети. Локализация лиц на вводимом изображении выполнена методом Виола-Джонса. Свёрточная нейронная сеть формирует признаки лица человека. Идентификация личности происходит вычислением косинусного коэффициента сходства векторов признаков лиц. Процесс распознавания реализован при отсутствии режима реального времени.

Разработанная версия системы, использующая ресурсы библиотек Keras и OpenCV и может применяться для оперативного учёта присутствия студентов на занятиях по изображениям лиц из видеопотока.

Литература

1. Гудфеллоу Я., Бенджио И., Курвилль А. Глубокое обучение / пер. с англ. А.А.Слинкина. – 2-е изд., испр. – М.: ДМК Пресс, 2018. – 652 с.
2. Федяев О.И., Махно Ю.С. Система распознавания зашумлённых и искажённых графических образов на основе нейронной сети типа неокогнитрон // Одиннадцатая национальная конференция по искусственному интеллекту с международным участием КИИ-2008: Труды конференции. Т. 3. – М.: ЛЕНАНД, 2008. – С. 75–83.
3. 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.
4. OpenCV – библиотека компьютерного зрения с открытым исходным кодом [Электронный ресурс]. – Режим доступа: <http://software.intel.com/en-us/articles/>
5. Paul Viola, Michael Jones. Rapid Object Detection using a Boosted Cascade of Simple Features, 2001. – P. 1–8.
6. Le Cun Y, Bengio Y. Convolutional neural networks for Images, Speech and Time Series, AT&T Laboratories, 1995. – P. 1 – 14.

Суханов А.А., Ткачёв Н.М., Федяев О.И. Визуальный контроль присутствия студентов в аудитории на основе глубокой нейронной сети. В статье рассмотрена задача автоматизации учёта присутствия студентов на занятии в аудитории. Свёрточная нейронная сеть модели VGGFace формирует признаки лица человека. Идентификация личности происходит по сходству признаков лиц. Программная версия системы реализована с использованием библиотек Keras и OpenCV.

Ключевые слова: распознавание лиц человека, глубокая нейронная сеть, видеоконтроль.

Sukhanov A.A., Tkachev N.M., Fedyaev O.I. Visual presence control of students in a classroom based on a deep neural network. The paper considers the task of automating student presence detection in a classroom. A convolutional neural network of VGGFace model generates facial features of a person. Identification of a person is based on the similarity of the facial features. The software version of the system is implemented using Keras and OpenCV libraries.

Keywords: article: face recognition, deep neural network, videocontrol.

Информационно технологические системы подготовки принимаемых решений по запросам и реализации горно-шахтного оборудования

О.А. Криводубский*¹, Е.Н. Павлюк*²

^{*1} д.т.н, доцент, Донецкий национальный технический университет,
oleg.krivodubski.dn@gmail.com

^{*2} аспирант, Донецкий национальный технический университет,
luxury.l@yandex.ru

Криводубский О.А., Павлюк Е.Н. Информационно технологические системы подготовки принимаемых решений по запросам и реализации горно-шахтного оборудования. Данная статья представляет собой методический материал, основанный на методах математической статистики которая включает в себя последовательно виды анализа: дисперсионный, корреляционный, каскадный регрессионный. Эти методы лежат в основе математических моделей, позволяющих оценивать количество импортзамещающего оборудования.

Ключевые слова: горно-шахтное оборудование, математическая модель, система подготовки принимаемых решений, оформление, математическая статистика.

Krivodubsky Oleg, Pavlyuk Elena: Information technology systems for the preparation of decisions made on requests and the implementation of mining equipment. This article is a methodological material based on the methods of mathematical statistics, which includes sequentially the types of analysis: dispersion, correlation, cascade regression. These methods underlie mathematical models that allow estimating the amount of import-substituting equipment.

Key words: mining equipment, mathematical model, decision-making system, design, mathematical statistics.

**Полный текст статьи опубликован в выпуске № 3 (29), 2022 г.
научного журнала «Информатика и кибернетика»**

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

УДК 004.93

Современное состояние методов компьютерной обработки информации при визуализации объектов

С. В. Иваница^{*1}, В. А. Мишустин^{*2}

^{*1} канд. техн. наук, доцент кафедры компьютерной инженерии,
ГОУ ВПО «Донецкий национальный технический университет»,
sergey.ivanitsa@cs.donntu.org, OrcID: 0000-0001-5235-1804, SPIN-код: 1474-9499

^{*2} аспирант, ГОУ ВПО «Донецкий национальный технический университет»,
mishustin.post@yandex.ru

Иваница С.В., Мишустин В.А. Современное состояние методов компьютерной обработки информации при визуализации объектов. В работе дается определение компьютерной визуализации. Рассматриваются подобласти компьютерной визуализации. Приводится краткий обзор приложений каждой из подобластей. Детально проводится анализ существующих методов научной визуализации. В ходе анализа выявлены проблемы, возникающие в ходе визуализации данных, также объясняются причины возникновения данных проблем. Анализ выявленных проблем определяет дальнейшее направление исследований по каждой из проблем.

Ключевые слова: визуализация, многомерная визуализации данных, моделирование.

Введение

Известно, что бурное развитие компьютерных наук повлияло на множество междисциплинарных отраслей науки. Этот факт привел к тому, что процесс визуализации данных ускорился и выполнение большого количества сложных математических операций свелось до долей секунд. Однако графическое представление количественной информации имеет более глубокие корни. Эти корни уходят в историю статистики и визуального отображения информации. Компьютерная визуализация решает задачи в разных направлениях деятельности человека. Среди них можно выделить: ученых в различных инженерных областях и дисциплинах; дизайнеров; архитекторов; инженеров; создателей мультимедийной информации; фотографов; модельеров тканей и одежды; медиков; специалистов в области теле- и видеомонтажа и др. [1].

В современной науке объектами исследований становятся новые и не простые предметы, исследование которых требует более сложные средства визуального представления данных. Классические методы графического представления данных становятся неактуальными для отображения всех качеств изучаемого объекта, на первое место выступают более реалистичные методы визуализации данных. Современные ЭВМ предлагают множество мультимедийных возможностей: средства двумерной, трехмерной графики, и многомерной графики, звуковое сопровождение, построение стереоскопических изображений, а также технологии дополненной реальности.

Задача данной работы заключается в рассмотрении наиболее распространенных методов визуализации данных, а также их особенностей, с целью определения возникающих проблем при визуализации данных.

Задачи визуализации данных возлагаются на вполне самостоятельную дисциплину – компьютерная визуализация, однако в рамках вычислительных наук считается вспомогательной. В данном случае компьютерная визуализация – процесс передачи свойств, качеств и абстрактных представлений об объекте в геометрические образы. Это дает возможность пользователю анализировать собранные данные об объекте способом восприятия зрением.

Считается, что компьютерная визуализация традиционно имеет подобласти [2]:

- визуализация программного обеспечения;
- информационная визуализация;

- научная визуализация.

Визуализация программного обеспечения

Это совокупность технологий, которые используют компьютерную графику и возможности взаимодействия человека с компьютерной техникой, с целью более глубоко постижения и обслуживания программного обеспечения ЭВМ, а также для описания элементов программного кода во время разработки программ [3]. Широко применяется в области параллельных и распределенных вычислений.

Результатом работы исследователей и ученых в области визуализации программного обеспечения является разработка, тестирование, и отладка визуальных языков программирования.

Разделяют системы визуализации программирования и системы визуального программирования.

Изначально, ожидалось, что перспективы применения визуального параллельного программирования будут соответствовать мощностям параллельных вычислителей. Однако, на данный момент наблюдается задержка темпов прогресса, с этим связывают недостаток новых идей и результатов [3].

Информационная визуализация

Это наиболее общая подобласть компьютерной визуализации, которая объединяет все возможные варианты представления информации графическим способом. В данной области компьютерной графики визуализация рассматривается как информационная связь в системе «потребитель – медиа». Исходя из этого визуальная информация должна выполнять следующие функции [4]:

1. Информационно-коммуникативная.
2. Идеологическая функция.
3. Культурно-образовательная функция.
4. Рекламная.
5. Рекреативная.

Таким образом данная область исследует возможности и способы качественного выполнения перечисленных выше функций.

Научная визуализация

Научная визуализация как часть компьютерного моделирования понимается как совокупность систематизированных инструментов, методов и операций над геометрическими и, в первую очередь, над физическими данными, а также над функциями изображения, позволяющих отразить на экране монитора поведение и развитие физических или каких-либо иных процессов с использованием машинной графики. При этом большой объем результатов моделирования представляется в компактной и легко воспринимаемой форме [5].

Результатом компьютерного моделирования могут выступать разнообразные графики, диаграммы, снимки, модели и другое. Таким образом, можно выделить три группы методов научной визуализации: двумерная визуализация, трехмерная визуализация, стереовизуализация данных.

Для проведения различных исследований и компьютерных экспериментов исследователи зачастую используют актуальные научно-технические системы и программные пакеты. Научная визуализация способна быть как самостоятельной целью исследования или эксперимента, так и частью анализа данных. Поэтому считаем данную область компьютерной визуализации наиболее актуальной и требующей более детального рассмотрения. Рассмотрим подробнее методы научной визуализации.

Методы двумерной визуализации данных

Для создания двумерной визуализации используются векторные или растровые изображения. Растровые изображения используются геометрические примитивы (простые фигуры). Растровое изображение представляет собой набор пикселей, которые задаются двумерным массивом данных. Рассмотрим наиболее распространенные формы представления данных методами двумерной визуализации.

График, как средство визуализации, применяют с целью показать функциональную зависимость, связать математические понятия, или связать между собой параметры. В зависимости от целей использования графика могут различаться виды графиков, что также обуславливается системой координат и масштабов отчета по осям.

Что бы показать возможное соотношение значений нескольких показателей применяют способ диаграмм. Диаграммы широко применяются в разных задачах, что объясняет большое разнообразие типов диаграмм, применяемых в различных задачах [6]. Самыми распространенными являются диаграмма Ганта и сетевой график.

Диаграмма Ганта (рис.1) использует полосы, расположенные продольно оси времени. Данные полосы символизируют время, которое предположительно необходимо затратить на определенную работу, где начало полосы определяет время начало проекта, и конец полосы – время завершение работы. Однако у такого способа представления информации есть недостаток – невозможность отображения ресурсоемкости и приоритетности определенной работы.

Для отображения взаимосвязи между несколькими проектами используют сетевой график (рис.2), также данный способ позволяет отображать информацию о состоянии проекта.

В астрономии данные также можно визуализировать способом диаграмм, для этого используют диаграмму Герцшпрунга-Рассела (рис.3). Данный способ позволяет графически показать зависимость абсолютной величины звезды от ее спектрального класса. Также данная способ имеет свои варианты, с помощью которых можно показать, например, зависимость светимости от температуры.

В результате определенных экспериментом приборы передают характеристики объекта в качестве снимков в самых различных спектральных диапазонах. Подобные снимки являются также методом двухмерной визуализации данных. Также исследователь может комбинировать полученные данные, объединяя и фильтруя по определенным параметрам.

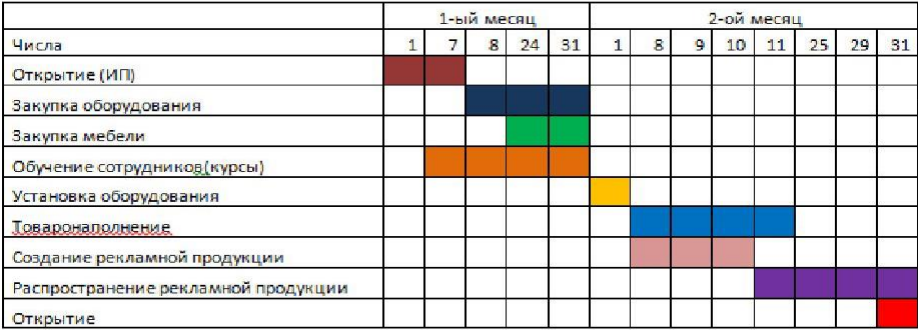


Рисунок 1 – Диаграмма Ганта (2D – визуализация)

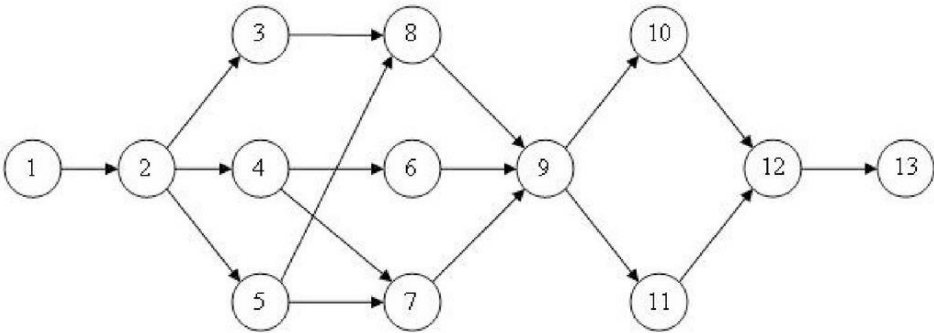


Рисунок 2 – Сетевой график (2D – визуализация)

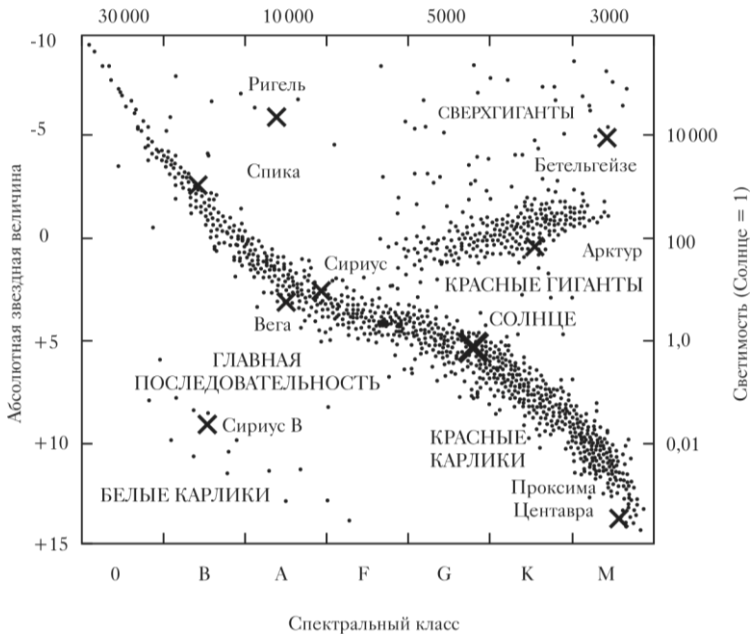


Рисунок 3 – Диаграмма Герцшпрунга-Рассела (2D – визуализация)

Проблемы использования методов двухмерной визуализации данных при анализе больших объемов данных

Простая визуализация целого большого набора данных может привести к тому, что исследователь увидит на экране большое скопление точек, которое будет сливаться в одно большое пятно. Такое явление при визуализации большого набора данных называется Visual noise / визуальный шум [7]. Данная проблема заключается в том, что многие объекты графика тесно связаны друг с другом и на экране пользователя не могут быть разделены в виде отдельных объектов. Тогда задача получения важной и полезной информации, при анализе является затруднительной задачей. В данном случае «визуальный шум» необходимо рассматривать не как результат некорректных или поврежденных данных, а как результат перенасыщения данных.

Следующей характерной проблемой при визуализации большого количества входных данных является Large image perception / восприятие большого изображения. Эта проблема объясняется тем, что у человеческого мозга существует определенный предел восприятия определенных графических данных. Тогда после перехода через данный предел человек теряет способность анализировать и получать информацию из представленных графическим образом данных. К тому визуализация данных ограничена устройством отображения графической информации. Таким образом приходим к тому, методы визуализации данных ограничены возможностями восприятия человеком графических данных и техническими показателями устройства вывода.

Некоторые подходы к решению двух предыдущих проблем основываются на уменьшении отображаемых данных. Однако подобные решения приводят к новой проблеме – Information loss / потеря информации.

Большие данные можно визуализировать статически, когда они были собраны и обработаны ранее. Однако на практике применяются и методы динамической визуализации, что в свою очередь порождает проблему High performance requirements / высокие требования производительности.

Динамическая визуализация также порождает проблему High rate of image change / высокая скорость изменения изображения. Данная проблема проявляется во время мониторинга за появляющимися данными, и выражается тем, что наблюдатель не успевает реагировать на новые данные.

Многие из рассмотренных проблем присущи и другим методам визуализации больших данных.

Методы трехмерной визуализации данных

Методы трехмерной визуализации базируются на трехмерном изображении. Трехмерное изображение создается на основе модели, сформированной на основе данных визуализации. В данном случае моделью являются данные, представленные с помощью определенного языка или с помощью структуры. Когда модель создана производится построение трехмерного изображения с учетом выбранных точек обзора, угла и уровня освещения. Процесс построения модели и рендеринга изображения определяются условиями задачи, и выбранного способа визуализации данных.

Наиболее популярными моделями при построении трехмерной графики считаются: проволочная каркасная модель, модель «поверхность-ребро-вершина», трехмерная каркасная модель, воксельная модель [8].

Проволочная каркасная модель — использование данной модели, предполагает, что площадь моделируемого предмета содержит плоские куски, при этом все ребра предмета можно представить в виде прямолинейных отрезков.

Также при визуализации данных можно использовать обобщенную проволочно-каркасную модель – модель «поверхность-ребро-вершина». В данном случае, данные о всех элементах объекта (ребра, вершины, плоскости) хранятся структурированно. При этом структура данных хранит данные о связях элементов объекта между собой.

При использовании трехмерной каркасной модели необходимо задать множество вершин и ребер, которые образуют многоугольники, чаще треугольники. Данная модель нашла широко применения в задачах визуализации рельефа (рис.4).

Воксельная модель — это аналог растровой графики для трехмерной графики. В данном случае вместо пикселей используются воксели и используется не двумерный массив, а трехмерный. Данная модель нашла широкое применение в области геологии (рис.5), сейсмологии и т. п.

Задача визуализации по построенной ранее модели сводится к определению видимой области объекта с заданной точки обзора, расчет значения пикселей, их отображение с учетом ранее заданных свойств элементов объекта.

Также существует возможность создания трехмерной графики без создания ее геометрической модели. В работе [9] автор привел методы визуализации на основе большого набора изображений.

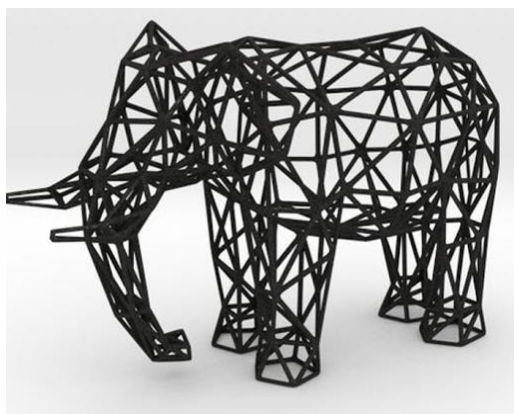


Рисунок 4 – 3D-визуализация модели на основе каркасной модели

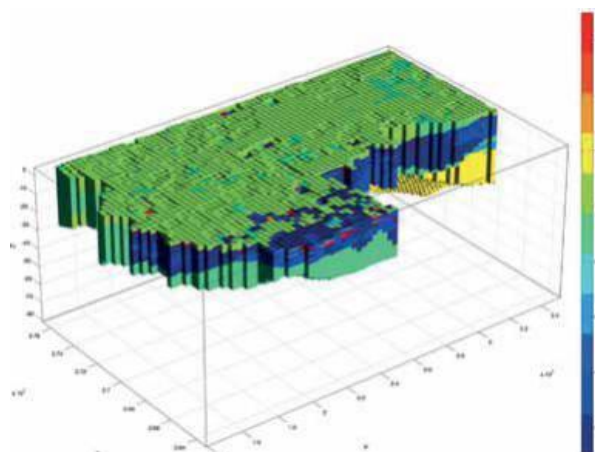


Рисунок 5 – 3D-визуализация геологической модели на основе воксельной модели

Проблемы системного сглаживания границ

При решении инженерных и наукоемких задач с использованием методов трехмерной визуализации данных часто приходится работать с множеством графических объектов, также ученым и инженерам необходимо иметь возможность динамически менять угол обзора при просмотре трехмерных моделей. Нередко границы объектов трехмерной графики имеют рваные границы (алиасинг), что имеет негативное влияние при визуальном анализе.

Наиболее простой метод обработки изображения, чтобы убрать алиасинг, является использование системных функций антиалиасинга встроенного в большинство современных платформ. Наиболее стандартным алгоритмом антиалиасинга является сглаживание на основе множественной выборки [10]

Несмотря на то, что стандартные алгоритмы дают хороший результат, они имеют ряд недостатков:

1. Значительное сокращение эффективности визуализации.
2. Повышенная платформенно-зависимость алгоритма.
3. Нежелательное сглаживание определенных объектов визуализации.

Сокращение эффективности является основной проблемой, так исследования показывают, что производительность может проседать почти в три раза [10]. Так необходимо учитывать, что при динамическом изменении угла обзора системный антиалиасинг не будет позволять происходить изменениям плавно.

Вторая проблема объясняется тем, что программный код алгоритма системного сглаживания отличается у каждой из операционных систем [10].

Третья проблема проиллюстрирована на рисунке 6, то есть если на сцене находятся объекты, границы которых нет необходимости сглаживать, но стороны данного объекта параллельны сторонам экрана, то к данному объекту будет применяться алгоритм антиалиасинга.

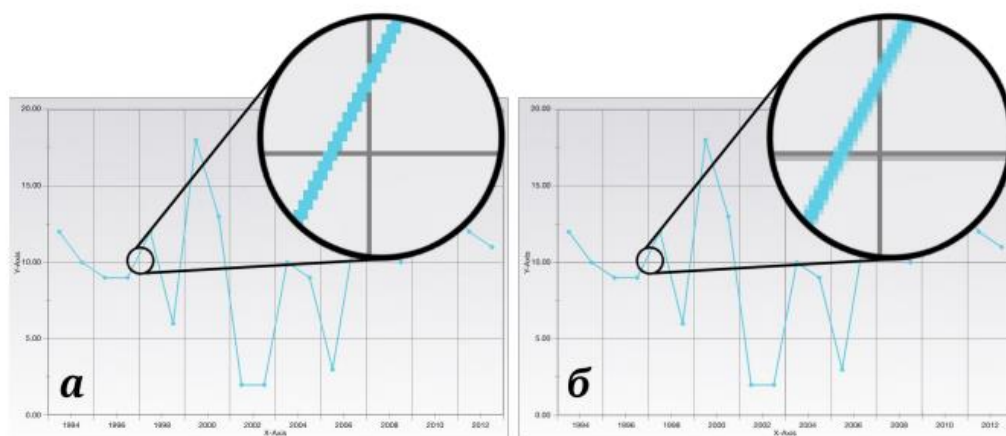


Рисунок 6 – Сцена без антиалиасинга (а) и сцена с системным антиалиасингом, порождающим нежелательное размытие вертикальных и горизонтальных линий (б)

Методы стереовизуализации данных

Одна из целей трехмерной визуализации – создание ощущения объемности визуализируемого объекта. Однако объемности можно достичь и путем разрозненных наблюдений ракурсов визуализируемых данных, таким образом можно создать стереоэффект у наблюдателя. Однако возможность визуализации данных данным методом ограничивается наличием специальной компьютерной техники.

Методы стереовизуализации данных являются необходимыми во время решения научных и инженерных задач. Широкое применение данные методы нашли в области виртуальной и дополненной реальности. Наиболее распространенными способами стереовизуализации данных являются: эклипсный метод, поляризационный метод, растровый метод, анаглифический метод [11].

Эклипсный метод заключается в попеременном выводе на экран правого и левого ракурсов. Вместе с отображением визуализированных данных затемняется дисплей, через который наблюдатель смотрит на стереопары.

Поляризационный метод заключается в построении схем стереовидеопроекции. Данный метод использует два видеопроектора, которые параллельно проектируют на один недеполяризующий экран разные стереопары ракурсов [11]. Для того что бы зритель мог наблюдать стереоэффект визуализированных данных на проекторы устанавливаются поляризационные фильтры, а наблюдателю необходимо наблюдать сквозь поляризационные очки.

Особенностью растрового метода является пространственное распределение стереопары. Таким образом изображение строится из множества вертикальных полосок, с чередованием изображений стереопары. Для того что бы полноценно наблюдать объемность визуализированных данных перед экраном помещается растр, который позволяет каждому глазу улавливать только определенные полоски изображения [11].

Анаглифический метод также основывается на разделении ракурсом. Однако это происходит посредством цветового дифференцирования изображения, улавливаемого левым и правым глазом (рис.7). Данный метод имеет преимущества, учитывая его дешевизну, простоту, и применимость. Но подобная универсальность накладывает определенные трудности при разработке программного обеспечения, учитывая разные характеристики дисплеев, и поляризованных очков. В итоге не всегда фильтрация цветов оказывается идеальной, а полученные в ходе обработки цветовые каналы могут быть не полными.

Необходимо учитывать, что незначительная помеха при отображении ракурса на дисплей может привести к разрушению ощущения объемности. Также необходимым условием для всех методов стереовизуализации – наличие специального программного и аппаратного обеспечения.

Анаглифический метод визуализации данных является одним из самых доступных. Однако данный метод обладает недостатками, выражающимися потерей в цветопередаче и повышенной утомляемостью зрителя.

Растровый способа позволяет наблюдать визуализацию без специальных очков, однако разрешение меньше в двое по сравнению с методами двухмерной визуализации, также имеются другие проблемы при анализе визуализированного объекта.

Другие рассмотренные методы лишены подобных недостатков, но имеют более высокую цену.

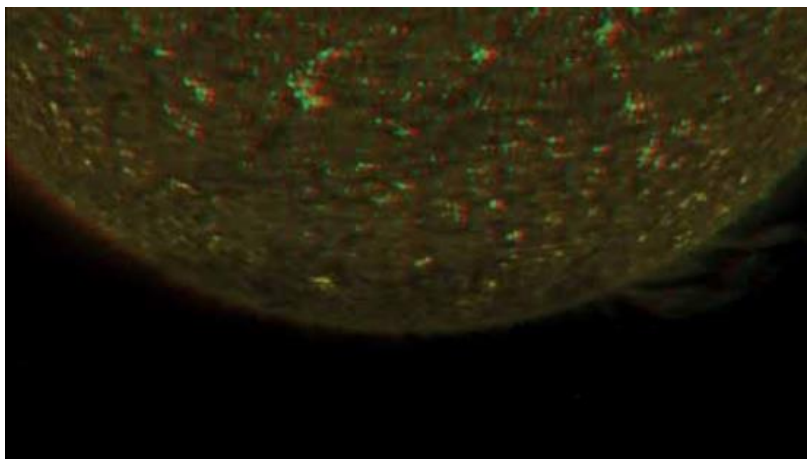


Рисунок 7 – Анаглиф (алгоритм Dubois) по стереопаре Солнца

Перспективы дальнейших исследований

В данной работе был проведен детальный обзор существующих методов графического представления данных подбласти компьютерной визуализации – научной визуализации. В ходе обзора были выявлены следующие проблемы:

- 1) Visual noise / визуальный шум.
- 2) Large image perception / восприятие большого изображения.
- 3) Information loss / потеря информации.

- 4) High performance requirements / высокие требования производительности.
- 5) High rate of image change / высокая скорость изменения изображения.
- 6) Значительное сокращение эффективности визуализации при использовании стандартного алгоритма сглаживания.

7) Повышенная платформенно-зависимость алгоритма антиалиасинга.

8) Нежелательное сглаживание определенных объектов визуализации алгоритма антиалиасинга.

Проблемы 1-3 связаны с процессом визуализации большого количества данных. Понятно, что вначале, большой объем данных необходимо отфильтровать, структурировать, и обработать, и только после этого переходить к процессу визуализации. Детальное изучение способов фильтрации, структурирования, и обработки больших объемов данных даст понять, на какую область, обработка больших данных или визуализация больших данных, необходимо возложить задачи решения выявленных проблем.

Проблемы 4-5 напрямую связаны с производительностью систем генерации изображения. Решение данных проблем может лежать в области постбинарного компьютинга и постбинарной логики [12]. Планируется провести детальное изучение данной области и проведение исследование возможностей применения постбинарной логики при визуализации данных.

Решение проблем 6-8 возможно только путем разработки нового алгоритма сглаживания границ визуализируемого объекта. Планируется провести детальное изучение процесса рендеринга для разных платформ операционных систем, выдвинуть требования к новому алгоритму антиалиасинга, и разработка нового алгоритма решения проблем сглаживания.

Литература

- 1.Цыбина, Е. Ю. Компьютерная графика. Сферы применения и перспективы развития // Вестник Хакасского государственного университета им. Н.Ф. Катанова. – 2022. – №2(40). – С. 113–116.
2. Авербух, В. Л. Разработка средств визуализации программного обеспечения параллельных вычислений. Визуальное программирование и визуальная отладка параллельных программ. / В. Л. Авербух, А. Ю. Байдалин // ВАНТ. Сер. Математическое моделирование физических процессов. – 2003. – вып. 4. – С. 68–80.
3. Авербух, В. Л. Компьютерная визуализация как самостоятельная дисциплина // SCVRT'2011 Труды международных научных конференций «Ситуационные центры и информационно-аналитические системы класса 4i (SC-IAS4i-2011) и «Системы виртуального окружения для комплексной безопасности и антитеррористической защищенности зданий и сооружений» (VRTerro2011). (Сборник посвящается 70-летию Станислава Клименко и Бориса Долговесова). Москва-Протвино. Институт физико-технической информатики. – 2011. – С. 140-145.
4. Борисов, Е. Е. Визуализация контента информационных ресурсов органов местного самоуправления (на материале Белгородской области) : специальность 42.04.02 «42.04.02 Журналистика. Коммуникативистика и связи с общественностью» : выпускная квалификационная работа / Борисов Евгений Егорович ; рук. Крюкова Светлана Валентиновна ; ФГАОУ ВО НИУ «БелГУ». – Белгород, 2019. – 72 с.
5. Баснова Г.М. Вопросы построения базовой библиотеки научной визуализации для windows // Наука и техника Казахстана. – 2009. – No 1. – С. 34-43.
6. Большой энциклопедический словарь : [А-Я] / гл. ред. А. М. Прохоров. – 2-е изд., перераб. и доп. - М. : Большая рос. энцикл. ; СПб. : Норинт, 2000. – 1434,[16] с.; ISBN 5-7711-0004-8.
7. Gorodov Analytical Review of Data Visualization Methods in Application to Big Data / Gorodov, V.Gubarev // Journal of Electrical and Computer Engineering. – 2013. – pp7
8. Шапиро, Л. Компьютерное зрение / Л. Шапиро, Дж. Стокман ; пер. с англ. А. А. Богуславского ; ред. С. М. Соколова. М.: Бином. Лаборатория знаний. 2006. С. 608–614, 685–688.
9. Форсайт, Д. Компьютерное зрение. Современный подход / Д. Форсайт, Ж. Понс. М.: Изд. дом «Вильямс», 2004. С. 837.
10. Рябинин, К. В. Методы и средства разработки адаптивных мультиплатформенных систем визуализации научных экспериментов : специальность 05.13.11 «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей» : дис. на соиск. ученой степени канд. физ.-мат. наук / Рябинин Константин Валентинович; Пермский государственный национальный исследовательский университет. – Пермь, 2014. – 208 с.
11. Корнева, Н. Н. Визуализация данных в задачах наземного сегмента научных космических проектов / Н. Н. Корнева, В. Н. Назаров. - Москва : ИКИ, 2013. - 44, [3] с. : цв. ил.; (Федеральное государственное бюджетное учреждение науки Институт космических исследований Российской академии наук (ИКИ РАН); Пр-2173).
12. Аноприенко, А. Я. Введение в постбинарный компьютинг арифметико-логические основы и программно-аппаратная реализация / А. Я. Аноприенко, С. В. Иваница ; ГОУВПО «ДОННТУ». – Донецк : ДОННТУ, 2017.

Иваница С.В., Мишустин В.А. Современное состояние методов компьютерной обработки информации при визуализации объектов. В работе дается определение компьютерной визуализации. Рассматриваются подобласти компьютерной визуализации. Приводится краткий обзор приложений каждой из подобластей. Детально проводится анализ существующих методов научной визуализации. В ходе анализа выявлены проблемы, возникающие в ходе визуализации данных, также объясняются причины возникновения данных проблем. Анализ выявленных проблем определяет дальнейшее направление исследований по каждой из проблем.

Ключевые слова: визуализация, многомерная визуализации данных, моделирование.

Ivanitsa S., Mishustin V. The current state of computer information processing methods for object visualization. The paper defines computer visualization. Subdomains of computer visualization were considered. A brief overview of the applications of each of the subdomains was given. The analysis of existing methods of scientific visualization is carried out in detail. During the analysis, problems arising during data visualization are identified, and the causes of these problems are also explained. The analysis of the identified problems determines the further direction of research on each of the problems.

Key words: visualization, multidimensional data visualization, modeling.

Проблематика поиска изображений по их содержимому и подходы к её разрешению

О.А. Серёженко^{*1}, С.А. Зори^{*2},

^{*1} аспирант кафедры программной инженерии им. Л.П. Фельдмана, Донецкий национальный технический университет,
seryozhenko-94@mail.ru, OrcID: 0000-0001-9760-3438, SPIN-код: 8231-2521

^{*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

Серёженко О.А., Зори С.А. Проблематика поиска изображений по их содержимому и подходы к её разрешению. В статье выполнен анализ существующих подходов и методов решения задач поиска изображений по их содержимому, определены нерешённые проблемы поиска. Проведен сравнительный анализ существующих алгоритмов, используемых при проектировании систем обратного поиска, определены приоритетные направления дальнейших исследований и разработок.

Ключевые слова: CBIR, дескриптор, гистограмма, признаки изображения, цвет, текстура, форма, свёрточная нейронная сеть.

Введение

Большой рост Интернета и цифровых технологий привел к значительному увеличению количества доступных цифровых изображений. Для хранения изображений не требуются большие вычислительные затраты, однако простой поиск и извлечение уже подразумевают наличие полноценной системы, которая может эффективно организовать эти данные. Необходимо иметь универсальную систему для организации большой графической базы данных, которая является частью системы поиска изображений на основе их содержимого (CBIR). Методы CBIR относятся к различным областям, таких как сегментация изображения, извлечение и представление признаков, хранение и индексирование, измерение подобия изображений и поиск.

Система CBIR использует визуальное содержимое изображений, описанное в виде характеристик низкого уровня, таких как цвет, текстура, форма и пространственное положение. Далее происходит извлечение похожих изображений, когда образец представлен в качестве входных данных для системы. Подобный запрос устраняет необходимость описания визуального содержимого изображений словами и приближается к человеческому восприятию визуальных данных [1].

Семантика изображения относится к значению содержимого изображения. В отличие от низкоуровневых визуальных признаков изображения, семантика является высокоуровневым понятием. Часто изображения со схожими низкоуровневыми характеристиками могут содержать разную концепцию, например, изображение Солнца и изображение апельсина имеют схожие цветовые характеристики, но содержат совершенно разные концепции.

Точно так же изображения, имеющие разные низкоуровневые визуальные признаки, могут содержать одну и ту же концепцию, например, изображения таких объектов, как транспортное средство, могут иметь разный цвет и форму, следовательно, разные низкоуровневые характеристики, но при этом они представляют одну и ту же концепцию.

Это несовпадение между элементами низкого уровня и понятиями высокого уровня, присутствующими в изображениях, известно, как семантический разрыв. Одна из самых важных проблем систем CBIR — как преодолеть этот семантический разрыв.

К настоящему времени разработано множество способов автоматического извлечения характеристик содержимого изображения из визуальных данных и использования извлеченной информации о содержании изображения для поиска изображений в ответ на запрос изображения.

Анализ литературных источников по проблематике CBIR

В [2] была предложена быстрая система CBIR, основанная на слиянии взвешенного цвета, текстуры, моментов цветности, цветового перцентиля и локального бинарного шаблона с последующей классификацией изображений с контролируемым запросом и моделью поиска.

В [3] для выравнивания вектора признаков был предложен метод нормализации в виде линейного масштабирования до единичной дисперсии. На основе этого метода был использован метод поиска K ближайших соседей с равным средним для поиска первых K ближайших «соседей» запроса. Эти «соседи» считались извлечёнными изображениями.

В [4] была предложена система CBIR, состоящая из двухэтапной индексации и извлечения изображений. На первом этапе для ускорения задачи индексации использовались распределенная модель MapReduce на Spark и распределенная система хранения, ориентированная на память Tachyon. На втором этапе в Apache Spark была реализована параллельная модель MapReduce с k -ближайшими соседями (k -NN).

В [5] предложен эффективный метод, основанный на разложении и обнаружении границ для поиска и извлечения изображений. Пиксели были классифицированы на основе краевых и внутренних пикселей для выбора функции.

В [6] предложен метод поиска, который основан на классификаторе опорных векторов. Кодирование цветовой строки используется для извлечения признаков. Это сравнение строк полезно для уменьшения вычислительной сложности.

В [7] предложен алгоритм, основанный на получении векторов признаков для решения задачи поиска в динамической среде. Для эффективного поиска проводится сохранение соответствующих векторов признаков в файле протокола.

В [8] предложен метод, основанный на методах «мягких» вычислений. Была создана структура, основанная на этих методах для уменьшения семантики. Эта система основана на механизме опорных векторов, которая классифицирует изображения на две части: одна является релевантной, а другая — нерелевантной.

В [9] предложен интегрированный метод поиска для извлечения признаков цвета и текстуры. Был использован цветовой момент для извлечения цветовых признаков, а локальный бинарный шаблон — для извлечения текстурных признаков на изображениях в градациях серого. Затем он объединяется для формирования вектора признаков. Затем на основе меры подобия сравниваются евклидовы расстояния изображений базы данных с изображением-образца.

В [10] исследуются и сравниваются методы поиска изображений на основе цвета, текстуры и формы, а также время, как метод обратной связи релевантности, точность системы. Особое внимание уделяется концепции «методов поиска изображений на основе семантики», которые были предложены и которые лучше подходят для преодоления проблемы «семантического разрыва» в контексте поиска изображений.

В [11] использовали преимущества цвета, текстуры и особенности гистограммы для поиска изображения. Для извлечения текстуры использовалась матрица совпадения уровней серого. Гистограмма затенения вычисляет количество пикселей в тусклом уровне. После этого применяется евклидово расстояние, нейронная сеть и алгоритм методов целевого поиска для извлечения изображений из базы данных.

Однако бывает так, что пользователь не в состоянии связать полученные результаты поиска и введенный запрос. Нередко возникает ситуация, когда пользователь не может понять, насколько хорошо результаты поиска соответствуют запросу, или получает нерелевантные, но подходящие по внешним признакам (например, по цветовой гамме) результаты [11].

Такие случаи, как правило, являются следствием двух основных причин. Первая из них — ориентация технологии CBIR на класс цифровых изображений, представляющих только типичную визуальную информацию — изображения, полученные только в видимом свете. Вторая причина — «семантический разрыв» между низкоуровневым обобщенным представлением изображения и семантикой этого изображения.

Неоднозначность запроса на поиск и его результата в совокупности с отмеченными выше причинами стимулируют появлению новых тенденций в технологии поиска информации по содержанию [11].

Помимо семантического разрыва, на сегодняшний день можно выделить ещё один важный аспект CBIR — сенсорный разрыв [13].

Сенсорный разрыв — это неизбежное различие между реальным объектом и информацией, которую наблюдатели могут собрать об объекте.

Технология CBIR сводится к двум фундаментальным проблемам:

1. Математическое описание содержимого изображения (характеристики изображения).
2. Оценить сходство между парой изображений на основе их подписей.

Проблема CBIR мотивирована необходимостью эффективного и действенного поиска в экспоненциально растущем пространстве баз данных изображений.

Анализ рассмотренных методов систем поиска изображений позволил выделить следующие особенности:

1. Рассмотренные системы поиска изображений по содержанию можно разделить на семантические (использующие выделение отдельных регионов/объектов для индексации) и несемантические (анализирующие характеристики изображения целиком).

2. При построении новой системы поиска в графических базах данных изображений следует обратить внимание на задачи, для решения которых она проектируется, и в зависимости от них – выбрать общий подход к построению механизма поиска, а также набор особенностей, которые стоит реализовать.

3. При выборе визуальных примитивов для поиска необходимо учитывать, что быстродействие системы поиска напрямую связано с количеством характеристик (чем больше примитивов, тем медленнее поиск), которое должно быть достаточным для полного описания всех особенностей анализируемых изображений.

Поиск изображений на основе содержимого с использованием комбинированной гистограммы

Исследования показывают, что информация о цвете и ориентации краёв играет важную роль в восприятии изображения человеком. В [12-14] были предложены подходы для извлечения признаков цвета или ориентации краёв. Тем не менее, эффективная интеграция цветов и краёв по-прежнему остается сложной задачей.

В [15] предложен метод, который эффективно извлекает и интегрирует цветовые и краевые особенности изображения. При описании предлагаемой комбинированной гистограммы сначала кратко описываются цветовое пространство HSV и цветовое квантование. Затем описывается обнаружение ориентации края и квантование ориентации в цветовом пространстве HSV, затем производится ввод признаков.

Цветовое пространство — это математическая модель для представления цветных изображений. Цветовое пространство HSV является одним из наиболее распространенных цветовых пространств, используемых в приложениях цифровой обработки изображений и машинного зрения. Цветовое пространство HSV пытается имитировать то, как люди воспринимают цвет. HSV означает оттенок, насыщенность и значение. Оттенок представляет собой чистый цвет, насыщенность — это количество серого, добавленное к чистому цвету, а значение описывает яркость цвета.

В предлагаемом методе используется цветовое пространство HSV, поскольку предлагаемый метод основан на цветовой гистограмме, и в таких методах использование цветового пространства HSV приводит к лучшей производительности поиска, когда цветовое пространство равномерно квантовано. Основная причина заключается в том, что при равномерном квантовании цветового пространства количество похожих цветов, лежащих в разных ячейках, минимально в цветовом пространстве HSV по сравнению с другими цветовыми пространствами.

Чтобы извлечь цветовые характеристики и упростить реализацию, цветовое пространство сначала квантуется. Задача цветового квантования состоит в том, чтобы выбрать ограниченный набор цветов из всего цветового пространства для представления цветного изображения.

Цветные изображения в основном включают 3 цветовых канала, таких как тон, насыщенность и значение в цветовом пространстве HSV. Чтобы определить ориентацию краёв на изображении, сначала необходимо вычислить градиент изображения. Существует множество методов вычисления градиента цветового изображения. Одним из распространенных методов является сначала преобразование цветного изображения в изображение в оттенках серого, а затем вычисление градиента изображения на основе изображения в оттенках серого.

В этом случае большая часть хроматической информации будет потеряна. С другой стороны, обработка трёх отдельных плоскостей путем применения оператора градиента отдельно в каждом канале для формирования составного градиентного изображения может привести к ошибочным результатам. Поскольку обычные операторы градиента неприменимы к векторным функциям, понятие градиента цветного изображения распространяется на вектор максимальной скорости изменения скалярной функции $f(x,y)$ в координатах (x,y) . Чтобы вычислить направления рёбер в цветовом пространстве HSV можно использовать оператор Собеля, так как он менее чувствителен к шуму и хорош с точки зрения вычислений.

Поиск изображений, основанный на сочетании признаков цвета, текстуры и формы

Чтобы повысить производительность поиска изображений, необходимо объединить более одного визуального признака. Простейшим методом является объединение двух или трёх дескрипторов для создания одного дескриптора с большим количеством элементов. Также можно объединить дескрипторы, относящиеся к признакам цвета, текстуры и формы, для создания одного завершённого бинарного дескриптора области.

В иной комбинированной схеме различные типы дескрипторов можно применить к базе данных изображений каскадным или параллельным образом. В первом случае сначала применяется дескриптор на основе цвета для извлечения подмножества релевантных изображений, а затем дескриптор на основе формы, но только к изображениям, принадлежащим только ранее извлеченному подмножеству. В параллельной структуре несколько подмножеств релевантных изображений извлекаются независимо с использованием различных типов дескрипторов, а затем на этапе принятия решения создается окончательный набор релевантных изображений из всех извлеченных подмножеств изображений.

Визуальные признаки также можно линейно комбинировать для повышения производительности поиска, в котором сходство каждого признака вычисляется независимо с использованием метрики расстояния. Затем определяется адекватный вес для каждого признака для создания взвешенной линейной комбинации оценок

сходства различных признаков. Эта комбинированная схема может обеспечить построение гибкой схемы CBIR в зависимости от назначения приложения. Например, в системе поиска медицинских изображений, в которой признак текстуры может быть более важным, чем другие признаки, значение веса, присвоенное признаку текстуры, может быть более важным, чем признаки цвета и формы.

Дескрипторы цвета делятся на две категории:

- глобальные дескрипторы цвета, которые учитывают все изображения для извлечения информации о цвете, этот процесс не включает предварительную обработку или сегментацию изображения;
- локальные дескрипторы цвета, которые извлекают пространственную информацию о том, как пиксели распределены в определенной области, и это делается с помощью предварительной обработки или сегментации изображения.

Далее описаны различные методы извлечения цветовых признаков и приведена их сравнительная характеристика (см. табл. 1).

1. Цветовая гистограмма является наиболее часто используемым методом и проста в реализации. Гистограмма изображений строится путем подсчета количества пикселей каждого цвета в изображении, и пиксели каждого цвета содержатся в отдельных ячейках. Гистограмма изображения не изменяется при повороте и перемещении плоскости изображения, а изменяется медленно при изменении угла обзора.

2. Цветовая коррелограмма дает информацию о том, как пары цветов меняются с расстоянием. Цветовая коррелограмма не является методом уточнения гистограммы или методом разделения изображения. Особенности цветовой коррелограммы являются то, что она описывает соотношение цветов в пространственной плоскости и используется для описания глобального распределения посредством локальной пространственной корреляции цветов.

3. Дескриптор доминирующего цвета описывает типичные цвета в изображении или области изображения. Он используется для извлечения похожих изображений из базы данных на основе одного или нескольких значений цвета. Данный метод может обеспечить мощное и компактное представление ярко выраженного цвета по сравнению с обычными дескрипторами на основе гистограмм.

4. Матрица совпадения цветов — это распространенный метод, используемый для захвата цветовых вариаций в изображении, который придает характеристику цвета. Он используется для расчета вероятности появления одного и того же цвета пикселя между каждым пикселем и соседним пикселем. Каждый пиксель в изображении соответствует четырем соседним пикселям, поэтому каждое изображение может быть представлено четырьмя мотивами изображения шаблона сканирования, которые могут быть дополнительно сконструированы в четыре двумерные матрицы. Из этих четырех матриц генерируются мотивы шаблона сканирования, чтобы зафиксировать изменение цвета в изображении.

Таблица 1 – Сравнительный анализ методов выделения цветовых признаков

Метод	Достоинства	Недостатки
Цветовой гистограммы	Простой в использовании и быстрый расчёт	Потеря пространственной информации и отсутствие цветового сходства
Цветовой коррелограммы	Даёт пространственную информацию	Медленные вычисления и высокая размерность
Дескриптора доминирующего цвета	Обеспечивает эффективное представление цветов	Генерация неправильного ранжирования при индексации изображений
Матрицы совпадения цветов	Фиксирует цветовые вариации изображения	Нет конкретного недостатка

Текстура является важным свойством для характеристики и распознавания изображения.

Под текстурами понимаются видимые узоры, которые обладают свойствами подобия и невозможны из-за существования только одного цвета или интенсивности. Текстура содержит ценную информацию о структурном расположении поверхностей и их отношении к окружающей среде. Различные свойства текстуры распознаются человеческим глазом, такие как регулярность, направленность, гладкость и грубость.

Далее описаны различные методы распознавания текстур и приведена их сравнительная характеристика (см. табл. 2).

1. Признаки Тамуры основаны на визуальном распознавании человеком. В основном выделяют шесть текстурных признаков: грубость, контраст, направленность, линейность, регулярность и зернистость. По результатам экспериментов в [17] был сделан вывод, что первые три признака являются наиболее важными признаками.

Грубость — самая основная характеристика текстуры. Она изменяется непосредственно при изменении масштаба и частоты повторения. Изображение может иметь текстуры в нескольких масштабах. Целью грубости является определение наибольшего размера, при котором существует текстура.

Контраст — мера того, насколько сильно и резко может меняться цвет на изображении. Контраст изображения используется для захвата динамического диапазона уровней серого с поляризацией распределения чёрного и белого.

Направленность — признак, измеряемый с помощью гистограммы локальных направлений контуров. Есть две основные маски, используемые для обнаружения краёв изображения: угол и величина подсчитываются для каждого пикселя. Затем создается гистограмма вероятности края путем подсчета всех пикселей с величиной, превышающей пороговое значение. Гистограмма будет отражать степень направленности.

2. **Вейвлет-преобразование** включает в себя анализ текстуры и классификацию с использованием подхода с несколькими разрешениями. При вейвлет-преобразовании изображение раскладывается в соответствии со сдвинутой и расширенной вейвлет-функциями. Двумерное дискретное вейвлет-преобразование эффективно получается путём применения банка фильтров к каждому столбцу изображения, а также применения банка фильтров к каждой строке результирующего коэффициента.

3. **Вейвлет-преобразование Габора** является наиболее широко используемым методом для извлечения признаков текстуры. Он использует подход с несколькими ориентациями и несколькими разрешениями для анализа текстуры. Функция Габора получается с помощью Гаусса, модулированного комплексной синусоидой. Характеристики изображения вычисляются путём применения фильтра при различной ориентации и масштабе, получается массив величин. При вычислении среднего значения и стандартного отклонения величины преобразованных коэффициентов используются для представления похожей текстуры изображения. Следующее среднее значение и стандартное отклонение преобразованных коэффициентов дают аналогичную текстуру области.

Таблица 2 – Сравнительный анализ методов выделения текстурных признаков

Метод	Достоинства	Недостатки
Признаки Тамуры	Создаёт гистограмму для особенностей текстуры	Представление определенных типов текстуры
Вейвлет-преобразования	Меньше времени поиска	Низкая производительность по сравнению с вейвлет-преобразованием Габора
Вейвлет-преобразования Габора	Достигает самых высоких результатов поиска	Вычислительно интенсивный

Форма играет важную роль в распознавании и восприятии человека, обеспечивая мощный ключ к идентификации объекта.

В отличие от анализа признаков пространственной взаимосвязи, преобразование формы проецирует контур или область формы в другую область, чтобы получить некоторые из его внутренних характеристик. Для описания формы всегда существует компромисс между точностью и эффективностью. С одной стороны, форма должна быть описана максимально точно; с другой стороны, описание формы должно быть максимально компактным, чтобы упростить индексацию и поиск. Для алгоритма анализа преобразования формы очень гибко выполнять описание формы с различной точностью и эффективностью, выбирая количество коэффициентов преобразования.

Извлечение признака формы в соответствии с человеческим восприятием — непростая задача. Из-за того, что человеческое зрение и восприятие представляют собой необычайно сложную систему, нельзя надеяться на то, что машинное зрение обладает превосходными характеристиками при небольшой сложности. Кроме того, при выборе подходящих функций для системы распознавания формы необходимо учитывать, какие функции подходят для данной задачи. Не существует общей функции, которая лучше всего работала бы для каждого типа изображений.

С одной стороны, форма должна быть описана максимально точно; с другой стороны, описание формы должно быть максимально компактным, чтобы упростить индексацию и поиск.

Дескрипторы на основе формы можно разделить на два класса:

– дескрипторы на основе контуров, которые используют только информацию о границах, игнорируя важную информацию внутри объектов;

– дескрипторы на основе областей, которые используют обе, границы и внутреннюю информацию объектов.

Далее описаны различные методы распознавания текстур и приведена их сравнительная характеристика (см. табл. 3).

Геометрические моменты рассматриваются как некоторое конкретное средневзвешенное значение (момент) интенсивности пикселей изображения. Моменты изображения полезны для описания объектов после сегментации. Некоторыми свойствами изображения, которые можно найти с помощью моментов изображения, являются площадь (или общая интенсивность), информация об ориентации и его центр тяжести. Функция

моментов использует только одно значение для вектора признаков, а производительность текущей реализации плохо масштабируется, поэтому это означает, что вычисление вектора признаков занимает много времени, когда размер изображения становится относительно большим.

Алгебраические моменты вычисляются из первых m центральных моментов, и они рассматриваются как собственные значения некоторых предопределённых матриц $M_{[j,k]}$, элементы которых являются масштабными коэффициентами этих m центральных моментов. Алгебраические моменты инвариантны к аффинным преобразованиям, а также могут быть построены до произвольного порядка.

Дескрипторы Фурье являются одной из надёжных функций для представления различных границ и объектов. Дескрипторы Фурье получаются путём применения преобразования Фурье к сигнатуре формы, которая представляет собой одномерную функцию, полученную из координат границы формы [16]. С дескрипторами Фурье глобальные признаки формы фиксируются первыми несколькими низкочастотными элементами, в то время как элементы с более высокой частотой охватывают более тонкие особенности формы.

Дескриптор Фурье на основе области генерируются путём применения модифицированного преобразования Фурье к изображению. Аппроксимированное нормализованное изображение поворачивается против часовой стрелки на достаточно малый угол, а значения пикселей вдоль положительного x -направления, начиная с центра изображения, копируются и вставляются в новую матрицу в виде элементов строки. Данные шаги повторяются до тех пор, пока изображение не будет повернуто на 360° .

Таблица 3 – Сравнительный анализ методов выделения признаков формы

Метод	Достоинства	Недостатки
Геометрические моменты	Вычислительная сложность меньше Сочетание преимуществ с другими методами для обеспечения лучшего результата [18]	Чувствительность к изменению яркости Низкая шумоустойчивость Производительность плохо масштабируется
Алгебраические моменты	Вычислительная сложность меньше Могут быть построены до произвольного порядка	Низкая устойчивость к затенению
Дескрипторы Фурье	Прост в реализации и основан на хорошо разработанной теории анализа Фурье	Низкая шумоустойчивость
Дескрипторы Фурье на основе области	Высокая шумоустойчивость Можно использовать для общих приложений	Вычислительно интенсивный

Таким образом, из сравнения вышеописанных методов видно, что эффективнее всего использовать следующие методы:

- матриц совпадения цветов извлечения цветовых признаков;
 - вейвлет-преобразования Габора для извлечения текстурных признаков;
 - дескрипторы Фурье на основе области для извлечения признаков формы.
- Эти методы лучше всего подходят для проектирования CBIR.

Поиск изображений на основе содержимого с использованием нейросетей

При использовании низкоуровневых методов, в основе которых лежат признаки изображений, нередки случаи появления достаточно большого разрыва между низкоуровневыми представлениями и высокоуровневым семантическим значением. Для решения этой проблемы предлагается ряд методов с использованием свёрточных нейронных сетей (convolutional neural network, CNN).

Свёрточные нейронные сети специально нацелены на распознавание образов. В их основе лежит чередование слоёв свёртки и подвыборки. Для различных слоёв используется одна и та же матрица весов (ядро свёртки), которая интерпретируется как графическое кодирование какого-либо признака. В дальнейшем происходит как свёртка – проход ядром свёртки по всем нейронам нижнего слоя с целью получения карты признаков (активация нейронов верхнего слоя), так и подвыборка – уменьшение размерности. Пример архитектуры свёрточной нейронной сети изображён на рисунке 1.

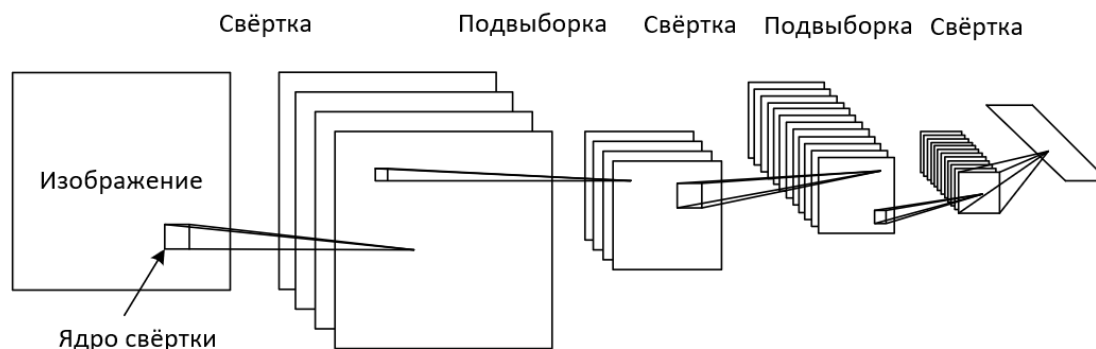


Рисунок 1 – Архитектура свёрточной нейронной сети

Использование таких сетей может дать возможность извлечения компактного представления изображения с учётом семантики для задачи CBIR. Например, предварительно обученная нейронная сеть AlexNet (на основе архитектуры свёрточных нейронных сетей) настраивается посредством алгоритма на наборе данных. Затем выходные данные полностью связанных слоёв используются в качестве вектора признаков для поиска изображений.

Однако существуют и недостатки: все методы на основе CNN, используемых для поиска изображений на основе содержимого, полагаются на равномерное преобразование выходных данных конечного слоя свёртки в вектор признаков. При этом они не имеют возможности локализовать и сфокусироваться на интересующей области изображения. Следовательно, при использовании таких подходов могут возникать серьёзные погрешности из-за нерелевантных данных. Релевантная же информация будет упущена во время поиска изображения. Поэтому необходимо учитывать механизмы взвешивания или визуальное внимание.

Выводы

Огромный объём накопленных человечеством визуальных данных в цифровом виде требует наличия быстрых и эффективных алгоритмов для возможности своевременного поиска по содержимому с учетом низких погрешностей. С этой целью рассмотрены 3 подхода, позволяющие осуществлять эту процедуру с учётом вышеописанных требований:

- поиск изображений на основе содержимого с использованием комбинированной гистограммы;
- поиск изображений, основанный на сочетании признаков цвета, текстуры и формы;
- поиск изображений на основе содержимого с использованием нейросетей.

Тем не менее, данные подходы обладают недостатками. Так, в первом могут быть потери в точности, например, из-за цветовых преобразований. Использование третьего подхода повышает риски потерь релевантной информации в связи с определёнными архитектурными особенностями свёрточных нейронных сетей. Второй же подход содержит в себе целый ряд методов, каждый из которых также обладает своими слабыми сторонами.

Исходя из вышесказанного, следует, что проблема поиска изображений по содержимому на текущий момент сохраняет свою актуальность и требует поиска решения в виде улучшения существующих методов и подходов, либо их эффективной комбинации.

Для достижения этой цели предполагается составить сравнительную характеристику на базе экспериментального исследования точности и быстродействия работы наиболее перспективных методов - матрицы совпадения цветов, вейвлет-преобразования Габора и дескрипторы Фурье, так и свёрточных нейронных сетей на основе различных архитектур. Также необходима проработка стратегии поиска существующих изображений, совокупность которых будет обладать репрезентативностью для данного аналитического эксперимента. Впоследствии, на основе полученных результатов, представится возможным определить как необходимость улучшения конкретного метода, так и сам вектор его модернизации.

Литература

1. Alhassan A.K., Alfaki A.A. Fusion-Based Method for Content Based Image Retrieval. URL: https://www.researchgate.net/profile/Abdolraheem-Khader/publication/314203620_Color_and_texture_fusion-based_method_for_content-based_Image_Retrieval/links/5dcb8c4692851c818049f69f/Color-and-texture-fusion-based-method-for-content-based-Image-Retrieval.pdf DOI: 10.1109/ICCCCEE.2017.7867649
2. Singh V.S., Srivastava S., Srivastava R. An efficient image retrieval based on fusion of fast features and query image classification // International Journal of Rough Sets and Data Analysis (IJRSDA). – 2017. – Vol. 4 (1). – pp. 19–37. DOI:10.4018/IJRSDA.2017010102
3. Zhe-Ming L., Burkhardt H. and Boehmer S. Fast content-based image retrieval based on equal-average k-nearest-neighbor search schemes // Pacific-Rim Conference on Multimedia, Springer. – 2006. – pp. 167–174. DOI:10.1007/11922162_20

4. Mezzoudj S., Rachid S., Yassmina S. A Parallel Content-Based Image Retrieval System Using Spark and Tachyon Frameworks // Journal of King Saud University-Computer and Information Sciences. – 2019. – Vol. 33 (1). – pp. 1–9. DOI: 10.1016/j.jksuci.2019.01.003
5. Enescu F.M., Știrbu C., Liță A.I. Image search algorithms // Conference: 2015 7th International Conference on Electronics, Computers and Artificial Intelligence (ECAI). – pp. 35–40. DOI: 10.1109/ECAI.2015.7301248
6. Ladjel B., Patrick V., Tadeusz M. Advances in Databases and Information Systems. URL: <https://link.springer.com/book/10.1007/978-3-030-82472-3> DOI: 10.1007/978-3-030-82472-3.
7. Lingxi X., Jingdong W., Bo Z., Qi T. Fine-Grained Image Search // IEEE Transactions On Multimedia. – 2015. – Vol. 17 (5). – pp. 636-647. DOI: 10.1109/TMM.2015.2408566
8. Smelyakov K., Ruban I., Sandrkin D., Martovytskyi V., Romanenkov Y. Search by Image. New Search Engine Service Model. URL: https://www.researchgate.net/publication/330162417_Search_by_Image_New_Search_Engine_Service_Model/citation/download DOI: 10.1109/INFOCOMMST.2018.8632117.
9. Yasmin M., Sharif M., Irum I., Mohsin S. An Efficient Content Based Image Retrieval using EI Classification and Color Features // Journal of applied research and technology. – 2014. – Vol. 12 (5). – pp. 877-885. DOI: 10.1016/S1665-6423(14)70594-2
10. Ulape S.B., Takale S.A. Representative and Diverse Image Set Gathering for Geographic Area and its Surrounding Region // International Journal of Advanced Computer Research (IJACR). – 2014. – Vol. 4 (15). – pp. 549-554.
11. Choudhary R., Raina N., Chaudhary N., Chauhan R., Goudar R.H. An integrated approach to Content Based Image Retrieval // Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference). – pp. 2404-2410. DOI: 10.1109/ICACCI.2014.6968394
12. C. W. Liang and W. Y. Chung, "Color feature extraction and selection for image retrieval," in 2016 International Conference on Advanced Materials for Science and Engineering (ICAMSE), 2016, pp. 589-592.
13. M. Li and J. Xiuhua, "An Improved Algorithm Based on Color Feature Extraction for Image Retrieval," in 2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2016, pp. 281-285.
14. A. Anandh, K. Mala, and S. Suganya, "Content based image retrieval system based on semantic information using color, texture and shape features," in 2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16), 2016, pp. 1-8.
15. Rezaei M., Ahmadi A., Naderi N. Content-based image retrieval using Mix histogram. URL: <https://arxiv.org/ftp/arxiv/papers/1909/1909.09722.pdf>
16. Dipalee Gupta, Siddhartha Choubey, "Discrete Wavelet Transform for Image Processing", International Journal of Emerging Technology and Advanced Engineering, March 2015.
17. J. M. Patel and N. C. Gamit, "A review on feature extraction techniques in Content Based Image Retrieval," 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), 2016, pp. 2259-2263, doi: 10.1109/WiSPNET.2016.7566544.
18. A. Chadha, S. Mallik and R. Johar, "Comparative Study and Optimization of Feature-Extraction Techniques for Content based Image Retrieval", International Journal of Computer Applications, Volume 52– No.20, August 2012.

Серёженко О.А., Зори С.А. Проблематика поиска изображений по их содержимому и подходы к её разрешению. В статье выполнен анализ существующих подходов и методов решения задач поиска изображений по их содержимому, определены нерешённые проблемы поиска. Проведен сравнительный анализ существующих алгоритмов, используемых при проектировании систем обратного поиска, определены приоритетные направления дальнейших исследований и разработок.

Ключевые слова: CBIR, дескриптор, гистограмма, признаки изображения, цвет, текстура, форма, свёрточная нейронная сеть.

Seryozhenko O.A., Zori S.A. The problems of content-based image search and approaches to their solution. The article analyzes the existing approaches and methods for solving the problems of image search by their content, and identifies unsolved search problems. A comparative analysis of existing algorithms used in the design of reverse search systems has been carried out, and priority areas for further research and development have been identified.

Ключевые слова: CBIR, descriptor, histogram, image features, color, texture, shape, convolutional neural network.

Разработка модели и симуляции распространения эпидемии вируса с использованием игрового движка Unity

Бездетный Н. А.^{*1}, Зори С. А.^{*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. Разработка модели и симуляции распространения эпидемии вируса с использованием игрового движка Unity. В статье представлена реализация модифицированной модели симуляции распространения эпидемии вируса, отличающейся большей точностью, простой и гибкой регуляцией настроек, для повышения эффективности процесса моделирования, и применение современных средств трёхмерной компьютерной графики для повышения наглядности процесса симуляции и визуализации его результатов.

Ключевые слова: вирус, эпидемия, прогнозирование, моделирование, компьютерная симуляция, SIR-модели, программирование, игровой движок, Unity, C#.

Zori S. A., Bezdetniy N. A. Development of a model and simulation the spread of a virus epidemic using the Unity game engine. The article presents the implementation of a modified virus epidemic simulation model with greater accuracy, simple and flexible regulation of settings, to improve the efficiency of the simulation process, and the use of modern 3D computer graphics to make the simulation process more visible and visualize its results.

Keywords: virus, epidemic, forecasting, modeling, computer simulation, SIR models, programming, game engine, Unity, C#.

**Полный текст статьи опубликован в выпуске № 3 (29), 2022 г.
научного журнала «Информатика и кибернетика»**

Методика исследования нейросетевой технологии реализации E2E систем синтеза сигнала

И.Н. Паламарь^{*1}, А.И. Гагарина^{*2}

^{*1} к.т.н, доцент, Рыбинский государственный авиационный технический университет имени П.А. Соловьева, irina.palamar@mail.ru, OrcID: 0000-0002-2186-6116, SPIN-код: 5589-5060

^{*2} аспирант, Рыбинский государственный авиационный технический университет имени П.А. Соловьева, aig.rsatu@mail.ru

Паламарь И.Н., Гагарина А.И. Методика исследования нейросетевой технологии реализации E2E систем синтеза сигнала. В статье рассмотрена проблема повышения качества E2E системы, реализуемой на основе нейросетевой технологии синтеза сигнала. Рассмотрены модели архитектуры системы синтеза речи на основе модели свёрточной нейронной сети. Предложены методики включения этапа разработки датасета и оптимизации всех компонентов потока обработки данных. Проведен анализ результатов моделирования системы по методике исследования. Результаты экспериментов по модификации архитектуры и реализации компонентов по оценке MOS показали повышение качества синтеза речевого сигнала.

Ключевые слова: E2E система, системный анализ, нейросетевая технология, глубокое обучение, синтез сигнала, методика исследования.

Введение

E2E (end-to-end) – «сквозная» система, где этапы решения задачи объединены в единую модель, связывая вход и выход. С эффективной реализацией и распространением нейросетевых технологий глубокого обучения стало возможным обучать E2E модели, которые решают задачи преобразования «последовательность – в – последовательность» (seq2seq) [1]. Необходимость таких систем возникает в областях, связанных с генерацией управляющих сигналов для промышленного оборудования, с аугментацией данных, с анализом диагностических сигналов, реализацией голосовых технологий. Системы E2E синтеза сигнала в реализации голосовых технологий напрямую предсказывают акустические признаки по буквам. Так, с использованием нейронных сетей основные этапы классического параметрического подхода могут быть заменены одной нейросетью. Основными преимуществами применения нейросетевой E2E системы синтеза речи являются: сокращение времени обработки данных и объема требуемой памяти, а также объединение внутренних этапов синтеза речи в единую модель, связывая вход и выход, в отличие от классических систем синтеза речи, которые состоят из различных подмодулей (генератор F0, текстовый анализатор, генератор спектра, вокодер) [2]. Спецификой работы с E2E системами синтеза речи является настройка ее параметров под конкретный язык и подбор входного алфавита, что делает данную систему универсальной.

Речевое общение является частью человеко-машинного интеллектуального интерфейса и создает возможность организации естественной для человека связи при взаимодействии с машиной. В настоящее время существует необходимость озвучивания текстов большого объема для решения различных видов задач. Синтез речи широко используется в телефонии, в автомобильных навигационных системах, торговле, информационных, интеллектуальных системах для озвучивания различной информации. Успешная реализация технологии стала возможной благодаря развитию семейства подходов к машинному обучению seq2seq, превращающих одну последовательность в другую, что отражает суть языковой обработки. В случае решения задачи TTS (Text-to-Speech) входной последовательностью является текст, а выходной – акустический сигнал, озвучивающий данный текст.

Системы TTS вышли уже на промышленный уровень внедрения продуктов, однако существуют определенные проблемы, как качества синтезированной речи, так и доступности речевого синтеза для конечных пользователей.

Учитывая уровень развития технологии нейронных сетей, исследователь в настоящее время может сам разработать и обучить математическую модель. Однако необходимо отметить следующие проблемы при реализации доступного нейросетевого синтеза русскоязычной речи:

- низкое качество нейросетевого синтеза русскоязычной речи при ограничении аппаратно-временных ресурсов;
- отсутствие методик по подбору гиперпараметров обучения нейросетевых моделей;

– отсутствие данных о программной реализации технологии.

Целью исследования является выработка рекомендаций по улучшению технологии нейросетевого синтеза речи на основе модели свёрточной сети в системе *E2E*.

Для достижения приемлемого качества речевого синтеза необходимо выполнить планирование и проведение собственных экспериментов с использованием определенных затрат: небольшого объема датасета и ограничении аппаратно-временных ресурсов. Реализация синтеза русскоязычной речи, как обладающей большим количеством вариативных параметров, еще больше осложняет задачу обучения нейронной сети.

Основная часть

В результате анализа групп методов синтеза речи для исследования был выделен нейросетевой подход, который в настоящее время является *state-of-the-art* и позволяет обеспечивать достижение более высокого качества речи по сравнению с классическими методами синтеза. Преимуществами применения нейросетевого подхода в задачах синтеза речи являются:

- скрытое извлечение признаков. Это позволяет эффективно моделировать сильно взаимосвязанные признаки большой размерности;

- нейронные сети могут использовать распределенное представление. Таким образом, захватывая большой участок пространства данных, отсутствует высокая фрагментация, в отличие от таких методов как деревья решений, *k*-средних и другие. Меньшее количество параметров улучшает описательные качества.

В нейросетевой технологии синтеза речи *state-of-art* моделями являются рекуррентная (RNN) и сверточная (CNN) нейронные сети. Были рассмотрены наиболее известные модели такие как: Tacotron 2, WaveNet, Deep Voice 3, DCTTS [3, 4, 5].

RNN является стандартным методом для подхода *seq2seq*, особенно в области машинного перевода или диалоговой системы. Однако такой подход имеет недостатки, например, модель кодер-декодер не может эффективно кодировать слишком длинную последовательность в вектор фиксированной длины. Эта проблема была решена с помощью механизма внимания. Вторая проблема заключалась в том, что рекуррентная нейронная сеть обычно требует много времени для обучения, потому не подходит для решения таких задач как параллельные вычисления с использованием графических процессоров. Чтобы решить эту проблему ряд ученых предложили использовать сверточные нейронные сети вместо рекуррентных сетей. Некоторые из этих исследований показали, что сети на основе CNN могут обучаться намного быстрее и даже превосходить по вычислительным мощностям методы, основанные на RNN.

В результате сравнительного анализа была выбрана модель DCTTS (*Deep Convolutional TTS*.), которая является полностью сверточной, что позволяет обучаться намного быстрее, чем моделям основанных на рекуррентных сетях, не теряя в качестве звука [6]. Архитектура сети похожа на Tacotron [7], но основана полностью на сверточной модели обучения с подходом *seq2seq*. Особенностью модели является использование механизма внимания. Авторы предлагают использовать быструю тренировку механизма внимания и дают ей определение «направленное внимание». На момент выхода этой модели уже были представлены работы, в которых системы TTS основаны на RNN, такие как DeepVoice 1 & 2, Tacotron и система, основанная на CNN – WaveNet. Некоторые из них попытались уменьшить зависимость от ручного выделения признаков. Самым продвинутым методом оказался Tacotron, который зависит только от мел и линейных спектрограмм, а не от других речевых и лингвистических характеристик.

Модель DCTTS является полностью сверточной, как и WaveNet, но отличие в том, что WaveNet – это вокодер (серверная часть), который синтезирует сигнал из некоторой информации, которая передается внешними компонентами, а DCTTS является фронтендом и большей частью серверной обработки. CNN нужен для синтеза спектрограммы, из которой простой вокодер может синтезировать сигнал. На данном этапе нет единого мнения об оптимальной архитектуре нейронной сети для TTS. Однако по существующим характеристикам моделей и связанным с ними результатам исследований можно сделать вывод о том, что для текущих экспериментов работе больше всего подходит модель глубоких сверточных нейронных сетей. Данная модель является полностью CNN, что позволяет обучаться намного быстрее, чем модели на основе RNN, не теряя в качестве звука. Выбор модели также зависит и от имеющегося аппаратного обеспечения таким образом, чтобы было возможно провести вычислительные сложные эксперименты в установленные сроки.

Использование больших объемов данных, благодаря развитию нейросетевых архитектур, происходит с минимальной предварительной обработкой. При этом нейросетевые архитектуры требуют особого подхода к подбору их гиперпараметров. Для повышения качества синтезируемой речи необходимо проведение многочисленных опытов для того, чтобы определить наилучший набор параметров нейросетевой архитектуры. После выбора модели был проведён анализ существующих датасетов в открытом доступе. Проведенные эксперименты показали, что эти датасеты не обеспечивают приемлемое качество речи либо из-за их недостаточного объема, либо потому что они основаны на произвольных текстах.

Рассмотрим структуру системы E2E синтеза речи на основе сверточной нейронной сети (см. рис. 1). Сплошными стрелками обозначен процесс обучения системы, а пунктирными – процесс синтеза сигнала. Функциональная схема системы синтеза речи состоит в следующем. После создания и подготовки датасета, данные попадают на этап их преобразования в вид, пригодный для подачи на нейронную сеть. На этапе обучения

данные проходят через сеть Text2Mel, состоящей из текстового энкодера, аудиоэнкодера, механизма внимания и аудиодекодера, далее проходя через сеть SSRN, которая повышает разрешающую способность спектрограммы, по которой вокодер восстанавливает сигнал. В отличие от аналогичных исследований, использующих существующие готовые датасеты как «черный ящик», необходимо выполнить разработку собственного датасета. Как показали результаты экспериментов, качество используемого датасета влияет на качество синтезируемой речи. Особенностью методики исследования системы синтеза речи на основе сверточной нейронной сети является включение этапа подготовки датасета в технологию синтеза речи. В рассматриваемой системе именно этот этап управляет процессом обучения. Этап оценки качества также включён в процесс обучения и позволяет определять наилучшие параметры модели.

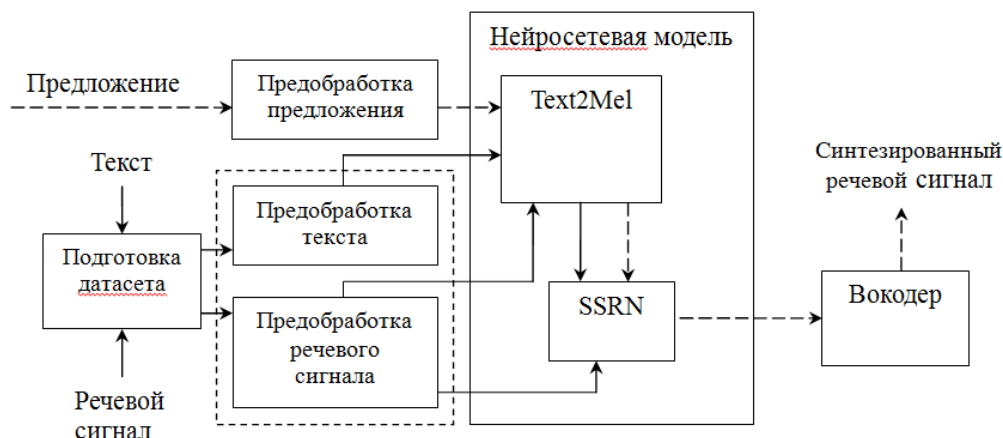


Рисунок 1 – Структура системы синтеза речи на основе сверточной нейронной сети

Схема процесса исследования системы синтеза речи разработана на основе обратной связи на все компоненты E2E системы через оценку качества синтезированного речевого сигнала (см. рис. 2).

Процесс исследования включал в себя этапы создания и разработки датасета, предварительной обработки текста и речевого сигнала, применения нейросетевой модели и её усовершенствование, применения вокодера для восстановления сигнала. Реализация системы синтеза выполнена на языке *Python* с использованием библиотеки *TensorFlow*. Дополнительное исследование параметров модели и алгоритма обучения позволило получить значения, позволяющие обеспечить приемлемое качество синтеза (способность сети к обобщению) и сходимость сети за ограниченное время обучения.



Рисунок 2 – Схема процесса исследования системы синтеза речи

Первоначально в эксперименте были исключены этапы нейросетевого синтеза и оставлены только этапы цифровой обработки сигнала. В результате было установлено, что восстановленная речь заметно потеряла в естественности и в ней появлялись искажения. Поэтому было принято решение, заменяя элементы обработки, оценивать их вклад, которые они вносят в потерю качества речи.

Архитектура модели при замене базового вокодера *Griffin-Lim* отражает сквозной поток преобразований данных (см. рис. 3). Целью эксперимента являлась оценка влияния технологии реализации вокодера на качество синтезированной речи. В блок синтеза нейронной сети подается подготовленный и нормализованный файл, разбитый на предложения. После векторизации и пакетирования (сборки в батчи) они подаются на нейронную сеть *Text2Mel* работающую в режиме вывода, которая предсказывает децимированные мел-спектрограммы предложений. По отдельности каждая децимированная мел-спектрограмма предложения подается на сеть, восстанавливающую амплитудную спектрограмму высокого разрешения. Из-за того, что восстановленная амплитудная спектрограмма лишена фазовой составляющей, она не может быть напрямую использована для восстановления осциллограммы аудио сигнала. Для восстановления осциллограммы аудио сигнала применяют вокодер, который восстанавливает (предсказывает – для нейронного вокодера) исходную форму сигнала. В этой системе базовым вокодером является *Griffin-Lim* [7]. Финальным этапом синтеза речи является сборка

фрагментов синтезированных аудио сигналов в единый аудио файл с помощью библиотеки *ffmpeg* и при необходимости шумоподавления с помощью библиотеки *logmmse*.

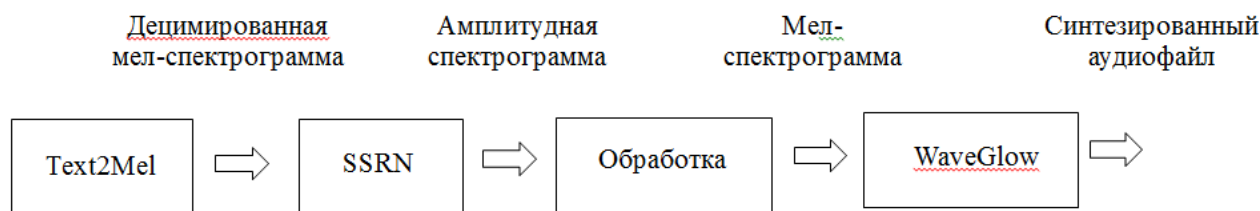


Рисунок 3 – Архитектура модели при замене вокодера

Для этого было предложено использовать вокодер *WaveGlow* и проверить относительно вокодера *Griffin-Lim*, сколько теряется качества речи. В результате восстановленная речь вокодером *WaveGlow* практически не имеет искажений, тогда как восстановление речи вокодером *Griffin-Lim* вносит существенные и заметные искажения. Работа вокодера требует этапа предварительной обработки сигнала, поэтому нужно проводить эксперимент, чтобы установить, какой именно этап влияет на качество речи и на сколько (распределение потери качества между этими этапами).

Первоначально из этапа обработки был исключен вокодер *Griffin-Lim* и заменен на вокодер *WaveGlow*, который преобразует мел-спектрограмму в осциллограмму. Для этого был модифицирован модуль, в котором синтезируемую нейросетью речь сохранили в виде амплитудного спектра в файле, в формате *numpy* массивов. Полученные амплитудные спектрограммы не могли быть напрямую использованы вокодером *WaveGlow*. Поэтому исходный код *WaveGlow* был дополнен функцией предобработки, который производил этапы денормализации спектрограммы (переводил из логарифмической шкалы в абсолютную), динамической декомпрессии, и после получения восстановленной речи от *WaveGlow* были добавлены этапы обрезания тишины (является поиском конца синтеза речи) и де-преэмпасинг (пост частотная фильтрация). Качество речи сильно улучшилось, и резко возросла скорость работы системы. Из-за того, что итоговое качество улучшилось, сделан вывод, что вокодеры вносят существенный вклад в искажение синтезируемого сигнала.

Следующим этапом исключим из системы синтеза речи вторую нейронную сеть. Архитектура модели при замене сети *SSRN* содержит два компонента (см. рис. 4).

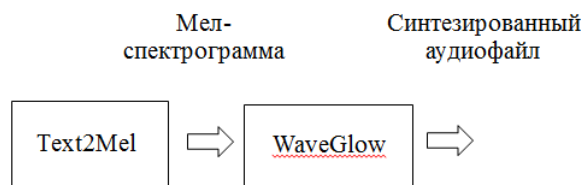


Рисунок 4 – Архитектура модели при замене сети *SSRN*

Однако для его реализации необходимо провести эксперимент обучения первой сети на полных мел-спектрограммах и подтвердить в нем возможность использования в качестве обучающих и предсказываемых мел-спектров полное их представление (без редукции). Был создан собственный модуль, который соответствовал измененным этапам синтезом речи. В результате первая нейронная сеть корректно обучилась на полных мел-спектрограммах (сошлась). Поскольку эксперимент дал положительный результат исключаем вторую сеть из системы синтеза речи. В результате качество синтеза речи немного ухудшилось, поэтому делается вывод о том, что вторая нейронная сеть улучшает качество сигнала после прохождения первой нейронной сети. Первоначально в системе синтеза речи основной вклад в естественность речи вносят первые два блока: предварительной цифровой обработки сигнала и первая нейронная сеть.

Архитектура сети с использованием вокодера *WaveGlow* даёт наилучшее качество речи. Так как вторая нейронная сеть восстанавливает амплитудную спектрограмму, а вокодер *WaveGlow* принимает на вход мел-спектрограмму, поэтому нужно провести преобразование. Для получения мел-спектрограммы из амплитудной спектрограммы нужно применить следующие операции: де-нормализация, выравнивание мощности, редукция, преобразование в мел-шкалу.

Оценка качества синтезируемой речи выполнялась с помощью *MOS*. Оценку лучшего всего проводить в максимально звукоизолированном помещении, но это не является обязательным условием, достаточно, чтобы респонденты слушали речи в наушниках. Выбирается определенное количество образцов для прослушивания, синтезированных на основе текстовой выборки, которые не участвовали в процессе обучения. Респондентам предлагается оценить качество речи по нескольким критериям: разборчивость и естественность. Оценка ставится в диапазоне от 1 до 5 балла с шагом 0,1 в соответствии оценкам качества [8].

Результаты оценки качества синтезируемой речи в зависимости от выбранной архитектуры приведены в таблице 1. Из таблицы 1 видно, что наилучшее качество речи достигается за счёт использования нейросетевой модели на основе глубоких сверточных сетей с механизмом внимания, состоящей из двух последовательно работающих нейронных сетей, и вокодером *WaveGlow*. Таким образом, за счёт модификации существующей архитектуры качество синтезируемой речи улучшилось с 2,6 до 3,6 по оценке MOS.

Таблица 1 – Результаты оценки MOS в зависимости от выбранной архитектуры

Название эксперимента	Разборчивость речи	Естественность речи	MOS
Исходный женский голос	4,6	4,4	4,5
Синтезируемый женский голос с помощью DCTTS + Griffin-Lim	3,0	2,2	2,6
Синтезируемый женский голос с помощью DCTTS + WaveGlow	3,8	3,5	3,6

В ходе анализа полученных оценок была установлена связь между оценками и тем, знаком ли респондент с синтезированной речью. Респонденты, не привыкшие к синтезируемой речи, склонны давать более низкие оценки естественности синтезируемой речи, чем остальные. Для проверки значимости данной закономерности воспользуемся *U*-критерием Манна-Уитни. При заданной численности сравниваемых групп критическое значение *U*-критерия Манна-Уитни составляет 0. Поэтому делается вывод, что различия в оценках в сравниваемых группах статистически незначимы (показатель $p > 0.05$). Доли потерь качества речи на каждом этапе нейросетевого синтеза при исследовании архитектуры E2E системы приведены в таблице 2.

Таблица 2 – Доли потерь качества речи на каждом этапе нейросетевого синтеза

Критерий	Вид голоса	Сигнал без обработки, исходный голос	Только цифровая обработка сигнала с вокодером Griffin-Lim	На вход второй сети SSRN мел-спектрограмма из исходного сигнала	Синтезированная речь со всеми этапами
MOS	Женский	4,6	3,6	3,9	3,8
	Мужской	4,7	3,8	3,7	3,4
Доля потерь качества, %	Женский	–	21	15	17
	Мужской	–	19	21	27

Для определения доли потери качества на каждом этапе нейросетевого синтеза был проведен опрос респондентов для вычисления оценки MOS. Эксперимент был проведен на двух голосах: женском и мужском. Получившиеся убывающие оценки MOS характеризуют постепенное падение качества синтезированного голоса на каждом этапе. Но для женского голоса выяснилось, что вторая нейросеть подавляет появившиеся искажения вследствие прохождения этапа цифровой обработки сигнала, поэтому общая оценка MOS для первого этапа оказалась ниже, чем для второго.

Выводы

В результате работы был выполнен системный анализ всех аспектов технологии синтеза речи на основе сверточных нейронных сетей и разработаны рекомендации для создания системы E2E синтеза.

Полученные оценки MOS для различных архитектур показывают, что предложенная замена вокодера Griffin-Lim на вокодер WaveGlow улучшает качество синтеза при резком возрастании скорости работы системы. Следовательно, вокодеры вносят существенный вклад в искажение синтезируемого сигнала, и дальнейшее улучшение качества сигнала следует выполнять для данного компонента. При модификации системы синтеза речи с исключением второй нейронной сети был убран этап редукции мел-спектров из блока цифровой обработки сигнала и модифицирована первая сеть. В результате качество синтезируемой речи немного ухудшилось,

следовательно, наилучшая архитектура системы синтеза речи состоит из двух последовательно работающих нейронных сетей с вокодером WaveGlow. Проведенный эксперимент по модификации архитектуры модели с заменой вокодера на нейросетевой показал, что по оценке MOS качество синтезируемой речи улучшилось, следовательно, имеется возможность повышения качества голоса в дальнейших исследованиях.

Итоговые оценки качества речи относятся к 1 классу качества согласно ГОСТ Р 50840-95 [8], что говорит о том, что цель исследования достигнута. Разработанная методика оценки потерь качества на каждом этапе синтеза речи позволила выявить по результатам оценки MOS этапы, оптимизации которых должно быть уделено особое внимание для улучшения качества голоса.

Литература

1. Wang Yuan (Gary). Deep Text-to-Speech System with Seq2Seq Model. // arXiv preprint arXiv:1903.07398v1. – 2019.
2. Sotelo Jose, Mehri Soroush, Kundan Kumar, Joao Felipe Santos, Kyle Kastner, Aaron Courville, Yoshua Bengio. Char2Wav: End-to-End Speech Synthesis. In ICLR workshop. – 2017.
3. Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Skerry-Ryan RJ, Rif A. Saurous, Yannis Agiomyrgiannakis, Yonghui Wu. Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions. // arXiv preprint arXiv:1712.05884. – 2017.
4. Arik S. O., Chrzanowski M., Coates A., Diamos G., Gibiansky A., Kang Y., Li X., Miller J., Raiman J., Sengupta S., and Shoneybi M. Deep voice: Real-time neural text-to-speech. In ICML. – 2017.
5. Tachibana Hideyuki, Uenoyama Katsuya. Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention. // arXiv preprint arXiv:1710.08969v1. – 2017.
6. Мирзаянова С.В. Исследование системы сквозного нейросетевого синтеза речи на основе системы Tacotron. Сборник трудов VII Конгресса молодых ученых. Том 2. – СПб.: Университет ИТМО, 2018. – С. 49 – 52.
7. Griffin D. and Lim J. Signal estimation from modified shorttime fourier transform / IEEE Trans. ASSP, vol. 32, no. 2. 1984. – Pp. 236 – 243.
8. ГОСТ Р 50840-95. Передача речи по трактам связи. Методы оценки качества, разборчивости и узнаваемости. – М.: Госстандарт России, 1995. – 234 с.

Паламарь И.Н., Гагарина А.И. Методика исследования нейросетевой технологии реализации E2E систем синтеза сигнала. В статье рассмотрена проблема повышения качества E2E системы, реализуемой на основе нейросетевой технологии синтеза сигнала. Рассмотрены модели архитектуры системы синтеза речи на основе модели свёрточной нейронной сети. Предложены методики включения этапа разработки датасета и оптимизации всех компонентов потока обработки данных. Проведен анализ результатов моделирования системы по методике исследования. Результаты экспериментов по модификации архитектуры и реализации компонентов по оценке MOS показали повышение качества синтеза речевого сигнала.

Ключевые слова: E2E система, системный анализ, нейросетевая технология, глубокое обучение, синтез сигнала, методика исследования.

Palamar Irina, Gagarina Alisa. Research technique of neural network technology for the implementation of E2E signal synthesis systems. The article considers the problem of improving the quality of an E2E system implemented on the basis of neural network signal synthesis technology. The models of the architecture of the speech synthesis system based on the convolutional neural network model are considered. The techniques of inclusion of the dataset development stage and optimization of all components of the data processing flow are proposed. The analysis of the results of modeling the system according to the research technique is carried out. The results of experiments on the modification of the architecture and the implementation of components for the evaluation of MOS showed an increase in the quality of speech signal synthesis.

Key words: E2E system, system analysis, neural network technology, deep learning, signal synthesis, research technique.

Компьютерное моделирование методом Монте-Карло

Б.Ю. Кныш^{*1} И.Ю. Анохина^{*2}

^{*1} бакалавр, Донецкий национальный технический университет, school42don@mail.ru,

^{*2} к.т.н, доцент, Донецкий национальный технический университет,
OrcID: 0000-0003-1073-6333, SPIN-код: 3555-6335
ingatula@mail.ru

Кныш Б.Ю., Анохина И.Ю. Компьютерное моделирование методом Монте-Карло. Рассмотрены вопросы разработки алгоритма компьютерного моделирования методом Монте-Карло, а также его программной реализации на языке программирования Python. В качестве объекта исследования использованы спутниковые фотографии пожароопасных территорий Земли.

Ключевые слова: метод Монте-Карло, пожарная активность, Python, цветовая модель RGB.

Введение

Многие интересные задачи и их модели часто имеют чрезвычайно сложную структуру и не могут быть описаны с помощью традиционных методов. В случае, когда аналитические методы неприменимы или не обеспечивают достаточную степень точности, прибегают к универсальному методу статистического моделирования, методу Монте-Карло.

Метод Монте-Карло как метод анализа данных используется в тех ситуациях, когда система действует, в том числе и под влиянием случайных величин. Используя метод, проводя повторные независимые испытания и сопоставляя их друг с другом, можно получить представление о статистических свойствах системы. Метод получил широкое распространение в таких областях, как инженерия, физика, финансы, наука о данных, машинное обучение и т. д.

Вместо описания процесса средствами аналитического аппарата, производится моделирование случайного явления. При многократном повторении процедуры получают новую, случайную реализацию исследуемого процесса. Множество полученных реализаций можно рассматривать как искусственно полученные статистические данные, которые впоследствии обрабатываются статистическими методами [1].

Характерными особенностями метода является с одной стороны простота алгоритма, а с другой – сильная зависимость точности расчетов от числа смоделированных испытаний и среднего квадратического отклонения исследуемой случайной величины.

Эффективность метода Монте-Карло можно оценить на примере вычисления площадей фигур произвольной формы.

Постановка задачи

Национальным управлением по аэронавтике и исследованию космического пространства NASA запущено более десятка спутников Земли, которые предоставляют информацию, позволяющую изучать океаны, сушу и атмосферу Земли. В интернете представлены карты с информацией о растительном покрове, льдах, поверхности почвы, степени ее засоленности и пр.

Карта с данными о пожарах существует практически в режиме реального времени. Спутники создают карту в среднем четыре раза за сутки, что позволяет оценить пожарную активность, спрогнозировать последствия для экосистем, создать архив пожароопасных территорий Земли. Информация о пожарах выкладывается на сайт Fire Map [2].

Поставлена задача разработки алгоритма и его программной реализации с целью определения площадей лесных пожаров на карте мира. В качестве метода решения выбран метод Монте-Карло.

Описание алгоритма

На рис. 1 показана карта пожаров за 18 октября 2022г. В правой части экрана представлена область настройки карты, позволяющая выбрать временной интервал в пределах от одних суток до месяца, в нижней

части – календарь, используя который можно получить архивные данные. Карты доступны для скачивания в формате jpg.



Рисунок 1 – Карта пожаров за 18 октября 2022 г.

На первом шаге алгоритма изображение представляли в виде трехмерного массива пикселей. Для работы использовали библиотеки PIL и NumPy.

```
import numpy as np # для манипуляций с массивами
import pandas as pd # для удобной работы с данными в табличном виде
from PIL import Image # для работы с изображениями
```

Изображение загружали с помощью функции `open()` подмодуля `Image`, затем данные преобразовывались в массив NumPy, который сохраняли в переменной `arr`.

```
img = Image.open('FIRMS_24hrs_2022-10-18.jpg')
arr = np.asarray(img, dtype='int16')
```

Размер изображения 1920x825 пикселей.

```
img.size
Out:
(1920, 825)
```

```
arr[0]
```

В результате получен массив размером 1920x825, причем каждый элемент массива является массивом из трех элементов (значений RGB¹ модели пикселя).

```
Out:
array([[ 20,  63, 106],
       [ 20,  63, 106],
       [ 20,  63, 106],
       ...,
       [ 17,  65, 103],
       [ 17,  65, 105],
       [ 17,  65, 105]], dtype=int16)
```

Полученные значения пикселей считывались в список.

¹ RGB — цветовая модель, названная так по трём заглавным буквам названий цветов, лежащих в ее основе: **Red** (красный), **Green** (зеленый), **Blue** (синий). Цвет выражается как триплет RGB (r, g, b), каждый компонент которого может варьироваться от нуля до 255.

```

pixels = []
for i in range(img.size[1]):
    for j in range(img.size[0]):
        pixels.append(arr[i][j])

```

Преобразуем список в DataFrame для дальнейшей работы с данными. Длина полученного датафрейма df должна получиться 1920*825, причем каждая строка данных представляет собой значение цветов RGB модели.

```

df = pd.DataFrame(data = pixels, columns = ['red', 'green', 'blue'])
df

```

Полученный набор данных размером 1584000*3 показан на рис. 2.

Out[23]:

	red	green	blue
0	20	63	106
1	20	63	106
2	20	63	106
3	20	63	106
4	20	63	106
...	...	...	...
1583995	218	221	230
1583996	238	242	253
1583997	231	235	246
1583998	221	225	236
1583999	206	210	221

1584000 rows × 3 columns

Рисунок 2 – Полученный набор пикселей изображения

В зависимости от плотности пожаров оттенки красного цвета разнятся. Например, одна из наиболее ярких областей пожара имеет следующие значения RGB модели: R = 238, G = 0, B = 4. Области пожаров с меньшей плотностью соответствуют значения RGB: R = 225, G = 102, B = 77, см. рис. 3.

Чтобы определить площади, занимаемые пожарами, необходимо отметить именно их, не затрагивая соседних областей.

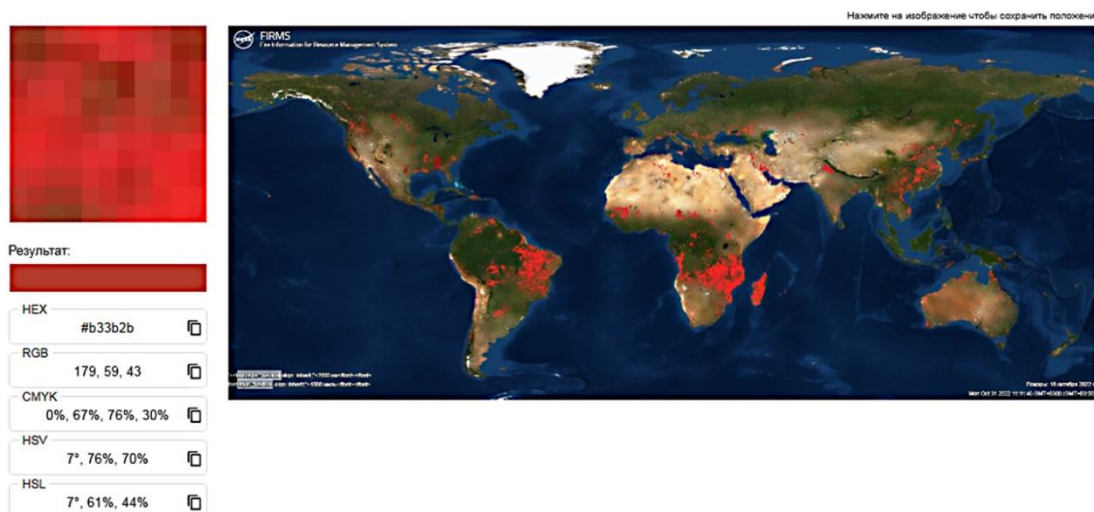


Рисунок 3 – Интенсивность пожаров и значения RGB модели

Анализ имеющихся данных позволил сделать вывод, что в оттенках красного цвета именно значение составляющей Red в RGB модели должно быть не менее, чем в 1.8 раза больше, чем значения других составляющих.

Булева маска, позволяющая проверять это условие, имеет вид:

```
(df['red'] > df['green'] * 1.8) & (df['red'] > df['blue'] * 1.8)
```

Расчеты показали, что лишь 2.4% площади изображения удовлетворяют поставленным условиям.

```
np.mean((df['red'] > df['green'] * 1.8) & (df['red'] > df['blue'] * 1.8))
Out:
0.0243794191919192
```

Чтобы проверить гипотезу о правильном определении оттенков красного цвета был введен коэффициент scaler. Его значения подбирались эмпирическим путем.

```
[np.mean((df['red'] > df['green'] * scaler) & (df['red'] > df['blue'] * scaler)) for scaler in np.arange(1, 2.5, 0.1)]
```

При значении scaler= 1 модель становится слишком чувствительной и около 17% карты отмечаются как пожары. С увеличением scaler чувствительность уменьшается, при этом скорость уменьшения value падает после значения scaler =1.3

```
df_scaler = pd.Series([np.mean((df['red'] > df['green'] * scaler) & (df['red'] > df['blue'] * scaler))
                        for scaler in np.arange(1, 2.5, 0.1)]).to_frame().rename(columns = { 0 : 'value'})
df_scaler['scaler'] = np.arange(1, 2.5, 0.1)
```

Для определения величины scaler использовали методику, схожую с «методом локтя» при определении количества кластеров в кластеризации. При кластеризации одной из основных проблем является определение числа кластеров. Методы k – средних, каменистой осыпи не дают однозначного ответа. Часто для определения количества кластеров используют «метод локтя», который подразумевает циклическое выполнение алгоритма с изменяемым количеством кластеров и оценкой некоторого показателя, который, как правило, определяется как отношение внутрикластерного расстояния к межкластерному [3].

На рис.4. показана таблица данных, определяющих долю «красных» пикселей на текущем и следующем шагах алгоритма.

```
df_scaler['value_lag'] = df_scaler['value'].shift(-1)
```

	value	scaler	value_lag	growth_rate
0	0.166830	1.0	0.134730	0.238247
1	0.134730	1.1	0.092801	0.451819
2	0.092801	1.2	0.055737	0.664971
3	0.055737	1.3	0.041508	0.342824
4	0.041508	1.4	0.034685	0.196702
5	0.034685	1.5	0.029622	0.170901
6	0.029622	1.6	0.026502	0.117749
7	0.026502	1.7	0.024379	0.087060
8	0.024379	1.8	0.022762	0.071058
9	0.022762	1.9	0.021287	0.069278
10	0.021287	2.0	0.020213	0.053159
11	0.020213	2.1	0.019186	0.053503
12	0.019186	2.2	0.018311	0.047821
13	0.018311	2.3	0.017490	0.046925
14	0.017490	2.4	NaN	NaN

Рисунок 4 –Данные, содержащие долю «красных» пикселей при текущем значении scaler и отношение к доле красных пикселей при следующем значении scaler

На рис.5. показан график прироста значений value при изменении значений scaler.

```
import matplotlib.pyplot as plt
plt.style.use('dark_background') # По оси x отложим значения коэффициента
df_scaler.set_index('scaler')['growth_rate'].plot() # по оси y прирост значений value
```

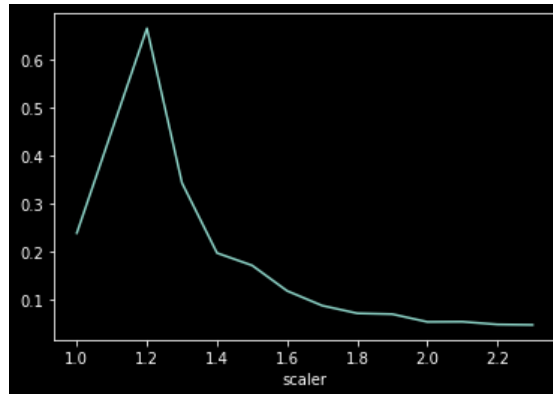


Рисунок 5 – График прироста значение value при изменении scaler

Плавное течение графика после значения scaler равного 1.8 означает, что значимых изменений переменной `growth_rate` при увеличении коэффициента не наблюдается, следовательно, чувствительность метки оттенка цвета существенно не изменяется, поэтому значение scaler приняли равным 1.8. Отметим, что метод каменистой осыпи, используемый при статистических расчетах, работает по аналогичному принципу [4].

Полученное значение scaler использовано при расчете значений переменной, которая определяет, следует ли относить анализируемый пиксель к красным или нет.

```
df['fire_flag'] = (df['red'] > df['green']* 1.8) & (df['red'] > df['blue']* 1.8)
```

Площадь пожаров в пикселях определяли с помощью симуляций Монте-Карло. В качестве переменной для симуляций принята `fire_flag`, имеющая биномиальное распределение, так как представлена всего двумя значениями True/False.

Функция `value_counts` возвращает количество каждого из уникальных значений. Для оценки площади пожара определяли отношение количества пикселей, отмеченных как красные к общему количеству пикселей.

Общее число пикселей составило больше 1.5 млн. Нами рассматривались выборки размером от 500 наблюдений до 10млн. и оценивалась влияние их размерности на точность результатов. Результаты симуляций заносили в DataFrame.

```
from tqdm.notebook import tqdm
sample_size = [500, 1000, 10000, 50000, 500000, 1000000, 5000000, 10000000]
result = []
for n in tqdm(sample_size):
    simulation = df['fire_flag'].sample(n, replace = True).value_counts()
    result.append(simulation[True]/simulation.sum())
df_simulation = pd.Series(sample_size).to_frame().rename(columns = {0: 'sample_size'})
df_simulation['square'] = pd.Series(result).values
```

На рис.6 показано изменение оценочной площади в зависимости от размера выборки.

	sample_size	square
0	500	0.034000
1	1000	0.018000
2	10000	0.027200
3	50000	0.023220
4	500000	0.024500
5	1000000	0.024125
6	5000000	0.024467
7	10000000	0.024370

Рисунок 6 – Оценочная площадь пожара в зависимости от размера выборки

При размере выборки размером от 50 тысяч, получены достаточно точные результаты. Выборка размером в 10млн. позволила определить площадь пожаров с точностью до 5-ого знака после запятой.

Выполненные расчеты наглядно показали действие закона больших чисел: с увеличением размера выборки точность расчета площади существенно увеличивалась.

Для определения площади пожара в километрах умножали количество красных пикселей на общую площадь суши и на масштаб исходного изображения.

Выводы

В статье рассмотрены вопросы разработки алгоритма компьютерного моделирования методом Монте-Карло, выполнена реализация на языке программирования Python. В качестве проекта для реализации алгоритма была взята проблема оценки площади пожаров на Земле по спутниковым изображениям.

Получены следующие результаты: разработан алгоритм поиска площади пожаров в пикселях с помощью симуляций Монте-Карло, оценена точность получаемых результатов в зависимости от изменяемых размеров выборок.

Предложенный алгоритм может использоваться при расчетах любых площадей изображений.

Литература

1. Вентцель Е.С. Исследование операций.– М. : Советское радио, 1972. – 552 с.
2. Fire Map. Fire Information for resource management system [Электронный ресурс] – Режим доступа: <https://firms.modaps.eosdis.nasa.gov/map> - Загл. с экрана.
3. Хабр. Сообщество IT – специалистов. Кластеризуем лучше, чем метод локтя. [Электронный ресурс] – Режим доступа: <https://habr.com/ru/company/jetinfosystems/blog/467745/> - Загл. с экрана.
4. В. П. Боровиков, Г. И. Ивченко Учебник по математической статистике с упражнениями в системе STATISTICA, - Москва: StatSoft, 2003г., 320с.

Кныш Б.Ю., Анохина И.Ю. Компьютерное моделирование методом Монте-Карло. В статье рассматриваются вопросы разработки алгоритма компьютерного моделирования методом Монте-Карло, а также его программной реализации на языке программирования Python. В качестве объекта исследования использованы спутниковые фотографии пожароопасных территорий Земли.

Ключевые слова: метод Монте-Карло, пожарная активность, Python, цветовая модель RGB.

Knysh B.Yu., Anokhina I.Yu. Computer modeling by the Monte Carlo method. The article discusses the development of a Monte Carlo computer simulation algorithm, as well as its software implementation in the Python programming language. Satellite photographs of the Earth's fire-hazardous territories were used as the object of the study.

Key words: Monte Carlo method, fire activity, Python, RGB color model.

Применение моделирования процесса индивидуального подбора деталей при сборке высокоточных изделий

Н.А. Задорина^{*1}, В.В. Непомилуев^{*2}

^{*1} старший преподаватель, Рыбинский государственный авиационный технический университет имени П. А. Соловьева,
zadorina@rsatu.ru,

^{*2} д.т.н., профессор, Рыбинский государственный авиационный технический университет имени П. А. Соловьева,
valerij.vas.nepomiluev@gmail.com

Задорина Н.А., Непомилуев В.В. Применение моделирования процесса индивидуального подбора деталей при сборке высокоточных изделий. В статье приведены результаты исследования метода индивидуального подбора деталей при сборке машин, содержащих многозвенные размерные цепи. Представлена модель процесса индивидуального подбора деталей на основе генетического алгоритма поиска оптимального решения и проведено ее исследование. Показано, что использование предлагаемого алгоритма подбора деталей позволяет существенно повысить точность таких машин, что дает возможность обеспечивать требуемое качество их сборки с меньшими затратами.

Ключевые слова: сборка, качество, размер, звено, размерная цепь, индивидуальный подбор деталей, компьютерное моделирование.

Введение

Важную роль в обеспечении эксплуатационных показателей качества (надёжности) изделий играет геометрическая точность, которая зависит от точности изготовления деталей и применяемых технологических методов сборки. Самыми распространенными в промышленности методами обеспечения точности сборки являются методы взаимозаменяемости. Однако в ряде случаев, например, для сборки высокоточных изделий, взаимозаменяемость не позволяет получить нужный результат. Для таких задач может использоваться метод индивидуального подбора деталей (ИПД), идея которого заключается в рассмотрении возможных вариантов сочетаний деталей и выборе лучшего варианта с точки зрения обеспечения требуемых показателей качества. В настоящее время метод подбора реализуется в основном вручную, что неизбежно ограничивает количество рассматриваемых вариантов и, следовательно, достигаемую точность.

Однако развитие и внедрение на промышленных предприятиях информационных технологий, включающих способы создания, обработки, передачи и хранения специфической информации о промышленной продукции, методах её разработки, испытаний и производства, делает возможным дополнение или замену традиционных этапов различных видов работ эффективными процедурами синтеза информационных моделей, как продукции, так и процессов производства. Разработка информационных моделей объектов и моделей процессов для определения оптимальных вариантов комплектования и взаимного положения деталей является перспективным направлением совершенствования сборочных процессов высокоточных изделий. Анализ существующих методов подбора деталей при сборке и разработка новых алгоритмов и методов являются важной составляющей более глобальных задач, связанных с цифровизацией и информатизацией производства.

Формализация задачи индивидуального подбора деталей

Решение задачи индивидуального подбора деталей основано на построении модели процесса и проведении моделирования, в ходе которого происходит поиск оптимального решения по обеспечению показателей точности.

Эффективность моделирования процесса подбора деталей оценивается точностью результатов моделирования, а также затратами машинных ресурсов (времени и памяти).

В [1] приведены алгоритмы подбора, основанные на упорядочивании линейных размеров деталей: алгоритм сложной сортировки и алгоритм сложной направленной сортировки.

Алгоритм сложной сортировки размеров деталей предусматривает сортировку по возрастанию элементов массивов с четными номерами, по убыванию массивов с нечетными номерами. Затем выполняется построчное суммирование массивов данных.

Алгоритм сложной направленной сортировки предполагает упорядочивание по возрастанию массива размеров детали, на которую назначен наибольший допуск, размеры остальных деталей – по возрастанию. Суммирование выполняется построчно.

Для исследования алгоритмов сложной сортировки, сложной направленной сортировки и полного перебора было проведено компьютерное моделирование процесса сборки, для чего сгенерированы массивы объемом в 100 единиц, имитировавшие размеры деталей. При проведении экспериментов варьировались допуски деталей. Было отмечено, что метод полного перебора неприменим для задач большой размерности из-за эффекта комбинаторного взрыва. Методы на основе сортировки дают хороший результат только для двухзвенных размерных цепей, для многозвенных цепей результат лучше, чем при случайном выборе, но далек от оптимального.

В связи с этим возникает задача разработки эффективного алгоритма. Решение этой задачи требует применения методов, используемых в теории сложности вычислений. Для того, чтобы успешно применять аппарат теории сложности вычислений, необходимо дать формальное определение задачи индивидуального подбора деталей как задачи оптимизации.

Задача формирования сборочных комплектов методом индивидуального подбора деталей подразумевает комплектование деталей по результатам измерений истинных значений параметров компонентов таким образом, чтобы были получены необходимые заданные величины выходных параметров изделия.

Предлагается следующее формальное описание задачи: Пусть изделие состоит из m деталей различного вида: $A_1, A_2, \dots, A_m$, тогда $U_1, U_2, \dots, U_m$ – конечные мультимножества истинных размеров деталей $A_1, A_2, \dots, A_m$ мощности n [2], $U_j = \{x_{1j}, x_{2j}, \dots, x_{nj}\}$, $j = 1, \dots, m$. Необходимо найти такую комбинацию векторов $\bar{x}$, чтобы некоторый заданный функционал качества от $\bar{x}$ достигал экстремума.

В такой постановке задачу подбора деталей при сборке размерной цепи можно формализовать как задачу комбинаторной оптимизации. В приложении к задаче подбора деталей необходимо найти набор сборочных комплектов, чтобы некоторый функционал показателя качества сборки был оптимален. Например, в качестве такого функционала может выступать величина разброса замыкающего звена. В этом случае необходимо решить задачу минимизации.

Множество всех вариантов сборочных комплектов D является конечным множеством, так как оно равномощно с отрезком натурального ряда или, иными словами, состоит из конечного числа элементов. Множество, равномощное множеству натуральных чисел N , называется счетным множеством. Из определений следует, что конечное множество является конечным подмножеством счетного множества. Поэтому задача подбора деталей при сборке размерной цепи является задачей дискретной оптимизации.

Согласно [3] задачей комбинаторной оптимизации $P = (X, f)$ называется задача оптимизации, в которой задано конечное множество объектов X и целевая функция $f : X \rightarrow R^+$, которая каждому объекту $x \in X$ ставит в соответствие некоторое положительное число – значение стоимости, необходимо найти такое $x^* \in X$, что

$$x^* = \arg \min_{x \in D \subseteq X} f(x), \quad (1)$$

где X – пространство решений задачи, элементами которого являются комбинаторные объекты;

$D \subseteq X$ – подпространство допустимых решений;

$f : X \rightarrow R^+$ – целевая функция задачи.

Применительно к задаче индивидуального подбора деталей множество X – это множество m -мерных векторов. Элементы множества D , на котором осуществляется поиск экстремума целевой функции, можно рассматривать как векторы действительных размеров деталей, входящих в сборочный комплект.

В постановке (1) задача заключается в поиске минимума целевой функции:

$$f(\bar{x}) \rightarrow \min, \bar{x} \in D, \quad (2)$$

где D – конечное множество допустимых решений, мощность которого n^m , а объектами являются $\bar{x} = \{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n\}$;

$\bar{x}_i = \{x_{i1}, x_{i2}, \dots, x_{im}\}$, – сборочный комплект для i -го изделия;

$i = 1, \dots, n$;

$x_{i1}, x_{i2}, \dots, x_{im}$ – действительные размеры деталей $A_1, A_2, \dots, A_m$.

Целевая функция $f(\bar{x})$ может представлять собой значение разброса Δx_i на множестве размеров замыкающего звена x_i всех полученных изделий:

$$f(\bar{x}) = \max_i \left(\sum_{j=1}^m x_{ij} \right) - \min_i \left(\sum_{j=1}^m x_{ij} \right) \rightarrow \min, \bar{x} \in D, \quad (3)$$

где $\max_i \left(\sum_{j=1}^m x_{ij} \right)$ – максимальный размер замыкающего звена размерной цепи, $\min_i \left(\sum_{j=1}^m x_{ij} \right)$ – минимальный размер замыкающего звена размерной цепи для полученного варианта решения.

Можно предложить и другие варианты определения целевой функции $f(\bar{x})$, которые будут зависеть от схемы размерной цепи или цели оптимизации.

Формализация задачи подбора деталей при сборке как задачи дискретной комбинаторной оптимизации позволяет использовать классические подходы к решению подобных задач.

Все известные алгоритмы комбинаторной оптимизации принято делить на два класса: точные и приближенные.

Точные методы обладают свойством гарантировано находить оптимальное решение. Точное решение задачи подбора деталей можно получить путем полного перебора всех возможных вариантов решения, то есть всех возможных сочетаний деталей каждого вида в сборочном пакете. Подходы, относящиеся ко второй категории, включают прием, который можно назвать «снижение требований». Он заключается в отказе от поиска оптимального решения и в нахождении вместо этого хорошего решения за приемлемое время.

Для того, чтобы сделать выбор между точными и приближенными методами решения данной задачи необходимо рассмотреть сложность предлагаемых алгоритмов, которая является одним из критериев оценки алгоритма и определяет количество ресурсов, необходимых алгоритму для решения данного экземпляра задачи. Сложность также определяет порядок роста необходимого для решения задачи времени и емкости памяти при увеличении входных данных. Наиболее важным ресурсом является время. С каждой конкретной задачей обычно связано некоторое число – размер, которое выражает меру количества входных данных. Для задачи подбора деталей размер задачи определяется количеством видов деталей, входящих в состав сборочного комплекта, и объемами партий. Временной сложностью алгоритма называется функция, описывающая зависимость максимального числа основных операций, выполняемых алгоритмом на абстрактной вычислительной машине (машине Тьюринга) от размера задачи $O(f(n))$. Поведение этой функции в пределе при увеличении размера задачи называется асимптотической временной сложностью [4]. Любую вычислительную задачу можно считать решенной в том и только в том случае, если существует алгоритм ее решения с приемлемой временной сложности. В теории сложности выделяют два класса алгоритмов: полиномиальные (polynomial time) и экспоненциальные (exponential time или EXPTIME). Полиномиальным (или алгоритмом полиномиальной временной сложности) называется алгоритм, у которого временная сложность есть $O(p(n))$, где $p(n)$ – полином от n . Задачи с экспоненциальной сложностью времени работы образуют класс EXPTIME. В теории сложности вычислений, класс сложности EXPTIME (иногда называемый просто EXP) – это множество задач, решаемых с помощью детерминированной машины Тьюринга за время $O(2^{p(n)})$, где $p(n)$ – это полиномиальная функция от n [4]. Можно сказать, что экспоненциальный алгоритм имеет сложность $O(2^n)$ или более высокую, например $O(n^n)$.

По мнению IT-специалистов алгоритм является практически полезным для вычислительной задачи только в том случае, если его сложность растет полиномиально относительно размера входа [4]. Таким образом, алгоритм считается эффективным, если он полиномиален. По классу временной сложности алгоритма задача подбора деталей методом полного перебора относится к типу EXPTIME [5]. Для малых значений n полиномиальное время может даже превосходить экспоненциальное. Таким образом, с увеличением n время нахождения оптимального решения растёт экспоненциально и быстро наступает такой момент, когда при выделенных вычислительных ресурсах поиск решения становится экономически нецелесообразным.

Таким образом, практическое применение метода полного перебора для решения задачи индивидуального подбора деталей ограничено из-за существенного недостатка, которым является «комбинаторный взрыв» – эффект резкого роста временной сложности алгоритма при увеличении размерности задачи. Поэтому применимость метода полного перебора зависит от размерности задачи. При малой размерности применение оправдано, однако при $m > 3$ и $n \geq 5$, что соответствует реальным данным, полный перебор нереален. Возможны два пути решения проблемы. Первый путь – попытаться разработать алгоритм точного решения задачи с полиномиальной временной сложностью; второй – отказаться от попыток найти точное решение и предложить эффективный приближенный алгоритм.

Анализ сложности задачи индивидуального подбора деталей

Для любой алгоритмически разрешимой задачи существует множество различных алгоритмов, позволяющих получить ее решение. С практической точки зрения важно не просто разработать алгоритм, а найти среди множества алгоритмов наименее трудоемкий. Однако в теории сложности нет строгой математической постановки задачи минимизации сложности алгоритма, однако дается понятие не только сложности алгоритмов, но и сложности задач.

Сложность задачи – это минимальная из сложностей всех возможных алгоритмов решения этой задачи. Пусть $\{F_1, F_2, \dots, F_k\}$ – множество алгоритмов решения задачи Z , каждый из которых характеризуется асимптотически точной верхней оценкой функции трудоемкости $g_1(n), g_2(n), \dots, g_k(n)$, то сложность $g(n)$ задачи Z определяется, как $g(n) = \min(g_1(n), g_2(n), \dots, g_k(n))$.

При разработке алгоритмов можно наблюдать, что для некоторых задач можно построить алгоритм полиномиальной сложности. Такие задачи называют полиномиальными. Полиномиально разрешимые задачи можно успешно решать на компьютере даже в тех случаях, когда они имеют большую размерность. Для других задач не удастся найти полиномиальный алгоритм, поэтому их называют трудноразрешимыми.

Почти все комбинаторные задачи являются NP -трудными, то есть алгоритмы их решения имеют экспоненциальную сложность **Ошибка! Источник ссылки не найден.**[6]. В классе NP выделяют подкласс NP -полных задач. Основы теории NP -полных задач были заложены С. Куком [7]. Класс NP -полных задач включает в себя многие «классические» задачи комбинаторики (задача о коммивояжере, задача об упаковке в контейнеры, задача целочисленного линейного программирования и т.д.). В теории сложности доказано, что все задачи этого класса эквивалентны, то есть могут сводиться одна к другой [8]. Однако, несмотря на то, что многие из этих задач изучались специалистами десятилетиями, полиномиальные алгоритмы решения этих задач пока не найдены.

Представляется перспективным попытаться доказать, что для задачи индивидуального подбора не существует эффективного точного алгоритма решения. И если это так, обратиться к приближенным методам решения.

Теория NP -полных задач дает методы доказательства того, что конкретная задача столь же трудна, как и ряд других задач, признанных очень трудными. Наиболее простым способом доказательства является метод сужения [8]. Этот прием доказательства самый простой. Доказательство методом сужения некоторой задачи Π заключается в установлении того, что задача Π включает в качестве частного случая известную NP -полную задачу Π' .

В качестве задачи Π' рассмотрим задачу комбинаторики – «Задачу о куче камней» [9], суть которой в классической постановке заключается в разбиении множества камней различного веса на кучи максимально одинакового веса. Задача о куче камней относится к классу NP -полных задач, для которых точное решение может быть получено только методом полного перебора.

Дадим общую постановку задачи Π' .

Имеется множество камней N , каждому из которых j сопоставлено положительное число w_j . Требуется разделить это множество на заданное множество куч m таким образом, чтобы вес самой тяжелой кучи был минимален. Иными словами, требуется найти такое разбиение $\mathfrak{R} : N = \bigcup_{i \in 1:m} N_i$, которое минимизирует

$$\max \{ |N_i| \times w[N_i] \mid i \in 1:m \}.$$

Сопоставим формулировку задачи о куче камней с задачей индивидуального подбора деталей. Рассмотрим частный случай задачи Π , когда изделие состоит из произвольного количества деталей одного вида. Тогда формулировка задачи будет выглядеть так:

Имеется N деталей, каждой из которых j сопоставлено положительное число u_j , тогда U – множество истинных размеров деталей мощности n . Необходимо из этих деталей собрать m изделий так, чтобы размер самого большого изделия был бы минимален.

Очевидно, что задача Π включает в качестве частного случая известную NP -полную задачу Π' – задачу о куче камней. Таким образом, задача индивидуального подбора деталей является NP -полной задачей.

Решение задачи индивидуального подбора деталей может быть найдено путем полного перебора вариантов (в общем случае с экспоненциальной трудоемкостью), а также с помощью других методов, применяемых для класса NP -полных задач: динамического программирования и методов с возвратом, позволяющим сократить полный перебор, а также с помощью приближенных методов. Однако в большинстве случаев тот факт, что задача принадлежит классу NP , является достаточным основанием для отказа от поиска эффективного и точного алгоритма в пользу эффективного приближенного решения.

Алгоритмы решения задачи подбора деталей

Практическая необходимость в решении NP -полных задач заставляет искать эффективные алгоритмы их решения. Ряд комбинаторных задач теоретически могут быть решены методом полного перебора метод полного

перебора. Однако переборные алгоритмы позволяют найти решение только для случаев малой размерности. Это исключает возможность нахождения оптимального решения методом полного перебора даже для задач среднего размера за приемлемое время.

Известным и широко применяемым методом сокращения перебора является метод ветвей и границ, который представляет собой развитие метода полного перебора. В основе метода ветвей и границ лежит идея последовательного разбиения (ветвления) множества допустимых решений на подмножества. В свою очередь каждое из подмножеств снова разбивается на подмножества до тех пор, пока каждое подмножество не будет представлять собой точку в многомерном пространстве. Полученные подмножества образуют дерево поиска или дерево ветвей и границ, узлами которого являются подмножества допустимых значений. В силу конечности наборов значений переменных дерево подмножеств (схема ветвлений) конечно. Каждое подмножество подвергается анализу – содержит ли оно оптимальное решение или нет. Применение метода ветвей и границ для задачи индивидуального подбора не имеет смысла в силу, во-первых, сложной структуры дерева поиска (допустимое решение представляет собой набор векторов), во-вторых, неочевидной процедуры ветвления, в-третьих, несоблюдения необходимых условий оптимальности решения, в-четвертых, значительных временных затрат для задач большой размерности из-за необходимости решать задачу нахождения оценок для каждого узла дерева поиска.

Еще одной стратегией сокращения перебора является построение алгоритмов с самообучением, т. е. алгоритмов с запоминанием некоторых найденных ранее решений и использованием их в дальнейшем. Эта стратегия положена в основу метода динамического программирования. Основная идея заключается в разбиении сложных задач на более простые подзадачи и последующем решении этих подзадач, используя принцип оптимальности Беллмана [10]. Однако для задачи индивидуального подбора метод динамического программирования применять нецелесообразно, так как при большой размерности метод требует больших вычислительных ресурсов. Кроме того, задача индивидуального подбора не обладает необходимыми свойствами для эффективного применения метода.

Таким образом, для решения задачи индивидуального подбора метод ветвей и границ и динамическое программирование не позволяют найти решение задачи за приемлемое время. Метод полного перебора позволяет гарантированно получить точное решение задачи, однако применение метода оправдано только при малой размерности из-за резкого роста временной сложности алгоритма при увеличении размерности задачи. Поэтому необходимо применять подход, связанный с поиском приближенных решений, которые строят допустимые решения за полиномиальное время.

Современные приближенные алгоритмы комбинаторной оптимизации позволяют получать решения «высокого качества» за приемлемое (с практической точки зрения) время. В настоящее время для решения задач оптимизации часто применяют эвристические методы. Они строятся на основе правдоподобных идей (эвристик) и характеризуются неосознанным (интуитивным) способом действий для достижения осознанных целей. Алгоритмы, построенные на основе таких предположений, называют эвристическими. Эвристические алгоритмы не гарантируют точное решение задачи. Однако для большинства труднорешаемых задач эвристики представляют собой единственный способ получить «хорошее» решение за приемлемое время. Примером эвристических алгоритмов являются жадные алгоритмы.

Жадные алгоритмы (Greedy algorithm) – однопроходные итерационные алгоритмы, на каждом шаге делают выбор, который является самым лучшим в данный момент, т.е. производится локально оптимальный выбор в предположении, что он приведет к оптимальному решению глобальной задачи. Однако для многих задач жадный алгоритм не позволяет найти оптимального решения.

Предложенные в [1], [11] и [12] алгоритмы на основе сортировок представляют собой жадные алгоритмы, так как на каждом шаге получают локально оптимальное решение для задачи меньшей размерности, при разработке алгоритма используется ошибочное предположение об индуктивности функции показателя качества сборки.

Временная сложность данных алгоритмов зависит от применяемого метода сортировки массивов. Для простых методов сортировки (простыми обменами, простыми вставками и т.д.) имеет место квадратичная сложность. Поэтому можно сделать вывод, алгоритмы сложной и сложной направленной сортировки по временной сложности относятся к классу полиномиальных. Применение алгоритмов подбора, основанных на упорядочивании линейных размеров деталей, имеет ряд преимуществ: высокая скорость, возможность применения для решения задач большой размерности и простота реализации. Однако задача подбора деталей не обладает необходимыми условиями, поэтому применение алгоритмов не позволяет получить оптимальное решение.

Проведенные автором эксперименты показали, что применение алгоритмов, основанных на жадных стратегиях, не позволяет получить оптимального решения для задачи индивидуального подбора при $m > 3$. Общим недостатком является недостаточно хорошее качество решения. Это означает, что выбор оптимально локального решения на каждом шаге не гарантирует оптимального решения задачи.

Популярным в последнее время становятся метаэвристические методы, которые представляют собой стратегии поиска приближенного решения. Метаэвристики используют эвристики в рамках алгоритмических схем более высокого уровня [13]. Для всех метаэвристических методов характерно намеренное введение элемента

случайности в алгоритм поиска. Наиболее перспективными для решения задачи индивидуального подбора представляется группа эволюционных методов.

Достоинствами методов являются высокое быстродействие, сравнительно простая внутренняя реализация, малочувствительность к росту размерности.

Эволюционные методы относятся к популярному в настоящее время научному направлению, которое называют «природные вычисления», объединяющему математические методы, в которых заложены принципы природных механизмов поиска наилучших решений [14, 15].

Основная идея эволюционных алгоритмов состоит в использовании базовых принципов теории биологической эволюции – отбора, мутации и воспроизводства особей. Подклассом класса эволюционных методов являются генетические алгоритмы [13]. Генетические алгоритмы образуют группу алгоритмов поисковой оптимизации, основанных на математическом моделировании биологических механизмов и процессов в живой природе с помощью принципов популяционной генетики, которые позволяют находить оптимальные или близкие к ним (субоптимальные) решения. Работа алгоритма представляет собой последовательную смену поколений, состоящих из фиксированного числа особей – точек пространства решений. При формировании следующего поколения часть потомков полностью идентична родителям, а часть изменяется некоторым случайным образом в результате мутации и кроссинговера (скрещивания) [16, 17]. Интерпретируя задачу подбора, как задачу генетического отбора, получаем, что каждая особь – это вариант сборочного комплекта, ген – деталь. Начальная популяция представляет собой допустимое решение задачи, которое будет улучшаться на каждом шаге алгоритма. Инициализация начальной популяции происходит случайным образом. В ходе выполнения генетических операторов объем популяции остается неизменным. В результате выполнения алгоритма получается одно решение – популяция n особей (n сборочных комплектов). Отбор особей для скрещивания может проводиться любым способом. В простейшем варианте алгоритма используем панмиксию (случайный выбор). Оператор скрещивания двух хромосом порождает двух потомков. Для скрещивания может использоваться любой вид кроссинговера, будем использовать одноточечный кроссинговер. Мутации в силу ограничений предметной области не выполняются. Для создания нового поколения применяется принцип элитарности, то есть после получения пары новых особей на основе конкурентного подхода с локальным состязанием принимается решение, кто попадает в новую популяцию. Критерием завершения порождения новых популяций могут быть различные события, например, выход на «плато» (значение целевой функции не изменяется или не происходит изменений популяции), исчерпание числа поколений, отпущенных на эволюцию, исчерпание времени, отпущенного на эволюцию и т.д.

Анализ моделей процесса подбора деталей

В ходе проведения компьютерного эксперимента имитация процесса сборки с использованием индивидуального подбора деталей производилась по нескольким алгоритмам:

1) алгоритм со случайным выбором основан на простом суммировании массивов размеров и соответствует методу полной взаимозаменяемости;

2) алгоритм с сортировкой основан на простом упорядочении массивов размеров;

3) алгоритм частичного перебора.

В процессе проведения компьютерного эксперимента генерировались соответствующие массивы данных, имитирующие составляющие звенья рассматриваемой размерной цепи, распределения размеров соответствовали нормальному закону распределения с параметрами распределения размеров деталей – среднее значение 1, стандартное отклонение $\sigma = 0,3$, количество значений в выборке варьировалось от 5 до 50. Результаты компьютерного моделирования приведены в таблице 1.

Результаты экспериментов демонстрируют, что при достаточном объеме партий по мере увеличения количества составляющих звеньев требования к точности их при использовании индивидуального подбора деталей могут быть даже снижены. Это объясняется тем, что при подборе погрешности составляющих звеньев компенсируют друг друга, а при увеличении количества звеньев в размерной цепи возможности взаимной компенсации увеличиваются.

Одним из факторов, влияющим на эффективность алгоритма, является оператор отбора пар для обмена деталями. В [18, 19] предложен случайный равновероятный отбор. Однако могут быть предложены и другие варианты отбора пар изделий, например, инбридинг или аутбридинг. Инбридинг предполагает выбор первого изделия пары случайным образом, а второе изделие – ближайшее к первому (в смысле минимального евклидова расстояния между вещественными векторами). При аутбридинге также используют понятие схожести, но пары формируют из максимально далеких изделий. Для определения схожести изделий будем использовать размер замыкающего звена цепи: наиболее удаленными будем считать изделия, имеющие наибольший и наименьший размер.

Таблица 1 – Величины поля рассеяния размеров замыкающего звена по результатам компьютерного моделирования

Номер эксперимента	Количество звеньев	Объем выборки	Поле рассеяния замыкающего звена		
			Алгоритм со случайным выбором	Алгоритм с сортировкой	Алгоритм с частичным перебором
1	2	50	1,70	0,46	0,46
2	3		2,53	0,89	0,07
3	4		2,95	0,39	0,03
4	5		3,22	1,16	0,03
5	2	25	1,67	0,28	0,28
6	3		1,84	0,61	0,09
7	4		1,71	0,22	0,06
8	5		2,06	1,17	0,07
9	2	10	1,27	0,60	0,60
10	3		1,47	0,37	0,15
11	4		1,26	0,26	0,17
12	5		1,41	1,05	0,15
13	2	5	1,27	0,71	0,71
14	3		0,96	0,40	0,38
15	4		1,26	0,21	0,20
16	5		1,09	1,08	0,45

На рис. 1 показаны результаты эксперимента для инбридинга и аутбридинга, количество деталей в партии – 5, варьируется количество звеньев.

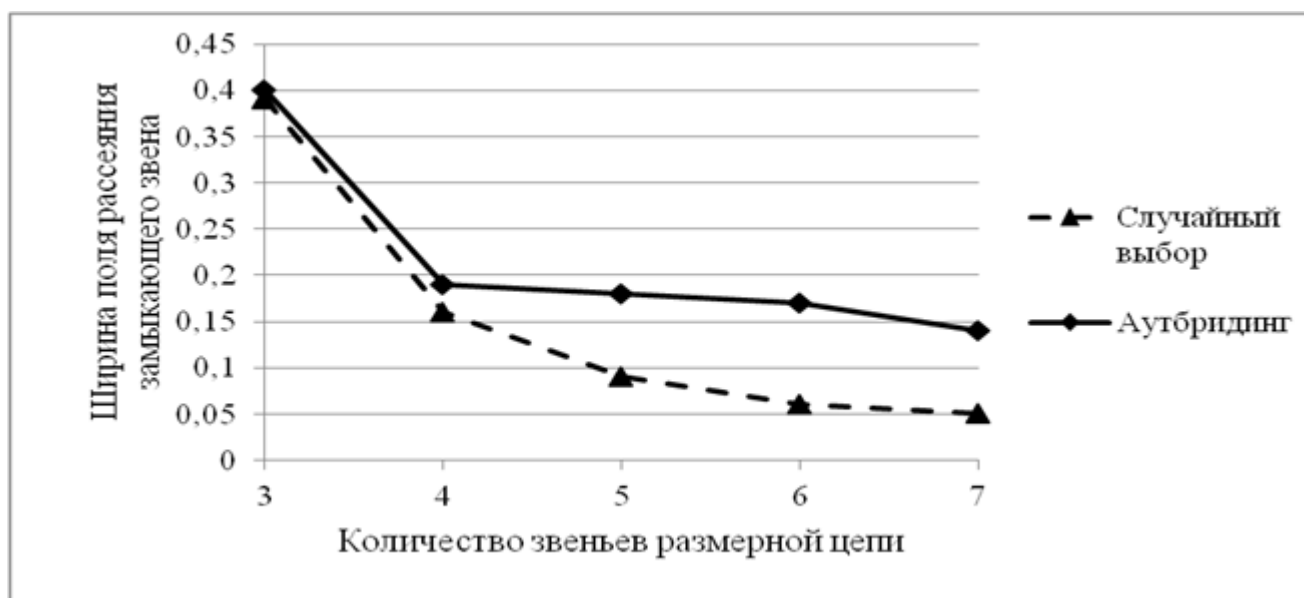


Рисунок 1 – Влияние правила отбора «родительских» пар на точность решения

Анализируя результаты экспериментов, можно сделать вывод, что оператор отбора «родительских» пар сказывается на точности полученного решения. Получение более качественного результата ведет к замедлению поиска и достигается при случайном выборе пар для обмена. Это происходит за счет расширения области поиска решения. Применение аутбридинга обеспечивает скорейшее нахождение решения, однако недостатком является сужение области поиска и как следствие нахождение локального минимума. Однако в этом случае временная сложность алгоритма по результатам эксперимента является константой и не зависит от размерности задачи.

На скорость работы алгоритма оказывает влияние критерий окончания поиска (рис.2).

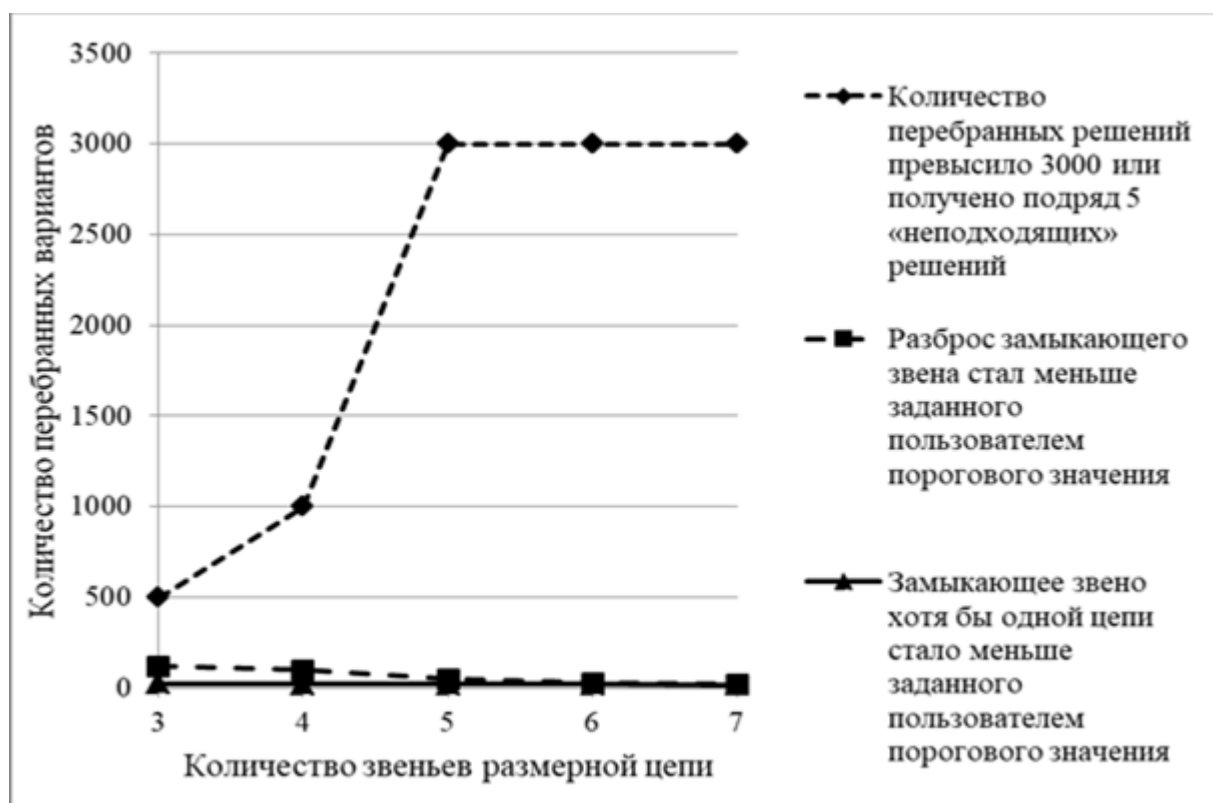


Рисунок 2 – Зависимость трудоемкости алгоритма от критерия окончания

Критерий окончания поиска оказывает влияние на трудоемкость и точность. Поэтому при росте размерности целесообразно применение комбинированного критерия окончания поиска решения, что позволяет контролировать вычислительную сложность алгоритма, которая не может превышать некоторую константу.

Можно сказать, что предлагаемый алгоритм частичного перебора является модификацией классического генетического алгоритма. Необходимость модификации связана с особенностями предметной области. Однако все преимущества генетических алгоритмов сохранены. К ним относятся способность получать эффективные решения (быстрее и эффективнее по сравнению с традиционными методами), хорошие параллельные возможности, наличие на каждом шаге решения, которое на каждой итерации улучшается.

Тем не менее, алгоритм не является эффективным для задач малой размерности (лучше применять полный перебор) и не гарантирует оптимальности полученного решения.

Выводы

В работе показаны возможности метода индивидуального подбора деталей при сборке высокоточных изделий. Для исследования задачи индивидуального подбора деталей была выполнена ее постановка, как задачи комбинаторной оптимизации. Исследована сложность задачи индивидуального подбора деталей, доказано, что задача относится к классу NP -полных задач. Следствием из этого факта является утверждение, что найти эффективный с точки зрения временной сложности алгоритм точного решения задачи невозможно. Предлагаемая авторами модель процесса использует для подбора алгоритм частичного случайного перебора, в основе которого лежат генетические алгоритмы. Рассмотрены параметры алгоритма, позволяющие получить «хорошее» решение при приемлемых вычислительных затратах. Выполнено сравнение предлагаемой модели процесса с моделями, использующими другие алгоритмы поиска.

Литература

1. Непомилуев В.В., Майорова Е.А. Исследование возможности повышения качества изготовления высокоточных изделий машиностроения путем учета компенсирующей способности деталей при использовании метода индивидуального подбора [Текст] // Сборка в машиностроении, приборостроении. - № 9. - 2009. С. 11-14.
2. Грэхем Р., Кнут Д., Паташник О. Конкретная математика. Основание информатики: Пер. с англ. – М.: Мир, 1998. – 703 с.
3. Пападимитриу Х., Стайглиц К. Комбинаторная оптимизация. Алгоритмы и сложность. – М.: Мир, 1982. – 512 с.
4. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. - М.: Мир, 1979.

- 535 с.

5. Тимофеева Е.В. Алгоритм выбора оптимального углового положения деталей ротора ГТД // Известия Волгоградского государственного технического университета. - 2007. - № 3 (29) – С. 119-121

6. Андерсен Дж. Дискретная математика и комбинаторика: пер. с англ. – М.: ООО «И.Д. Вильямс», 2016. – 960 с.

7. Cook S.A. The complexity of theorem-proving procedures. Proc. 3rd Ann. ACM Symp. On Theory of Computing, Association for Computing Machinery, New York, 151-158.

8. Гэри М. Вычислительные машины и труднорешаемые задачи / М. Гэри, Д. Джонсон. – М.: Мир, 1982. – 416 с.

9. Хопкрофт Дж., Мотвани Р., Ульман Дж. Введение в теорию автоматов, языков и вычислений. - М.: «Вильямс», 2002. – 528 с.

10. Bellman R., Dreyfus S. Applied Dynamic Programming. - Princeton: Princeton University Press, 1962

11. Куприянов, А. В. Комплектование при сборке с подбором деталей [Текст] / А. В. Куприянов // Сборка в машиностроении, приборостроении. – 2001. – № 11. – С. 8–10

12. Ламнауэр Н. Ю. Метод сборки деталей машин, обеспечивающий точность соединения / Н. Ю. Ламнауэр // Восточно-Европейский журнал передовых технологий. – 2014 – № 6/7(72). – С. 45-49.

13. Щербина О.А. Метаэвристические алгоритмы для задач комбинаторной оптимизации // Таврический вестник информатики и математики. - 2014. - № 1 (24). - С. 56-72.

14. Агеев Е.В., Бендерская Е.Н. Обзор природных вычислений: основные направления и тенденции // Научно-технические ведомости Санкт-Петербургского государственного политехнического университета. Информатика. Телекоммуникации. Управление. - СПб.: Изд-во СПбГПУ, 2014. - №2 (193). – С.9-22.

15. Матренин П.В., Гриф М.Г., Секаев В.Г. Методы стохастической оптимизации: учебное пособие. - Новосибирск: Изд-во НГТУ, 2016. - 67 с.

16. Вирсански Э. Генетические алгоритмы на Python / пер. с англ. А. А. Слинкина. – М.: ДМК Пресс, 2020. – 286 с.

17. Гладков Л.А., Курейчик В.В., Курейчик В.М., Сороколетов П.В. Биоинспирированные методы в оптимизации. - М.: ФИЗМАТЛИТ, 2009. - 384 с.

18. Задорина Н.А. Разработка модели процесса индивидуального подбора деталей с использованием методов стохастической оптимизации [Текст] // Сборник научных трудов 5-й Международной молодежной научно-практической конференции КАЧЕСТВО ПРОДУКЦИИ: КОНТРОЛЬ, УПРАВЛЕНИЕ, ПОВЫШЕНИЕ, ПЛАНИРОВАНИЕ Т.1, Юго-Зап. гос. ун-т., Курск: Из-во ЗАО «Университетская книга», 2018. С.213-217.

19. Задорина Н.А., Непомилуев В.В. Исследование влияния критерия окончания итерационного процесса подбора деталей на качество сборки изделий [Текст] // Сборник научных трудов 7-й Международной научно-технической конференции Управление качеством на этапах жизненного цикла технических и технологических систем Т.1, Юго-Зап. гос. ун-т., Курск: Из-во ЗАО «Университетская книга», 2019. С.284-288.

Задорина Н.А., Непомилуев В.В. Применение моделирования процесса индивидуального подбора деталей при сборке высокоточных изделий. В статье приведены результаты исследования метода индивидуального подбора деталей при сборке машин, содержащих многозвенные размерные цепи. Представлена модель процесса индивидуального подбора деталей на основе генетического алгоритма поиска оптимального решения и проведено ее исследование. Показано, что использование предлагаемого алгоритма подбора деталей позволяет существенно повысить точность таких машин, что дает возможность обеспечивать требуемое качество их сборки с меньшими затратами.

Ключевые слова: сборка, качество, размер, звено, размерная цепь, индивидуальный подбор деталей, компьютерное моделирование.

Zadorina N.A., Nepomiluev V.V. Application of the individual selection process simulation of parts in the assembly of high-precision products. In the article, the results of the study of the possibilities of the method of individual selection of parts when assembling machines containing multi-level chains. A process model of individual selection of parts based on a genetic algorithm for finding the optimal solution is presented and its study is carried out. The use of the proposed parts selection algorithm can significantly improve the accuracy of such machines, which makes it possible to ensure the required quality of their assembly at a lower cost.

Key words: Assembly, quality, size, link, dimensional chain, individual selection of parts, computer simulation.

Разработка и исследование моделей процесса распространения вредоносного программного обеспечения в корпоративных сетях

Н.А. Задорина^{*1}, А.А. Малышева^{*2}

^{*1} старший преподаватель, Рыбинский государственный авиационный технический университет имени П. А. Соловьева,
zadorina@rsatu.ru,

^{*2} магистрант, Рыбинский государственный авиационный технический университет имени П. А. Соловьева,
tzaryova.alina1998@gmail.com

Задорина Н.А., Малышева А.А. Разработка и исследование моделей процесса распространения вредоносного программного обеспечения в корпоративных сетях. Для организации информационного взаимодействия на современных предприятиях широко используются корпоративные вычислительные сети. Однако сетевая интеграция приводит к увеличению рисков, связанных с возможностью распространения по сети вредоносного программного обеспечения. Анализ защищенности сети и прогнозирование может проводиться с помощью моделирования процесса распространения вредоносного программного обеспечения в корпоративной сети. В статье рассматриваются модели процесса, особое внимание уделено имитационной модели и её параметрам, представляющим собой характеристики рабочих станций корпоративной сети.

Ключевые слова: корпоративная сеть, вредоносное программное обеспечение, имитационная модель процесса, моделирование, игровая модель.

Введение

В настоящий момент приоритетным направлением государственной поддержки является создание и развитие сервисов и технологий, направленных на формирование и поддержание производственного, экономического и технологического суверенитета страны. В этих условиях информация представляет для предприятий и организаций особую ценность, поскольку получение, создание и восстановление информации может быть достаточно дорогостоящим и трудоемким процессом. Противоправные действия с информацией приводят к нарушению ее свойств: конфиденциальности, достоверности, целостности и доступности [1].

Необходимо особо отметить опасность вредоносного программного обеспечения (ПО): троянское ПО, сетевые черви и т.д. Защита сети от вредоносного ПО является частью комплексной системы защиты информации. Задача защиты данных и противодействия распространению вредоносных программ очень сложна. В настоящее время атаки вредоносного программного обеспечения и целевые атаки с применением вредоносного ПО представляют наибольшую опасность для корпоративных сетей предприятий [2]. По данным [3] еженедельно более 34% предприятий подвергаются вредоносным атакам, а 80% финансовых учреждений ежегодно становятся мишенью для вредоносных программ. Одной из причин успешных атак вредоносного ПО в корпоративной сети являются устаревшие технологии безопасности, как следствие отсутствие профилактики и повышенная оценка возможностей имеющейся защиты. По мнению специалистов Лаборатории Касперского грамотная защита в IT-средах предполагает выполнение действий в 4 областях: предотвращение, обнаружение, реагирование и прогнозирование. Прогнозирование направлено на проактивный поиск угроз, а также на анализ потенциальных уязвимостей корпоративной сети с целью совершенствования мер противодействия и снижению риска. С помощью математических и имитационных моделей можно изучить динамику числа зараженных компьютеров в компьютерной сети и оценить необходимость усиления защиты сети.

В связи с этим возникает необходимость исследования применимости различных моделей прогнозирования процесса распространения вредоносного ПО для оценки защищенности сети предприятия.

Для решения проблемы необходимо проанализировать способы защиты сети и выявить место моделирования в системе защиты данных, осуществить сравнительный анализ моделей распространения вредоносного ПО в сетях и выявить наиболее перспективные из них.

Существующие подходы и решения

Прогнозирование распространения вредоносного ПО относится к классу упреждающих средств защиты.

Существуют различные модели прогнозирования процесса распространения вредоносного ПО [4], все многообразие которых можно разбить на три класса: аналитические, имитационные и натурные модели.

Аналитические модели предполагают замену исходной системы на некоторые функциональные соотношения, которые отражают только общие характеристики системы. Основными моделями в аналитическом подходе являются SI-модель, SIR-модель, SAIR-модель, PSIDR-модель.

Задачей аналитического моделирования является решение систем дифференциальных уравнений для получения теоретических результатов и графиков развития эпидемии и сопоставление этих результатов с практикой [5].

К достоинствам аналитического моделирования можно отнести большую силу обобщения, многократность использования, получение результата в короткие сроки, отсутствие значимых затрат на прогнозирование. Однако использование аналитического метода предполагает существенное упрощение, что не позволяет учитывать многие факторы.

Имитационное моделирование заменяет исходную систему моделью, достаточно точно описывающей ее в значимых для данного исследования характеристиках. Для разработки имитационной модели необходимо специальное программное обеспечение, необходима система моделирования.

Натурное моделирование предполагает проведение эксперимента на самом исследуемом объекте. Проводится построение реального сегмента сети и его заражение. Важным достоинством подобных моделей является легкость соблюдения технических особенностей моделируемых систем. Однако, с увеличением масштабов модели существенно возрастает стоимость эксперимента.

В последние годы наблюдается интерес к нейронным сетям, которые успешно применяются в самых различных областях. Очевидно, прогнозирование процесса распространения вредоносного ПО в корпоративной сети возможно с помощью искусственных нейронных сетей. Задача прогнозирования может быть решена следующими типами нейронных сетей: многослойным персептроном (MLP), радиально-базисной сетью (RBF), обобщенно-регрессионной сетью (GRNN), сетью Вольтерри и сетью Эльмана. Рассматривая задачу прогнозирования процесса распространения вредоносного ПО в корпоративной сети, как задачу прогнозирования временного ряда, нейросетевая модель должна выдавать количество зараженных узлов в заданные моменты времени.

Для обучения нейросети необходим датасет, содержащий данные, полученные, как вариант, с помощью других моделей (имитационной и аналитических). Датасет должен быть разделен на три части: тестовая выборка, обучающая выборка и предиктивная выборка.

На сегодняшний день существуют программные решения, позволяющие выполнять анализ защищенности корпоративной сети и отдельных ее узлов. Это Trend Virus Control System, Retina, Internet Scanner, Nessus Security Scanner, SAINT.

Основным недостатком существующих программных решений является невозможность рассматривать сеть целиком, рассматриваются отдельные узлы без связи друг с другом. Кроме того, программные продукты не позволяют оценить защищенность узла от некоего абстрактного, а не конкретного известного вируса. Этот недостаток является актуальным, так как ежегодно создаются новые виды вредоносного программного обеспечения, использующего все новые уязвимости узлов сети.

Таким образом, на данный момент не существует программных аналогов, выполняющих имитационное моделирование процесса распространения вредоносного ПО. Существующие подходы к моделированию в литературе описаны достаточно поверхностно и без рекомендаций к практическому применению. Прогнозирование распространения вредоносного ПО является актуальной задачей, поскольку позволяет выявить насколько быстро сеть предприятия будет заражена и насколько быстро необходимо принимать меры в случае начала развития эпидемии.

Имитационная модель процесса распространения вредоносного ПО в корпоративной сети

Авторами предложена имитационная модель на основе агентного подхода с использованием графовой модели сети [6].

Вредоносное ПО и рабочие станции рассматриваются как агенты. Каждая рабочая станция может иметь свои отличительные характеристики, такие как различное антивирусное ПО, различные операционные системы (ОС), связи с другими узлами. Как наиболее важные были выделены следующие параметры модели: топология сети, ранг сети, количество узлов, наличие антивирусного ПО, ОС, алгоритм поведения вредоносного ПО, частота атак, вероятность заражения узла.

Следует отметить, что наиболее распространенными средствами защиты рабочих станций являются: антивирусное ПО и встроенные средства защиты операционных систем. Антивирусное ПО для рабочих станций позволяют отлавливать те виды вредоносных программ, которые по тем или иным причинам не были обнаружены защитой на других уровнях. Здесь важно отметить, что вредоносное ПО постоянно совершенствуется, используя для своего распространения различные уязвимости, поэтому важно систематически выполнять обновление антивирусного ПО. Отсутствие установленных обновлений ПО, а также отключенный/не установленный антивирус, антивирус с не обновленными базами входят в ТОП-10 уязвимостей корпоративной

сети.

Операционная система (ОС), как правило, предусматривает некоторый набор средств защиты от основных классов угроз, в том числе и от атак вредоносного ПО. Например, в ОС Windows Vista предусмотрена случайная выборка адресного пространства (ASLR) для загрузки системных файлов по случайным адресам памяти. В этом случае вредоносному коду трудно узнать, где расположены привилегированные системные функции, что делает маловероятным их использование. В Windows 10 предусмотрен ранний запуск антивредоносной программы, реализующий установку защиты до старта всех сторонних драйверов и приложений.

Немаловажное значение имеет защита ядра ОС. В Mac OS X, например, это «песочница» (Sandbox) - изолированная среда для приложений, которые могут навредить системе. OS X обеспечивает «защиту песочницей» в Safari, изолируя встроенное средство просмотра PDF и плагины типа Adobe Flash Player, Silverlight, QuickTime и Oracle Java, как наиболее уязвимое и эксплуатируемое ПО. На уровне ядра имеется также защита в реальном времени: встроенная в процессор функция XD (выполнить/отключить), которая создает преграду между памятью, используемой для данных, и памятью для исполняемых инструкций. Согласно описанию Apple, это защищает от вредоносных программ, пытающихся обмануть Mac, заставив компьютер поступать с данными тем же образом, что и с программами, для того чтобы скомпрометировать систему.

Кроме того, ОС содержат в своем составе универсальные встроенные защитники, которые предотвращают проникновение в систему вредоносных программ. В состав ОС семейства Windows включены специальные функции, позволяющие удалить или заблокировать шпионские и другие нежелательные программы в режиме реального времени (Windows Defender). Инструментом обеспечения безопасности Linux является Tripwire - система обнаружения вторжений (HIDS). Задача Tripwire – мониторинг файловой системы и фиксирование изменений файлов.

Функция OS X Gatekeeper отвечает за защиту Mac от вредоносного ПО и «неподобающих приложений, загруженных из Интернета». Функция по своим целям и поведению напоминает контроль учетных записей Windows (UAC).

Таким образом, очевидно, что ОС обладают различной степенью защищенности от атак вредоносного ПО. Поэтому выбор и соответствующая настройка ОС влияют на уязвимость узла сети и, как следствие, на скорость и масштаб распространения вредоносного ПО в сети.

Еще одним значимым параметром процесса распространения является стратегия поведения вредоносного ПО [7], а именно возможность повторного заражения узла, создание и рассылка собственных копий вредоносным ПО.

Модель предусматривает 4 возможных алгоритма поведения вредоносного ПО: вредоносное ПО может пытаться заразить уже зараженный узел, при этом не создает своих копий; вредоносное ПО заражает только ранее не зараженные узлы и не создает собственных копий; вредоносное ПО может заразить уже зараженный узел и создает для рассылки собственные копии; вредоносное ПО заражает только ранее не зараженные узлы и создает свои копии.

Разработанная модель предполагает случайный выбор жертвы – заражаемого узла сети. Однако стратегия захвата узлов сети также оказывает существенное влияние на развитие эпидемии.

При случайном распространении заражаются произвольные узлы адресного пространства без его разделения на подмножества [7]. При распространении с разделением адресного пространства возможны два способа: RandomWithDif и CodeRedII. RandomWithDif предполагает деление всего адресного пространства сети на подпространства. Далее вредоносное ПО начинает рассылку своих копий по выделенным подпространствам адресов, пытаясь заразить в каждом из них узел. Как только ПО заражает узел, оно перестает отправлять свои копии в данное подпространство. Каждый экземпляр вредоносного ПО – потомок прародителя получает определенную часть адресного подпространства (сегмент) и также делит его на подпространства. Далее потомок начинает работать со своим сегментом. Таким образом, на всех этапах развития эпидемии каждый из потомков повторяет действия своего прародителя, действуя параллельно с остальными. CodeRedII – способ распространения более сложный, чем RandomWithDif. Каждый зараженный узел на следующем шаге с вероятностью 1/8 заражает совершенно случайно выбранный из всей сети узел. С вероятностью 3/8 заражает узлы из своего адресного подмножества. С вероятностью 1/2 заражает узел в одном из своих, уже им зараженных сегментов.

Для каждой стратегии распространения вредоносного ПО, использующих разделение адресного пространства, возможны различные методы деления: Sqrt, Binary, LoadBalance, GoldSection). Метод Sqrt делит адресное пространство всей сети размера N на число подпространств, равное $\sqrt{N}$. На следующем шаге каждый сегмент делится на подпространства $\sqrt{\sqrt{N}}$ и т.д. до тех пор, пока остается не более 2-х узлов в сегменте. Binary разбивает все пространство на первом этапе на 2 сегмента, затем каждый сегмент делится пополам (бинарное разделение). LoadBalance является более общим случаем бинарного разделения адресного пространства. В нем все пространство делится на заданное число подпространств (на k подпространств в общем случае). Каждый сегмент в свою очередь также делится на такое же число подмножеств. И так до момента, когда размер подмножества не будет меньше, чем заданная граница. GoldSection разбивает адресное пространство в соответствии с пропорцией золотого сечения на 2 части.

Следует также учитывать, что отправка копий вредоносного ПО осуществляется не между ближайшими,

физически связанными узлами сети, а в адресном пространстве узлов (размер которого зависит только от числа узлов), причем заражаться могут любые узлы (как соседние, так и разделенные друг от друга большим числом физических связей). Поэтому при разработке модели следует учитывать связку Стратегия распространения + Способ разделения адресного пространства.

Проведенные эксперименты с моделью показали, что наилучшей с точки зрения скорости захвата сети оказалась стратегия CodeRedII + Binary. Однако слишком быстрое распространение может быть быстро обнаружено, а эпидемия остановлена блокировкой зараженных узлов. Вредоносное ПО имеет определенный способ распространения и для каждой копии он одинаков. Однако каждый экземпляр вредоносного ПО может принимать решение о дальнейшем заражении сети (выбирать следующий узел для заражения) [8, 9]. Поэтому авторами сделан вывод, что наиболее значимым является алгоритм «поиска жертвы» со случайным выбором узла заражения.

Имитационное моделирование распространения вредоносного ПО позволяет спрогнозировать время заражения сети и другие параметры заражения. Эти данные помогают сделать вывод о степени устойчивости сети к атакам вредоносного ПО.

Теория игр для выработки стратегий защиты сети

Для выбора комплекса средств защиты авторы предлагают использовать аппарат теории игр. Существует множество способов защиты рабочих станций от вредоносного ПО, отличающихся по надежности, спектру угроз, от которых оно может защитить, и стоимости данной защиты. Задачу повышения защищенности отдельных узлов корпоративной сети в общем виде можно определить как задачу поиска оптимального компромисса между потребностями в защите и необходимыми для этих целей ресурсами. Потребности в защите обуславливаются, прежде всего, важностью и объемами защищаемой информации, а также условиями ее хранения, обработки и использования. Поэтому при выработке рекомендаций необходимо учитывать важность и критичность информационных активов и адекватность предпринимаемых мер безопасности по отношению к их значимости. В связи с этим необходимо выбирать различные методы защиты в зависимости от конкретной сети [10].

Защиту от атак вредоносного ПО можно рассматривать как игру. Хакера и системного администратора – как игроков. Каждая из сторон имеет свой набор стратегий. Необходимо выбрать оптимальную стратегию (чистую или смешанную) [11]. Наиболее точно описывают процесс защиты сети от атак вредоносного ПО парная бескоалиционная конечная игра (игроки системный администратор и хакер) с неполной информацией.

Применение аппарата теории игр позволило описать процесс выбора стратегии с помощью точного математического аппарата, вычислить затраты и прогнозируемый ущерб, рассчитать стратегии с различными критериями оптимальности.

К недостаткам модели можно отнести предположение, что участники конфликта ведут себя рационально, что не всегда соответствует действительности, а также сложность оценки рисков.

Практическая реализация результатов исследования

Авторами было разработано программное обеспечение, позволяющее спрогнозировать время заражения сети и оценить вероятность заражения каждого из узлов сети, а также получить рекомендации по защите отдельных узлов сети. Полученные данные позволят определить необходимость и оценить объем дальнейших работ по защите сети.

При разработке программных модулей системы анализа защищенности корпоративной сети предприятия использовалось общедоступное программное обеспечение широкого распространения. Для базовой программной платформы была использована операционная система MS Windows 10. Используемым языком программирования был выбран C++ (среда VisualStudio). Для хранения и работы с исходными данными выбрана СУБД SQLite.

На основе данных, характеризующих корпоративную сеть предприятия, система позволяет построить модели (аналитические модели SAIR и PSIDR, имитационную модель) и по результатам моделирования получить средние значения времени заражения, вероятность «захвата» каждого из узлов, а также графики развития эпидемии по каждой модели (см. рис.1).

Прогнозирование распространения вредоносного программного обеспечения с помощью аналитических моделей организуется с помощью решения систем дифференциальных уравнений. Для решения систем дифференциальных уравнений используется метод Рунге-Кутты 4 порядка.

Для получения результатов имитационного моделирования необходимо выбрать начальную вершину заражения, относительную максимальную длительность заражения, количество экспериментов, частоту атак, а также указать, возможно ли повторное заражение, копирует ли вирус себя, учитывать ли антивирус и учитывать ли общую защищенность узла (см. рис.2). Результаты моделирования выдаются в виде таблицы, где указывается имя узла, вероятность его заражения в случае атаки и среднее относительное время заражения. Также указывается количество полных заражений сети из общего количества экспериментов и среднее относительное время заражения всей сети.



Рисунок 1 – Схема функционирования системы

Рисунок 2 – Задание параметров имитационной модели

Эксперименты с модулем выдачи рекомендаций предполагали моделирование на сети до выдачи рекомендаций и моделирование с учетом выданных рекомендаций. Оценивалось среднее время заражения сети и количество полных заражений сети.

Результаты показали, что выданные рекомендации увеличивают защищенность сети. В случае, когда защита узла не учитывается, показатели среднего времени заражения сети и количество полных заражений не меняется, как и предполагалось. В случае, когда защита узла учитывается при тех параметрах сети, которые были рекомендованы, снижается количество полных заражений сети и значительно увеличивается среднее время заражения сети.

Выводы

В работе показана возможность применения различных моделей прогнозирования процесса распространения вредоносного ПО для оценки защищенности сети предприятия.

Анализ основных подходов к прогнозированию процесса распространения вредоносного ПО позволил выявить их достоинства и недостатки. Достоинством аналитического подхода является простота использования и скорость работы. Однако модели не позволяют учитывать особенности конкретной сети. Недостатком натурного моделирования является сложность проведения экспериментов и высокие затраты. Имитационная модель может учитывать большое количество параметров, свойственных конкретной сети, с ее помощью можно оценить степень влияния тех или иных параметров на процесс развития эпидемии. При этом данный подход является гораздо менее затратным, чем натурное моделирование. Предлагаемая авторами имитационная модель реализована на основе агентного подхода, как наиболее перспективного. Рассмотрены параметры процесса распространения вредоносного ПО, выявлены наиболее значимые.

Разработанное программное обеспечение позволило провести серию экспериментов с моделями. Результаты показали, что предложенная имитационная модель отвечает требованиям и может применяться для моделирования процесса распространения вредоносного ПО, при этом в отличие от аналитических моделей, разработанная модель учитывает особенности конкретной сети.

Литература

1. Ярочкин В.И. Информационная безопасность: Учебник для вузов. – 5-е изд. - М: Академический проект, 2008. – 544 с.
2. Передовая защита от сложных угроз и снижение риска целевых атак. [Электронный ресурс] – режим доступа: https://media.kaspersky.com/ru/business-security/Kaspersky_Anti_Targeted_Attack_Platform_White-paper_RU.pdf.
3. 13 различных типов вредоносных программ [Электронный ресурс] – режим доступа: New-Science.ru <https://new-science.ru/13-razlichnyh-tipov-vredonosnyh-programm>.
4. Абрамов К.Г. Модели угрозы распространения запрещенной информации в информационно-телекоммуникационных сетях [Текст]: дис. на соиск. учен. степ. канд. техн. наук / Абрамов Константин Германович; ВлГУ. – Владимир, 2014. – 117с.
5. Zesheng Chen, LixinGao, and Kevin Kwait. Modeling the spread of active worms. [Электронный ресурс] – режим доступа: <http://www.ece.ucf.edu/~yukse/teaching/nae/reading/2003-chen.pdf>.
6. Царева А.А., Задорина Н.А., Смирнов М.Ю. Имитационное моделирование процесса распространения сетевых червей в компьютерных сетях. - Технические науки: тенденции, перспективы и технологии развития./ Сборник научных трудов по итогам международной научно-практической конференции. No 5. Волгоград, –НН: ИЦРОН, 2018.
7. Возможные решения антивирусной защиты корпоративной сети [Электронный ресурс] – режим доступа: https://ozlib.com/825147/informatika/vozmozhnye_resheniya_antivirusnoy_zaschity_korporativnoy_seti
8. RETINA [Электронный ресурс] – режим доступа: <https://www.securitylab.ru/software/233019.php>
9. Система анализа защищенности Internet Scanner [Электронный ресурс] – режим доступа: http://citforum.ru/security/internet/internet_scan.shtml
10. Теория игр и защита компьютерных систем: учебное пособие / А.К. Гуц, Т.В. Вахний. – Омск: Изд-во ОмГУ, 2013. – 160 с.
11. Царева А.А., Задорина Н.А. Применение математического аппарата теории игр для разработки стратегии защиты компьютерной сети от эпидемий вредоносного программного обеспечения. - Молодежь. Техника. Космос: труды двенадцатой общерос. молодежн. науч.-техн. конф. В 4т. Т. 3. / Балт. гос. техн. ун-т. – СПб., 2020. – С.186-189.

Задорина Н.А., Малышева А.А. Разработка и исследование моделей процесса распространения вредоносного программного обеспечения в корпоративных сетях. Для организации информационного взаимодействия на современных предприятиях широко используются корпоративные вычислительные сети. Однако сетевая интеграция приводит к увеличению рисков, связанных с возможностью распространения по сети вредоносного программного обеспечения. Анализ защищенности сети и прогнозирование может проводиться с помощью моделирования процесса распространения вредоносного программного обеспечения в корпоративной сети. В статье рассматриваются модели процесса, особое внимание уделено имитационной модели и её параметрам, представляющим собой характеристики рабочих станций корпоративной сети.

Ключевые слова: корпоративная сеть, вредоносное программное обеспечение, имитационная модель процесса, моделирование, игровая модель.

Zadorina N.A., Malysheva A.A. Development and study of process models of spreading malware in corporate networks. To organize information interaction at modern enterprises, corporate computer networks are widely used. However, network integration increases the risks associated with the possibility of malware spreading over the network. Network security analysis and forecasting can be carried out by modeling the process of spreading malicious software in a corporate network. The article discusses the process models, special attention is paid to the simulation model and its parameters, which are the characteristics of the workstations of the corporate network.

Key words: corporate network, malware, process simulation model, simulation, game theory model.

Самоорганизация автомобильного потока в критическое состояние

Д.В. Бельков^{*1}, В.И. Зензеров^{*1}, В.Н. Лавренюк^{*2}

^{*1} к.т.н, доцент, Донецкий национальный технический университет,
belkovdv@list.ru

^{*2} магистрант, Донецкий национальный технический университет,
knaiz0f@gmail.com

Бельков Д.В., Зензеров В.И., Лавренюк В.Н. Самоорганизация автомобильного потока в критическое состояние. Выполнено исследование модели BML (Biham-Middleton-Levine) движения автотранспорта с учетом самоорганизованной критичности. В результате определено, что в синхронизированном потоке имеется набор метастабильных критических состояний, переходящих друг в друга посредством лавин, процесс уменьшения потока при изменении плотности в пределах 0,5 – 0,8 происходит по степенному закону с достоверностью не ниже 0,77, а также получена ступенчатая зависимость параметра порядка d от управляющего параметра p при пороговом значении равно 0,5.

Ключевые слова: синхронизированный автомобильный поток, затор, метастабильные состояния, самоорганизованная критичность.

Введение

При поиске оптимальных решений по проектированию дорожной сети и организации движения необходимо учитывать широкий круг характеристик автомобильного потока и закономерности влияния различных факторов на динамические свойства потока. Для моделирования движения автомобилей широко используются клеточные автоматы. Поэтому в теории транспортных потоков предложено множество моделей [1,2,3]. Целью данной статьи является моделирование плотного транспортного потока с учетом фаз синхронизированного потока и затора. Задача – исследование модели BML (Biham-Middleton-Levine) движения автотранспорта с учетом самоорганизованной критичности (SOC).

Эта теория объясняет явления масштабной инвариантности и прерывистого равновесия [4]. Согласно SOC динамика сложных систем приводит их к критическим состояниям. Системы самоорганизуют себя так, чтобы оказаться в критическом состоянии. Самоорганизованная критичность представляет собой набор метастабильных критических состояний, переходящих друг в друга посредством «лаvin». Лавины могут быть как малыми, так и охватывающими всю систему, и те, и другие вызываются одинаково малыми внешними возмущениями.

Фазовые состояния автомобильного потока

В современной теории автомобильного потока важное место занимает фундаментальная диаграмма. Это зависимость потока от его плотности. Величина потока $q(x,t)$ равна числу машин, пересекающих сечение автомагистрали в точке x за единицу времени. Идеальная фундаментальная диаграмма показана на рис. 1.

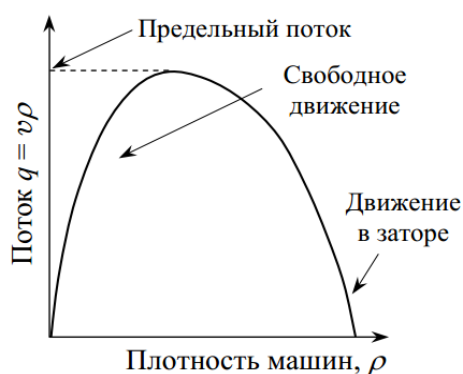


Рисунок 1 – Фундаментальная диаграмма

В автомобильном потоке выделяют свободный и плотный потоки. В модели Б.С. Кернера [5] плотный поток состоит из двух фаз: синхронизированный поток и широкие перемещающиеся пробки. Таким образом, различаются три фазы движения транспортного потока и его свойства объясняются с помощью фазовых переходов между перечисленными состояниями потока. Эти состояния соответствуют различным фрагментам фундаментальной диаграммы.

При свободном движении дорога не загружена и водители придерживаются желаемой скорости, свободно переходя на соседние полосы. Корреляции в движении отдельных автомобилей малы, и средняя скорость машин определяется средним расстоянием между ними. Синхронизированное движение, соответствующее убывающей ветви фундаментальной диаграммы, возникает, когда водители не могут свободно маневрировать, и вынуждены согласовывать свою скорость со скоростью потока. Автомобили мешают друг другу разогнаться до оптимальной скорости, но поток не переходит в состояние затора. В таком потоке формируются большие кластеры автомобилей, которые перемещаются по автомагистрали как единое целое. Такой кластер может распространяться на все ряды магистрали и содержать десятки машин по направлению движения. В этом случае корреляции в движении отдельных автомобилей достаточно велики. С ростом плотности потока движение становится неустойчивым, и поток распадается на отдельные группы автомобилей с неравномерным движением. При уменьшении средней скорости до 20 км/ч начинается формирование заторов. Фазовые состояния в реальном транспортном потоке показаны на рис. 2.

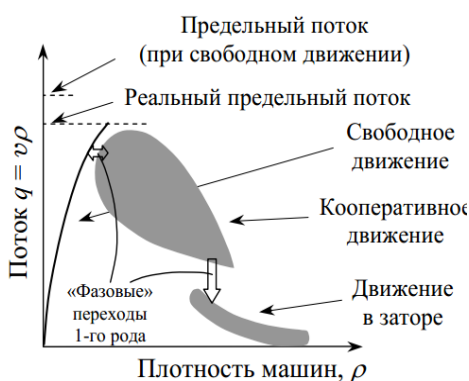


Рисунок 2 – Фазовые состояния автомобильного потока

Переход от свободного (F) к плотному потоку в теории трех фаз соответствует возникновению фазы синхронизированного потока, т.е. фазовому переходу $F \rightarrow S$. Природу $F \rightarrow S$ перехода можно объяснить с помощью «соревнования» во времени и пространстве двух противоположных процессов: ускорения машин при обгоне более медленного автомобиля (переускорение), и в случае, когда обгон невозможен, торможения машин до скорости более медленного автомобиля (адаптация скорости). Переускорение поддерживает дальнейшее существование свободного потока. Адаптация скорости приводит к синхронизированному потоку.

Спонтанный $F \rightarrow S$ переход может произойти в широком диапазоне значений величины потока q в свободном транспортном потоке. Основываясь на эмпирических данных измерений, Б.С. Кернером был сделан вывод, что существует бесконечное число значений пропускной способности автомагистрали в свободном потоке. Как показано на рис. 3, это бесконечное число значений пропускной способности находится в диапазоне между минимальным q_{th} и максимальным q_{max} значениями пропускной способности. Если величина потока близка к максимальному значению пропускной способности, то достаточно малое возмущение в свободном потоке вблизи узкого места приведет к спонтанному $F \rightarrow S$ переходу. Если величина потока близка к минимальному значению пропускной способности, то только возмущение большой амплитуды способно привести к спонтанному $F \rightarrow S$ переходу.

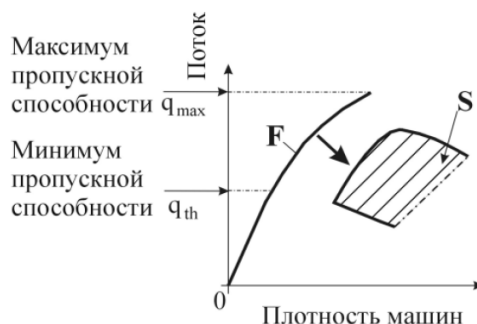


Рисунок 3 – Максимум и минимум пропускной способности

В теории трех фаз движение в заторе соответствует широкому движущемуся кластеру автомобилей. Такое состояние (wide moving jam, фаза J) не возникает в свободном потоке, но может возникнуть в области синхронизованного потока [6,7]. Образование затора является результатом каскада $F \rightarrow S \rightarrow J$ фазовых переходов, показанного на рис. 4.

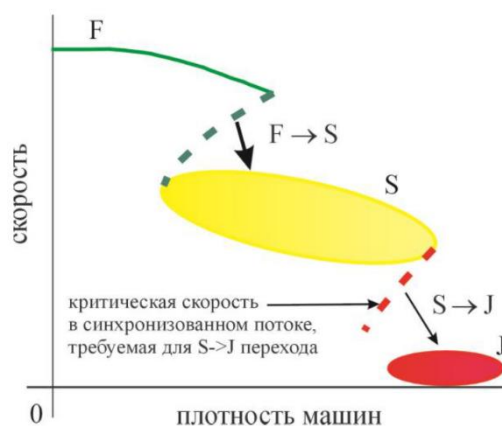


Рисунок 4 – Каскад фазовых переходов

Важной особенностью является переход автомобильного потока в затор через последовательность метастабильных состояний синхронизованного потока. Область метастабильных состояний показана на рис. 5.

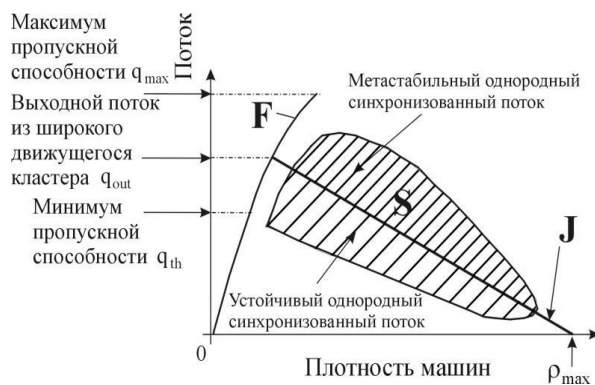


Рисунок 5 – Область метастабильных состояний

Как показано на рис. 6 при переходе от свободного потока к затору возникает последовательность метастабильных состояний [8]. Скорость потока со временем уменьшается и возникает несколько метастабильных состояний синхронизованного потока.

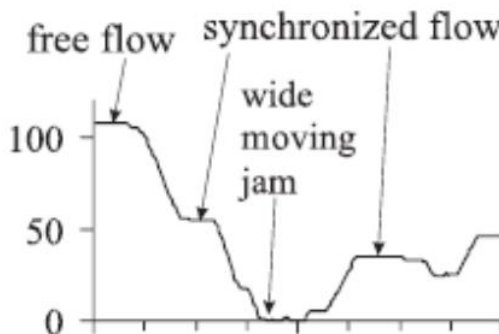


Рисунок 6 – Метастабильные состояния синхронизованного потока

Состояние самоорганизованной критичности

В данной работе предполагается, что последовательность метастабильных состояний синхронизованного потока соответствует самоорганизованной критичности. Реализация в динамической системе такого состояния

означает, что она под влиянием малых внешних возбудителей приходит в критическое состояние, которое в процессе дальнейшей эволюции является самоподдерживающимся. По структуре оно представляет собой последовательность метастабильных состояний, переходящих друг в друга посредством лавин, инициируемых малыми внешними возбудителями. Размеры возникающих лавин демонстрируют степенное распределение.

Базовой моделью теории SOC является куча песка. Рассмотрим кучу песка, изображенную на рис. 7.

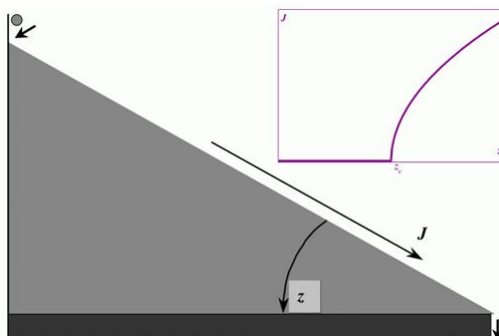


Рисунок 7 – Куча песка

Состояние системы полностью определяется наклоном поверхности. В тех местах, где локальный наклон больше порога устойчивости, происходит осыпание песчинок вниз по склону на соседние участки поверхности, т.е. возникает лавина. Пусть угол наклона поверхности равен u . При его изменении происходит фазовый переход от неподвижного состояния к состоянию непрерывного тока песка. Если средний наклон поверхности u меньше чем u_c , то песок неподвижен. Если наклон превышает значение u_c , возникает спонтанный ток песка J по поверхности. Возникает фазовый переход, в котором управляющим параметром является наклон u , а параметром порядка – ток песка J . Критическое значение наклона u_c разделяет хаотическую ($u < u_c$) и упорядоченную ($u > u_c$) фазы. Обе эти фазы соответствуют состоянию системы устойчивому к малым возмущениям. В хаотической фазе они быстро затухают, а в упорядоченной – не могут повлиять на величину тока. Зависимость параметра порядка от управляющего параметра приведена на врезке рис. 7.

Одной из областей, где теория самоорганизованной критичности позволила добиться значительного успеха при моделировании процесса, является биологическая эволюция. Объектом эволюции является вид, представляющий собой группу особей, упорядоченных по возрастанию приспособленности к условиям окружающей среды. Приспособленность видов изменяется в результате двух различных процессов. Во-первых, переход от одной особи к соседней особи увеличивает приспособленность вида, и он поднимается выше по эволюционному ландшафту. Этот процесс можно назвать мутацией особей. Во-вторых, переход от одного вида к другому обеспечивает связь между видами. В результате такой мутации видов изменяется эволюционный ландшафт. В критическом состоянии экосистема демонстрирует прерывистое равновесие.

Аналогия между эволюцией видов и процессом перехода в затор показана в табл. 1.

Таблица 1 – Аналогия процессов эволюции и перехода в затор

Эволюция видов	Переход в затор
Особь	Автомобиль
Вид	Метастабильное состояние потока
Мутация особей	Переход между метастабильными состояниями (лавина)
Мутация видов	Движение в метастабильных состояниях потока
Общая накопленная приспособленность	Скорость свободного потока
Эволюционный ландшафт	Ландшафт скорости

Вычислительный эксперимент

При исследовании фазовых состояний транспортного потока в данной статье используется модель BML [9,10], реализованная в среде программирования R. Она известна как самоорганизующаяся модель транспортного потока на основе детерминированного клеточного автомата. Подвижные элементы представлены точками на

решетке со случайным стартовым положением. Модель BML является двумерной аналогией движения по правилу 184. Согласно этому правилу подвижные элементы подобно частицам движутся в одном направлении, останавливаясь и продолжая движение в зависимости от наличия элементов впереди. Модель имеет следующие правила перехода: если ячейка i не пуста, а $(i + 1)$ – пуста, то на следующем шаге ячейка $(i + 1)$ оказывается занята. Значение ячейки i для новой итерации определяется её левым соседом, для которого схема определения состояния аналогична.

BML-модель позволяет в зависимости от значения параметра p (плотность машин) получать различные метастабильные состояния величины v (количество подвижных машин). Результаты моделирования показаны на рис. 8–15.

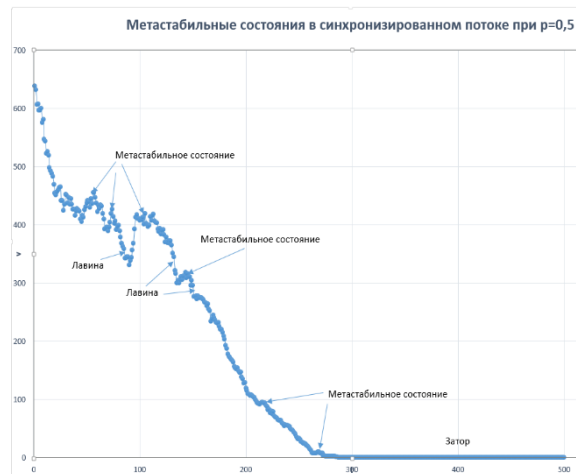


Рисунок 8 – Метастабильные состояния в синхронизированном потоке при $p=0,5$

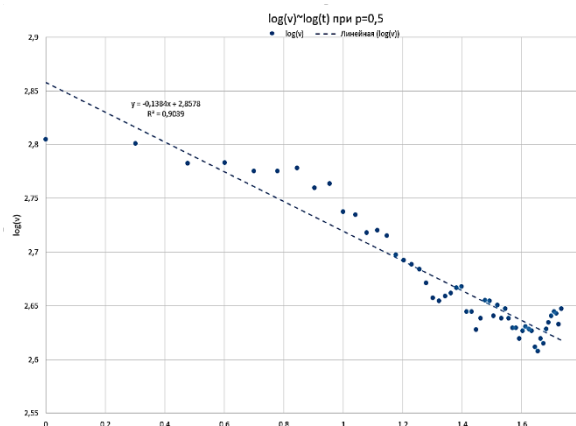


Рисунок 9 – Степенной закон уменьшения скорости при $p=0,5$

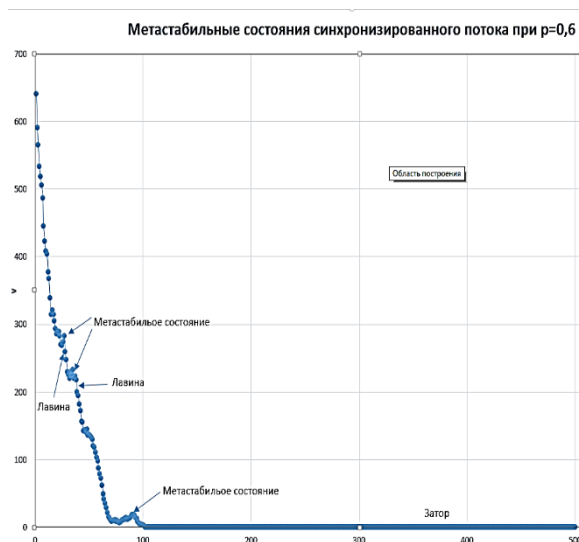


Рисунок 10 – Метастабильные состояния в синхронизированном потоке при $p=0,6$

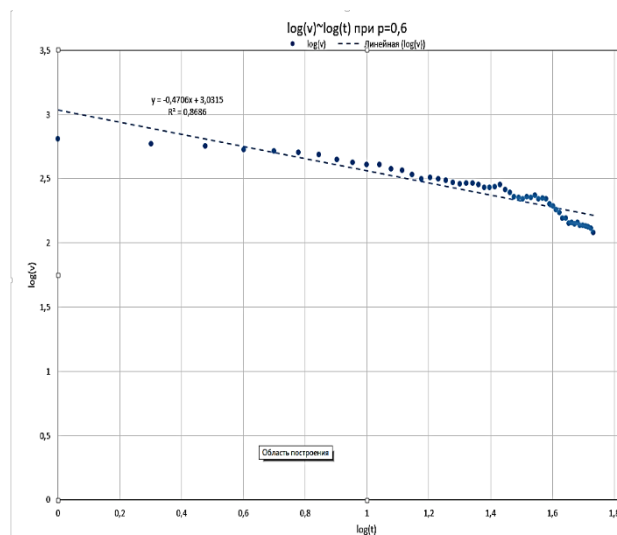


Рисунок 11 – Степенной закон уменьшения скорости при $p=0,6$

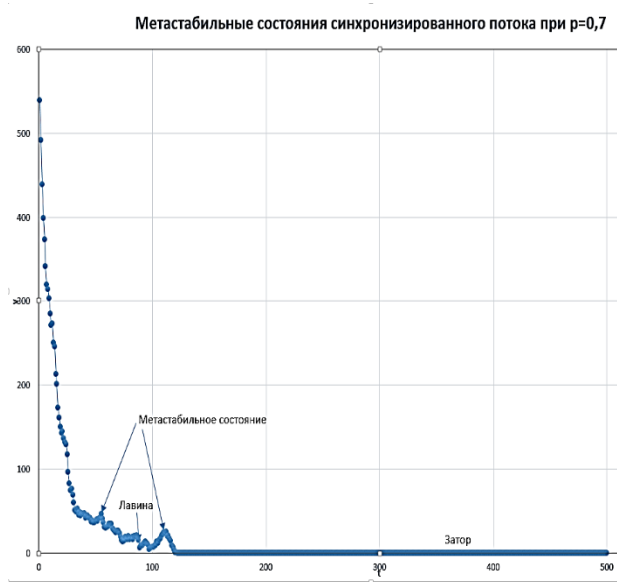


Рисунок 12 – Метаустойчивые состояния в синхронизированном потоке при $p=0,7$

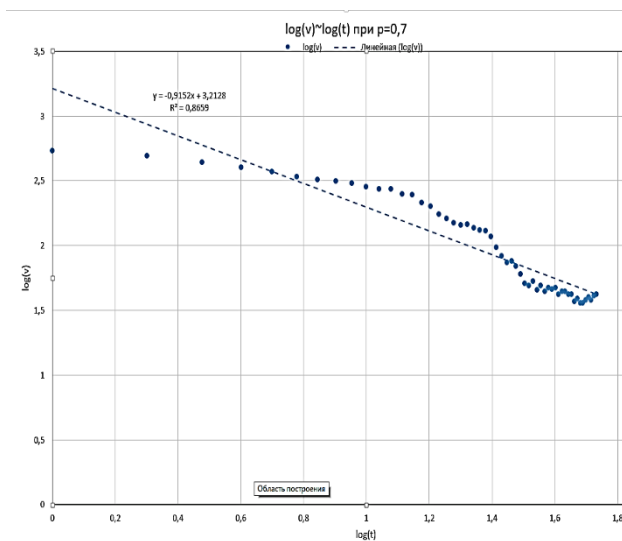


Рисунок 13 – Степенной закон уменьшения скорости при $p=0,7$



Рисунок 14 – Метабильные состояния в синхронизированном потоке при $p=0,8$

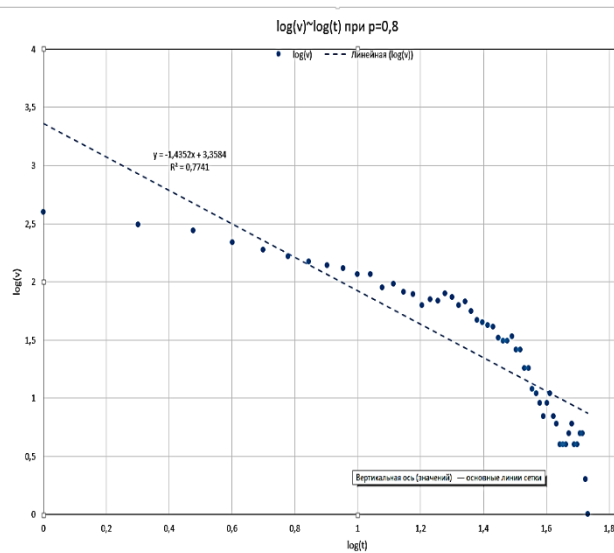


Рисунок 15 – Степенной закон уменьшения скорости при $p=0,8$

При различных значениях плотности автомобилей величина затора, измеряемая количеством шагов моделирования, где автомобили неподвижны, имеет разную длительность. На рис. 16 показана зависимость длительности затора d от плотности машин p .

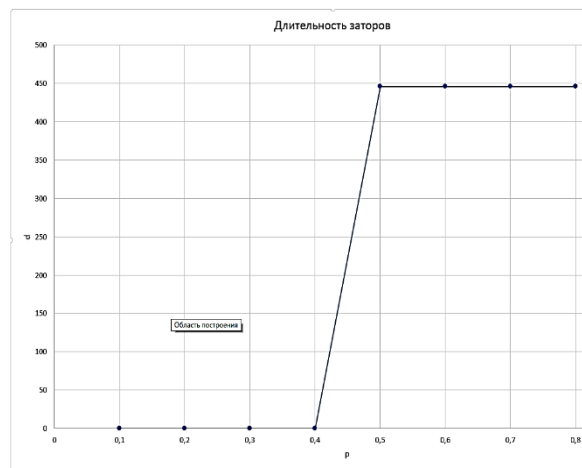


Рисунок 16 – Зависимость параметра порядка d от управляющего параметра p

Выводы

В статье выполнено исследование модели BML (Biham-Middleton-Levine) движения автотранспорта с учетом самоорганизованной критичности. Получены следующие результаты: в синхронизированном потоке имеется последовательность метастабильных критических состояний, переходящих друг в друга посредством лавин; процесс уменьшения потока v при плотности $p=0,5$ происходит по степенному закону $v=t^{-0,14}$, достоверность – $R^2=0,9$; процесс уменьшения потока v при плотности $p=0,6$ происходит по степенному закону $v=t^{-0,47}$, достоверность – $R^2=0,87$; процесс уменьшения потока v при плотности $p=0,7$ происходит по степенному закону $v=t^{-0,92}$, достоверность – $R^2=0,87$; процесс уменьшения потока v при плотности $p=0,8$ происходит по степенному закону $v=t^{-1,44}$, достоверность – $R^2=0,77$. Получена ступенчатая зависимость параметра порядка d от управляющего параметра p , пороговое значение – $p=0,5$.

Литература

1. Долгушин Д.Ю. Многофакторное моделирование автотранспортных потоков на основе клеточных автоматов. Автореферат диссертации на соискание ученой степени кандидата технических наук. – Омск: СибАДИ, 2011. – 21 с.
2. Долгушин Д.Ю., Мызникова Т.А. Применение клеточных автоматов к моделированию автотранспортных потоков. – Омск: СибАДИ, 2012. – 112 с.
3. Лубашевский И.А., Гусейн-Заде Н.Г., Гарнисов К.Г. Макроскопические фазовые состояния автотранспортного потока в туннелях. <https://b-ok.org/book/2980831/71d08c>
4. Подлазов А.В. Теория самоорганизованной критичности – наука о сложности. [Электронный ресурс], 2008. – Режим доступа: <http://www.nonlin.ru/articles/podlazov/soc>
5. Семенов В.В. Математическое моделирование автотранспортных потоков. http://www.mathnet.ru/php/archive.phtml?wshow=paper&jmid=ipmp&paperid=817&option_lang=rus
6. Введение в математическое моделирование транспортных потоков: учеб. пособие / Гасников А.В., Кленов С.Л., Нурминский Е.А., Холодов Я.А., Шамрай Н.Б.; Под ред. А.В. Гасникова. — М.: МФТИ, 2012. — 362 с.
7. Кленов С.Л. Стохастические математические модели транспортного потока в рамках теории трех фаз. Автореферат диссертации на соискание ученой степени доктора физико-математических наук. Москва – 2019. – 34 с.
8. B.S. Kerner, S.L. Klenov, M. Schreckenberg. Simple cellular automaton model for traffic breakdown, highway capacity, and synchronized flow. // <https://arxiv.org/ftp/arxiv/papers/1805/1805.05555.pdf>
9. Бак П. Как работает природа. Теория самоорганизованной критичности. Москва. – 2013. – 276 с.
10. Модель движения Biham-leton-Levine. [http://ru.knowledgr.com/18079938/Транспортная МодельBihamMiddletonLevine](http://ru.knowledgr.com/18079938/Транспортная%20МодельBihamMiddletonLevine)

Бельков Д.В., Зензеров В.И., Лавренюк В.Н. Самоорганизация автомобильного потока в критическое состояние. Выполнено исследование модели BML (Biham-Middleton-Levine) движения автотранспорта с учетом самоорганизованной критичности. В результате определено, что в синхронизированном потоке имеется набор метастабильных критических состояний, переходящих друг в друга посредством лавин, процесс уменьшения потока при изменении плотности в пределах $0,5 - 0,8$ происходит по степенному закону с достоверностью не ниже $0,77$, а также получена ступенчатая зависимость параметра порядка d от управляющего параметра p при пороговом значении равном $0,5$.

Ключевые слова: синхронизированный автомобильный поток, затор, метастабильные состояния, самоорганизованная критичность.

Belkov D.V., Zenzerov V.I., Lavrenuk V.N. Self-organization of the traffic flow into a critical state. Investigates the BML (Biham-Middleton-Levine) model of vehicular traffic, taking into account self-organized criticality. As a result, it was determined that in a synchronized flow there is a set of metastable critical states that pass into each other through avalanches, the process of reducing the flow with a change in density within $0.5 - 0.8$ occurs according to a power law with a confidence of at least 0.77 , and a step dependence of the order parameter d on the control parameter p is obtained with a threshold value of 0.5 .

Keywords: synchronized traffic flow, congestion, metastable states, self-organized criticality.

Анализ трафика беспроводной сети

Д.В. Бельков^{*1}, Д.Д. Лосев^{*2}

^{*1} к.т.н, доцент, Донецкий национальный технический университет,
belkovdv@list.ru

^{*2} студент, Донецкий национальный технический университет,
daniill.tf@gmail.com

Бельков Д.В., Лосев Д.Д. Анализ трафика беспроводной сети. Исследования современного сетевого трафика показали, что это - самоподобный процесс и его Марковские модели приводят к некорректным результатам. Поэтому важной задачей является дальнейшее исследование трафика. Данная статья представляет результаты статистического анализа временных рядов задержки в сети с GPRS-доступом на стороне отправителя и UMTS-доступом со стороны получателя.

***Ключевые слова:** самоподобие, трафик, задержка, функция автокорреляции, спектральная плотность, параметр Херста.*

Введение

Для процессов пакетной передачи характерным является обнаруженное на практике свойство фрактальности. В процессе передачи возникают большие всплески при относительно низком среднем уровне трафика. Данное явление, которое значительно влияет на потери и задержки пакетов, не учитывается в теории телетрафика. Расчет параметров системы распределения информации по классическим формулам дает некорректные, неоправданно оптимистические результаты. Алгоритмы обработки трафика, созданные для работы с простейшим потоком неэффективны для фрактальных потоков в современных сетях.

В связи с тенденциями объединения различных телекоммуникационных приложений на базе универсальной сетевой инфраструктуры актуальной проблемой является разработка методов анализа и синтеза информационно-управляющих систем. Данная работа посвящена одной из важных задач развития этого направления – исследованию процессов в компьютерных сетях с беспроводным доступом.

Появление новых свойств сетевого трафика, необходимость обеспечения высокого качества обслуживания приложений, делают актуальным статистический анализ трафика.

Целью данной работы является исследование структуры трафика сети с беспроводным доступом, направленное на выявление его характерных особенностей. Изучается один из основных сетевых процессов – процесс RTT-задержки, который служит для получения информации о состоянии сети методом «черного ящика», когда через сеть пропускается последовательность пакетов, и на основании времени их прохождения до удаленного узла и обратно делаются выводы о загрузке сети.

В работе решаются следующие задачи: оцениваются плотности распределения, автокорреляционные функции (АКФ), энергетические спектры изучаемых временных рядов, определяется степень фрактальности (показатель Херста). Вычислительный эксперимент выполнен в среде Octave.

Реализации сетевого трафика

Для изучения выбраны шесть реализаций сетевого трафика, полученные в университете города Наполи (Италия). Согласно лицензии, данные свободно доступны для анализа. Измерения проводились каждые 10 миллисекунд, получено свыше 3000 отсчетов. Отправитель имел GPRS-доступ, получатель - UMTS-доступ, скорость передачи 100 pps, операционная система Windows32, на каждой из сторон. Для процесса RTT-задержки на маршруте прохождения пакета можно выделить постоянную минимальную составляющую, обусловленную отсутствием очередей и переменную составляющую, возникающую из-за задержек в очередях. RTT-задержка измеряется на стороне источника и складывается из времени обработки, времени распространения, времени коммутации, времени ожидания в очереди. Изучаемые временные ряды показаны на рис. 1-3. Условные обозначения, принятые в работе, показаны в табл. 1.

Таблица 1 – Условные обозначения

Обозначение	Описание
TCP_d64	Ряд измерений задержки TCP-пакетов объемом 64 байт
TCP_d256	Ряд измерений задержки TCP-пакетов объемом 256 байт
TCP_d1024	Ряд измерений задержки TCP-пакетов объемом 1024 байт

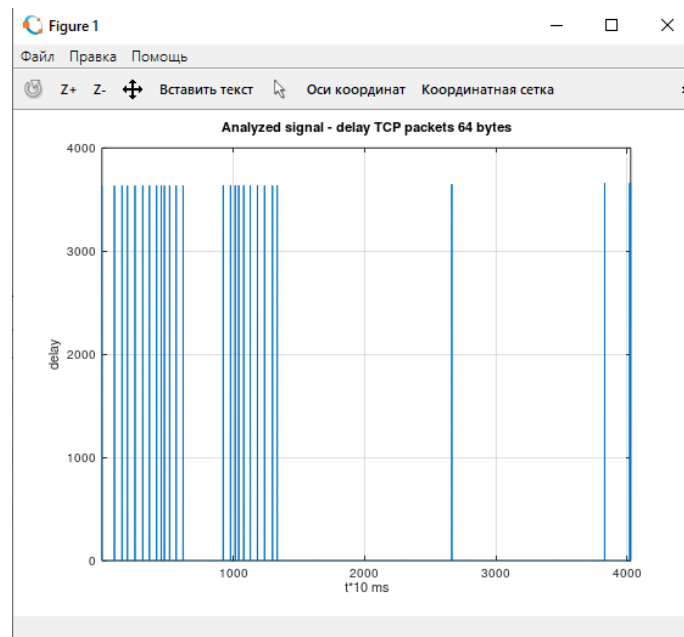


Рисунок 1 – Задержка при передаче пакетов объемом 64 байт

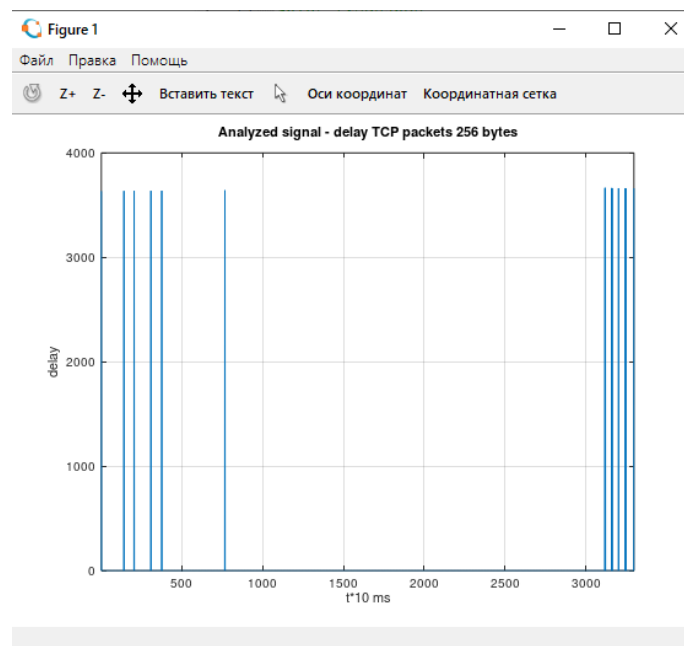


Рисунок 2 - Задержка при передаче пакетов объемом 256 байт

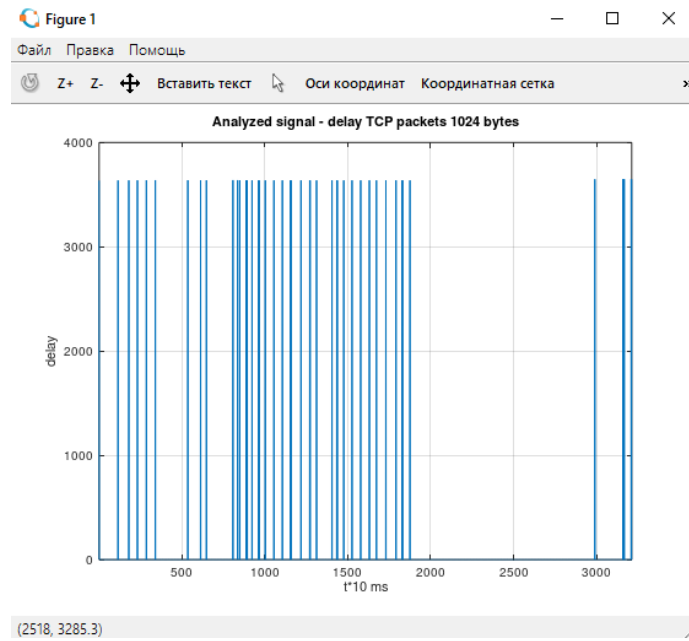


Рисунок 3 - Задержка при передаче пакетов объемом 1024 байт

Анализ плотности распределения

В этом разделе выполнен анализ плотности распределения. Оценка проводится на основании гистограмм относительных частот, показанных на рис. 4-6. Визуальный анализ позволяет сделать вывод, что исследуемые ряды не подчиняются нормальному распределению.

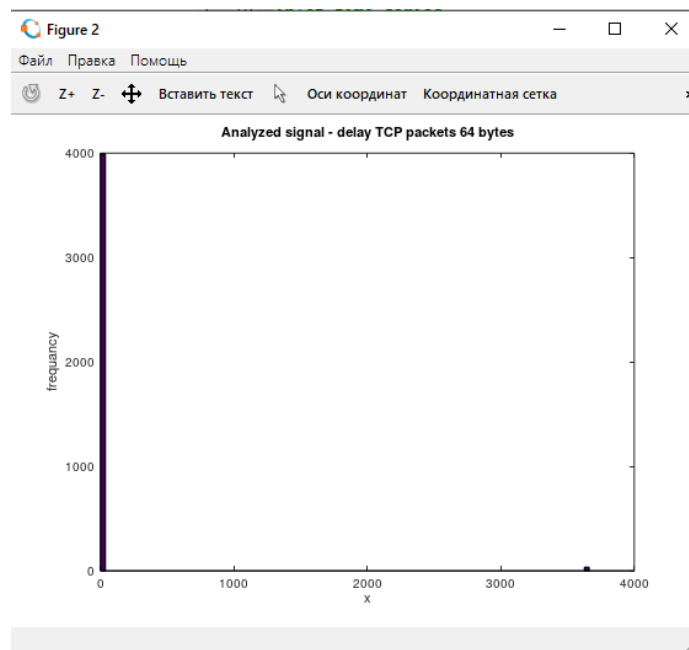


Рисунок 4 - Плотность распределения временного ряда TCP_d64

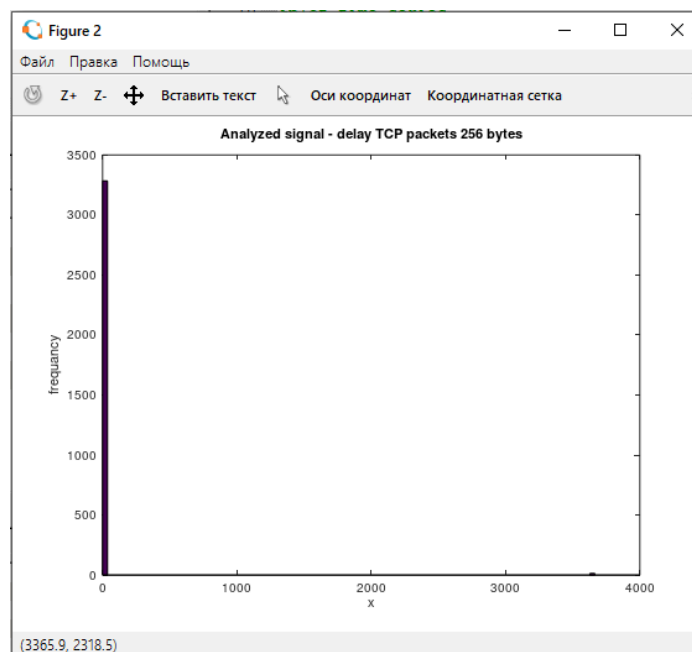


Рисунок 5 - Плотность распределения временного ряда TCP_d256

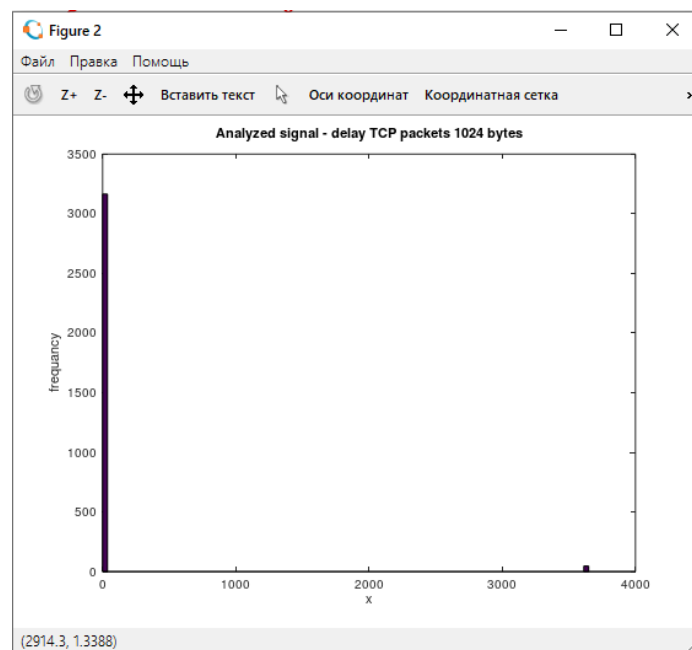


Рисунок 6 - Плотность распределения временного ряда TCP_d1024

Анализ автокорреляционных функций

В этом разделе необходимо определить обладают ли временные ряды медленно убывающей зависимостью или быстро убывающей зависимостью. По определению процесс X обладает медленно убывающей зависимостью (МУЗ) [long-range dependence], если для его АКФ выполняется условие $r(k) \sim k^{-\beta} \cdot L_1(k), k \rightarrow \infty$, где $0 < \beta < 1$ и L_1 - медленно меняющаяся на бесконечности функция: $\lim_{t \rightarrow \infty} \frac{L_1(tx)}{L_1(t)} = 1$ для всех x . Процессы с МУЗ характеризуются автокорреляционной функцией, которая убывает по степенному закону при увеличении временной задержки (лага). В отличие от процессов с МУЗ, процессы с быстро убывающей зависимостью (БУЗ) [short-range dependence] обладают экспоненциально спадающей АКФ вида $r(k) \sim e^{-k}, k \rightarrow \infty$.

Вычисления АКФ выполнены по формуле $r(k) = \frac{\sum_{i=1}^{N-\tau} (X_i - \bar{X})(X_{i+k} - \bar{X})}{(N - \tau)\sigma^2(X)}$, где $\bar{X}$ - выборочное среднее

ряда X , $\sigma^2(X)$ - выборочная дисперсия ряда X , $k=0,1,\dots$. Автокорреляционные функции исследуемых временных рядов приведены на рис. 7-9.

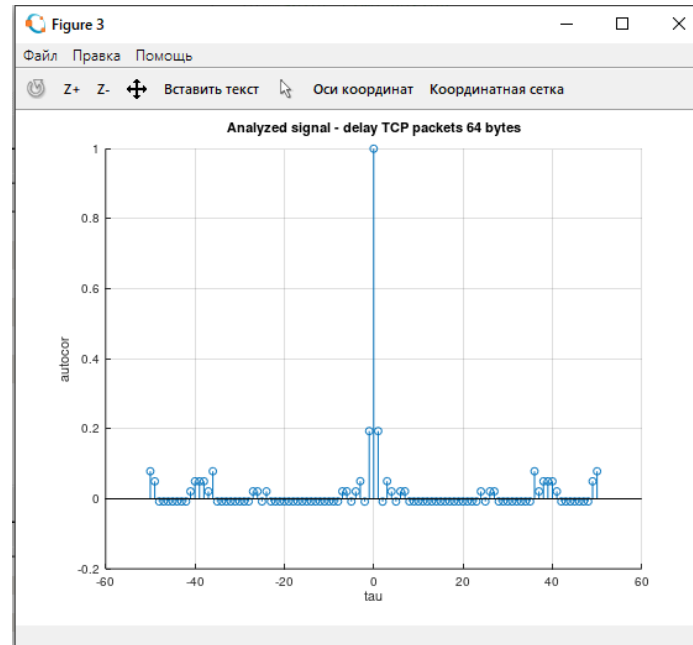


Рисунок 7 - АКФ временного ряда TCP_d64

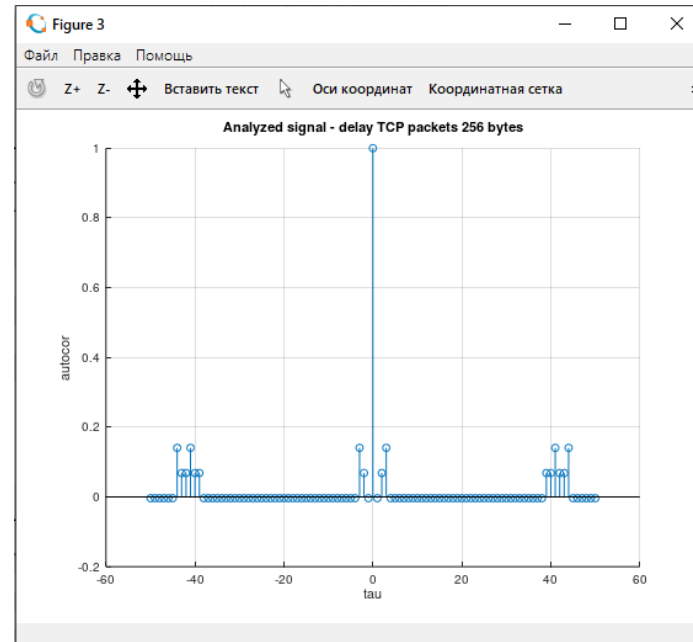


Рисунок 8 - АКФ временного ряда TCP_d256

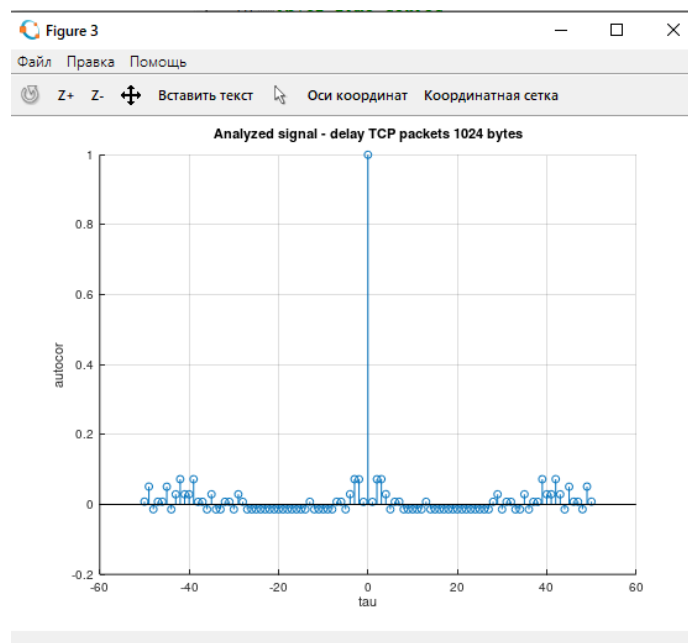


Рисунок 9 - АКФ временного ряда TCP_d1024

Спектральный анализ

В этом разделе выполнен расчет энергетических спектров исследуемых временных рядов. Известно, что в частотной области МУЗ приводит к степенному закону поведения спектральной плотности процесса. Процесс X обладает МУЗ, если для спектральной плотности выполняется условие, где $f \rightarrow 0$, $i = \sqrt{-1}$, $0 < b < 4$ и L_2 - медленно изменяющаяся в нуле функция. Процесс с МУЗ обладает спектральной плотностью с особенностью в нуле: спектральная плотность стремится к бесконечности, по мере того как частота f стремится к нулю. Энергетические спектры реализаций представлены на рис. 10-12. Визуальный анализ позволяет сделать вывод, что временные ряды задержки самые мощные гармоники имеют на низких частотах.

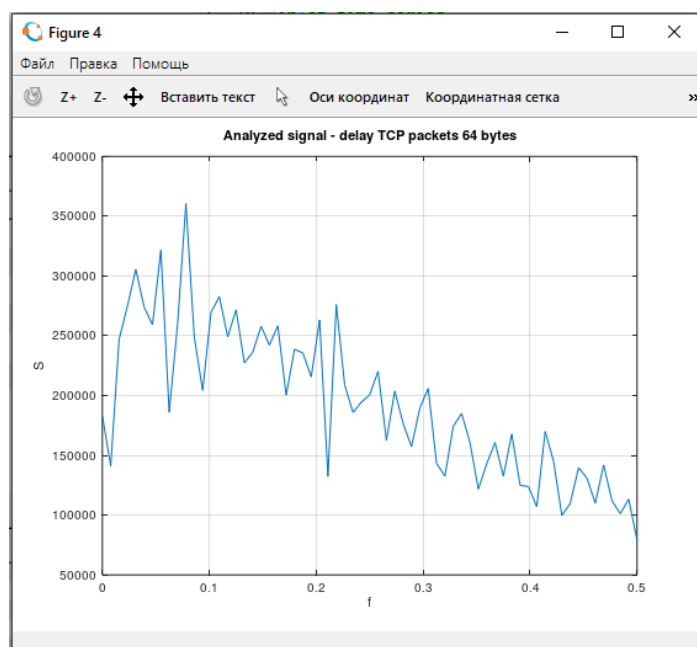


Рисунок 10 - Спектральная плотность временного ряда TCP_d64

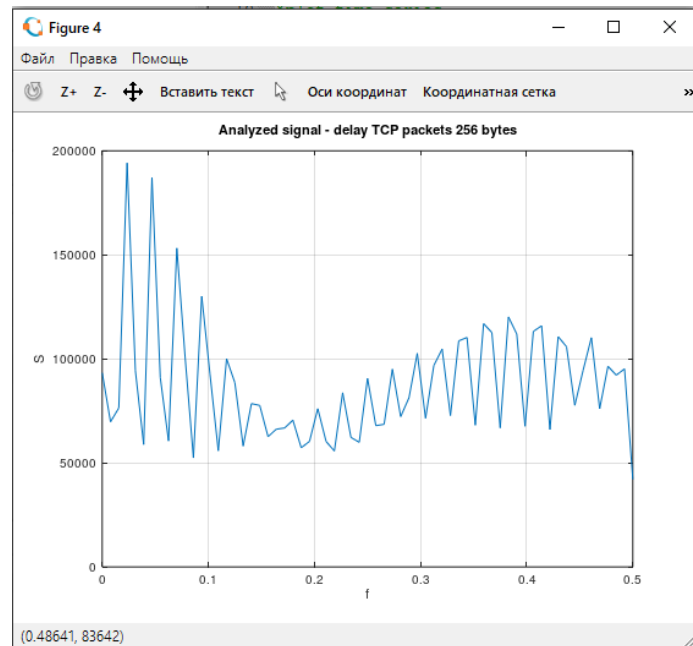


Рисунок 11 - Спектральная плотность временного ряда TCP_d256

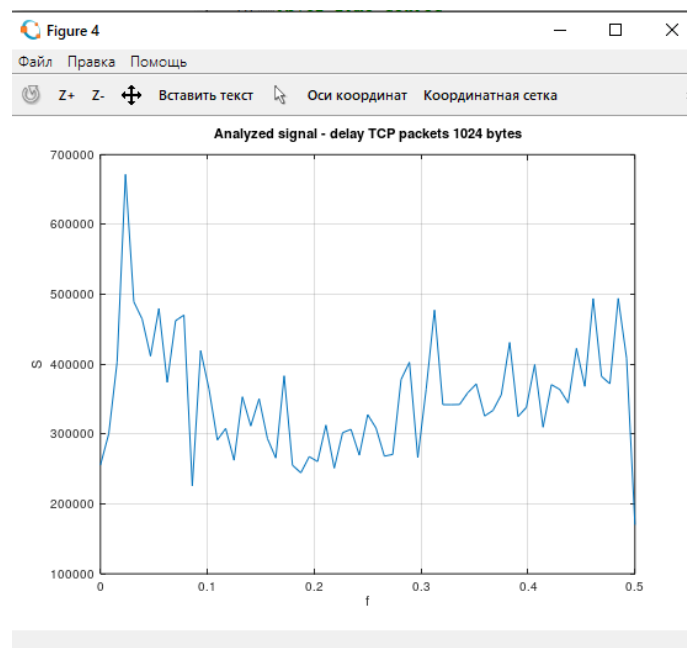


Рисунок 12 - Спектральная плотность временного ряда TCP_d1024

Исследование показателя Херста

В этом разделе оценивается показатель Херста H , который является мерой устойчивости статистического явления или мерой длительности долгосрочной зависимости процесса. Значение $H=0,5$ указывает на отсутствие долгосрочной зависимости. Корреляция между событиями отсутствует. Ряд является случайным, а не фрактальным. Чем ближе значение H к 1, тем выше степень устойчивости долгосрочной зависимости. При $0 \leq H < 0,5$ временной ряд является трендонеустойчивым (антиперсистентным). Он более изменчив, чем случайный ряд, поскольку состоит из частых реверсов спад-подъем. При $0,5 < H \leq 1$ ряд трендоустойчив. Тенденция его изменения может быть спрогнозирована. Показатель Херста связан с другими фрактальными параметрами формулами, показанными в табл. 2.

Таблица 2 – Фрактальные параметры

Параметр	Расчетная формула
Фрактальная размерность D	$D = 2 - H$
Корреляционный параметр β	$\beta = 2(1 - H)$
Спектральный показатель b	$b = 2H - 1$

Для определения показателя Херста в работе используется функция `hurst`. Значения H , полученные для изучаемых временных рядов, показаны в табл. 3.

Таблица 3 – Показатель Херста

Временной ряд	Показатель Херста
TCP_d64	0,65494
TCP_d256	0,62933
TCP_d1024	0,61896

Выводы

В данной работе для реализаций процесса задержки TCP-пакетов выполнен статистический анализ. Получены следующие результаты: изучаемые временные ряды не подчиняются нормальному распределению, имеют медленно убывающую зависимость АКФ, самые мощные гармоники находятся на низких частотах. Все изучаемые временные ряды являются трендоустойчивыми.

Литература

1. Городецкий А.Я., Заборовский В.С. Информатика. Фрактальные процессы в компьютерных сетях. – СПб.: СПбГТУ, 2000. – 102 с.
2. Шепухин О.И., Теняшев А.М., Осин А.В. М. Фрактальные процессы в телекоммуникациях Москва: Радиотехника, - 2003. - 480 с.
3. Ложковський А.Г. Аналіз і синтез систем розподілу інформації в умовах мультисервісного трафіка. Автореферат дисертації. – Одеса, 2010. – 38 с.
4. Шиленко И.С., Бельков Д.В., Лазебная Л.А. Моделирование видео трафика. Материалы XI Международной научно-технической конференции «Информатика, управляющие системы, математическое и компьютерное моделирование» (ИУСМКМ-2020). – Донецк: ДонНТУ, 2020. - С. 17-22.
5. Едемская Е.Н., Бельков Д.В., Родь А.С. Модель UDP-трафика компьютерной сети. Информационное пространство Донбасса: проблемы и перспективы. Материалы IV Республиканской с международным участием научно-практической конференции. Донецк. ГОУ ВПО «ДонНУЭТ». 2021. – С. 133-136.
6. Бельков Д.В., Лазебная Л.А. Самоорганизация TCP-трафика. Информатика и кибернетика № 21.- С. 29-37.
7. Network tools and traffic traces. [Электронный ресурс], 2007. – Режим доступа: <http://www.grid.unina.it/Traffic/Traces/traces.php>

Бельков Д.В., Лосев Д.Д. Анализ трафика беспроводной сети. Исследования современного сетевого трафика показали, что это - самоподобный процесс и его Марковские модели приводят к некорректным результатам. Поэтому важной задачей является дальнейшее исследование трафика. Данная статья представляет результаты статистического анализа временных рядов задержки в сети с GPRS-доступом на стороне отправителя и UMTS-доступом со стороны получателя.

Ключевые слова: самоподобие, трафик, задержка, функция автокорреляции, спектральная плотность, параметр Херста.

Belkov D.V., Losev D.D. Analysis of wireless network traffic. Studies of modern network traffic have shown that this is a self-similar process and its Markov models lead to incorrect results. Therefore, an important task is to further research traffic. This article presents the results of a statistical analysis of the time series of latency in a network with GPRS-access on the sender side and UMTS-access on the receiver side.

Keywords: self-similarity, traffic, delay, autocorrelation function, spectral density, Hearst parameter.

Модель структуры угольного пласта

Д.В. Бельков^{*1}, Д.Г. Рубанов^{*2}

^{*1} к.т.н, доцент, Донецкий национальный технический университет,
belkovdv@list.ru

^{*2} магистрант, Донецкий национальный технический университет,
rubanovdima1990@mail.ru

Бельков Д.В., Рубанов Д.Г. Модель структуры угольного пласта. Эффективным средством борьбы с опасными явлениями в шахтах является обработка угольных пластов жидкостями. Для исследования течения жидкости в неоднородной среде угольного пласта необходимо использовать модели его пустотной структуры. Одна из возможных моделей предложена в статье. В основу разработки модели положены следующие предположения. Трещины и поры соединяются между собой микропорами. Микропоры являются ответвлением от крупных пор. Исследования фильтрации и капиллярного равновесия, визуальные и микроскопические наблюдения, основанные на этих предположениях, привели к представлениям о пустотной структуре угля, как трещиновато-пористой блочной среде.

Ключевые слова: *пустотная структура угольного пласта, самоподобие, модель BML (Biham-Middleton-Levine).*

Введение

Важным средством борьбы с опасными явлениями в шахтах является предварительная обработка угольных пластов жидкостями. На характере движения жидкости сказываются все свойства угольного пласта как фильтрующей среды [1,2]. Поэтому для создания математических моделей течения жидкости в среде угольного пласта необходимо использовать модель его пустотной структуры.

Возможны [3] две модели пустотной структуры. В первой модели трещины и поры всех размеров распределены в угле случайным образом. Их диаметры в направлении движения жидкости значительно изменяются. Трещины и поры соединяются между собой микропорами. Такая модель использована в работе [4], где угольный пласт представляется как сильно сцементированная сплошная среда с трещиновато-пористой структурой и случайно распределёнными фильтрационными параметрами. Во второй модели пустотность представляется в виде разветвленной системы пор и трещин. Микропоры являются ответвлением от крупных пор. Исследования фильтрации и капиллярного равновесия, визуальные и микроскопические наблюдения, основанные на этой модели, привели к представлениям о пустотной структуре угля, как трещиновато-пористой блочной среде [5]. Эта среда имеет иерархический характер. Мелкие блоки вложены в более крупные блоки, а те, в свою очередь, вложены в ещё более крупные блоки. Таким образом, в пустотной структуре угольного пласта на разных масштабных уровнях наблюдается самоподобие (масштабная инвариантность). Это свойство является серьезным аргументом в пользу причисления ископаемых углей к естественным фракталам.

Целью данной статьи является моделирование пустотной структуры угольного пласта. Задачи работы: разработка BML-модели пустотной структуры угольного пласта в среде программирования R, мультифрактальный анализ в среде Octave пустотной структуры угольного пласта, полученной с помощью BML-модели.

BML-модель пустотной структуры угольного пласта

В данной статье для моделирования пустотной структуры угольного пласта предлагается модель BML (Biham-Middleton-Levine). Она известна как самоорганизующаяся модель транспортного потока на основе детерминированного клеточного автомата. Подвижные элементы представлены точками на решетке со случайным стартовым положением.

Модель имеет следующие правила перехода: если ячейка i не пуста, а $(i + 1)$ – пуста, то на следующем шаге ячейка $(i + 1)$ оказывается занята. Значение ячейки i для новой итерации определяется её левым соседом, для которого схема определения состояния аналогична.

Математически правила перехода для каждой клетки можно записать:

1. Ускорение и торможение:

$$v_i(t) = \min(g_i(t-1), 1). \quad (1)$$

2. Перемещение:

$$n_i(t) = n_i(t-1) + v_i(t). \quad (2)$$

Модель BML используется при описании транспортных потоков. Аналогия между структурой транспортного потока и пустотной структурой угольного пласта показана в таблице 1.

Таблица 1 – Аналогия между структурой транспортного потока и пустотной структурой угольного пласта

Структура транспортного потока	Пустотная структура угольного пласта
Подвижный элемент	Текучее вещество, заполняющее пустоты
Свободный поток	Пористая структура
Плотный поток	Трещиновато-пористая структура
Затор	Трещиноватая структура

В транспортном потоке различают три состояния: свободный поток, плотный поток и затор. При свободном движении, когда дорога не загружена, водители придерживаются желаемой скорости, свободно переходя на соседние полосы. Корреляции в движении отдельных автомобилей подавлены, и средняя скорость машин определяется только средним расстоянием между ними. В плотном потоке водители уже не могут свободно маневрировать, и вынуждены согласовывать свою скорость со скоростью потока. Машины мешают друг другу разогнаться до оптимальной скорости, но поток не переходит в состояние затора. В таком потоке формируются большие кластеры автомобилей, которые перемещаются по автомагистрали как единое целое. BML-модель позволяет получать фазы затора, свободного и кооперативного движения в плотном потоке.

Вычислительный эксперимент

Реализованная в среде R BML-модель структуры угольного пласта позволяет в зависимости от значения параметра density (концентрация пустот) получать различные пустотные структуры угольного пласта.

На рис. 1 показаны результаты моделирования пористой структуры при значении density=0,3. Синим цветом показаны поры, заполненные текучим веществом, белым цветом – неувлажненные участки пласта. Показано, что при низкой концентрации пустот подвижность текучего вещества стремится к нулю, т.е. влажность пласта незначительна.

На рис. 2 показаны результаты моделирования трещиновато-пористой структуры при значении density=0,4. Показано, что при значительной концентрации пустот текучее вещество обладает подвижностью достаточной для образования влажных кластеров.

На рис. 3 показаны результаты моделирования трещиноватой структуры при значении density=0,7. Показано, что при высокой концентрации пустот текучее вещество обладает значительной подвижностью по трещинам и пласт имеет высокую влажность.

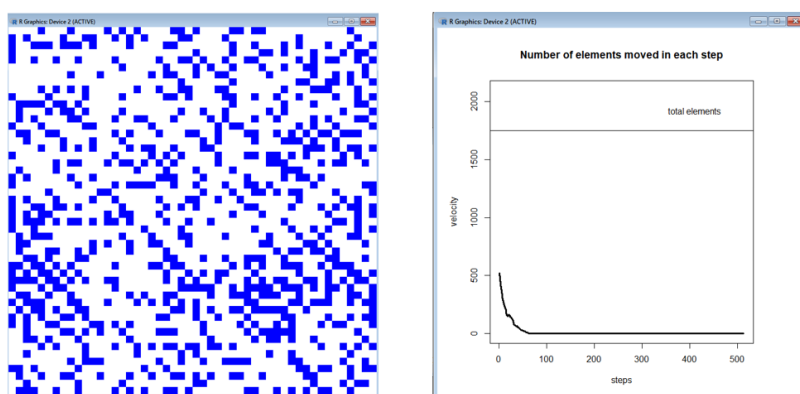


Рисунок 1 – Моделирование пористой структуры при density= 0,3

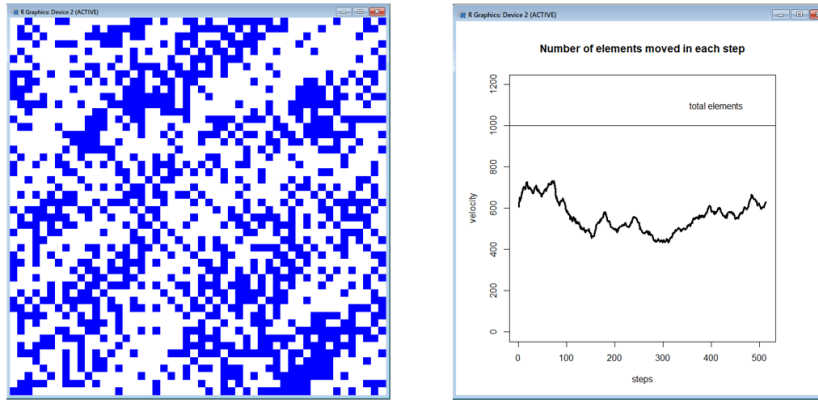


Рисунок 2 – Моделирование пористой структуры при density= 0,3

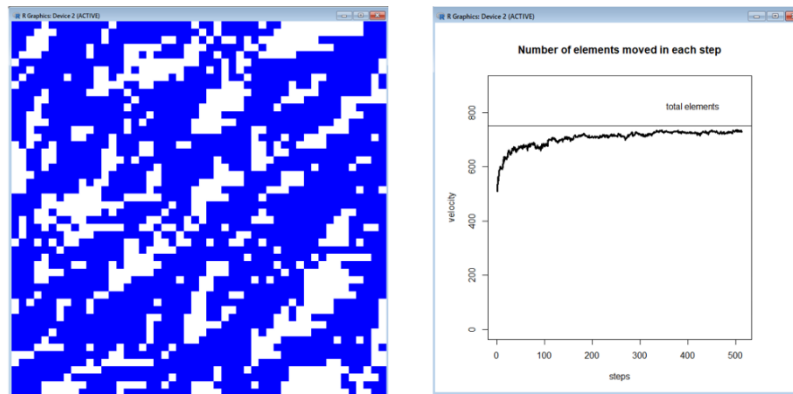


Рисунок 3 – Моделирование трещиноватой структуры при density= 0,7

Мультифрактальный анализ

В работе [4] предложено использовать мультифрактальный подход к описанию структур пространственной самоорганизации углей, поскольку ансамбль микродефектов пласта растет как мультифрактальный (статистически самоподобный) кластер. В данной статье для исследования фрактальности пустотной структуры угольного пласта, полученной с помощью BML-модели, в среде Octave выполнен мультифрактальный анализ трещиновато-пористой и трещиноватой структур. Метод мультифрактального анализа сводится к следующим шагам [4].

1. Для исследуемого ряда $x(i)$, $i=1, 2, \dots, N$ следует выделить флуктуационный профиль, где $\bar{x}$ – среднее значение.

2. Полученные значения $y(i)$ разделяются на непересекающихся сегментов равной длины s . При этом следует учесть, что длина ряда N не всегда кратна шагу s . Поэтому, чтобы не исключать из анализа последний участок, содержащий число элементов, меньшее s , следует повторить процедуру деления, начиная с противоположного конца ряда. В результате получаем $2N_s$ сегментов длины s .

3. Используя метод наименьших квадратов, для профиля $y(i)$, отвечающего каждому из этих сегментов, вычислить локальный тренд, представляющий полином, степень которого обеспечивает заданную точность. Затем для сегментов по формуле (3) определяется дисперсия:

$$F^2(\nu, s) = \frac{1}{s} \sum_{i=1}^s \{y[(\nu-1)s+i] - y_\nu(i)\}^2. \quad (3)$$

Для $\nu = N_s + 1, \dots, 2N_s$ используется формула (4)

$$F^2(\nu, s) = \frac{1}{s} \sum_{i=1}^s \{y[N - (\nu - N_s)s + i] - y_\nu(i)\}^2. \quad (4)$$

Усредняя значения (3), (4), деформированные произвольным показателем q , вычисляются моменты:

$$F_q(s) = \left\{ \frac{1}{2N_s} \sum_{\nu=1}^{2N_s} [F^2(\nu, s)]^{q/2} \right\}^{1/q}. \quad (5)$$

При $q \rightarrow 0$ вместо (5) нужно использовать формулу (6)

$$F_0(s) = \exp \left\{ \frac{1}{4N_s} \sum_{v=1}^{2N_s} \ln[F^2(v, s)] \right\}. \quad (6)$$

При положительных показателях q основной вклад в сумму по v дают сегменты, отвечающие большим отклонениям $F^2(v, s)$, а при отрицательных доминируют вклады малых флуктуаций.

Самоподобное поведение, означающее наличие далекодействующих степенных корреляций, представляется степенной зависимостью моментов (5), (6):

$$F_q(s) \propto s^{h(q)}. \quad (7)$$

При фиксированном значении q эта зависимость в двойных логарифмических координатах представляет собой прямую линию. При больших значениях s зависимость $F_q(s)$ не имеет статистической информативности, поскольку число сегментов N_s , используемых в процедуре усреднения (3), (4), становится малым. При обработке ряда нужно исключить значения $s > N/4$, а также малые сегменты ($s < 6$), для которых теряет статистическую достоверность усреднение (3), (4) по каждому из сегментов.

Если ряд экспериментальных данных является монофракталом, то обобщенный показатель Херста $h(q)$ в равенстве (7) принимает единственное значение $h(q) = H$. В случае мультифрактального ряда показатель h зависит от q . Для стационарных рядов $h(0)$ определяет топологическую размерность пространства, содержащего фрактальное множество, $h(1)$ – меру его беспорядка, $h(2)$ – показатель дальних корреляций.

В рамках стандартной фрактальной идеологии используется не только обобщенный показатель Херста $h(q)$, но массовый показатель $\tau(q)$ и мультифрактальный спектр $f(\alpha)$. Этот переход достигается преобразованиями Лежандра:

$$\tau(q) = qh(q) - 1, \quad (8)$$

$$\alpha = \tau'(q), f(\alpha) = q\alpha - \tau(q). \quad (9)$$

Для монофрактальных объектов функция $\tau(q)$ является прямолинейной зависимостью, которая с переходом к мультифракталам выгибается, сохраняя прямолинейные участки. Наиболее ярко строение самоподобного объекта представляется формой мультифрактального спектра $f(\alpha)$, ширина которого дает набор фрактальных размерностей. Для монофракталов функция $f(\alpha)$ имеет δ -образную форму.

Программная реализация метода мультифрактального анализа выполнена в среде Octave.

На рис. 4-7 показаны результаты, полученные при исследовании трещиноватой структуры для $\text{density} = 0,7$.

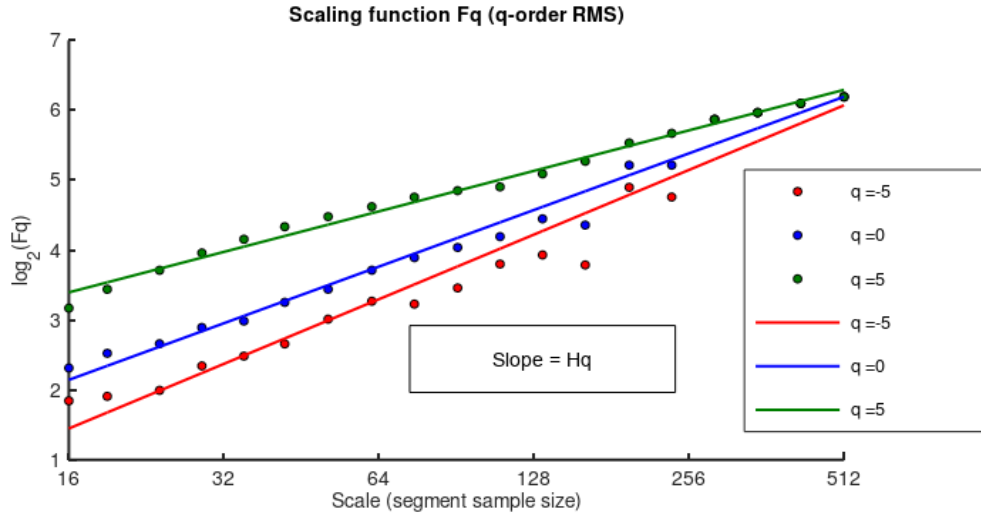


Рисунок 4 – Функция скейлинга для $\text{density} = 0,7$

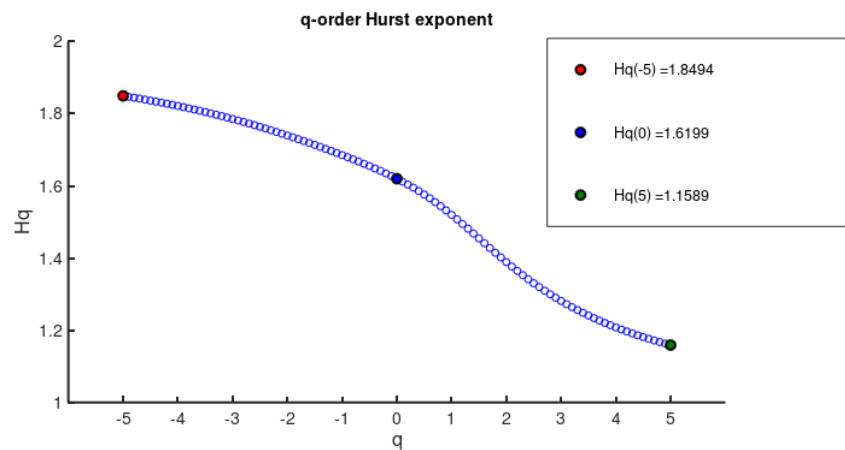


Рисунок 5 – Экспонента Херста для density= 0,7

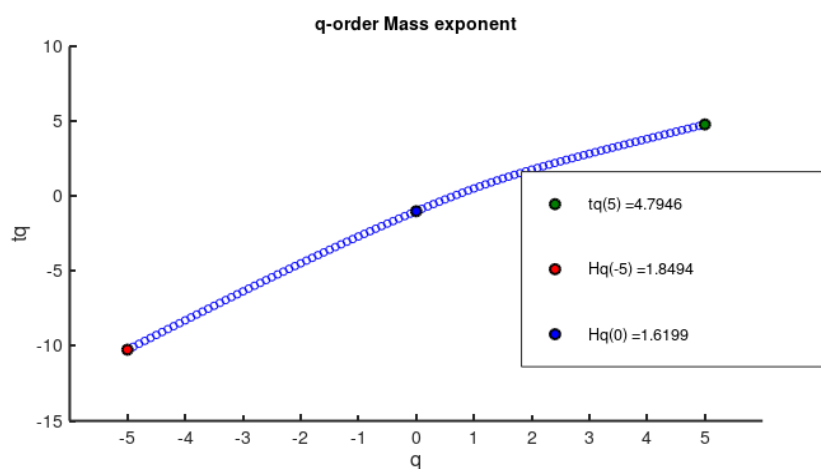


Рисунок 6 – Массовая экспонента для density= 0,7

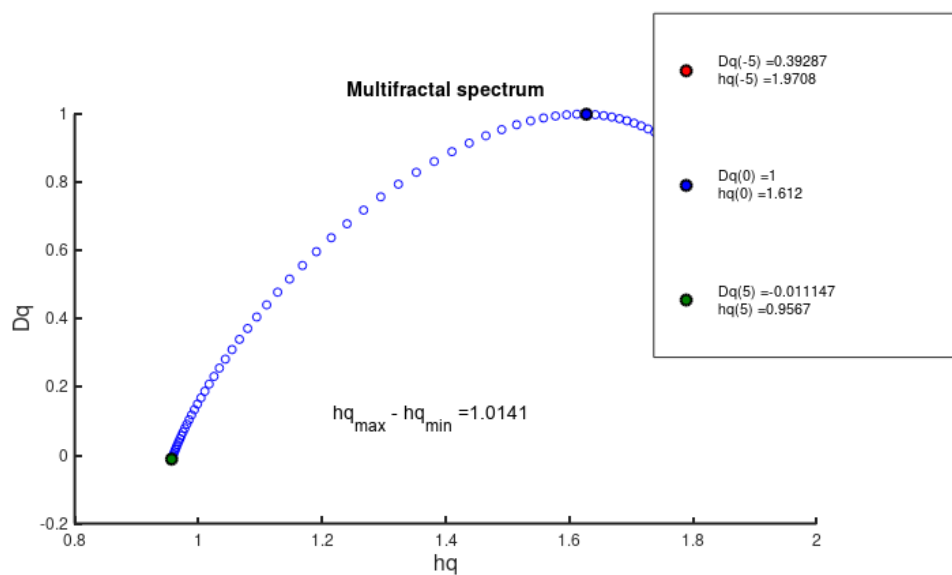


Рисунок 7 – Мультифрактальный спектр для density= 0,7

На рис. 8-11 показаны результаты, полученные при исследовании трещиновато-пористой структуры для density= 0,4.

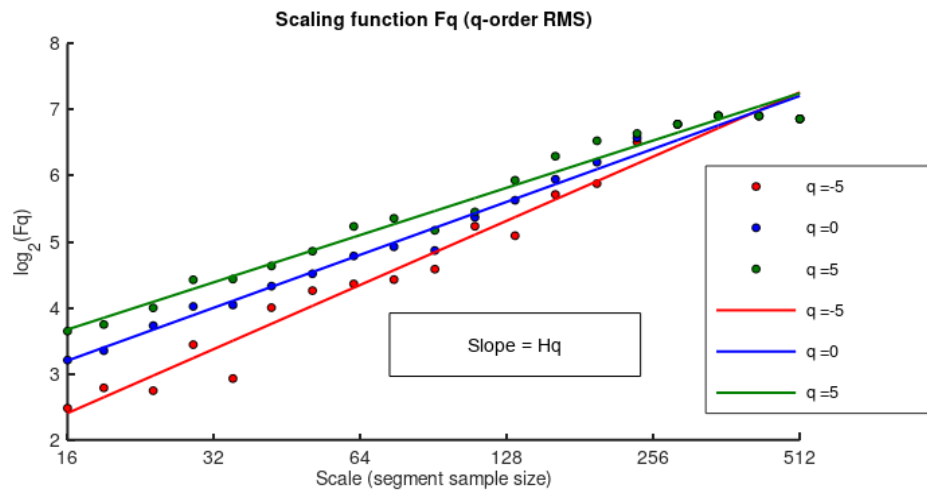


Рисунок 8 – Функция скейлинга для density= 0,4

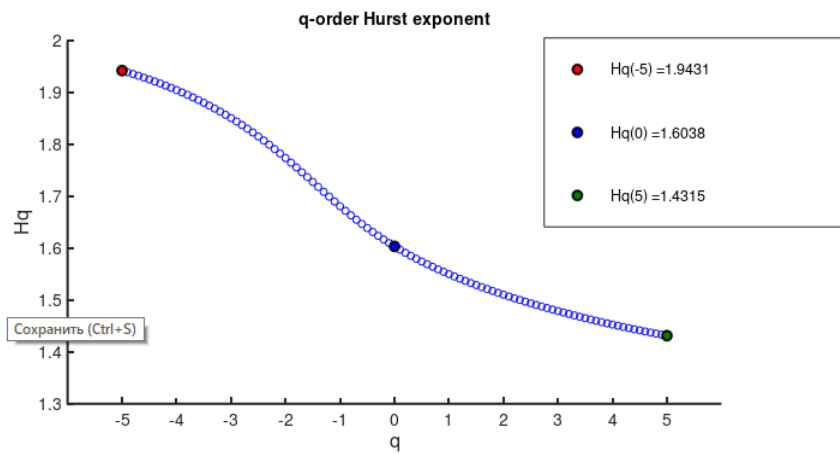


Рисунок 9 – Экспонента Херста для density= 0,4

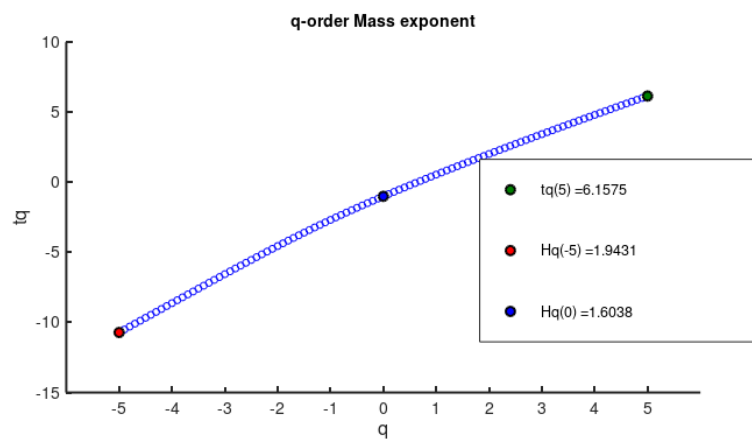


Рисунок 10 – Массовая экспонента для density= 0,4

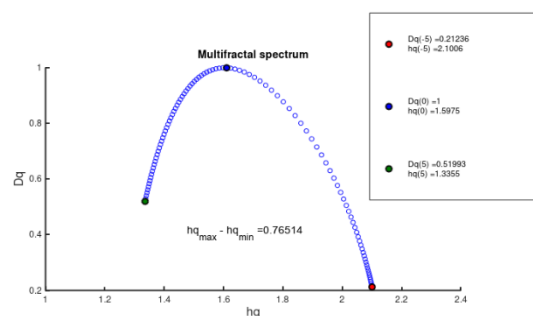


Рисунок 11 – Мультифрактальный спектр для density= 0,4

Выводы

Разработана BML-модель пустотной структуры угольного пласта. Программная реализация модели выполнена в среде программирования R. При работе программы в зависимости от параметра density получены следующие результаты: на интервале изменения density от 0,1 до 0,3 пласт имеет пористую структуру, на интервале изменения density от 0,4 до 0,6 пласт имеет кластерную трещиновато-пористую структуру, на интервале изменения density от 0,7 до 0,9 пласт имеет трещиноватую структуру. Выполнен мультифрактальный анализ пористой и трещиновато-пористой структур угольного пласта, полученных с помощью BML-модели.

Литература

1. Прогрессивные технологические решения по комплексному освоению ресурсного потенциала угольных месторождений: монография / В.В. Мельник, В.Н. Павлыш, С.С. Гребенкин и др. ; // под общ. ред. В.В. Мельника, В.Н. Павлыша; Донецк : ВИК, 2015. – 340с.
2. Захаров В.Н., Малинникова О.Н. Исследование структурных особенностей углей выбросоопасных пластов [Электронный ресурс]. – Режим доступа: <https://cyberleninka.ru/article/n/issledovanie-strukturnyh-osobennostey-ugley-vybrosopasnyh-plastov>
3. Булат А.Ф., Дырда В.И. Фракталы в геомеханике. /Монография. – Киев: «Наукова думка», 2005. – 358 с.
4. Лазебная Л.А. Принципы формирования математической модели процесса гидравлического воздействия на угольный пласт с учетом особенностей его структуры и управления параметрами. /Информатика и кибернетика, № 2 (20), 2020, Донецк, ДонНТУ. – С. 40–47.
5. Астахов А.В., Белый А.А., Минаев В.И., Подыногина А.В., Широчин Д.Л. Применение фрактального подхода и теории перколяции при исследованиях ископаемых углей. [Электронный ресурс]. – Режим доступа: <https://cyberleninka.ru/article/n/primenenie-fraktalnogo-podhoda-i-teorii-perkolyatsii-pri-issledovaniyah-iskopaemyh-ugley>

Бельков Д.В., Рубанов Д.Г. Модель структуры угольного пласта. Эффективным средством борьбы с опасными явлениями в шахтах является обработка угольных пластов жидкостями. Для исследования течения жидкости в неоднородной среде угольного пласта необходимо использовать модели его пустотной структуры. Одна из возможных моделей предложена в статье. В основу разработки модели положены следующие предположения. Трещины и поры соединяются между собой микропорами. Микропоры являются ответвлением от крупных пор. Исследования фильтрации и капиллярного равновесия, визуальные и микроскопические наблюдения, основанные на этих предположениях, привели к представлениям о пустотной структуре угля, как трещиновато-пористой блочной среде.

Ключевые слова: пустотная структура угольного пласта, самоподобие, модель BML (Biham-Middleton-Levine).

Belkov D.V., Rubanov D.G. Model of the structure of the coal seam. An effective means of combating hazards in mines is the treatment of coal seams with liquids. To study the flow of a liquid in a heterogeneous environment of a coal seam, it is necessary to use models of its hollow structure. One of the possible models is proposed in the article. The development of the model is based on the following assumptions: Cracks and pores are interconnected by micropores. Micropores are an offshoot of large pores. Studies of filtration and capillary equilibrium, visual and microscopic observations based on these assumptions, led to ideas about the hollow structure of coal as a cracked-porous block medium.

Key words: hollow structure of the coal seam, self-similarity, BML model (Biham-Middleton-Levine).

Исследование видео трафика компьютерной сети

Д.В. Бельков^{*1}, М.И. Трушкин^{*2}

^{*1} к.т.н, доцент, Донецкий национальный технический университет,
belkovdv@list.ru

^{*2} магистрант, Донецкий национальный технический университет,
kamitrushkin@gmail.com

Бельков Д.В., Трушкин М.И. Исследование видео трафика компьютерной сети. Анализ современного трафика показал самоподобный характер процессов в компьютерных сетях. Их Марковские модели приводят к некорректным результатам. В связи с этим актуальной задачей является исследование реального трафика сети с целью выявления его характерных особенностей. Данная статья представляет результаты статистического анализа временных рядов видео трафика компьютерной сети.

Ключевые слова: самоподобие, видео трафик, функция автокорреляции, спектральная плотность, параметр Херста.

Введение

Анализ реального трафика компьютерных сетей показывает, что трафик различных видов услуг является самоподобным (фрактальным). Известно множество экспериментальных исследований в этой области, однако число работ, посвященных изучению фрактальной природы трафика видео сервисов невелико. В настоящее время отсутствуют систематизированные исследования воздействия самоподобного видео трафика на качество обслуживания каждого подписчика таких сервисов. Анализ данной проблемы является важным, поскольку при самоподобном трафике в системах передачи видео качество обслуживания, как правило, ухудшается по сравнению с тем, что наблюдается в случае Пуассоновского трафика [1,2,3]. Фрактальный трафик сохраняет поведение и внешние признаки на разных временных масштабах. Поэтому методы моделирования и расчета сетевых систем, основанные на использовании Пуассоновских потоков, не дают точной картины того, что происходит в сети. Трафик имеет большое количество выбросов при небольшом среднем уровне передачи. Это явление увеличивает потери и задержки пакетов при прохождении трафика через узлы сети. Пакеты поступают на узел не по отдельности, а пачкой, что может приводить к их потерям из-за ограниченности буфера, рассчитанного по классическим методам теории телетрафика [4,5].

Появление новых свойств сетевого трафика, необходимость обеспечения высокого качества обслуживания видео приложений, делают актуальным исследование реального видео трафика.

Целью данной работы является исследование структуры видео трафика, направленное на выявление его характерных особенностей. В работе решаются следующие задачи: оцениваются плотности распределения, автокорреляционные функции (АКФ), энергетические спектры изучаемых временных рядов видео трафика, определяется степень фрактальности (показатель Херста). Вычислительный эксперимент выполнен в среде Octave.

Исходные данные

В работе исследуется поток фреймов фильма «Парк Юрского периода» (сжатое видео стандарта H.263). Первоначально видео данные были обработаны программой bttvgrab (Version 0.15.10) и представлены в виде фреймов с частотой 25 frames/sec в QCIF формате. С помощью кодера tmn (Version 2.0/3.2) был получен поток сжатого видео стандарта H.263 [6].

Изучаемые временные ряды представляют собой измерения количества байт последовательно поступающих фреймов. Исходные данные показаны на рис. 1-4. В первом случае (ряд parkH16) трафик имеет постоянную скорость 16 kbit/sec, во втором (ряд parkH64) - 64 kbit/sec. Третий ряд (parkH256) – передача со скоростью 256 kbit/sec, четвертый ряд (parkVBR) – передача без установки постоянной скорости.

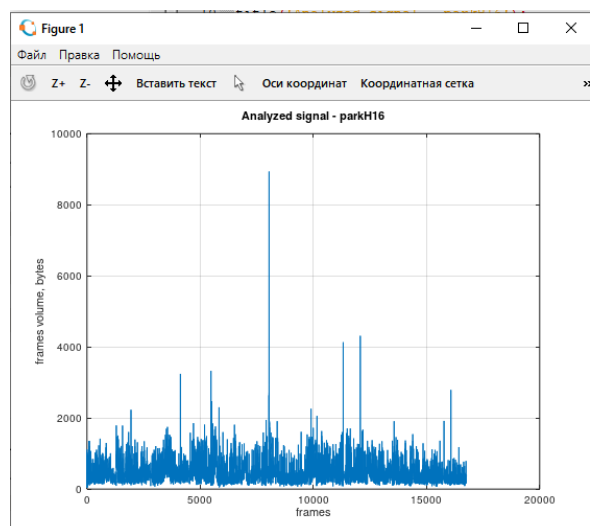


Рисунок 1 – Временной ряд parkH16

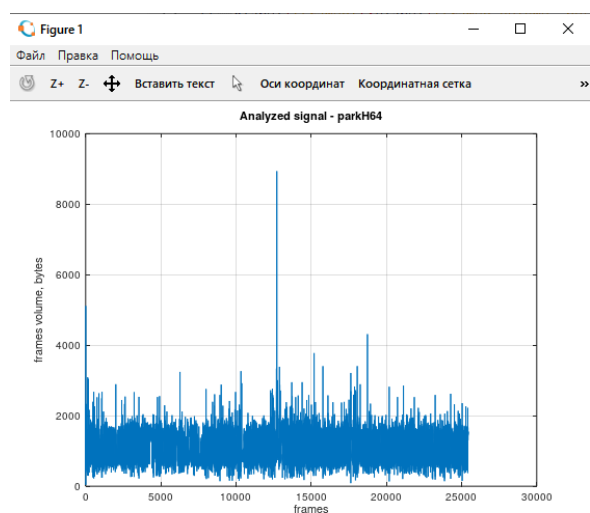


Рисунок 2 - Временной ряд parkH64

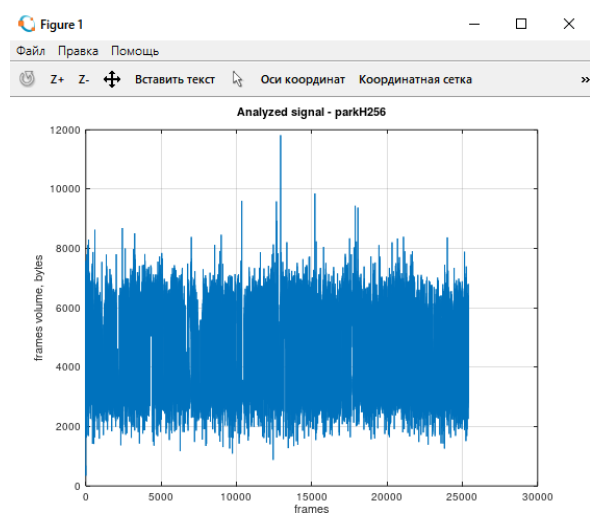


Рисунок 3 - Временной ряд parkH256

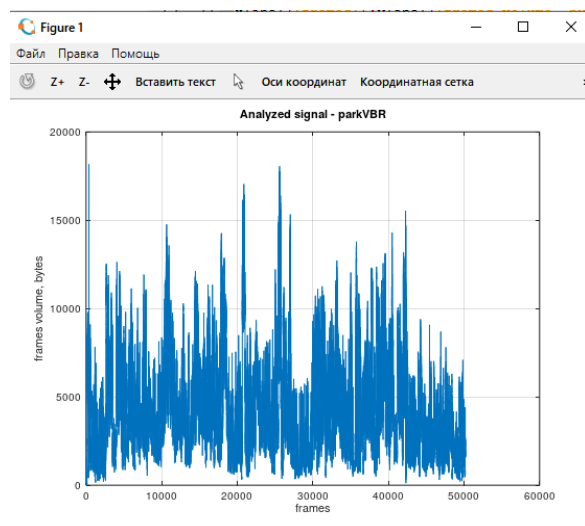


Рисунок 4 – Временной ряд parkVBR

Анализ плотности распределения

В работе выполнен анализ плотности распределения. Оценка выполняется с помощью гистограмм относительных частот, показанных на рис. 5-8. На изображениях видно, что изучаемые ряды не подчиняются нормальному распределению.

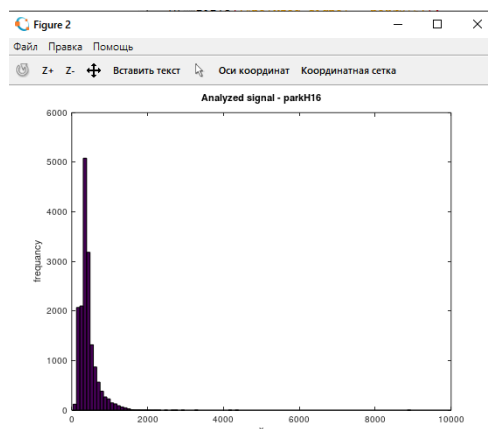


Рисунок 5 – Плотность распределения временного ряда parkH16

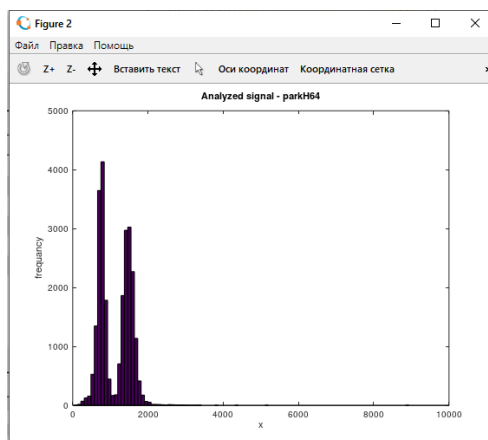


Рисунок 6 – Плотность распределения временного ряда parkH64

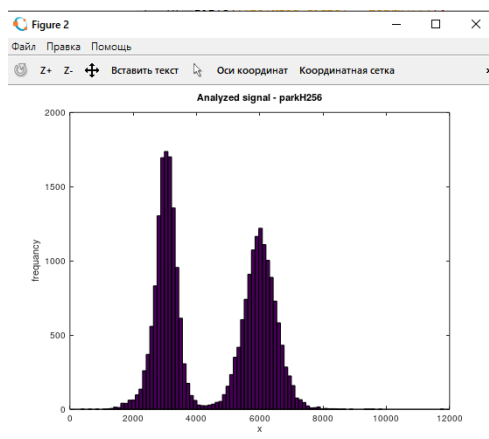


Рисунок 7 – Плотность распределения временного ряда parkH256

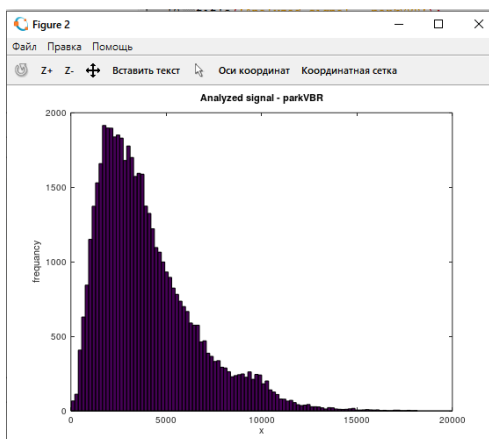


Рисунок 8 – Плотность распределения временного ряда parkVBR

Анализ автокорреляционных функций

Одна из задач исследования – определить обладают ли временные ряды медленно убывающей зависимостью или быстро убывающей зависимостью. Процесс X обладает медленно убывающей зависимостью (МУЗ), если его АКФ подчиняется условию $r(k) \sim k^{-\beta} \cdot L_1(k), k \rightarrow \infty$, где $0 < \beta < 1$ и L_1 – медленно меняющаяся на бесконечности функция: $\lim_{t \rightarrow \infty} \frac{L_1(tx)}{L_1(t)} = 1$ для всех x . В процессах с быстро убывающей зависимостью (БУЗ) АКФ спадает экспоненциально: $r(k) \sim e^{-k}, k \rightarrow \infty$. Автокорреляционные функции временных рядов приведены на рис. 9-12. Они показывают, что изучаемые ряды обладают медленно убывающей зависимостью.

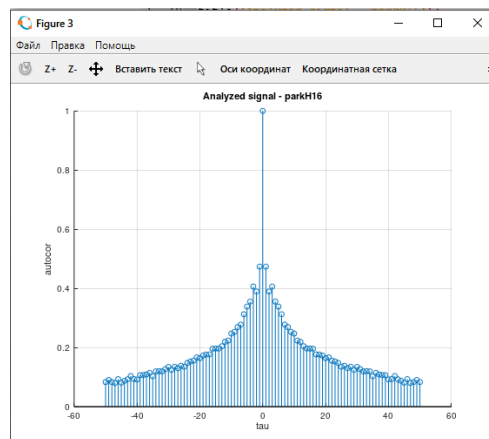


Рисунок 9 – АКФ временного ряда parkH16

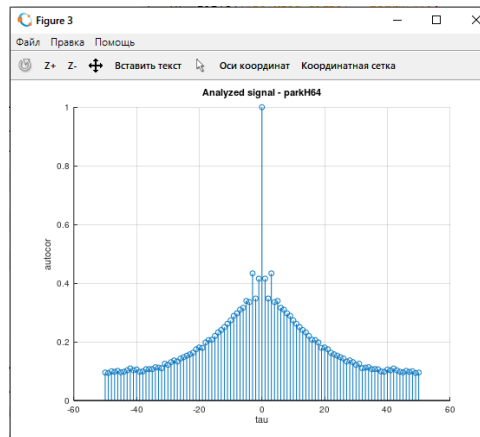


Рисунок 10 – АКФ временного ряда parkH64

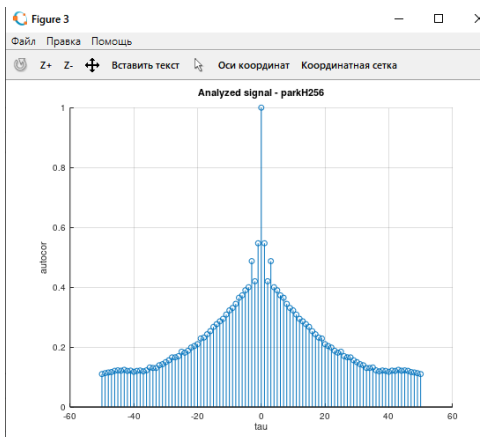


Рисунок 11 – АКФ временного ряда parkH256

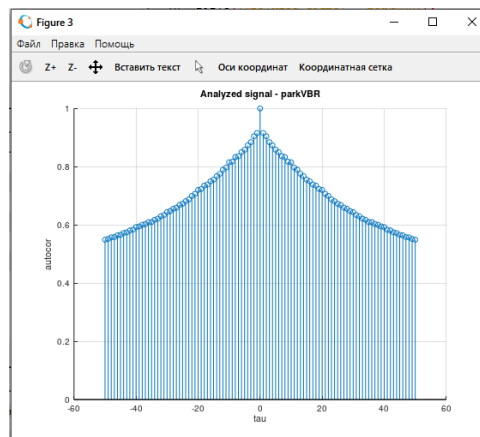


Рисунок 12 – АКФ временного ряда parkVBR

Спектральный анализ

В работе выполнен анализ энергетических спектров изучаемых временных рядов. В частотной области МУЗ приводит к степенному закону уменьшения спектральной плотности процесса. Процесс X обладает МУЗ, если для спектральной плотности выполняется условие, где $f \rightarrow 0$, $i = \sqrt{-1}$, $0 < b < 4$ и L_2 - медленно изменяющаяся в нуле функция. Процесс с МУЗ обладает спектральной плотностью с особенностью в нуле: спектральная плотность стремится к бесконечности, по мере того как частота f стремится к нулю. Энергетические спектры временных рядов представлены на рис. 13-16. Самые мощные гармоники ряды имеют на низких частотах.

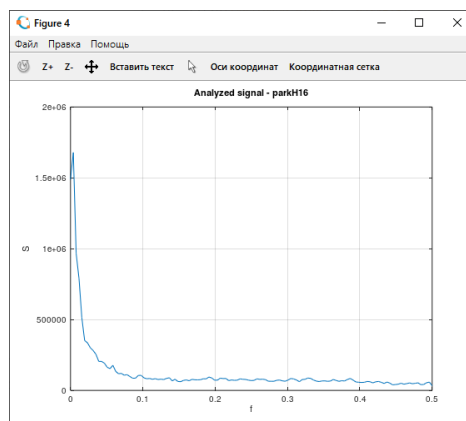


Рисунок 13 – Спектральная плотность временного ряда parkH16

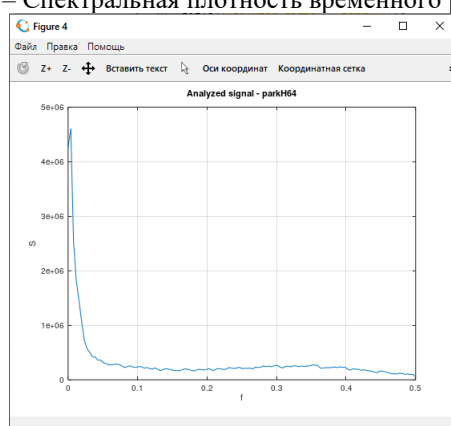


Рисунок 14 – Спектральная плотность временного ряда parkH64

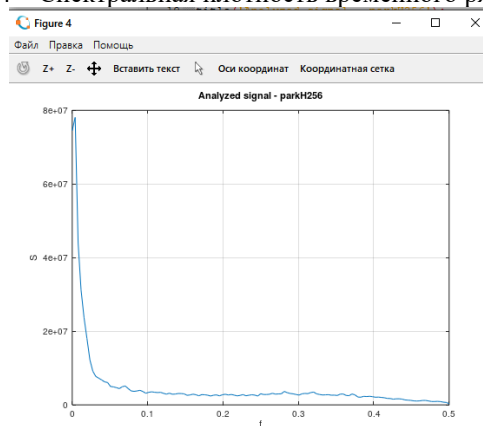


Рисунок 15 – Спектральная плотность временного ряда parkH256

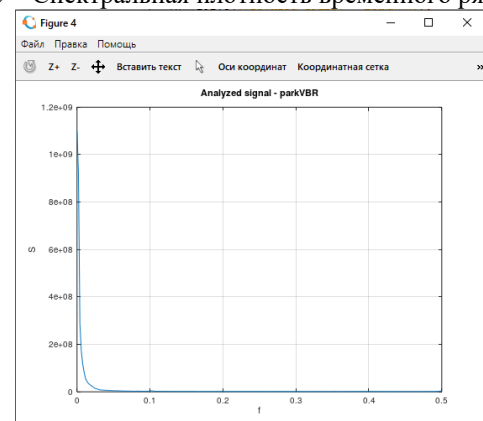


Рисунок 16 – Спектральная плотность временного ряда parkVBR

Анализ показателя Херста

Для определения степени фрактальности изучаемых рядов требуется найти величину показателя Херста H . Значение $H=0,5$ указывает на отсутствие долгосрочной зависимости среди членов ряда. В этом случае корреляции между событиями нет. Ряд является случайным, а не фрактальным. При $0 \leq H < 0,5$ временной ряд не имеет устойчивого тренда и называется антиперсистентным. Такой ряд более изменчив, чем случайный ряд. При $0,5 < H \leq 1$ ряд имеет устойчивый тренд и называется персистентным [6]. Значения H , полученные в результате вычислительного эксперимента, показаны в табл. 1.

Таблица 1 – Показатель Херста

Временной ряд	Показатель Херста H
parkH16	0,72411
parkH64	0,73487
parkH256	0,74013
parkVBR	0,80336

Выводы

В данной работе выполнен статистический анализ видео трафика компьютерной сети. Получены следующие результаты: изучаемые временные ряды не подчиняются нормальному распределению, имеют медленно убывающую зависимость АКФ, самые мощные гармоники находятся на низких частотах. Все изучаемые временные ряды являются персистентными. Наибольшая степень фрактальности наблюдается при передаче видео трафика без установки постоянной скорости.

Литература

1. Урьев Г.А. Исследование фрактальных свойств потоков трафика реального времени и оценка их влияния на характеристики обслуживания телекоммуникационных сетей. Автореферат диссертации. Москва. – 2007. – 21 с.
2. Шиленко И.С., Бельков Д.В., Лазебная Л.А. Моделирование видео трафика. Материалы XI Международной научно-технической конференции «Информатика, управляющие системы, математическое и компьютерное моделирование» (ИУСМКМ-2020). – Донецк: ДонНТУ, 2020. - С. 17-22.
3. Едемская Е.Н., Бельков Д.В. Исследование сетевого трафика с помощью функции Херста. Информатика и кибернетика, № 2, 2015, С. 39-46.
4. Едемская Е.Н., Бельков Д.В., Зензеров В.И. Мультифрактальный анализ трафика беспроводной сети. Материалы V Всероссийской научно-практической конференции «Информационные и инновационные технологии в науке и образовании». Таганрог, 2020, С. 301-306.
5. Бельков Д.В., Едемская Е.Н. Самоорганизация UDP-трафика компьютерной сети. Современные тенденции развития и перспективы внедрения инновационных технологий в машиностроении, образовании и экономике. VIII Международная научно-практическая конференция. Материалы и доклады Азов, 2021. Т.7. № 1 (6) –С. 80 -87.
6. MPEG-4 and H.263 Video Traces for Network Performance Evaluation. [Электронный ресурс], 2008. – Режим доступа: <http://www.tkn.ee.tu-berlin.de/research/trace/trace.html>.

Бельков Д.В., Трушкин М.И. Исследование видео трафика компьютерной сети. Анализ современного трафика показал самоподобный характер процессов в компьютерных сетях. Их Марковские модели приводят к некорректным результатам. В связи с этим актуальной задачей является исследование реального трафика сети с целью выявления его характерных особенностей. Данная статья представляет результаты статистического анализа временных рядов видео трафика компьютерной сети.

Ключевые слова: самоподобие, видео трафик, функция автокорреляции, спектральная плотность, параметр Херста.

Belkov D.V., Trushkin M.I. Study of video traffic of a computer network. Analysis of modern traffic showed the self-similar nature of the processes in computer networks. Their Markov models lead to incorrect results. In this regard, an urgent task is to study the real traffic of the network in order to identify its characteristic features. This article presents the results of statistical analysis of time series of video traffic of a computer network.

Keywords: self-similarity, video traffic, autocorrelation function, spectral density, Hearst parameter.

Один подход к формированию полигармонических колебаний в вибрационных машинах с инерционным возбуждением

В.Н. Беловодский^{*1}, С.Л. Букин^{*1}

^{*1} к.т.н, доцент, Донецкий национальный технический университет,
v.belovodskiy@gmail.com, orcid.org: 0000-0001-8345-7492, SPIN-код: 6679-3364

Беловодский В.Н., Букин С.Л. Один подход к формированию полигармонических колебаний в вибрационных машинах с инерционным возбуждением. Вибрационная техника широко используется для реализации и интенсификации технологических процессов в различных отраслях промышленности. И нередко, использование в них полигармонической вибрации, оказывается более эффективным по сравнению с гармонической. В работе предложен один подход к формированию полигармонических вибраций за счет излома валов в трансмиссии центробежного привода. Рассмотрена математическая модель одномассовой вибрационной машины с шарниром Гука в ее приводе и для одного из вариантов ее параметров проведена оценка полигармоничности в законах перемещений и ускорений рабочего органа.

Ключевые слова: вибрационная машина, инерционный привод, трансмиссия, шарнир Гука, полигармонические колебания.

Введение

Вибрационная техника широко используется для реализации и интенсификации технологических процессов в различных отраслях промышленности, в частности при изготовлении широкой номенклатуры сборных железобетонных элементов, подготовке строительных смесей, уплотнении грунтов, обогащении разнообразного сырья, подготовке лекарственных препаратов, транспортирования разнообразных грузов и т.д. И, нередко, для этих процессов полигармоническая вибрация оказывается более эффективной по сравнению с гармонической. Известными примерами ее формирования в механизмах с гармоническим возбуждением являются супергармонический привод, реализующий для этих целей неравномерность вращения двигателя, асимметричные виброплощадки с односторонними буферными ограничителями хода, вибромашины с нелинейными упругими связями, работающие в режиме комбинационных резонансов. Следует, также, отметить и принципиально новые «многочастотные» машины, разработанные компанией «Kroosh Technologies Ltd.» (Израиль) [1]. После применения грохотов нового типа на операциях разделения по крупности трудногрохотимых материалов [2], запатентованный принцип возбуждения многочастотных (полигармонических) колебаний с помощью моногармонического вибровозбудителя был использован и в ряде конструкций вибромашин разнообразного производственного назначения [3]. Однако, несмотря на достигнутые высокие показатели качества их конструкции отличаются сложностью, что отражается на стоимости и повышает затраты при эксплуатации. Поэтому актуальность поиска простых технических решений возбуждения полигармонических колебаний сохраняется.

Анализируя возможность возбуждения колебаний, отличных от гармонических, авторов заинтересовала особенность работы одинарного шарнира Гука в трансмиссии машины. Оказывается, что в таком приводе при перекосе ведомого вала относительно ведущего наблюдается неравномерность вращения ведомого вала при равномерном вращении ведущего [4], причём угловая скорость вращения ведомого вала $\dot{\phi}(t)$ описывается соотношением

$$\dot{\phi}(t) = \frac{\omega_d \cdot \cos \beta}{1 - \sin^2 \beta \cdot \cos^2(\omega_d \cdot t)}, \quad (1)$$

где ω_d - угловая скорость ведущего вала; β - угол разворота валов.

Если возбуждение полигармонических колебаний рабочего органа машины с таким виброприводом будет установлено, то существенным достоинством такого подхода может стать предельная простота конструкции со всеми вытекающими последствиями, в частности, возможностью реализации такой схемы не только в новых вибромашинах, но и в конструкциях действующих (после небольшой модернизации некоторых их узлов). Первый шаг в этом направлении был сделан в статье [5]. В данной работе предпринимается попытка несколько

расширить проведенные исследования и изучить возможность генерации таких колебаний в линейной вибромашине с изломами валов в узле трансмиссии.

1 Принципиальная схема, уравнения движения

Конструктивная схема рассматриваемой вибрационной машины показана на рис. 1. Вертикальные движения ее рабочего органа описываются уравнением

$$M\ddot{x} + \mu c\dot{x} + cx = m_v r \dot{\varphi}^2 \sin[\varphi(t)], \quad (2)$$

где M - масса рабочего органа совместно с вибровозбудителем; m_v - масса неуравновешенных частей вибровозбудителя; r - эксцентриситет дебалансного вибровозбудителя; c - суммарный коэффициент жёсткости опорных виброизоляторов в направлении оси X ; μ - коэффициент сопротивления опорных виброизоляторов.

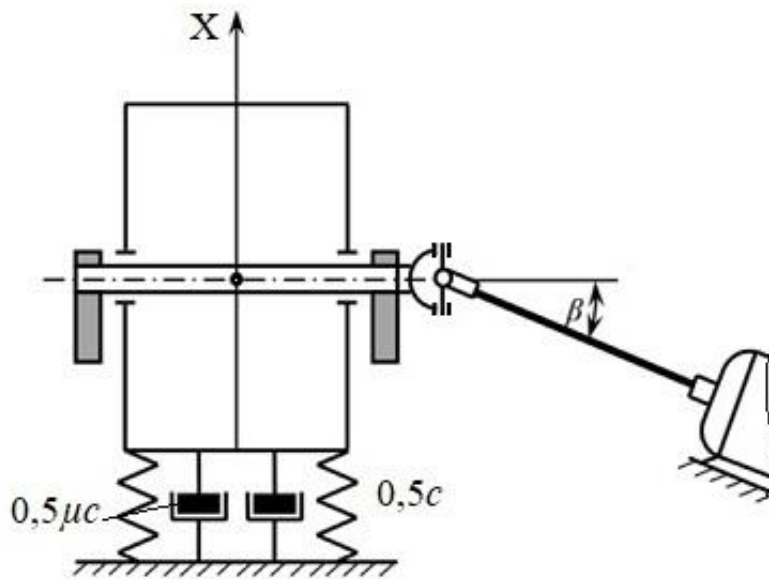


Рисунок 1 – Конструктивная схема вибрационной машины с шарниром Гука в трансмиссии центробежного вибропривода и несоосным соединением валов.

Здесь приняты традиционные допущения:

- рабочий орган является абсолютно твердым телом;
- ось вращения дебалансов вибровозбудителя горизонтальна;
- восстанавливающая сила упругих элементов подчиняется линейной зависимости;
- учёт внутренних сопротивлений в упругих элементах основывается на гипотезе вязкого трения;
- влиянием технологической нагрузки на колебания рабочего органа пренебрегаем.

Согласно (1), вращение дебалансного вала описывается интегралом

$$\varphi(t) = \int_0^t \frac{\omega_d \cdot \cos \beta}{1 - \sin^2 \beta \cdot \cos^2(\omega_d \cdot t_1)} dt_1, \quad (3)$$

где t_1 - переменная интегрирования

Выполним в уравнении (2) следующие замены: $x = \xi \cdot 10^{-3}$ м, $\omega_0 \cdot t = \tau$, где $\omega_0 = (c/M)^{1/2}$ - собственная частота колебаний вибромашины и перейдём в нем к дифференцированию по безразмерному времени, учитывая, что в этом случае

$$\frac{d\varphi}{d\tau} = \frac{\omega \cos \beta}{1 - \sin^2 \beta \cdot \cos^2(\omega \cdot \tau)}. \quad (4)$$

В результате этого, получим

$$M \omega_0^2 \cdot 10^{-3} \frac{d^2 \xi}{d\tau^2} + \mu c \omega_0 \cdot 10^{-3} \frac{d\xi}{d\tau} + c \cdot 10^{-3} \xi = m_v r \left[\frac{\omega_d \cos \beta}{1 - \sin^2 \beta \cdot \cos^2(\omega_d \cdot \tau / \omega_0)} \right]^2 \cdot \sin[\varphi(\tau)], \quad (5)$$

где, согласно (3),

$$\varphi(\tau) = \int_0^\tau \frac{\omega \cdot \cos \beta}{1 - \sin^2 \beta \cdot \cos^2(\omega \cdot \tau_1)} d\tau_1. \quad (6)$$

Разделив, теперь, обе части уравнения (5) на $M \cdot \omega_0^2 \cdot 10^{-3}$, получим

$$\frac{d^2 \xi}{d\tau^2} + b \frac{d\xi}{d\tau} + \xi = P_0 \cdot \left[\frac{\cos \beta}{1 - \sin^2 \beta \cdot \cos^2(\omega \cdot \tau)} \right]^2 \cdot \sin[\varphi(\tau)], \quad (7)$$

где $b = \mu \cdot \omega_0$, $P_0 = 10^3 m_v r \omega^2 / M$, $\omega = \omega_d / \omega_0$.

При проведении динамического анализа модели (7) в качестве основных исходных данных примем $b=0.1$, $P_0=54$, $\omega=3$, что соответствует реальным инерционным вибромашинам технологического назначения. Величину разворота валов, для определенности, будем считать равной $\beta=\pi/4$.

2 Предварительный анализ

Наибольший интерес представляют следующие динамические величины колебательной системы:

- угловая скорость вращения дебалансного вала вибровозбудителя, определяемая выражением (4);
- амплитуда вынуждающей силы, равная

$$P_a = P_0 \left[\frac{\cos \beta}{1 - \sin^2 \beta \cdot \cos^2(\omega \cdot \tau)} \right]^2; \quad (8)$$

- угол поворота дебалансов (6);
- вынуждающая сила

$$P(\tau) = P_0 \cdot \left[\frac{\cos \beta}{1 - \sin^2 \beta \cdot \cos^2(\omega \cdot \tau)} \right]^2 \cdot \sin[\varphi(\tau)]. \quad (9)$$

При принятых значениях параметров их графики представлены на рис. 2 и 3, а на рис. 4 и 5 показан закон изменения угла поворота дебалансов и вынуждающей силы.

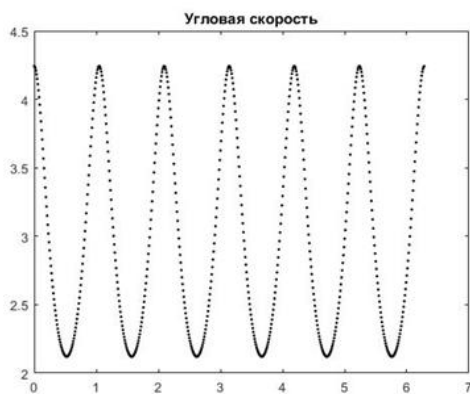


Рисунок 2 – График угловой скорости $\dot{\varphi}(\tau)$ в интервале $\tau \in [0, 2\pi]$

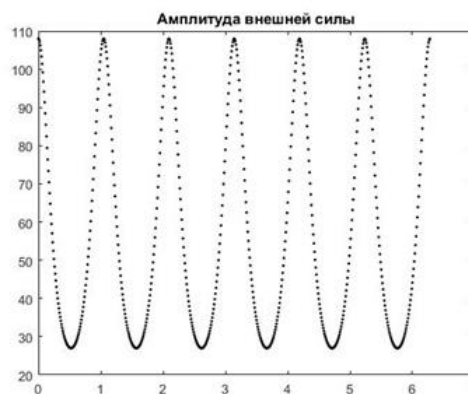


Рисунок 3 – График амплитуды вынуждающей силы P_a в интервале $\tau \in [0, 2\pi]$

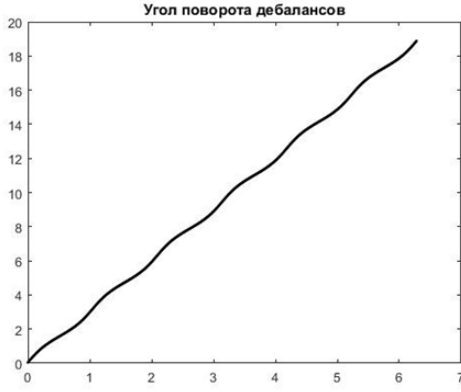


Рисунок 4 – График угла поворота дебаланса $\varphi(\tau)$ интервале $\tau \in [0, 2\pi]$

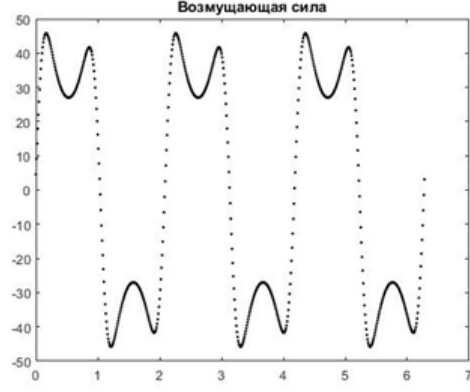


Рисунок 5 – График амплитуды вынуждающей в силы в интервале $\tau \in [0, 2\pi]$

3 Численно-аналитический анализ модели

Отметим, что уравнение (7) является линейным относительно перемещения. Поэтому, если его правую часть разложить в ряд Фурье (10)

$$P(\tau) = p_0 + p_1 \cdot \cos(\omega\tau - \alpha_1) + p_2 \cdot \cos(2\omega\tau - \alpha_2) + p_3 \cdot \cos(3\omega\tau - \alpha_3) + \dots, \quad (10)$$

то стационарные колебания модели будут складываться (в силу принципа суперпозиции) из частных колебаний, определяемых отдельными слагаемыми тригонометрического ряда (10). И в этом случае аналитическое описание закона движения рабочего органа будет иметь вид

$$\xi(\tau) = \xi_0 + \xi_1 \cdot \cos(\omega\tau - \beta_1) + \xi_2 \cdot \cos(2\omega\tau - \beta_2) + \xi_3 \cdot \cos(3\omega\tau - \beta_3) + \dots, \quad (11)$$

где

$$\xi_n = \frac{p_n}{\sqrt{(1-n^2\omega^2)^2 + b^2n^2\omega^2}}, \quad \beta_n = \arccos \frac{(1-n^2\omega^2)\cos\alpha_n - bn\omega\sin\alpha_n}{\sqrt{(1-n^2\omega^2)^2 + b^2n^2\omega^2}}.$$

Оценим период T вынуждающей силы (9), где множитель в квадратных скобках, представляющий ее амплитуду, имеет период $T_p = \pi/\omega$ и рассмотрим угол поворота дебалансов (6).

Заметим, что

$$\varphi\left(\frac{\pi}{\omega}\right) = \int_0^{\frac{\pi}{\omega}} \frac{\omega \cos \beta}{1 - \sin^2 \beta \cdot \cos^2(\omega \cdot \tau_1)} d\tau_1 = \int_0^{\frac{2\pi}{\omega}} \frac{\omega \cos \beta}{1 - \sin^2 \beta \cdot \cos^2(\omega \cdot \tau_1)} d\tau_1 = \int_0^{\frac{3\pi}{\omega}} \frac{\omega \cos \beta}{1 - \sin^2 \beta \cdot \cos^2(\omega \cdot \tau_1)} d\tau_1 = \dots \quad (12)$$

что устанавливается заменой переменной в каждом из интегралов.

Поэтому

$$\varphi\left(m \frac{\pi}{\omega}\right) = m \int_0^{\frac{\pi}{\omega}} \frac{\omega \cdot \cos \beta}{1 - \sin^2 \beta \cdot \cos^2(\omega \cdot \tau_1)} d\tau_1.$$

Таким образом, если между периодом синуса и значением $\varphi(\pi/\omega)$, т.е. при некоторых натуральных значениях n, m , существует рациональное соотношение, то имеет место

$$2\pi n = \varphi\left(m \frac{\pi}{\omega}\right) = m \int_0^{\frac{\pi}{\omega}} \frac{\omega \cdot \cos \beta}{1 - \sin^2 \beta \cdot \cos^2(\omega \cdot \tau_1)} d\tau_1,$$

а величина $T = m \cdot \pi/\omega$ и является периодом вынуждающей силы (9), причём, наименьшие значения параметров n, m определяют и наименьший её период. Но значение интеграла (12), вычисленное численно по правилу левых

прямоугольников с шагом 0.0001, оказывается равным

$$\varphi\left(\frac{\pi}{\omega}\right) = \varphi\left(\frac{\pi}{3}\right) = 3.1416,$$

что с точностью до 4-х знаков после запятой совпадает с π . Таким образом, в пределах этой точности можно считать, что $2\pi n = m\pi$, откуда $2n = m$ и, взяв, например, $n=1$, $m=2$, получим значение периода вынуждающей силы

$$T = m \cdot \frac{\pi}{\omega} = \frac{2\pi}{3}.$$

Выполним теперь разложение вынуждающей силы $P(\tau)$ в ряд Фурье (10) на интервале $[0, 2\pi/3]$. Разделим этот отрезок на $2^7=128$ частей и с использованием процедуры *ifft*, реализующей в среде Matlab обратное преобразование Фурье, получим

$$P(\tau) = 42.6745 \cos(3\tau - 1.4615) + 20.5806 \cos(9\tau - 1.4751) + \\ + 6.4055 \cos(15\tau - 1.4719) + 1.6773 \cos(21\tau - 1.4653)$$

(здесь учтены гармоники, амплитуды которых составляют не менее 1% наибольшей).

Для верификации полученного разложения на рис. 6 приводится результат его наложения на график $P(\tau)$. Тогда, в соответствии с (11), закон движения рабочего органа имеет вид

$$\xi(\tau) = 5.3306 \cos(3\tau - 1.7176) + 0.2572 \cos(9\tau - 1.6777) + \\ + 0.0286 \cos(15\tau - 1.6764) + 0.0038 \cos(21\tau - 1.6811). \quad (13)$$

Его график в интервале $[0, 2\pi/3]$ приведен на рис. 7.

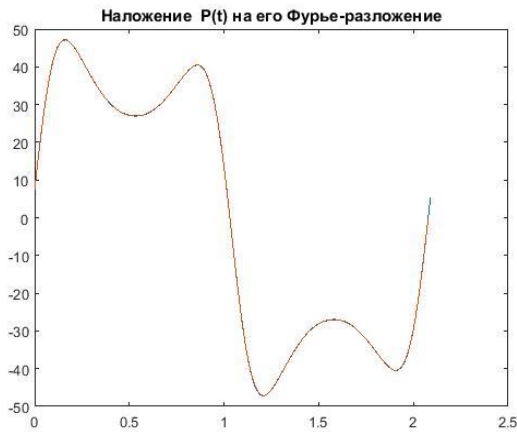


Рисунок 6 – Наложение графиков вынуждающей силы и его разложения Фурье

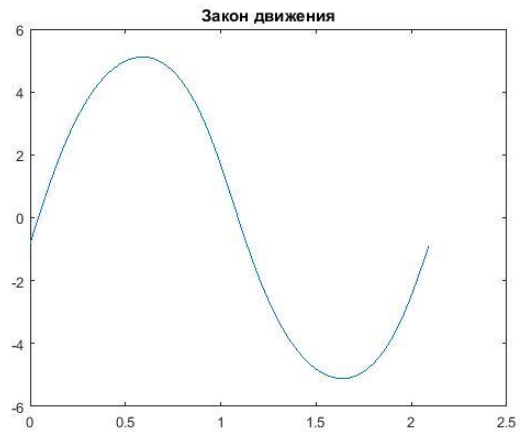


Рисунок 7 – График перемещения рабочего органа вибромашины

Дифференцируя (13), получим ускорение движения рабочего органа

$$\frac{d^2 \xi}{d\tau^2} = -9 \times 5.3306 \cos(3\tau - 1.7176) - 81 \times 0.2572 \cos(9\tau - 1.6777) - \\ - 225 \times 0.0286 \cos(15\tau - 1.6764) - 441 \times 0.0038 \cos(21\tau - 1.6811),$$

его график на участке $[0, 2\pi/3]$ приведен на рис. 8.

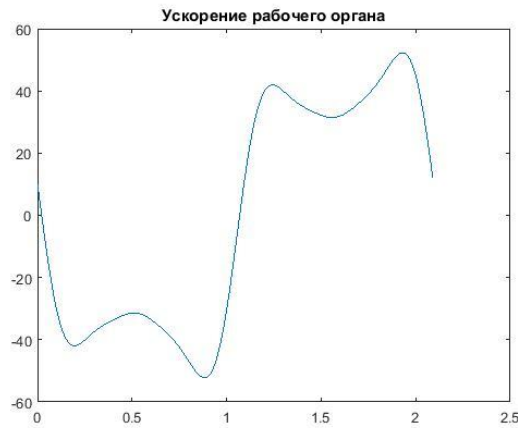


Рис. 8 – График ускорения рабочего органа вибромашины

4 Численный анализ модели

В форме дифференциальных уравнений первого порядка движение вибромашины (7) описывается системой трёх уравнений

$$\begin{cases} \frac{d\xi}{d\tau} = \xi_1; \\ \frac{d\xi_1}{d\tau} = -b\xi_1 - \xi + P_0 \cdot \left[\frac{\cos \beta}{1 - \sin^2 \beta \cdot \cos^2(\omega \cdot \tau)} \right]^2 \cdot \sin[\varphi(\tau)]; \\ \frac{d\varphi}{d\tau} = \frac{\omega \cos \beta}{1 - \sin^2 \beta \cdot \cos^2(\omega \cdot \tau)}. \end{cases} \quad (14)$$

Примем нулевые начальные условия

$$\xi(0) = \frac{d\xi(0)}{d(\tau)} = \varphi(0) = 0. \quad (15)$$

Решение задачи Коши (14), (15) было проведено в среде Matlab с использованием процедуры ode45, реализующей методы Рунге-Кутты 4-5-го порядка, с шагом интегрирования равным 0.01. Результаты, представленные на рис. 9 и 10, практически полностью совпадают с полученными выше.

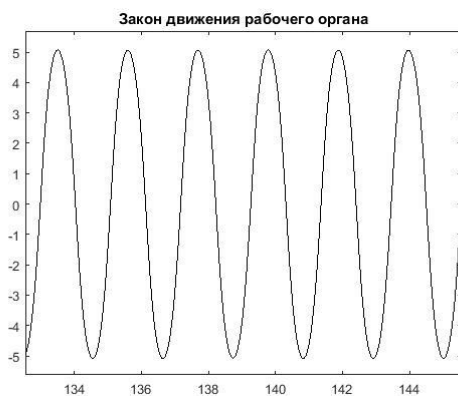


Рисунок 9 – График перемещения рабочего органа вибромашины.

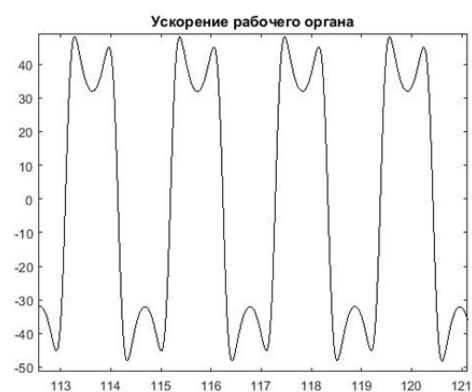


Рисунок 10 – График ускорения рабочего органа вибромашины.

Заключение

Полученные результаты, на наш взгляд, вполне заслуживают доверия, так как каждый шаг исследования вполне верифицирован. Таким образом, возможность формирования полигармонического возбуждения путем

несоосного соединения валов в приводе трансмиссии инерционной вибромашины можно считать теоретически обоснованным. Следствием появления полигармоничности в законе возбуждения является и полигармоничный характер колебаний рабочего органа. При выше выбранных соотношениях параметров величина третьей гармоники в спектре перемещений рабочего органа составляет около 4,8% основной, в спектре ускорений – 43,4%, что соответствует вибрационным машинам нелинейного типа в зоне основного резонанса. Обращает на себя внимание, также, что и амплитуда вынуждающей силы нелинейным образом зависит от угла излома валов, что ставит задачу выбора его оптимального значения, с учетом дополнительных требований, весьма интересной.

Литература

1. Продукция Kroosh [Электронный ресурс]. – Режим доступа: www.kroosh.ru.146-0-1. - Загл. с экрана.
2. Борохович, Д. Опыт применения технологии Kroosh для просеивания нерудных сыпучих материалов [Текст] / Д. Борохович, И. Круш, Ю. Ободан // Вибрации в технике и технологиях: Всеукр. науч.-техн. журнал. – Винница. – 2007. – No 48. – С. 61–64.
3. Круш, И. Применение технологии Kroosh для процессов разделения сыпучих материалов и полидисперсных суспензий [Электронный ресурс] / И. Круш, Д. Борохович, Г. Косой. – Режим доступа: <http://www.kroosh.com/ru/articles/primenenie-tehnologii-kroosh-dlya-processov-razdeleniya-sypuchih-materialov-i>. - Загл. с экрана.
4. Артоболевский, И. И. Механизмы в современной технике: Т. 2: Кулисно-рычажные и кривошипно-ползунные механизмы / И. И. Артоболевский. - М.: Наука, 1981. – 944 с.
5. Букин, С. Л. Динамическая модель одномассовой вибромашины с карданным валом в трансмиссии дебалансного возбудителя колебаний [Текст] / С. Л. Букин, А. С. Букина, В. В. Селивёрстов // Международ. сборник науч. трудов ДонНТУ «Прогрессивные технологии и системы машиностроения». – 2014. – No 4(50). – С. 65–73.

Беловодский В.Н., Букин С.Л. Один подход к формированию полигармонических колебаний в вибрационных машинах с инерционным возбуждением. Вибрационная техника широко используется для реализации и интенсификации технологических процессов в различных отраслях промышленности. И нередко, использование в них полигармонической вибрации, оказывается более эффективным по сравнению с гармонической. В работе предложен один подход к формированию полигармонических вибраций за счет излома валов в трансмиссии центробежного привода. Рассмотрена математическая модель одномассовой вибрационной машины с шарниром Гука в ее приводе и для одного из вариантов ее параметров проведена оценка полигармоничности в законах перемещений и ускорений рабочего органа.

Ключевые слова: вибрационная машина, инерционный привод, трансмиссия, шарнир Гука, полигармонические колебания.

Belovodskiy V.N., Bukin S.L. One approach to the formation of polyharmonic oscillations in vibration machines with inertial excitation. Vibration technology is widely used for the implementation and intensification of technological processes in various industries. And often, the use of polyharmonic vibration in them turns out to be more effective than harmonic. One approach to the formation of polyharmonic vibrations due to the fracture of shafts in the transmission of a centrifugal drive is proposed. A mathematical model of a single-mass vibrating machine with a Hook hinge in its drive is considered, and for one of the variants of its parameters, an assessment of polyharmonicity in the law of displacements and accelerations of the working body is carried out.

Key words: vibrating machine, inertial drive, transmission, Hook hinge, polyharmonic vibrations.

Неравенства в комплексной области и гипотеза Римана

З.Е. Филер

д.т.н., к.ф.-м.н., профессор, г. Нетания, Израиль

zalmenfilier3319@gmail.com, ORCID: 0000-0003-0804-6794, SPIN-код: 3555-6335

Филер З.Е. Неравенства в комплексной области и гипотеза Римана. Рассматривается знак $\zeta(z)$ функции в точках критической полосы. Используется представление об упорядочении в области комплексных чисел $z_1 < z_2 \Leftrightarrow \operatorname{Re}(z_1) < \operatorname{Re}(z_2)$ и при $\operatorname{Re}(z_1) = \operatorname{Re}(z_2)$ с $\operatorname{Im}(z_1) < \operatorname{Im}(z_2)$. Так как область, где $\zeta(z) < 0$ вмещает подобласть, где $\operatorname{Re}(z) < 0$, выделение последней может быть осуществлено на ПЭВМ. Точки пересечения границ областей $\operatorname{Re}(z) < 0$ и $\operatorname{Im}(z) < 0$ и являются корнями $\zeta(z)$. Отличие действительной части корня от $1/2$ невозможно, так как корню $z_1 < 1/2$ соответствует корень $z_2 = 1 - z_1$ в этой же полосе. Но он будет лежать в полосе, где $\zeta(z) > 0$, а не равно нулю. Если же $z_1 > 1/2$, то $1 - z_1$ будет лежать в области, где $\zeta(z) < 0$, что невозможно, если $\zeta(z_2) = 0$. Поэтому $z_2 = z_1 = 1/2$.

Ключевые слова. упорядочение области комплексных чисел, выделение области $\operatorname{Re}(z)$ на ПЭВМ, доказательство от противного.

Filer Z.E. Inequalities in the complex domain and the Riemann hypothesis. The sign $\zeta(z)$ of the function at the points of the critical band is considered. The representation of the ordering in the domain of complex numbers $z_1 < z_2 \Leftrightarrow \operatorname{Re}(z_1) < \operatorname{Re}(z_2)$ and at $\operatorname{Re}(z_1) = \operatorname{Re}(z_2)$ with $\operatorname{Im}(z_1) < \operatorname{Im}(z_2)$. Since the area where $\zeta(z) < 0$ accommodates the subdomain where $\operatorname{Re}(z) < 0$, the allocation of the latter can be carried out on a PC. The points of intersection of the boundaries of the regions $\operatorname{Re}(z) < 0$ and $\operatorname{Im}(z) < 0$ and are the roots of $\zeta(z)$. It is impossible to distinguish the real part of the root from $1/2$, since the root $z_1 < 1/2$ corresponds to the root $z_2 = 1 - z_1$ in the same band. But it will lie in the band where $\zeta(z) > 0$, and not equal to zero. If $z_1 > 1/2$, then $1 - z_1$ will lie in the region where $\zeta(z) < 0$, which is impossible if $\zeta(z_2) = 0$. Therefore, $z_2 = z_1 = 1/2$.

Keywords. ordering of the domain of complex numbers, allocation of the domain $\operatorname{Re}(z)$ on a PC, proof of the opposite.

**Полный текст статьи опубликован в выпуске № 4 (30), 2022 г.
научного журнала «Информатика и кибернетика»**

СЕКЦИЯ 5. «ПРОЕКТИРОВАНИЕ ПРОГРАММНЫХ И КОМПЬЮТЕРНЫХ СИСТЕМ, СРЕДСТВА АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ ПО И СИСТЕМ»

УДК 004.021

Классификация моделей САПР с помощью машинного обучения

А.В. Григорьев*¹, Е.С. Бондаренко*²

*¹ к.т.н, доцент, Донецкий национальный технический университет,
grigorievalvl@gmail.com

*² магистрант, Донецкий национальный технический университет,
kate.bond777@gmail.com

Григорьев А.В., Бондаренко Е.С. Классификация моделей САПР с помощью машинного обучения. В работе исследована возможность оценки сходства 3D-моделей в САПР с помощью нейронных сетей. Выявлены преимущества использования подхода машинного обучения при классификации 3D-моделей в САПР. Проведено сравнение трех различных алгоритмов контролируемого обучения. Предложен алгоритм создания надстройки в САПР для различия 3D-геометрии.

Ключевые слова: 3D-модель, машинное обучение, классификация, геометрия, нейронная сеть, САПР.

Введение

Оценка сходства между 3D-моделями является важной проблемой во многих областях, включая медицину, биологию и промышленность. Поскольку прямого метода сравнения 3D-геометрии не существует, разрабатываются различные представления моделей, позволяющие описывать формы объектов и проводить классификацию. Несмотря на то, что некоторые из этих дескрипторов обеспечивают высокую точность классификации, их применение часто ограничено.

За последние несколько десятилетий 3D-модели использовались в различных приложениях, включая медицину, биологию, химию, археологию, игры и промышленность. В то же время растущее число 3D-моделей, хранящихся в базах данных и системах управления жизненным циклом продукта (PLM), стимулирует разработку методов для сравнения и извлечения 3D-геометрии. В промышленном применении возможность обнаружения существующих аналогичных компонентов ускоряет процесс проектирования новых видов объектов. Это также исключает риск производства нескольких идентичных компонентов и, следовательно, более высокую стоимость единицы продукции.

Важность проблемы сравнения и извлечения трехмерной геометрии сделала ее областью обширного изучения с 1990 года. Однако первоначальных алгоритмов, основанных на тексте, использующих имя файла, его расширение и присвоенные теги, оказалось недостаточно. Они обладали низкой эффективностью и неудовлетворительной точностью поиска. Одновременно было доказано, что лучшие результаты могут быть получены путем включения поиска на основе данных, в которых для формулирования запроса используется 3D-модель или ее представление.

До сих пор не существует метода прямого сравнения 3D-моделей. Поэтому исследователи разработали серию дескрипторов, называемых сигнатурами формы, представляющих 3D-модель и ее атрибуты. Большинство работ сосредоточено на разработке единого «идеального» дескриптора, который позволит сравнивать объекты и точно определять их коэффициент сходства. В действительности использование только одной сигнатуры формы обычно приводит к описанию детали, которое подходит только для какой-то конкретной области применения.

В статье рассмотрен альтернативный метод, основанный на использовании комбинации различных сигнатур формы. Вектор легко извлекаемых атрибутов геометрии, включая разные параметры, используется в качестве входных данных для контролируемых алгоритмов машинного обучения, которые должны научиться группировать компоненты на основе свойств.

Целью работы является исследование возможности оценки сходства 3D-моделей в САПР с помощью нейронных сетей. Для этого были поставлены следующие задачи:

- выявление преимущества использования подхода машинного обучения при классификации 3D-моделей в САПР;

- сравнение трех различных алгоритмов контролируемого обучения;
- разработка алгоритма создания надстройки в САПР для различия 3D-геометрии.

Машинное обучение

Машинное обучение (ML) - это область компьютерных наук, использующая статистику и помогающая компьютеру обучаться без непосредственных инструкций [1]. Методы машинного обучения можно разделить на 2 основные категории – контролируемое обучение и неконтролируемые алгоритмы обучения. Алгоритмы контролируемого обучения, включая, например, регрессию, должны быть снабжены как входными данными, так и желаемыми выходными данными. Основываясь на этих двух наборах данных, алгоритм узнает, как получить желаемый результат на основе соответствующих входных данных. С другой стороны, при неконтролируемом обучении желаемые результаты не представляются модели, и предполагается, что алгоритм найдет оптимальный метод категоризации данных без каких-либо указаний.

Как контролируемые, так и неконтролируемые методы машинного обучения оказались полезными во многих реальных приложениях. Они могут быть использованы для задач типа классификации, регрессии и уменьшения размерности. Благодаря своей способности обобщать шаблоны, они не чувствительны к шуму во входном векторе. Кроме того, поскольку работа многих моделей ML, таких как нейронная сеть, основана на простых вычислениях, после обучения они работают быстрее, чем методы, использующие фильтрацию и набор логических правил [2].

Предполагается, что алгоритмы машинного обучения после обучения правильно прогнозируют желаемый результат на основе ранее невидимого входного вектора. Из-за этого точность алгоритмов ML должна определяться на основе ранее невидимых выборок. В практических приложениях набор данных обычно делится на три группы – обучающие, валидационные и тестовые наборы данных. Параметры алгоритма ML, называемые гиперпараметрами, настраиваются с использованием набора данных проверки, модель обучается с помощью набора данных обучения, тогда как конечная производительность модели (например, точность) оценивается на основе тестовых данных. Такой подход дает более точные прогнозы о том, как модель будет вести себя в будущем.

Существует три различных алгоритма контролируемого обучения. Они включают в себя классификатор случайного леса, классификатор опорных векторов (SVC) и полностью подключенную нейронную сеть.

Первый – классификатор случайного леса - на самом деле представляет собой набор более простых моделей прогнозирования, называемых деревьями решений. Каждое дерево решений представляет собой древовидный граф, использующий условные инструкции для входных переменных для получения желаемого результата. Поскольку случайный лес использует набор случайных деревьев, которые одновременно «голосуют» за конечный результат, этот метод менее чувствителен к выбору выборок, используемых для обучения, и, следовательно, менее уязвим для переобучения – получение идеальных прогнозов для обучающего набора данных и низкая производительность на тестовом наборе.

Важным гиперпараметром классификатора случайного леса является максимальная глубина дерева, определяющая максимальное количество шагов в процессе разделения набора данных. Слишком малое значение этого гиперпараметра приводит к низкой точности модели, поскольку впоследствии для разделения набора данных используется лишь несколько признаков. С другой стороны, слишком большое число приведет к переобучению – модель научится идеально разделять модели из обучающего набора данных, когда точность классификации для тестового набора данных будет значительно ниже. Средняя точность классификации, получаемая для различных значений гиперпараметра максимальной глубины, обычно является оптимальным значением.

Другой метод, называемый машиной опорных векторов (SVM), или его разновидностью, называемой классификатором опорных векторов (SVC), создает набор гиперплоскостей в многомерном пространстве, который, как предполагается, разбивает набор данных на желаемые кластеры, сохраняя при этом максимальное расстояние между гиперплоскостями и соседними выборками. Эти гиперплоскости затем могут быть использованы для регрессии (в приложении SVM, называемом Регрессия опорного вектора или SVR) или для целей классификации. Считается, что алгоритм SVC обеспечивает высокую точность классификации наборов данных, состоящих из относительно небольшого числа выборок по сравнению с количеством признаков, описывающих выборки.

Классификатор опорных векторов (SVC) - это алгоритм, который является полезным в случае задач, состоящих из большого количества объектов внутри входного вектора по сравнению с количеством выборок, используемых для обучения. Однако наилучшая точность алгоритма достигается с наборами данных, классы которых не перекрываются и могут быть легко разделены гиперплоскостями. Это потенциально может быть проблемой в случае рассматриваемого набора данных.

Последним методом машинного обучения, рассмотренным в этой работе, была полностью подключенная нейронная сеть (NN). NN состоит из нейронов, сгруппированных в слои, и связей, соединяющих нейроны, принадлежащие к соседним слоям. Каждая нейронная сеть содержит входные и выходные слои, разделенные одним или несколькими «скрытыми» слоями. NN, включающий более одного скрытого слоя, иногда называют «глубокой нейронной сетью». Каждое значение входного вектора, предоставляемого входному слою, умножается

на веса, присвоенные связям, соединяющим нейроны. Результирующие значения затем суммируются и обрабатываются функцией активации в нейронах следующего слоя. Схема структуры нейронной сети NN представлена на рис. 1.

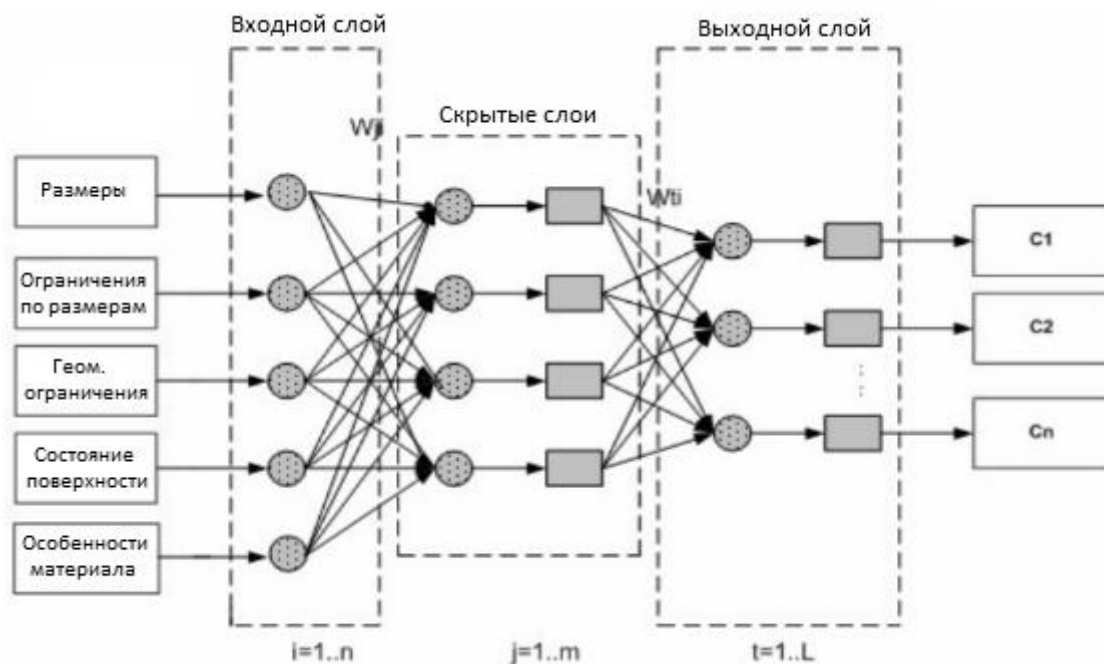


Рисунок 1 – Структура нейронных сетей NN

Следуя этой схеме, на выходном слое будут получены некоторые значения. Эти значения можно рассматривать как выходные данные нейронной сети. Таким образом, обучение NN оптимизирует веса, присвоенные конкретным ссылкам, чтобы они давали желаемые значения на выходном уровне.

Существует два важных гиперпараметра, управляющих топологией и архитектурой нейронной сети. Это количество скрытых слоев и количество нейронов в каждом слое. Нейронные сети, включающие более одного скрытого слоя, обычно называются глубокими нейронными сетями. Считается, что глубокие нейронные сети обычно превосходят обычные нейронные сети и лучше распознают сложные паттерны.

Сравнение 3D-моделей

В САПР 3D-модели нельзя сравнивать напрямую. Вместо этого из геометрии извлекаются дескрипторы тела, называемые представлениями формы или сигнатурами формы. Затем сходство между компонентами получается путем сравнения их дескрипторов формы. Набор требований к идеальному представлению формы обсуждался в [3]. Наиболее важные из них включают эффективность извлечения дескриптора формы из модели и количество значений, из которых он состоит.

Методы кластеризации 3D-форм часто классифицируются на основе характеристик, используемых для описания геометрии. Эти общие группы алгоритмов основаны на характеристиках, графиках, гистограммах и 2D-представлениях модели.

В области дескрипторов, использующих глобальные объекты, те, которые основаны на геометрических свойствах, обычно легче всего извлечь. В [4] была предложена поисковая система, фильтрующая 3D-модели на основе их геометрических свойств. Используемые характеристики включали объем, общую площадь поверхности, а также геометрические соотношения – извилистость и компактность.

Наконец, в [5] был использован коэффициент подобия, основанный на среднем соотношении таких свойств, как количество ребер, граней, впадин и т.д. Стоит отметить, что алгоритм был очень надежным и успешно работал с промышленными деталями.

Другая популярная группа дескрипторов использует графики. Метод преобразования 3D-модели в атрибутивный граф был представлен в [6]. При таком подходе грани модели САПР представлены узлами графа, тогда как ребра модели относятся к ее связям.

В последние годы анализ изображений все чаще используется для сравнения 3D-режимов. Первая работа в этой области, вероятно, была представлена в [7]. Авторы использовали полностью подключенную нейронную сеть для классификации шахт на основе их поперечных сечений. В [8] был определен и использован для поиска 3D-модели «дескриптор высот», полученный на основе 6 базовых видов в сером масштабе (сверху, снизу, спереди и т.д.). Был представлен алгоритм выбора наиболее репрезентативных видов, представляющих 3D-геометрию. Это повысило точность поиска, одновременно снизив вычислительные затраты. Наконец, сверточная

нейронная сеть была использована для обнаружения сходства между эскизами и 2D-видами 3D-геометрии. Такой подход дает возможность найти существующую модель на основе наброска от руки, а не 3D-геометрия, которая может быть неизвестна или недоступна.

Обзор методологии

Для создания надстройки в САПР для различия 3D-геометрии, рассмотрим основные этапы ее создания. На рис. 2 представлен общий обзор исследуемой методологии.



Рис. 2 – Обзор методологии

На первом этапе модели САПР загружаются из системы управления жизненным циклом продукта (PLM).

На втором этапе собранные модели обрабатываются с помощью приложения, разработанного на языке программирования VBA с применением API.

Последние версии многих САПР улучшили методы, поддерживаемые собственными объектами, и они были дополнены очень мощной математической утилитой. Эта функция позволяет легко выполнять базовые и расширенные точечные, векторные и матричные операции. Используя API в САПР, мы можем манипулировать тремя типами объектов: объектами, поступающими из САПР (собственные объекты модели), объектами, поступающими из базы данных математических утилит (математические объекты), и объектами, определяемыми пользователем (рис. 3). Собственные геометрические объекты относятся к объектам эскиза (точка, линия, круг, сплайн и т.д.) и их ограничениям, элементам (выдавливание, поворот, лофт и т.д.), управлению сборкой (сопряжение, вставка, перемещение и т.д.). Собственные математические объекты касаются точек, векторов и преобразований для манипулирования объектами (проецирование из пространства модели в пространство эскиза и наоборот, выполнение базовых операций с векторами и т.д.).

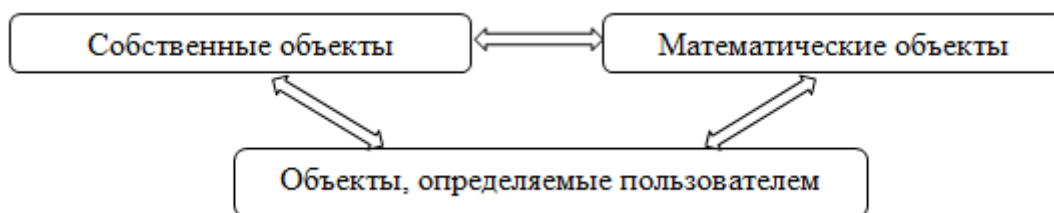


Рисунок 3 – Схема сущности API в САПР

Используя программное обеспечение без API, пользователь может получить доступ только к одному объекту модели, а прямой доступ к внутренней базе данных запрещен. Используя API, к базе данных объекта можно получить прямой доступ, экономя время на выполнение команды, а объекты модели могут быть чередованы с математическими и определяемыми пользователем объектами. Рассмотрим возможность импорта списка точек. Если количество этих точек велико, импорт вручную невозможен, поэтому можно сгенерировать макрокоманду для повторения операции импорта одной точки для всех точек, подлежащих обработке. Хотя эта операция возможна, это не самый лучший способ. Самый разумный способ импортировать точки - это получить прямой доступ к внутренней базе данных САПР, это можно сделать только с помощью методов программирования API.

С использованием вышеупомянутого программного обеспечения извлекаются различные переменные, характеризующие каждую модель САПР. Это, например, количество точек, граней и ребер, общая площадь поверхности, объем, отношение объема к площади, отношение площади к объему и т.д.

Изначально у нас есть только координаты точек объекта в трехмерном пространстве. Из этих координат можно вывести определенные свойства объекта, такие как площадь, соотношение сторон, длины ребер, радиус окружности, расстояние от центра тяжести до всех трех вершин, внутренние углы, тип многоугольника и многие другие. Примеры некоторых из производных характеристик: отношение угла между нормалью, отношение длины ребра, показ нижнего и верхнего квартилей числовых параметров и т.д.

Выбор набора подходящих входных переменных признаков является важной проблемой при построении нейронных, а также других классификаторов. Цель выбора переменной признака состоит в том, чтобы найти наименьший набор признаков, который может привести к удовлетворительной прогностической эффективности. Часто бывает необходимо и выгодно ограничить количество входных признаков в классификаторе, чтобы иметь хорошую прогностическую и менее трудоемкую в вычислениях модель. Производительность вне выборки можно повысить, используя только небольшое подмножество из всего набора доступных переменных.

Входные данные алгоритма обучения могут оказывать значительное влияние на производительность алгоритмов обучения. Поэтому предварительная обработка данных становится все более и более важной. Предварительная обработка данных - это собирательное название всех методов, направленных на обеспечение качества данных.

Собранные переменные, характеризующие модели САПР, затем сохраняются в файл.

Программа должна извлечь все переменные для каждой отдельной модели. Благодаря этому в наборе данных не будет пропущенных значений, что могло бы стать проблемой в случае некоторых алгоритмов машинного обучения.

Для использования алгоритмами машинного обучения, все числовые данные должны быть предварительно обработаны и нормализованы. Это нужно для того, чтобы избежать смещения алгоритмов из-за характеристик с более высокими абсолютными значениями. Вариант переменных представлен в таблице 1. Таких переменных может быть много, и они могут подбираться под каждый отдельный случай.

Таблица 1 - Функции, используемые алгоритмами машинного обучения

Характеристика	Тип данных
Количество точек	Числовой
Количество ребер	Числовой
Количество граней	Числовой
Объем	Числовой
Площадь поверхности	Числовой
Отношение площади к объему	Числовой
Отношение объема к площади	Числовой

На последнем шаге используются три различных алгоритма машинного обучения: классификатор случайного леса, классификатор опорных векторов и полностью подключенная нейронная сеть. Поскольку точность этих алгоритмов сильно зависит от значений гиперпараметров, их значения оптимизируются с использованием поиска по сетке для настройки модели и 5-кратной перекрестной проверки.

Чтобы объективно оценить результаты, полученные с помощью алгоритмов машинного обучения, необходимо рассчитать эталонную точность классификации. Ее можно определить по формуле 1:

$$\text{Точность} = \frac{\text{кол-во правильных решений}}{\text{кол-во рассмотренных решений}} \quad (1)$$

Преимущества методологии

Большинство отраслей промышленности по-прежнему используют текстовые поисковые системы для поиска моделей САПР. В фоновом режиме они используют имя файла и введенные вручную атрибуты, описывающие геометрию. Это делает эти методы уязвимыми для человеческих ошибок и трудными в обслуживании.

Подход машинного обучения, использующий геометрические атрибуты для классификации, обеспечивает удобную альтернативу. Используя различные обучающие наборы данных, модель машинного обучения может соответствовать требованиям конкретной отрасли. Она также может быть расширена новыми классификационными «правилами» без необходимости определять их вручную – алгоритм просто нужно переучить с расширенным или измененным набором данных. В результате алгоритм является простым в обслуживании. Наконец, обученная модель относительно невелика по сравнению с методами, основанными на операторах if-then или И-ИЛИ деревьях. Это позволяет использовать модель даже на компьютерах с ограниченной вычислительной мощностью.

Преимущества предлагаемой методологии обеспечивают множество потенциальных промышленных применений. Прежде всего, ее можно реализовать в жизненном цикле продукта Системы управления (PLM), помогающие дизайнерам находить аналогичные компоненты, уже находящиеся в производстве. В результате они

могли бы использовать знания, уже хранящиеся в производимых деталях. Это также предотвратило бы разработку очень похожих компонентов или дубликатов. Предлагаемый подход мог бы также позволить системам автоматизированного планирования технологических процессов (САПР) быстрее определять производственные процессы или оценивать производственные затраты на основе наиболее похожих компонентов, которые уже были рассмотрены.

Выводы

В этой статье рассмотрена методология использования алгоритмов машинного обучения (ML) для кластеризации моделей САПР. Свойства геометрии модели САПР (объем, площадь поверхности, количество точек и т.д.) используются в качестве входного вектора для трех различных моделей машинного обучения: классификатора случайного леса, классификатора опорных векторов и полностью связанной нейронной сети.

Исследование показало, что самой большой проблемой для алгоритмов является правильная классификация моделей САПР, которые отличаются только наличием некоторых незначительных особенностей (например, количеством точек). Благодаря гибкости методов машинного обучения, модели могут быть легко скорректированы в соответствии с конкретными требованиями при наличии некоторых особенностей.

Дальнейшая работа должна проверить надежность предлагаемого метода. Эта тема открыта для исследования, т.к. еще не создан универсальный метод сравнения графики в САПР, который позволил бы сократить время разработки новых моделей. Наконец, разработка надстройки в одной из САПР по предлагаемому методу позволит различать примитивные объекты графики и проверить корректность этого метода на практике. И в дальнейшем эту программу можно постепенно улучшать и доводить до совершенства.

Литература

1. V.B. Sunil and S.S. Pande. Automatic recognition of machining features using artificial neural networks. The International Journal of Advanced Manufacturing Technology, 41(9–10):932– 947, 2009.
2. A.C. Müller and S. Guido. Introduction to Machine Learning with Python: A Guide For Data Scientists. O'Reilly Media Inc., 2016.
3. Z. Qin, J. Jia, and J. Qin. Content based 3D model retrieval: A survey. In: 2008 International Workshop on Content-Based Multimedia Indexing, pages 249–256, London, UK, 18–20 June, 2008.
4. H.J. Rea, J.R. Corney, D.E.R. Clark, J. Pritchard, M.L. Breaks, and R.A. MacLeod. Part-sourcing in a global market. Concurrent Engineering, 10(4):325–333, 2002.
5. P. Cicconi, R. Raffaelli, and M. Germani. An approach to support model based definition by PMI annotations. Computer-Aided Design and Applications, 14(4):526–534, 2016.
6. M. El-Mehalawi and R.A. Miller. A database system of mechanical components based on geometric and topological similarity. Part I: representation. Computer-Aided Design, 35(1):83– 94, 2003.
7. H. Kaparthi and N.C. Suresh. A neural network system for shape-based classification and coding of rotational parts. International Journal of Production Research, 29(9):1771–1784, 1991.
8. J. Shih, C. Lee, and J.T. Wang. A new 3D model retrieval approach based on the elevation descriptor. Pattern Recognition, 40(1):283–295, 2007.

Григорьев А.В., Бондаренко Е.С. Классификация моделей САПР с помощью машинного обучения. В работе исследована возможность оценки сходства 3D-моделей в САПР с помощью нейронных сетей. Выявлены преимущества использования подхода машинного обучения при классификации 3D-моделей в САПР. Проведено сравнение трех различных алгоритмов контролируемого обучения. Предложен алгоритм создания надстройки в САПР для различия 3D-геометрии.

Ключевые слова: 3D-модель, машинное обучение, классификация, геометрия, нейронная сеть, САПР.

Grigoriev A.V., Bondarenko E.S. Classification of CAD models using machine learning. The paper investigates the possibility of assessing the similarity of 3D models in CAD using neural networks. The advantages of using the machine learning approach in the classification of 3D models in CAD are revealed. Three different algorithms of supervised learning are compared. An algorithm for creating an add-in in CAD to distinguish 3D geometry is proposed.

Keywords: 3D model, machine learning, classification, geometry, neural network, CAD.

Проблематика адаптивности системы поддержки принятия решений в области нейросетевых технологий

А.Д. Стальнов^{*1}, А.В. Григорьев^{*2}

^{*1} студент, Донецкий национальный технический университет,
anton.stalnov2000@mail.ru

^{*2} доцент, Донецкий национальный технический университет,
grigorievalvl@gmail.com

Стальнов А.Д., Григорьев А.В. Проблематика адаптивности системы поддержки принятия решений в области нейросетевых технологий. В данной статье рассмотрены базовые понятия, связанные с искусственными нейронными сетями. На основе данных, полученных в ходе анализа предметной области, был сформирован перечень ключевых критериев, которые необходимо учитывать при разработке адаптивной системы помощи принятия решений в области нейросетевых технологий. Определён набор принципов и технологических решений, способствующих адаптивности системы помощи.

Ключевые слова: искусственная нейронная сеть, адаптивность, искусственный нейрон, функция активации, слой, топология, перцептрон, система помощи принятия решений.

Введение

Изучая историю искусственных нейронных сетей, можно выделить ряд причин, которые снизили интерес научного сообщества к развитию данной отрасли. Основной причиной, вероятно, можно считать критику Марвином Минским самой модели перцептрона и демонстрация ограниченности его возможностей в книге «Перцептроны: введение в вычислительную геометрию» [1]. Кроме того, отсутствие внушительного количества данных для обучения моделей и ограниченность вычислительных мощностей также имели влияние. В настоящее время ситуация обстоит куда лучше: разработаны более эффективные модели, многократно возросли вычислительные мощности и сформировались достаточные для обучения наборы данных. Потому неудивительно, что в настоящее время мы можем наблюдать бурное развитие нейросетевых технологий и их повсеместное внедрение во многие сферы деятельности человека.

Из-за огромного числа разнообразных реализаций и, невозможности объективно оценить эффективность какой-либо из них, рядовые пользователи сталкиваются с трудностями в выборе программного обеспечения, которое могло бы решить поставленную перед пользователями задачу. Зачастую пользователь не разбирается в особенностях выбранного программного продукта и, опираясь на общие сведения (назначение сети), выбирает случайную нейронную технологию.

Следует отметить большую трудоемкость и сложность процессов подбора архитектуры нейронных сетей под специфику предметной области, в которой планируется применение нейронной сети, включая такие аспекты как: функция активации, топология сети, количество скрытых слоев, количество нейронов во входном слое и т.д.

Учитывая описанную выше ситуацию, возникает необходимость в разработке системы, позволяющей пользователю с меньшей трудоемкостью и пониженными требованиями к квалификации пользователя в области нейронных сетей, определиться с выбором архитектуры нейронной сети под задачу пользователя, т.е. – специфику предметной области.

Т.о., это делает актуальной задачу разработки методов и программных средств решения данной задачи.

Цели данной статьи: провести классификацию имеющихся типичных архитектур нейронных сетей, что позволит сформировать основные критерии, которые можно было бы использовать в адаптивной системе помощи принятия решений в области нейросетевых технологий в качестве характеристик архитектуры нейросети и выдвинуть предложения по обеспечению адаптивности описываемой системы с точки зрения специфики предметной области, где планируется применить данную нейронную сеть.

Для достижения цели предполагается выполнение следующих действий:

- провести анализ технологии искусственных нейронных сетей;
- сформировать ряд критериев, которые можно использовать в системе помощи принятия решений в области искусственных нейронных технологий в качестве характеристики нейропарадигм;
- определить ряд требований к инструментарию обработки признаков предметных областей с целью свести выбор нейронных сетей к специфике предметной областей;
- определить модели построения пользовательского интерфейса как способы автоматизировать поиск данных, характеризующих нейронные сети для данной предметной области.

Анализ технологии искусственных нейронных сетей

Адаптация — это способность приспосабливаться к условиям переменчивости окружающей среды. Транслируя данное понятие на пользовательский интерфейс, получаем следующее: способность пользовательского интерфейса приспосабливаться к условиям переменчивости набора данных, описываемых интерфейсом. Т.е. другими словами — возможность перестраиваться в случае, если пользователь внесёт какие-либо изменения в набор данных, которым оперирует данный интерфейс.

Искусственная нейронная сеть (ИНС), представляет собой математическую модель, основная идея которой — предоставить системе возможность самообучаться, посредством изменения внутреннего устройства системы. Идея модели навеяна биологическими нейронными сетями. ИНС это набор связанных искусственных нейронов.

Круглов В.В. и Борисов В.В. в своей книге «Искусственные нейронные сети: Теория и практика. – 2-е изд.» пишут об искусственном нейроне следующее: «Нейрон является составной частью нейронной сети. Он состоит из элементов трех типов: умножителей (синапсов), сумматора и нелинейного преобразователя» [2]. Таким образом, искусственные нейроны представляют собой вычислительные единицы, задача которых принимать данные (сигналы) от т.н. входных нейронов, связанных с ними синаптическими связями, производить вычисление и передавать сигнал, в случае необходимости. Каждая синаптическая связь обладает т.н. синапсом, который выступает в качестве веса связи. Решение об отправке сигнала зависит от т.н. нелинейного преобразователя — функции активации (иначе придаточной функции нейрона).

Производимые вычисление просты. Входное значение сигнала x_i умножается на синапс w_i , полученное произведение передаётся на вход, где складывается со всеми полученными ранее входными значениями. Далее сумма этих произведений передаётся в функцию активации, результат которой, в зависимости от вида функции, может быть как действительным, так и целым числом.

На рис. 1 приведено изображение искусственного нейрона и часто используемой функции активации.

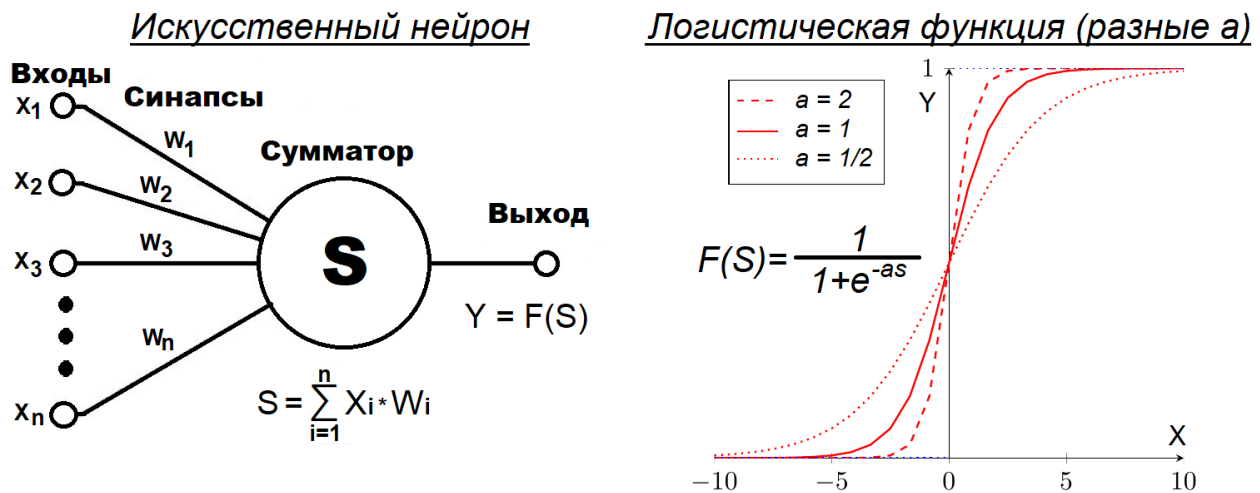


Рисунок 1 — Схема искусственного нейрона и одна из популярных функций активации

Для нашей задачи важно определить функцию активации искусственного нейрона, так по большей части именно от данной функции зависит поведение и обучаемость модели. В упомянутой ранее книге Круглов В.В. и Борисов В.В. приводят ряд примеров функций активации: линейная, полулинейная (иначе выпрямитель нейронной сети), логистическая (сигмоидальная), гиперболический тангенс (сигмоидальная), экспоненциальная, синусоидальная, сигмоидальная (рациональная), шаговая (линейная с насыщением), пороговая, модульная, знаковая (сигнатурная), квадратичная; особое внимания уделяя нелинейной функции активации с насыщением, так называемой логистической функции или сигмоиду (функция S-образного вида), поскольку функция возвращает значение в промежутке от 0 до 1, что делает данную функцию наиболее популярной (см. рис. 1).

Искусственные нейроны, собранные в сеть, представляют собой топологию сети (или архитектуру, если рассматривать уникальную топологию в качестве модели). Однако, прежде чем приступить к рассмотрению топологий, необходимо описать основные принципы нейропарадигм.

Парадигма (paradigma) переводится с греческого, как пример, модель, образец. Научная парадигма — принятая научным сообществом модель рациональной научной деятельности. Таким образом нейропарадигма является примером новой идеи механизма работы и обучения разума, и на практике представляет собой образец, состоящий из уникальной архитектуры искусственной нейронной сети и алгоритма её обучения.

Основные свойства нейропарадигм, сводятся к следующим принципам:

1. Коннекционизм, принцип которого представлен в идее, из которой следует, что мыслительные процессы могут быть описаны сетями из взаимосвязанных простых элементов, поскольку механизмы работы разума схожи с обработкой знакового языка средствами цифровой вычислительной машины. Формы связей зависят от конкретной модели. В рамках искусственных нейронных сетей элементами сети выступают искусственные нейроны, а связями — синапсы. В результате следования данному принципу в сети возникает нелинейность, которая разрушает линейную суперпозицию, из-за чего возможности сети многократно превосходят возможности отдельных искусственных нейронов, а специализация сетей не заложена заранее и обретается в ходе обучения, что обуславливает универсальность архитектур искусственных нейронных сетей.

2. Локальность и параллелизм вычислений — основной принцип, который заключается в том, что искусственные нейроны реагируют на поступающие в данный момент времени сигналы от связанных с ними элементов сети без привязки к общему жизненному циклу системы, как это обычно происходит в ЭВМ.

3. Программирование — обучение сети на основе поступающих данных. Данный принцип заключается в возможности изменять весовые значения связей, опираясь на результаты оценки эффективности системы, что в конечном счёте приводит к минимизации ошибок. Данный принцип увеличивает универсальность системы, поскольку позволяет формировать алгоритм в ходе самообучения с учётом специфики поставленной задачи.

4. Универсальность обучающих алгоритмов. Принцип заключается в общем назначении обучающих алгоритмов — минимизации эмпирической ошибки. Регулирующее значение задаётся извне в зависимости от целей обучения. В ходе обучения значения весов синаптических связей корректируется с целью минимизировать ошибку.

Fjodor Van Veen в своей статье «THE NEURAL NETWORK ZOO» [3], собрал, изучил и описал внушительное количество уникальных архитектур нейронных сетей. Внешний вид архитектур и краткие пояснения к элементам он запечатлел в виде диаграммы графов, предоставленной в рис. 6. Конечно, данный список не может считаться исчерпывающим, так как сфера нейросетевых технологий развивается и возникают новые идеи и реализации. Однако, этого достаточно, чтобы сформировать полноценное представление о том, с чем зачастую сталкивается пользователь при выборе нейросетевой технологии, и — с чем предстоит справиться при создании адаптивной системы помощи в выборе нейросетевой технологии. Рассмотрим некоторые из архитектур, которые привёл Fjodor.

Изучая архитектуру, изложенную Фрэнком Розенблаттом в своей работе «Перцептрон: вероятностная модель хранения и организации информации в мозгу» [4], Fjodor описывает искусственные нейронные сети с прямой связью (Feed forward neural networks, FF-сети или FFNN) и перцептроны (P), внешний вид которых изображен в рис. 3, как слоистую и однонаправленную структуру с, как правило, полносвязными элементами соседних слоёв. Слои бывают трёх видов: входные, скрытые и выходные. Достаточное количество скрытых нейронов, в теории позволяет моделировать взаимосвязь между входом и выходом, хотя на практике данная архитектура оказывается весьма ограниченной. Fjodor резюмирует, что данный вид архитектуры обычно объединяют с другими сетями, чтобы создавать новые.

Сети с радиальными базисными функциями (Radial basis function, RBF-сети) отличаются от FFNN только функциями активации (используются радиальные базисные функции).

Подобные сети обучают методом обратного распространения ошибки — метод вычисления градиента, который используется при обновлении весов. Данный метод был описан в работах А.И. Галушкина [5] и П.Дж. Вербоса [6]. Сети предоставляют данные о том, что в систему входит и что ожидается на выходе. Идея этого метода заключена в передаче сигналов ошибки от выхода сети к её входам. Подобный метод обучения называют «обучение с учителем».



Рисунок 3 — Искусственные нейронные сети с прямой связью (FF-сети слева, FFNN и RBF-сети справа)

Ещё одной вариацией FFNN можно считать т.н. «Машину экстремального обучения» [7] (Extreme learning machines, ELM). Данная архитектура выглядит как FFNN, однако связи и значения весов подобраны случайным образом. Метод обратного распространения ошибки не применим. Веса корректируются одним шагом в соответствии с методом наименьших квадратов. Гуан-Бинь Хуанг утверждает, что такой подход позволяет системе обучаться намного быстрее, чем при использовании метода обратного распространения ошибки.

Джеффри Элман в своей работе «Поиск структуры во времени» [8] описывает архитектуру рекуррентной нейронной сети (Recurrent neural networks, RNN). Идея архитектуры заключается в том, чтобы предоставить

скрытым нейронам возможность сохранять информацию о своём состоянии и обращаться к этим данным в будущем. Применение данного подхода приведёт к тому, что изменение порядок активации искусственных нейронов будет оказывать влияние на результат.

Fjodor Van Veen, описывая данную архитектуру, отмечает, что RNN обладает проблемой т.н. «Взрывающегося градиента» — это ситуация, при которой информация быстро теряется с течением времени. Подобная проблема также свойственна очень глубоким FFNN. И хотя потеря значения весов не столь критична, ведь это не состояние искусственного нейрона, но в RNN вес во времени хранит информацию из прошлого, а значит его потеря сделает предыдущее состояние менее информативным.

Майкл Джордан описал [9] схожую рекуррентную структуру. Отличие архитектуры Майкла от той, которую предоставил Джеффри Элман, в том, что памятью обладает вся система в целом. Перед выходом из сети данные сохраняются в памяти и передаются во входные искусственные нейроны. Изображение сети Элмана и Джордана предоставлено в рис. 4.

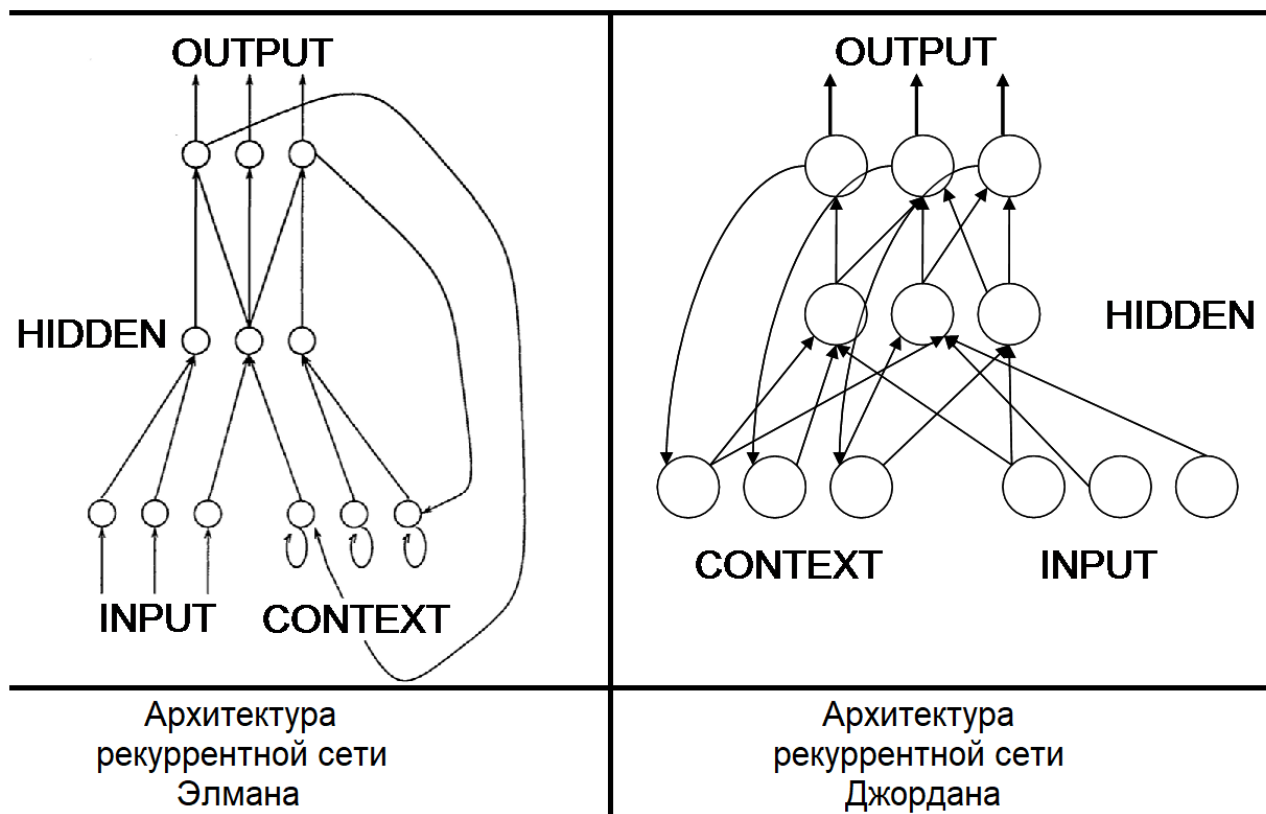


Рисунок 4 — Архитектура рекуррентных искусственных нейронных сетей Элмана и Джордана (RNN)

Найти применение RNN не сложно, в основном данная архитектура используется для областей, в которых данные не привязаны к временной шкале. RNN хорошо показывает себя в задачах генерации изображений и символов, в задачах распознавания образов, в задачах машинного перевода, автокоррекции и дополнения вводимых фраз. Основная сложность применения такой архитектуры в том, что её сложно обучить. Недостатком RNN является малый объём памяти. Ячейка памяти быстро перегружается, что приводит к возникновению ошибок.

С целью обойти упомянутые проблемы, свойственные традиционным рекуррентным сетям в 1997 году Зепп Хохрайтер и Юрген Шмидхубер в своей работе «Long short-term memory» [10] предоставили архитектуру, которая называется «Длинная цепь элементов краткосрочной памяти» (Long short-term memory, LSTM). Основной особенностью новой архитектуры было расширение возможностей ячейки памяти. В архитектуре сети присутствовали LSTM-модули, представляющие собой набор вентилях (иначе врат), которые, по назначению, можно именовать как вентили входа, выхода и забывания. Вентиль входа призван контролировать входящую информацию (решает какое количество информации должно быть сохранено в память), вентиль выхода призван контролировать выходящую информацию (решает какое количество данных необходимо передать следующему искусственному нейрону), а вентиль забывания призван контролировать объём памяти (решает есть ли необходимость помнить информацию). В результате сеть обрела способность запоминать информацию на произвольные промежутки времени, что позволило избежать от проблемы перегрузки памяти потери градиента при использовании метода обратного распространения ошибки во времени.

В 2014 году ряд учёных: Chung Junyoung, Gulcehre Caglar, Cho KyungHyun и Bengio Yoshua, в своей работе

«Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling» [11] предоставили вариацию LSTM, которую назвали «Управляемые рекуррентные блоки» (Gated Recurrent Units, GRU). В сравнении с LSTM, архитектура GRU быстрее, с ней проще работать, так как она имеет меньшее число параметров. Был пересмотрен механизм вентиля — вентили входа и выхода заменены вентилям обновления, а вентиль забывания заменён вентилям сброса. При этом, согласно приведенной информации, было доказано, что эффективность данной модели в задачах моделирования музыкальных и речевых сигналов сопоставима с LSTM.

Ещё одним из способов интерпретации FFNN выступает архитектура под названием «Автокодировщик» [12] (Autoencoder, AE). Основным отличием данной архитектуры от перцептрона является симметричность входа и выхода. Внешне топология данной архитектуры напоминает песочные часы. Подобная топология имеет чётко обозначенную задачу — заставить сеть выделять общие признаки из данных. Суть в том, чтобы получить на выходе отклик наиболее близкий ко входу. Для того, чтобы задача не была тривиальной, была изменена топология и введено ограничение на размерность данных. В скрытых слоях данные кодируются (сжимаются), при этом чем ближе к центру, тем более сжаты данные. Сеть можно обучить методом обратного распространения ошибки без учителя (что более естественно с точки зрения биологических сетей), если задавать ошибку равной разнице между входными и выходными данными. На рис. 5 представлено изображение описанной архитектуры, на котором X – входные данные, Z – сжатые данные, а X' – результат наиболее приближенные к значениям входа.

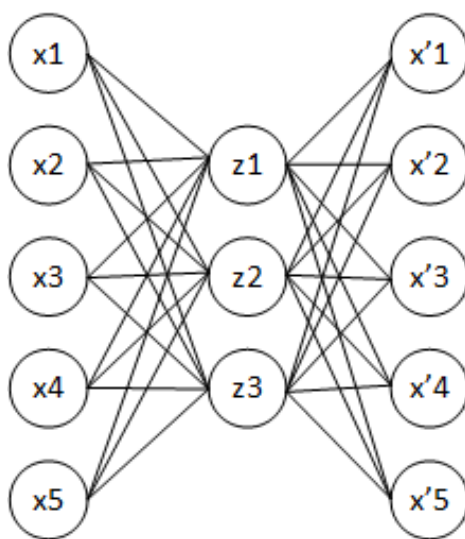


Рисунок 5 — Архитектура автокодировщика (AE)

В статье «Extracting and composing robust features with denoising autoencoders» [13] авторы описывают принцип работы т.н. «Шумоподавляющего автокодировщика» (Denoising autoencoders, DAE). Описанная архитектура является AE, на вход которого отправляют «зашумлённые» данные. Принцип обучения не меняется, т.е. данные выходного слоя сравниваются с подаваемыми на вход искаженными данными. Данный подход обладает некоторыми преимуществами — сеть учиться замечать более общие признаки, так как мелкие признаки теряются в «шуме».

Глубокая сеть доверия (Deep belief networks, DBN) — это многослойная глубокая нейросетевая архитектура, состоящая в основном из блоков AE или RBM (ограниченные машины Больцмана), соединённых между собой. В статье 2007 года «Greedy layer-wise training of deep networks» [14] была продемонстрирована эффективность поблочного обучения, причём каждый последующий блок должен лишь уметь кодировать (сжимать) предыдущий блок. Такой способ обучения называется методом жадного обучения, и заключается в выборе локальных оптимальных решений, что не может гарантировать оптимальный конечный результат. Сеть такой архитектуры можно обучить методом обратного распространения ошибки без учителя, что в конечном счёте позволит использовать сеть для генерации новых данных.

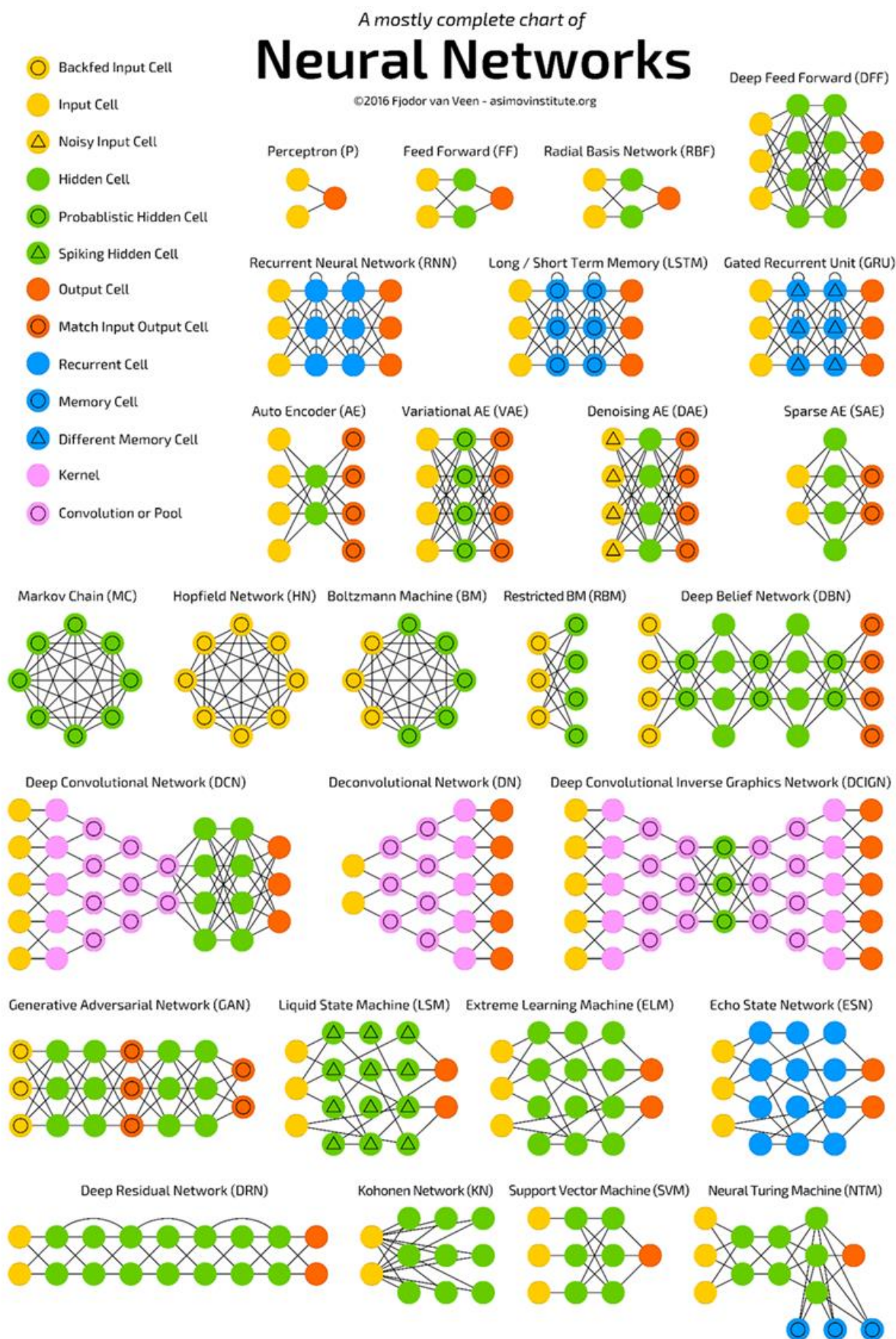


Рисунок 6 — «A mostly complete chart of Neural Networks»

Проблематика адаптивности системы помощи принятия решения в области искусственных нейронных технологий

В результате анализа технологий искусственных нейронных сетей было классифицировано некоторое множество популярных архитектур искусственных нейронных сетей. Информация, полученная в ходе классификации, позволяет выделить ряд структурных особенностей (признаков), однозначно характеризующих архитектуры. Таким образом, мы можем выделить следующий набор критериев, которые будут учитываться при разработке системы поддержки принятия решения в области нейросетевых технологий: архитектура, функция активации, метод обучения, количество скрытых слоёв, количество входных искусственных нейронов, количество скрытых искусственных нейронов, количество выходных искусственных нейронов, тип связности.

Важно отметить, что перечисленные выше критерии являются более узкими (техническими) и менее понятными, чем те, которые обычно привык использовать рядовой пользователь. Данные критерии, в основном, необходимы для более продвинутых пользователей, так как позволяют однозначно определиться с выбором. Поэтому систему помощи необходимо снабдить более общими и глобальными критериями, чтобы снизить порог квалификации пользователя. Примером таких критериев может выступать назначение сети и ряд характерных особенностей в сфере назначения, которые модель обретает в результате спецификации. Такие данные (специфические) помогут отсеять неподходящие результаты. Например, если назначением сети будет «художественное искусство», то характерными особенностями будут: принадлежность к стилю, вариант цветовой палитры, накладываемые эффекты и прочие. Конечно, набор подобных критериев зависит от назначения системы помощи. Однако, даже для самого базового уровня помощи, когда система должна помочь определиться с оптимальной топологией (архитектурой) сети, система должна уметь охарактеризовать цель, которую преследует пользователь. Нельзя абстрагироваться от задач пользователя, сформировав только базу характерных «технических» особенностей архитектур искусственных нейронных сетей, поскольку эффективность архитектур разнится от задачи к задаче. К примеру, в задачах классификации образов свёрточные сети проявляют себя гораздо лучше, чем любая из классических RNN. Очевидно, что подобные знания играют важную роль и умение оперировать ими необходимо для эффективности и адекватности выдаваемых системой результатов. Тем не менее, факт того, что свёрточная сеть имеет большую эффективность в задачах классификации образов, не позволяет игнорировать другие архитектуры. Пользователь должен иметь возможность выбора и, в случае возникновения неоднозначности, пользователь необходимо каким-либо образом подтолкнуть к правильному выбору. К примеру, можно внедрить в систему какой-либо отличительный маркер, указывающий на наиболее эффективное решение и добавить возможность сортировать список моделей по эффективности. Ещё одним вариантом стимуляции пользователя может выступать система сбора статистики использования моделей или система отзывов, в которой пользователи смогут описать свой опыт в использовании выбранной модели.

Таким образом, основной задачей разработчика системы помощи является необходимость сформировать мощную базу знаний. Важно, чтобы база знаний была актуальна, потому необходимо разработать методы, позволяющие легко пополнять базу новыми решениями. Такими методами могут выступать регулярные выражения поиска данных в структурированных (иерархичных) файлах формата XML, JSON, HDF и прочих подобных форматах. В таких файлах могут содержаться «технические» данные искусственной нейронной сети (функция активации, количество слоёв, метод обучения и прочие). Автоматизировать сбор «специфических» данных сложнее. У некоторых готовых продуктов может быть пользовательский интерфейс, который можно обработать синтаксическими анализаторами с целью получить назначение системы и перечень параметров, характеризующих особенности возвращаемого моделью результата. Хорошо, когда вместе с программным решением разработчик или генератор сети формирует и предоставляет файл или набор файлов со всеми искомыми данными. В крайнем случае, подобные файлы придётся формировать вручную для заполнения базы знаний.

Очевидно, что подобная система помощи должна обладать не менее мощным пользовательским интерфейсом. Интерфейс должен обладать системой поиска, которая должна пополняться новыми параметрами с ростом базы знаний. Существует множество разнообразных способов визуализировать модуль поиска, однако наиболее оптимальными, вероятно, можно считать следующие модели:

1. Помощь в выборе происходит посредством диалога с пользователем. Пользователю поэтапно предлагается определить спецификацию сети и технические характеристики. В ходе диалога пользователь выбирает предложенные варианты, которые хранятся в базе знаний, а не задаёт собственноручно, что увеличивает вероятность получить желаемый результат и снижает количество ошибок. Пользователь должен иметь возможность свободно перемещаться между этапами диалога, чтобы иметь возможность изменить свой выбор в любой момент. Как вариант, можно демонстрировать пользователю перечень выбранных критериев отдельно от диалога и предоставить возможность быстро переместиться в этап, соответствующий критерию, путём нажатия на него. Наличие возможности частично изменить набор параметров в модуле поиска неминуемо влечет возникновение исключительных ситуаций, связанных с необходимостью отменить выбор более поздних критериев. Конфликтующие параметры должны сбрасываться.

2. Помощь в выборе происходит путём выбора (активации чекбоксов) необходимых критериев из кучи. Пользователю предоставлены абсолютно все критерии. Их количество зависит от мощности базы знаний, что

может дезориентировать пользователя. Потому все критерии должны быть организованы в группы, например: назначение, специфика, размеры, методы обучения, функции активации и другие группы, в которых стандартным значением будет «не выбрано». Определённые критерии (например количество слоёв), должны сопровождаться списками, в которых можно будет выбрать доступное числовое значение. Конфликтующие критерии должны деактивироваться, если не использовались пользователем ранее или сбрасывать значение, в случае списков со значениями. Каждый деактивированный параметр, который ранее был использован пользователем должен подсвечиваться или выделяться каким-либо другим образом до тех пор, пока пользователь не обратит на него внимание.

Первая модель интерфейса более предпочтительна, поскольку имеет более строгую последовательность выбора и пользователю проще обратить внимание на сброс какого-либо критерия. Также можно выделить третий тип интерфейса, который будет совмещать описанные выше особенности, например, критерии в диалоге будут располагаться в виде кучи, соответствующей этапу диалога.

Заключение

В данной статье был проведен анализ технологий искусственных нейронных сетей, в ходе которого были рассмотрены основные понятия и некоторые архитектуры искусственных нейронных сетей. На основе полученных в результате анализа данных был сформирован ряд критериев, которые можно использовать в системе помощи принятия решений в области искусственных нейронных технологий в качестве характеристики нейропарадигм.

Фактически сформирован ряд структурных особенностей архитектур нейронных сетей, отличающихся функционалом, что открывает возможность выбора той или иной архитектуры, имеющей те или иные структурные особенности, исходя из значений ряда признаков предметной области.

В этом случае становится необходим инструментарий сравнения тех или признаков предметных областей, характеризующих те или иные архитектуры нейронных сетей с целью свести выбор нейронных сетей к специфике предметной области.

В случае неоднозначности выбора архитектуры нейронной сети необходимо обеспечить оптимизацию выбора по тем или иным критериям (быстродействие, минимальное число нейронов и т.д.).

Был сформирован ряд принципов, которые необходимо учитывать при создании адаптивной системы помощи принятия решений.

Для поддержания адаптивности было предложено несколько моделей пользовательского интерфейса и несколько способов автоматизировать поиск данных, характеризующих нейронные сети.

Результаты анализа и предложенные решения можно использовать при создании новых или модификации существующих редакторов искусственных нейронных сетей, что позволит потенциально повысить их эффективность и снизить требования к квалификации пользователя в области искусственных нейронных сетей.

Литература

1. М. Минский, С. Пейперт. Перцептроны пер. с англ. Г.Л. Гимельфарба и В.М. Шарыпанова; под ред. В.А. Ковалевского М.: Мир, 1971 261 с.: ил.
2. Круглов В.В., Борисов В.В. Искусственные нейронные сети: Теория и практика.— 2-е изд. стереотип. — М.: Горячая линия - Телеком, 2002. — 382 с.: ил. // Режим доступа: https://learn.ztu.edu.ua/pluginfile.php/91344/mod_folder/content/0/kruglov_v_v_borisov_v_v_iskusstvennye_neyronny_e_seti_teoriya.pdf — загл. с экрана.
3. The neural network zoo by Fjodor Van Veen, september 14, 2016 // [Электронный ресурс] — Режим доступа: <https://www.asimovinstitute.org/neural-network-zoo/> — загл. с экрана.
4. Rosenblatt Frank. The perceptron: a probabilistic model for information storage and organization in the brain. Psychological review 65.6 (1958): 386. // Режим доступа: <https://www.ling.upenn.edu/courses/cogs501/Rosenblatt1958.pdf> — загл. с экрана.
5. Галушкин А.И. Синтез многослойных систем распознавания образов. — М.: «Энергия», 1974.
6. Werbos P.J. Beyond regression: New tools for prediction and analysis in the behavioral sciences. Ph.D. thesis, Harvard University, Cambridge, MA, 1974.
7. Huang Guang-Bin, Zhu Qin-Yu, Siew Chee-Kheong (2006). "Extreme learning machine: theory and applications". Neurocomputing. 70(1): 489–501. // Режим доступа: <https://extreme-learning-machines.org/pdf/ELM-NC-2006.pdf> — загл. с экрана.
8. Elman Jeffrey L. Finding structure in time. Cognitive science 14.2 (1990): 179-211. // Режим доступа: <http://psych.colorado.edu/~kimlab/Elman1990.pdf> — загл. с экрана.
9. Jordan M.I. Serial Order: A Parallel Distributed Processing Approach: Tech. Rep. ICS Re-port 8604: Institute for Cognitive Science, University of California, San Diego, 1986. // Режим доступа: <https://cseweb.ucsd.edu/~gary/PAPER-SUGGESTIONS/Jordan-TR-8604-OCRed.pdf> — загл. с экрана.
10. Hochreiter Sepp, Jürgen Schmidhuber. Long short-term memory. Neural computation 9.8 (1997): 1735-1780. // Режим доступа: <https://www.bioinf.jku.at/publications/older/2604.pdf> — загл. с экрана.

11. Chung Junyoung, Gulcehre Caglar, Cho KyungHyun, Bengio Yoshua (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling, arXiv:1412.3555. // Режим доступа: <https://arxiv.org/pdf/1412.3555v1.pdf> – загл. с экрана.
12. Bourlard Hervé, Yves Kamp. Auto-association by multilayer perceptrons and singular value decomposition. Biological cybernetics 59.4-5 (1988): 291-294. // Режим доступа: http://data-science.wiki/images/Auto-association_by_multilayer_perceptrons_and_singular_value_decomposition_1548720901c89b3b7481f7500d7cd64e99bd.pdf – загл. с экрана.
13. Pascal Vincent, Hugo Larochelle, Yoshua Bengio. Extracting and composing robust features with denoising autoencoders. Proceedings of the 25th international conference on Machine learning. ACM, 2008. // Режим доступа: <http://machinelearning.org/archive/icml2008/papers/592.pdf> – загл. с экрана.
14. Yoshua Bengio, Pascal Lamblin, Dan Popovici, Hugo Larochelle. Greedy layer-wise training of deep networks. Advances in neural information processing systems 19 (2007): 153. // Режим доступа: <https://proceedings.neurips.cc/paper/2006/file/5da713a690c067105aeb2fae32403405-Paper.pdf> – загл. с экрана.

Стальнов А.Д., Григорьев А.В. Проблематика адаптивности системы поддержки принятия решений в области нейросетевых технологий. В данной статье рассмотрены базовые понятия, связанные с искусственными нейронными сетями. На основе данных, полученных в ходе анализа предметной области, был сформирован перечень ключевых критериев, которые необходимо учитывать при разработке адаптивной системы помощи принятия решений в области нейросетевых технологий. Определён набор принципов и технологических решений, способствующих адаптивности системы помощи.

Ключевые слова: искусственная нейронная сеть, адаптивность, искусственный нейрон, функция активации, слой, топология, перцептрон, система помощи принятия решений.

Stalnov A.D., Grigoriev A.V. The problem of adaptability of the decision support system in the field of neural network technologies. This article discusses the basic concepts associated with artificial neural networks. Based on the data obtained during the analysis of the subject area, a list of key criteria was formed that must be taken into account when developing an adaptive decision support system in the field of neural network technologies. A set of principles and technological solutions that contribute to the adaptability of the assistance system is defined.

Keywords: artificial neural network, adaptability, artificial neuron, activation function, layer, topology, perceptron, decision support system.

О задаче проектирования универсального конструктора нейронных сетей с возможностью экспорта и импорта моделей различных форматов

Д.А. Савельева^{*1}, А.В. Григорьев^{*2}

^{*1} студент, Донецкий национальный технический университет,
diana.saveleva.dos@gmail.com

^{*2} доцент, Донецкий национальный технический университет,
grigorievalvl@gmail.com

Савельева Д.А., Григорьев А.В. О задаче проектирования конструктора нейронных сетей с возможностью экспорта и импорта моделей различных форматов. Проанализированы лучшие практики конструкторов нейронных сетей. Выделена проблематика редакторов нейронных сетей. Рассмотрены основные проблемы и форматы сохранения моделей нейронных сетей. Выделены основные требования для нового универсального конструктора нейронных сетей с возможностью экспорта и импорта моделей различных форматов.

Ключевые слова: нейронная сеть, JSON, конструктор, модель, YAML, HDF5.

Введение

В настоящее время технологии «искусственного интеллекта» набирают всё большую популярность. Нейронные сети – один из наиболее используемых методов реализации искусственного интеллекта. Они широко применяются при решении различного типа задач, особенно при неэффективности или невозможности применения обыкновенных алгоритмических решений. Решение задач с применением искусственных нейронных сетей становится все более популярным. Вследствие, можно предположить, что в будущем искусственные нейронные сети будут широко применяться за счет лучшего понимания их основополагающих принципов [1].

Нейронные сети моделируют работу нервной системы человека, впоследствии перенимая основную её черту – способность к самообучению с учетом предыдущего опыта. Инструменты создания и применения нейронных сетей совершенствуются, а появление ошибок сводится к минимуму [2].

Нейронная сеть состоит из некоторых элементов – нейронов, имеющих определенные динамические параметры. Параметры нейронов могут менять свои значения в соответствии с полученными результатами. В общем случае, данный процесс называется обучением нейронной сети. Процедура, которая используется для процесса обучения, называется алгоритмом обучения. Эта процедура выстраивает в определенном порядке синаптические веса нейронной сети для обеспечения необходимой структуры взаимосвязей нейронов [3]. Выходной сигнал нейрона может посылаться другим элементам по взвешенным связям, с каждой из которых связан весовой коэффициент или вес. В зависимости от значения весового коэффициента передаваемый сигнал может усиливаться или подавляться [4]. Группы нейронов образуют нейронные слои.

Конструктор нейронных сетей - многофункциональный инструмент, среда разработки, обучения и тестирования нейронных сетей различного типа. Большинство программных продуктов, реализующих нейронные сети, направлено на реализацию вычислений и работает с сетями заданной структуры. Использование конструктора нейронных сетей в первую очередь направлено на формирование различных структур и конфигураций нейронных сетей [5].

Большое количество компаний работает над созданием уникального конструктора нейронных сетей, который будет включать не только функционал для работы с нейронными сетями и обширную библиотеку готовых алгоритмов, критериев, моделей и входных данных, но также будет позволять пользователю участвовать в процессе расширения данной предметной области посредством коммуникации с работами других пользователей. Разработка современного программного продукта конструирования нейронных сетей, простого в освоении, с наглядным и понятным интерфейсом, свободно распространяемого и с открытым исходным кодом, является актуальной задачей. Актуальной задачей на данный момент также является расширение списка форматов для сохранения нейронных моделей, внедрение функций экспорта и импорта готовых моделей в пользовательские программные продукты.

Целью данной статьи является выявление требований к универсальному конструктору нейронных сетей с основным упором на поиск решения проблем импорта и экспорта готовых моделей нейронных сетей.

Для достижения поставленной цели выполним следующие задачи:

- анализ существующих конструкторов нейронных сетей;
- определение проблематики развития редакторов нейронных сетей;
- выделение основных и расширенных требований для универсального (нового) нейронного конструктора;
- анализа основных форматов сохранения нейронных сетей;

- обсуждение проблем сохранения моделей нейронных сетей и в рамках универсального редактора нейронных сетей.

Анализ существующих конструкторов нейронных сетей

Проведем анализ существующих конструкторов нейронных сетей и нейронных пакетов для выявления недостатков и достоинств данного типа программного обеспечения.

Существует множество редакторов нейронных сетей: MemBrain, NeuroEmulator, Process Advisor, BrainMaker Pro, Neural Bench, SNNS, Significo, BrainMaker, NeuFrame, NeuroGenetic Optimizer, Braincel [6]. Для полноценного анализа в данной статье рассмотрим лучшие практики нейронных конструкторов: NeuroSolutions, NeuralWorks Professional, NeuroShell2.

NeuroSolutions - многофункциональный инструмент, разработанный специально для проектирования нейронных сетей. Нейронный редактор NeuroSolutions, разработанный фирмой NeuroDimension Inc, предназначен для моделирования широкого круга искусственных нейронных сетей. По оценкам специалистов является лучшим пакетом для взаимодействия с нейронными сетями. Ключевые особенности пакета: мощный функционал, позволяющий строить нейронные сети любой сложности; наличие комплекта готовых моделей, с кратким описанием к каждой; наличие функции генетической оптимизации; взаимодействие с документами Excel (извлечение/внесение данных); использование библиотек DLL, позволяющих интегрировать нейронные сети в собственные приложения. Преимущества и недостатки данного пакета представлены в табл. 1.

Таблица 1 – Преимущества и недостатки NeuroSolutions

Преимущества	Недостатки
Снабжен мощнейшим редактором визуального проектирования нейронных сетей, который позволяет создавать практически любые собственные нейронные структуры.	Включено малое количество критериев обучения. Пользователю необходимо создавать внешние модули для применения нестандартных критериев обучения.
Снабжен встроенным генератором кода и средствами, позволяющими использовать внешние модули при проектировании и обучении нейронной сети.	Реализовано малое количество методов обучения: алгоритм обратного распространения, сети Кохонена, самоорганизующиеся структуры.
Снабжен мощными и хорошо продуманными средствами визуализации.	
Широкий спектр средств для организации обучающих выборок.	

NeuralWorks Professional - мощное средство моделирования искусственных нейронных сетей. Нейронный конструктор разработан компанией NeuralWare. Основной упор сделан на применение стандартных нейронных парадигм и алгоритмов обучения. Содержит также большое число алгоритмов обучения нейронной сети. Дополнительно поставляемый модуль позволяет пользователю создавать свои собственные нейронные структуры и работать с ними. Модульная структура программы обеспечивает возможность подключения дополнительных стратегий управления, новые типы нейронных сетей и виды данных в графической форме. Преимущества и недостатки данного пакета представлены в табл. 2.

Таблица 2 – Преимущества и недостатки NeuralWorks Professional

Преимущества	Недостатки
В пакете реализовано 28 стандартных нейронных парадигм.	Отсутствие мощной системы предварительной обработки данных.
В пакете можно просмотреть корреляцию весов нейронной сети при обучении.	
Снабжен встроенным генератором кода, поддерживающим компилятор Microsoft Visual C++.	
Предоставляет возможность интегрирования внешних программных модулей, написанных пользователями.	

NeuroShell 2 является средой для создания, редактирования и обучения нейронных сетей. Данная система не считается универсальной. Она входит в состав пакета The AI Trilogy.

К особенностям нейропакета относят жестко реализованную последовательность действий при работе с нейронной сетью. Вследствие, невозможно определить структуру нейронной сети до того, как заданы входные данные. Данный нейронный пакет позволяет считывать данные, представленные в бинарном виде, а также в наиболее популярных финансовых форматах Mala Stock и Dow Jones. Преимущества и недостатки пакета представлены в табл. 3.

Таблица 3 – Преимущества и недостатки NeuroShell2

Преимущества	Недостатки
Мощная система обмена данными между различными приложениями.	Недостаточно продуманный интерфейс.
Имеет встроенный генератор исходного кода на языках Visual C и Visual Basic.	Недостаточно продуманная система визуализации данных.
	Нет возможности создания собственной архитектуры нейронной сети.
	Нельзя использовать специфичные алгоритмы обучения и изменять критерии обучения нейронной сети.

Проблематика редакторов нейронных сетей

Функционально оптимальный (далее – универсальный) конструктор нейронных сетей должен включать все преимущества существующих пакетов, а также устранить недостатки посредством реализации соответствующих им функций в новом нейронном конструкторе.

На основании анализа пакетов сформируем актуальные проблемы конструкторов нейронных сетей и возможные способы их решения.

Одной из наиболее важных проблем является отсутствие мощной системы предварительной обработки данных. Возможен следующий комплексный подход к решению данной проблемы. Перед обучением нейронной сети системе необходимо провести основные этапы обработки данных: сбор данных, анализ и очистка данных, преобразование данных.

Набор инструментальных средств - поддерживаемых нейронных моделей, критериев и алгоритмов обучения – требует постоянного расширения и повышения качества. Данную проблему необходимо решать внедрением максимально обширного количества нейронных моделей, критериев и алгоритмов обучения в результате проведения анализа требований пользователей.

Ограниченная система визуализации данных – требующая внимания проблема, препятствие на пути к пониманию структуры нейронной сети. Возможное решение - разработка максимально обширного функционала для работы с визуальной составляющей нейронного пакета. Структура нейронной сети должна отображаться в удобном для пользователя виде. Система должна предоставить пользователю возможность контроля представления нейронной структуры.

Отсутствие генератора исходного кода не позволяет пользователям внедрять модели нейронных сетей, построенные в конструкторе, в пользовательские программы. Внедрение генератора исходного кода нейронной сети (в виде файла на востребованных пользователями языках программирования) устранил недостаток многих конструкторов нейронных сетей.

Особое внимание должно быть уделено проблеме сохранения моделей нейронных сетей, в частности в различных форматах. Возможным решением является создание нового формата сохранения моделей, включающего лучшие свойства уже существующих форматов и ориентированного на соответствие требованиям любого приложения, реализующего функционал нейронных сетей.

Требования к проектируемому конструктору нейронных сетей

На основе анализа существующих редакторов нейронных сетей и проблематики их развития сформулируем требования к функционально полному, универсальному редактору нейронных сетей.

На данном этапе выделим два вида требований: основные, необходимо присущие всем существующим конструкторам нейронных сетей, и расширенные, добавляющие новые функции в нейронные конструкторы.

Перечислим основные требования к универсальному конструктору нейронных сетей:

1. Система должна предоставлять возможность создания нейронной матрицы связей;
2. Система должна предоставлять возможность удаления нейронных слоёв, нейронов, нейронных матриц связей;
3. Система должна предоставлять возможность настройки параметров каждого слоя для многослойных сетей. Система должна позволять ввод параметров: количество нейронов в слое, связи между нейронами в слое;
4. Система должна предоставлять возможность обучения нейронных сетей различного типа;
5. Система должна предоставлять возможность тестирования обученной нейронной сети;
6. Система должна предоставлять возможность масштабирования входных и выходных данных;
7. Система должна уметь визуализировать входные и выходные данные в виде графиков и диаграмм;
8. Система должна иметь встроенный генератор кода;
9. Система должна предоставлять возможность сохранения шаблона созданной нейронной структуры;
10. Система должна иметь мощную систему предварительной обработки данных;
11. Система должна предоставлять возможность импорта готовых входных данных из Microsoft Access,

Microsoft Excel, текстовых или CSV файлов;

12. Система должна предоставлять возможность экспорта результирующих данных в Microsoft Access, Microsoft Excel, текстовый или CSV файлы;

13. Система должна предоставлять возможность совместной разработки нескольким пользователям;

14. Система должна иметь комплект готовых алгоритмов, моделей нейронных сетей и данных для их обучения;

15. Система должна уметь решать задачи классификации, регрессии, уменьшения размерности или кластеризации;

16. Система должна содержать обширную элементную базу для моделирования нейронных сетей различных типов.

Выделим расширенные требования к универсальному конструктору нейронных сетей:

1. Система должна иметь открытый исходный код;

2. Система должна предусматривать загрузку пользователем внешних модулей в общую библиотеку модулей пакета;

3. Система должна предоставлять возможность создания различных типов нейронных сетей: неструктурированные нейронные сети, нейронные сети с локальными связями, однослойные нейронные сети, многослойные нейронные сети (последовательные, прямые, обратные), полносвязные нейронные сети;

4. Система должна предоставлять возможность настройки параметров обучения, поддержки классических нейросетевых моделей и алгоритмов обучения;

5. Система должна предоставлять возможность разметки (создания тренировочного, тестового и экзаменационного наборов данных) исходных данных;

6. Система должна поддерживать режим отладки при обучении, тестировании нейронных сетей;

7. Система должна предоставлять возможность оптимизации обучения сети;

8. Система должна иметь мощный редактор визуального проектирования нейронной сети, позволяющий создавать любые собственные нейронные структуры;

9. Система должна поддерживать инструмент статистического анализа и прогнозирования, основанный на использовании нейронной сети и позволяющий выявлять скрытые зависимости в больших массивах числовой информации;

10. Система должна предоставлять возможность рандомизирования нейронной сети;

11. Система должна осуществлять импорт и экспорт моделей нейронных сетей в формате HDF5, JSON, YAML;

12. Система должна предоставлять возможность выбора и минимизации функции ошибки;

13. Система должна предоставлять возможность просмотра оценки обучения нейронной сети;

14. Система должна предоставлять возможность создания нейронных структур с неограниченным количеством элементов;

15. Система должна предоставлять возможность подбора размера и архитектуры нейронной сети на основании размера пользовательского набора данных и типа решаемой проблемы (классификация, регрессия, снижение размерности или кластеризация).

Форматы сохранения нейронных сетей

Рассмотрим детальнее проблематику сохранения нейронных сетей в различных форматах.

Обсудим вначале достоинства и недостатки существующих форматов сохранения нейронных сетей.

Важной функцией универсального конструктора нейронных сетей является умение экспортировать и импортировать модели различных форматов. Обучение модели нейронной сети обычно занимает много времени, особенно если аппаратные возможности системы не соответствуют требованиям. После завершения обучения хорошим решением является сохранение модели в файл. Чтобы повторно использовать модель в более поздний момент времени для прогнозирования, мы загружаем сохраненную модель. Наиболее известные форматы для экспорта и импорта моделей нейронных сетей – HDF5, YAML, JSON. Рассмотрим характеристику данных форматов и реализацию сохранения моделей на примере языка Python. Python предоставляет обширный выбор библиотек, множество из которых ориентированы на машинное обучение.

Для работы с нейронными сетями и, в частности, их сохранения удобно использовать API Keras. Keras предоставляет возможность сохранения моделей нейронных сетей в трех форматах: формат YAML, формат JSON, формат HDF5. С помощью Keras определяются высокоуровневые нейронные сети, работающие поверх библиотеки TensorFlow или Theano [7].

Файлы YAML и JSON хранят только структуру модели, в то время как файл HDF5 хранит полную модель нейронной сети вместе со структурой и весами. Поэтому, если структура модели сохраняется с использованием формата YAML или JSON, веса должны быть сохранены в файле HDF5 для хранения всей модели.

Формат JSON - это простой формат файла для иерархического описания данных. За счёт своей лаконичности в сравнении с XML формат JSON может быть более подходящим для сериализации сложных структур. Keras предоставляет возможность описать любую модель, используя формат JSON и функцию `to_json()`. Позже модель может быть загружена с помощью функции `model_from_json()`. На рис. 1 приведено содержимое

JSON файла сохраненной модели.

```
{
  "keras_version": "2.0.2",
  "backend": "theano",
  "config": {
    "config": {
      "dtype": "float32",
      "bias_regularizer": null,
      "activation": "relu",
      "bias_constraint": null,
      "use_bias": true,
      "bias_initializer": {
        "config": {
          "maxval": 0.05,
          "minval": -0.05,
          "seed": null
        },
        "class_name": "Zeros"
      },
      "kernel_regularizer": null,
      "activity_regularizer": null,
      "kernel_constraint": null,
      "trainable": true,
      "name": "dense_1",
      "kernel_initializer": {
        "config": {
          "maxval": 0.05,
          "minval": -0.05,
          "seed": null
        },
        "class_name": "RandomUniform"
      },
      "batch_input_shape": [
        null,
        8
      ],
      "units": 12,
      "class_name": "Dense"
    },
    "config": {
      "kernel_regularizer": null,
      "bias_regularizer": null,
      "activation": "relu",
      "bias_constraint": null,
      "use_bias": true,
      "bias_initializer": {
        "config": {
          "maxval": 0.05,
          "minval": -0.05,
          "seed": null
        },
        "class_name": "Zeros"
      },
      "kernel_regularizer": null,
      "activity_regularizer": null,
      "kernel_constraint": null,
      "trainable": true,
      "name": "dense_2",
      "kernel_initializer": {
        "config": {
          "maxval": 0.05,
          "minval": -0.05,
          "seed": null
        },
        "class_name": "RandomUniform"
      },
      "units": 8,
      "class_name": "Dense"
    },
    "config": {
      "kernel_regularizer": null,
      "bias_regularizer": null,
      "activation": "sigmoid",
      "bias_constraint": null,
      "use_bias": true,
      "bias_initializer": {
        "config": {
          "maxval": 0.05,
          "minval": -0.05,
          "seed": null
        },
        "class_name": "Zeros"
      },
      "activity_regularizer": null,
      "kernel_constraint": null,
      "trainable": true,
      "name": "dense_3",
      "kernel_initializer": {
        "config": {
          "maxval": 0.05,
          "minval": -0.05,
          "seed": null
        },
        "class_name": "RandomUniform"
      },
      "units": 1,
      "class_name": "Dense"
    }
  },
  "class_name": "Sequential"
}
```

Рисунок 1 – Модель, сохраненная в формате JSON

Формат YAML, язык сериализации данных, используется для спецификации модели. Keras предоставляет возможность описать любую модель, используя формат YAML и функцию `to_yaml()`. Позже модель может быть загружена с помощью функции `model_from_yaml()`. На рис. 2 приведено содержимое JSON файла сохраненной модели.

```
backend: theano
class_name: Sequential
config:
- class_name: Dense
  config:
    activation: relu
    activity_regularizer: null
    batch_input_shape: !python/tuple
    [null, 8]
    bias_constraint: null
    bias_initializer:
      class_name: Zeros
      config: {}
    bias_regularizer: null
    dtype: float32
    kernel_constraint: null
    kernel_initializer:
      class_name: RandomUniform
      config: {maxval: 0.05, minval: -
0.05, seed: null}
    kernel_regularizer: null
    name: dense_1
    trainable: true
    units: 12
    use_bias: true
- class_name: Dense
  config:
    activation: relu
    activity_regularizer: null
    bias_constraint: null
    bias_initializer:
      class_name: Zeros
      config: {}
    bias_regularizer: null
    kernel_constraint: null
    kernel_initializer:
      class_name: RandomUniform
      config: {maxval: 0.05, minval: -
0.05, seed: null}
    kernel_regularizer: null
    name: dense_2
    trainable: true
    units: 8
    use_bias: true
- class_name: Dense
  config:
    activation: sigmoid
    activity_regularizer: null
    bias_constraint: null
    bias_initializer:
      class_name: Zeros
      config: {}
    bias_regularizer: null
    kernel_constraint: null
    kernel_initializer:
      class_name: RandomUniform
      config: {maxval: 0.05, minval: -
0.05, seed: null}
    kernel_regularizer: null
    name: dense_3
    trainable: true
    units: 1
    use_bias: true
keras_version: 2.0.2
```

Рисунок 2 – Модель, сохраненная в формате YAML

Формат HDF5 (h5) - формат файлов, разработанного для хранения большого объема цифровой информации. Данный формат является наиболее популярным, так как выполняет сохранение всей модели с помощью одной функции. Сохранение модели в данном формате в API Keras выполняется с помощью функции `save()`. Файл с сохраненной моделью является бинарным, поэтому просмотреть его структуру полностью в понятном виде невозможно. На рис. 3 представлена часть структуры сохраненной модели в формате HDF5.

[illegible]

Рисунок 3 – Модель, сохраненная в формате HDF5 (h5)

Обсуждение проблематики сохранения моделей нейронных сетей в рамках универсального редактора нейронных сетей

Форматы сохранения нейронных сетей – необходимая составляющая приложения, работающего с нейронными структурами.

В этом случае они используются для хранения созданных моделей, передачи данных между различными приложениями.

Важную функцию форматы нейронных сетей выполняют в приложениях оптимизации и синтеза нейронных сетей. Так, приложения генетической оптимизации или синтеза нейронных сетей предполагают автоматическую генерацию нейронной сети по заданным параметрам. Параметрами для синтеза нейронной структуры могут выступать: количество нейронов, количество нейронных слоёв и др.

Важная роль при синтезе нейронной сети может отводиться задаче, при которой по заданной специфике конкретной предметной области использования нейронной сети (т.е. ее назначению), должна быть сгенерирована оптимальная структура нейронной сети. Основными задачами, решаемыми с помощью нейронных сетей, являются задачи классификации, кластеризации, регрессии и снижения размерности.

Результатом генерации нейронной структуры является файл модели, сохраненный в некотором формате.

В этом случае формат представления нейронной сети должен соответствовать специфике представления нейронной сети в такого рода приложениях.

Предлагается сделать доступной пользователю функции выбора из существующих форматов или - задание собственного формата, в котором необходимо сохранить нейронную сеть.

Можно отметить, что существующие форматы сохранения нейронных сетей в полной мере исполняют свои функции. Однако, одной из наиболее перспективных задач является создание универсального формата сохранения моделей. Важной чертой такого формата является сохранение не только структуры нейронной сети, но и её весов. Сохранение модели и ее весов должно быть доступно как после завершения обучения нейронной сети, так и в процессе ее обучения. Такой формат должен быть приемлем для любого приложения, реализующего функционал нейронных сетей.

Далее пользователь может взаимодействовать с моделью посредством использования нейронного конструктора.

Как частный путь решения данной проблематики можно рассмотреть введение требования, при котором приложение также должно иметь открытый исходный код, что позволит пользователям использовать его фрагменты и функции в качестве API для работы с нейронными сетями и, в частности, с экспортом и импортом моделей нейронных сетей.

Заключение

В данной статье был проведен анализ существующих конструкторов нейронных сетей: NeuroSolutions, NeuralWorks Professional, NeuroShell2. Также были рассмотрены проблемы сохранения моделей нейронных сетей, изучены основные форматы сохранения моделей нейронных сетей: JSON, YAML, HDF5.

Вследствие анализа выделены основные и расширенные требования к проектируемому конструктору нейронных сетей.

Была достигнута цель данной статьи - выявление требований к универсальному конструктору нейронных сетей с основным упором на поиск решения проблем импорта и экспорта готовых моделей нейронных сетей.

В статье были выполнены все поставленные задачи: анализ существующих конструкторов нейронных сетей; определение проблематики развития редакторов нейронных сетей; выделение основных и расширенных требований для универсального (нового) нейронного конструктора; анализ основных форматов сохранения нейронных сетей; обсуждение проблем сохранения моделей нейронных сетей и в рамках универсального редактора нейронных сетей.

На данном этапе можно приступить к реализации некоторых перечисленных требований с основным упором на решение проблем импорта и экспорта готовых моделей нейронных сетей.

Литература

1. Цаунит, А. Н. Перспективы развития и применения нейронных сетей / А. Н. Цаунит. — Текст: непосредственный // Молодой ученый. — 2021. — № 23 (365). — С. 114-117. — Режим доступа: <https://moluch.ru/archive/365/81791/>
2. Нейронные сети и глубокое обучение. Учебный курс / Чару Аггарвал - Изд. 2-е, испр. - Москва [и др.] : Вильямс, 2020. - 752 с.
3. Нейронные сети: полный курс / Саймон Хайкин ; [пер. с англ. Н. Н. Куссуль, А. Ю. Шелестова]. - Изд. 2-е, испр. - Москва [и др.] : Вильямс, 2008. - 1103 с.
4. Основные концепции нейронных сетей / Роберт Каллан; [Пер. с англ. А.Г. Сивака]. - М. [и др.] : Вильямс, 2001. - 287 с.
5. Денисов С.В. Конструктор нейронных сетей/ С.В. Денисов, С.Ю. Иванов // Новые информационные технологии в автоматизированных системах. – 2014. – С. 86 – 92.
6. Круглов В.В. Искусственные нейронные сети. Теория и практика / В.В. Круглов, В.В. Борисов. - 2-е изд., стереотип. – М.: Горячая линия-Телеком, 2002. – 382 с.
7. Джулли А., Пал С. 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.: ил. — Режим доступа: https://www.rulit.me/data/programs/resources/pdf/Dzhulli_Biblioteka-Keras-instrument-glubokogo-obucheniya_RuLit_Me_603503.pdf

Савельева Д.А., Григорьев А.В. О задаче проектирования конструктора нейронных сетей с возможностью экспорта и импорта моделей различных форматов. Проанализированы лучшие практики конструкторов нейронных сетей. Выделена проблематика редакторов нейронных сетей. Рассмотрены основные проблемы и форматы сохранения моделей нейронных сетей. Выделены основные требования для нового универсального конструктора нейронных сетей с возможностью экспорта и импорта моделей различных форматов.

Ключевые слова: нейронная сеть, JSON, конструктор, модель, YAML, HDF5.

Savelyeva D.A., Grigoriev A.V. About the task of designing a neural network constructor with the ability to export and import models of various formats. The best practices of neural network designers are analyzed. The problems of neural network editors are highlighted. The main problems and formats of saving neural network models are considered. The main requirements for a new universal neural network designer with the ability to export and import models of various formats are highlighted.

Keywords: neural network, JSON, constructor, model, YAML, HDF5.

Динамичный расчет твердотельной детали с применением метода конечных элементов

Н.А.Горбунов *1, А.В. Григорьев *2

*1 бакалавриат, Донецкий национальный технический университет,
vrra.email@gmail.com

*2 к.т.н., доцент, Донецкий национальный технический университет,
grigorievalvl@gmail.com

Горбунов Н.А., Григорьев А.В. Динамичный расчет твердотельной детали с применением метода конечных элементов. В данной статье рассмотрен порядок практического применения метода конечных элементов, описано преимущества метода и его особенности, рассмотрен метод структурного анализа методом конечных элементов. Были затронуты метод прямого равновесия, методы работы или энергии, принцип виртуальной работы, методы взвешенных остатков, компьютерные методы решения задач. Сделаны выводы.

Ключевые слова: МКЭ, уравнение, узел, точка, тело, элемент, анализ, функция, матрица, метод, деформация, напряжение, задача.

Введение

Метод конечных элементов - это численный метод решения задач инженерной и математической физики. Типичные проблемные области, представляющие интерес в инженерной и математической физике, которые решаются с помощью метода конечных элементов, включают структурный анализ, теплопередачу, поток жидкости, массоперенос и электромагнитный потенциал.

Конечно-элементная постановка задачи приводит к системе одновременных алгебраических уравнений для решения, а не требует решения дифференциальных уравнений. Эти численные методы дают приближительные значения неизвестных в дискретном числе точек континуума. Следовательно, этот процесс моделирования тела путем разделения его на эквивалентную систему меньших тел или единиц (конечных элементов), соединенных в точках, общих для двух или более элементов (узловых точек или узлов) и / или граничных линий и/или поверхности называется дискретизацией. В методе конечных элементов вместо решения задачи для всего тела за одну операцию мы формулируем уравнения для каждого конечного элемента и объединяем их, чтобы получить решение для всего тела.

Как правило, для решения задачи анализа структурных напряжений инженер стремится определить перемещения и напряжения по всей конструкции, которая находится в равновесии и подвергается приложенным нагрузкам. Для многих конструкций трудно определить распределение деформации с помощью обычных методов, и поэтому обязательно используется метод конечных элементов.

Метод конечных элементов является одним из наиболее распространенных методов решения задач математической физики. Это связано с большой универсальностью метода, сочетающего в себе лучшие качества вариационных и разностных методов.

Актуальной является задачи популяризация метода, его модификация для улучшения его эффективности, а так же - поиск новых областей его применения.

Цель предлагаемой работы заключается в том, что бы донести до широкой аудитории принцип работы данного метода, его сильные и слабые стороны.

Предполагается решение следующих задач:

- подробно проанализировать метод конечных элементов,
- описать математическую часть данного метода;
- рассмотреть варианты решения задач, возникающих в ходе работы метода;
- описать основные преимущества и перспективы применения метода.

Порядок практического применения метода конечных элементов

Ниже представлены шаги, цель изложения этих общих шагов состоит в том, чтобы познакомить вас с процедурой, которой обычно следуют при постановке задачи с использованием конечных элементов.

Шаг 1. Дискретизация и выбор типа элементов. Включает в себя разделение тела на эквивалентную систему конечных элементов со связанными узлами и выбор наиболее подходящего типа элемента для наиболее точного моделирования фактического физического поведения. Элементы должны быть сделаны достаточно

маленькими, чтобы давать полезные результаты, и в то же время достаточно большими, чтобы уменьшить вычислительные затраты. Дискретизированное тело или сетка часто создаются с помощью программ генерации сетки или программ препроцессора, доступных пользователю. Выбор элементов, используемых при анализе методом конечных элементов, зависит от физического строения корпуса в реальных условиях нагрузки и от того, насколько близким к фактическому поведению аналитик хочет видеть результаты.

Шаг 2. Выбор функции перемещения. Функция определяется внутри элемента с использованием узловых значений элемента. Линейные, квадратичные и кубические многочлены являются часто используемыми функциями, поскольку с ними просто работать в формулировке конечных элементов. Однако можно также использовать тригонометрические ряды. Для двумерного элемента функция смещения является функцией координат в его плоскости (скажем, плоскости x-y). Одна и та же общая функция перемещения может использоваться повторно для каждого элемента. Следовательно, метод конечных элементов - это метод, в котором непрерывная величина, такая как перемещение по всему телу, аппроксимируется дискретной моделью, состоящей из набора кусочно-непрерывных функций, определенных в каждой конечной области или конечном элементе.

Шаг 3. Определение деформации/смещения и напряжения/деформация отношения. Необходимы для получения уравнений для каждого конечного элемента. В случае одномерной деформации, скажем, в направлении x, мы имеем деформацию ϵx , связанную со смещением u посредством

$$\epsilon x = \frac{du}{dx} \quad (1)$$

для небольших деформаций.

Кроме того, напряжения должны быть связаны с деформациями с помощью закона напряжений/деформаций, обычно называемого определяющим законом. Способность точно определять поведение материала наиболее важна для получения приемлемых результатов. Простейший из законов напряжения / деформации, закон Гука, который часто используется в анализе напряжений, задается

$$\sigma_x = E \epsilon_x \quad (2)$$

где σ_x = напряжение в направлении x и E модуль упругости.

Шаг 4. Выведение матрицы жесткости элемента и уравнения. Первоначально основывалась на концепции коэффициентов влияния жесткости, которая предполагает опыт структурного анализа. Мы немного затронем альтернативные методы, которые не требуют такой специальной подготовки.

Метод прямого равновесия. В соответствии с этим методом матрица жесткости и уравнения элементов, связывающие узловые усилия с узловыми перемещениями, получаются с использованием условий силового равновесия для базового элемента, наряду с соотношениями сила/деформация.

Методы работы или энергии. Для разработки матрицы жесткости и уравнений для двух и трехмерных элементов гораздо проще применить метод работы или энергии. Принцип виртуальной работы (с использованием виртуальных перемещений), принцип минимальной потенциальной энергии и теорема Кастильяно - это методы, часто используемые с целью вывода уравнений элементов.

Принцип виртуальной работы, применим к любому поведению материала, тогда как принцип минимальной потенциальной энергии и теоремы Кастильяно применимы только к эластичным материалам. Кроме того, принцип виртуальной работы может быть использован даже тогда, когда потенциальная функция не существует. Однако все три принципа приводят к идентичным элементным уравнениям для линейно-упругих материалов; таким образом, какой метод использовать для такого рода материалов при структурном анализе, в значительной степени зависит от удобства и личных предпочтений.

Методы взвешенных остатков. Полезны для разработки уравнений элементов; особой популярностью пользуется метод Галеркина. Эти методы дают те же результаты, что и энергетические методы, везде, где энергетические методы применимы. Они особенно полезны, когда такой функционал как потенциальная энергия недоступен. Методы взвешенных остатков позволяют применять метод конечных элементов непосредственно к любому дифференциальному уравнению. Использование таких методов приведет к получению уравнений для описания поведения элемента. Эти уравнения удобно записать в матричном виде в виде

$$\begin{Bmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_n \end{Bmatrix} = \begin{bmatrix} k_{11} & k_{12} & k_{13} & \dots & k_{1n} \\ k_{21} & k_{22} & k_{23} & \dots & k_{2n} \\ k_{31} & k_{32} & k_{33} & \dots & k_{3n} \\ \vdots & & & & \vdots \\ k_{n1} & & & \dots & k_{nn} \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_n \end{Bmatrix} \quad (3)$$

или в компактной матричной форме в виде

$$\{f\} = [k]\{d\} \quad (4)$$

где $\{f\}$ - вектор узловых сил элемента, $[k]$ - матрица жесткости элемента (обычно квадратная и симметричная), а $\{d\}$ - вектор неизвестных узловых степеней свободы элемента или обобщенных перемещений, п. Здесь обобщенные перемещения могут включать такие величины, как фактические перемещения, наклоны или даже искривления.

Шаг 5. Сбор уравнения элементов, чтобы получить глобальные или суммарные уравнения и ввести граничные условия. На этом этапе уравнения узлового равновесия отдельных элементов, сгенерированные на шаге 4, объединяются в глобальные уравнения узлового равновесия. Окончательное собранное или глобальное уравнение, записанное в матричном виде, является

$$\{F\} = [K] \{d\} \quad (5).$$

где $\{F\}$ - вектор глобальных узловых сил, $[K]$ - глобальная матрица структуры или общая матрица жесткости (для большинства задач глобальная матрица жесткости квадратная и симметричная), а $\{d\}$ теперь является вектором известных и неизвестных узловых степеней свободы структуры или обобщенных перемещений. Можно сказать, что на данном этапе глобальная матрица жесткости $[K]$ является сингулярной матрицей, поскольку ее определитель равен нулю. Чтобы устранить эту проблему сингулярности, мы должны вызвать определенные граничные условия (или ограничения или опоры), чтобы конструкция оставалась на месте, а не перемещалась как твердое тело. На данный момент достаточно отметить, что вызов граничных или вспомогательных условий приводит к модификации глобального уравнения (5). Также необходимо подчеркиваем, что приложенные известные нагрузки были учтены в глобальной матрице сил $\{F\}$.

Шаг 6. Решение для неизвестных степени свободы (или Обобщенные перемещения) Уравнение (5), модифицированное для учета граничных условий, представляет собой набор одновременных алгебраических уравнений, которые могут быть записаны в развернутой матричной форме как

$$\begin{Bmatrix} F_1 \\ F_2 \\ \vdots \\ F_n \end{Bmatrix} = \begin{bmatrix} K_{11} & K_{12} & \dots & K_{1n} \\ K_{21} & K_{22} & \dots & K_{2n} \\ \vdots & \vdots & & \vdots \\ K_{n1} & K_{n2} & \dots & K_{nn} \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{Bmatrix} \quad (6)$$

где n - общее число неизвестных узловых степеней свободы структуры. Эти уравнения могут быть решены для d_s с помощью метода исключения (такого как Метод Гаусса) или итеративный метод (такой как метод Гаусса–Зайделя). d_s называются первичными неизвестными, потому что они являются первыми величинами, определенными с использованием метода конечных элементов жесткости (или смещения).

Шаг 7. Решение для деформаций и напряжений элемента. Для задачи анализа структурных напряжений могут быть получены важные вторичные величины деформации и напряжения (или момента и силы сдвига), поскольку они могут быть непосредственно выражены в терминах перемещений, определенных на шаге 6. Типичные соотношения между деформацией и смещением и между напряжением и деформацией - такие как уравнения (1) и (2) для одномерного напряжения, приведенного на шаге 3, — могут быть использованы.

Шаг 8. Интерпретирование результата. Конечная цель состоит в интерпретации и анализе результатов для использования в процессе проектирования / анализа. Определение мест в конструкции, где возникают большие деформации и напряжения, как правило, важно при принятии проектных/аналитических решений. Компьютерные программы постпроцессора помогают пользователю интерпретировать результаты, отображая их в графической форме.

Преимущества метода

Этот метод обладает рядом преимуществ, которые сделали его очень популярным. Они включают в себя способность:

1. Довольно легко моделировать тела неправильной формы;
2. Справляться с общими условиями нагрузки без затруднений;
3. Моделировать тела, состоящие из нескольких различных материалов, поскольку уравнения элементов оцениваются индивидуально;
4. Обработать неограниченное количество и виды граничных условий;
5. Варьировать размер элементов, чтобы при необходимости можно было использовать небольшие элементы;
6. Изменять модель конечных элементов относительно легко и дешево;
7. Включать динамические эффекты;
8. Справляться с нелинейным поведением, существующим при больших деформациях и нелинейных материалах.

Метод структурного анализа методом конечных элементов позволяет проектировщику выявлять проблемы с напряжением, вибрацией и температурой в процессе проектирования и оценивать изменения в конструкции до создания возможного прототипа. Таким образом, повышается уверенность в приемлемости прототипа. Более того, при правильном использовании этот метод может сократить количество прототипов, которые необходимо построить. Несмотря на то, что метод конечных элементов первоначально использовался для структурного анализа, с тех пор он был адаптирован ко многим другим инженерным и математическим дисциплинам физика, такая как поток жидкости, теплопередача, электромагнитные потенциалы, механика грунта и акустика.

Существует два общих компьютерных метода подхода к решению задач методом конечных элементов. Один из них заключается в использовании крупных коммерческих программ, многие из которых были сконфигурированы для запуска на персональных компьютерах (ПК); эти программы общего назначения предназначены для решения многих типов проблем. Другой заключается в разработке множества небольших программ специального назначения для решения конкретных проблем.

Метод конечных элементов уже реализован во многих системах автоматического проектирования для анализа деформаций, таких как CATIA где над деталью можно проводить эксперименты с разными видами воздействия, на данный момент доступны три вида анализа:

1) Частотный анализ. Для проведения анализа достаточно создать деталь и назначить ей материал, после чего перейти через подпункт меню на экран эксперимента. Необходимо зафиксировать деталь с помощью соответствующего инструмента и указать массу (если необходимо) (см. рис. 1).

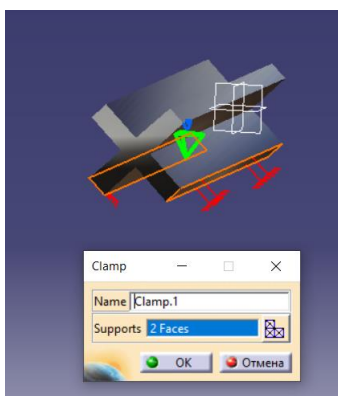


Рисунок 1

После завершения подготовки можем провести вычисления. Для этого необходимо воспользоваться инструментом «Compute», можно так же составить детальный отчет с расчетами с помощью инструмента Analysis Results», где в первой части отчета будут указаны параметры сетки, тип элемента и его качество, расчет материала. Во второй части будет приведен расчет структуры, сдержанности и структурной массы (см. рис. 2).

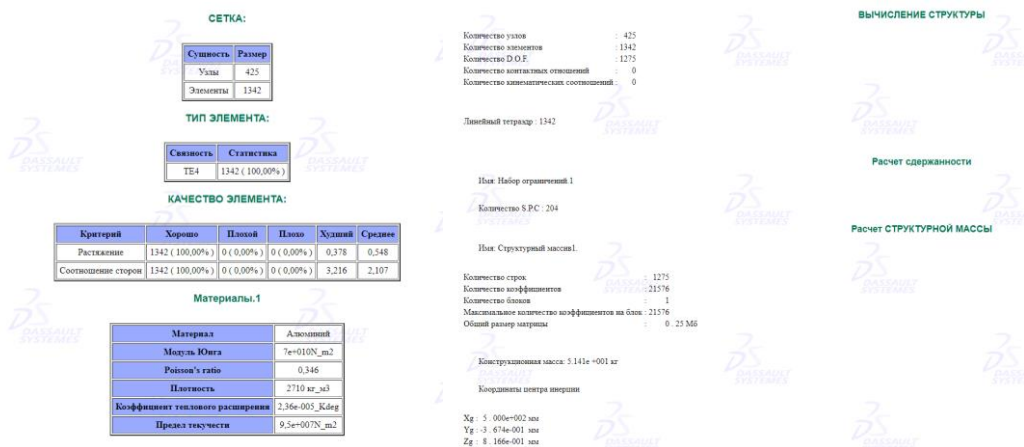


Рисунок 2

После расчетов формируются три таблицы: Минимальный и максимальный разворот, минимальный разворот, поступательное распределение точек поворота (см. рис. 3).

Матричный и массовый разворот					
Заголовок	Dof	Узел	x (мм)	y (мм)	z (мм)
5.7841e+008	Ty	332	4.3663e+002	-3.2270e+001	6.2270e+001
2.1444e+010	Tz	139	2.4995e+002	1.0000e+002	7.0000e+001

Матричный разворот					
Заголовок	Dof	Узел	x (мм)	y (мм)	z (мм)
6.7090e+008	Ty	106	0.0000e+000	2.2466e+001	-2.2466e+001
6.9738e+008	Ty	192	2.9026e+002	-2.1212e+000	3.2121e+001
7.6369e+008	Tz	283	2.4995e+002	6.2513e+001	-3.2513e+001
7.9094e+008	Tz	194	4.1589e+002	-2.1212e+000	3.2121e+001
1.0174e+009	Tz	202	9.3759e+002	-2.1212e+000	3.2121e+001
1.0333e+009	Tz	364	8.6807e+002	-6.6205e+001	6.9420e+001
1.0800e+009	Tz	138	0.0000e+000	7.1656e+001	-4.1656e+001
1.0931e+009	Tz	332	4.3663e+002	-3.2270e+001	6.2270e+001
1.1141e+009	Ty	332	4.3663e+002	-3.2270e+001	6.2270e+001

Заголовок	В процентах
10.E9 -> 10.E9	4.665e+001
10.E9 -> 10.E10	8.608e+001
10.E10 -> 10.E11	1.3445e+001

Рисунок 3

Третья часть отчета эта вычисление частоты. Генерируется две таблицы с десятью строками (по числу экспериментов). Последняя часть отчета показывает, как деталь деформируется во всех 10 сценариях (см. рис. 4). Похожие отчеты будут генерироваться и для других анализов.

Визир режима	Частота Гц	Стабильность
1	2.7781e+003	3.8114e+010
2	3.0128e+003	1.0050e+008
3	3.5055e+003	1.1080e+007
4	3.5542e+003	2.3693e+007
5	4.1181e+003	6.9453e+007
6	4.3731e+003	2.9383e+008
7	4.4454e+003	2.7935e+008
8	4.5746e+003	2.2366e+007
9	5.0868e+003	1.5106e+004
10	5.4140e+003	4.5533e+004

Модальное участие:

Режим	Частота Гц	Tx (%)	Ty (%)	Tz (%)	Rx (%)	Ry (%)	Rz (%)
1	2.7781e+003	0.02	21.18	2.13	45.92	1.14	17.45
2	3.0128e+003	0.24	1.87	0.07	2.37	0.96	16.71
3	3.5055e+003	71.82	0.00	0.00	0.00	1.62	0.12
4	3.5542e+003	1.42	0.07	0.00	0.04	0.02	0.23
5	4.1181e+003	0.01	0.25	8.87	0.23	6.54	0.16
6	4.3731e+003	0.04	0.02	0.07	0.03	0.44	0.71
7	4.4454e+003	0.01	1.34	33.23	1.22	23.83	0.74
8	4.5746e+003	3.30	0.01	0.01	0.00	10.70	0.22
9	5.0868e+003	0.00	0.01	0.02	0.00	0.03	0.02
10	5.4140e+003	0.12	0.11	0.00	0.00	0.01	0.16
Всего		76.61	34.86	44.41	49.80	45.28	36.64

Рисунок 4

После закрытия отчета мы увидим деформированную деталь, полученную при первом эксперименте. CATIA позволяет переключать вид детали, изменив его на отображение сетки, стрессовых зон или деформации по точкам (см. рис. 5).

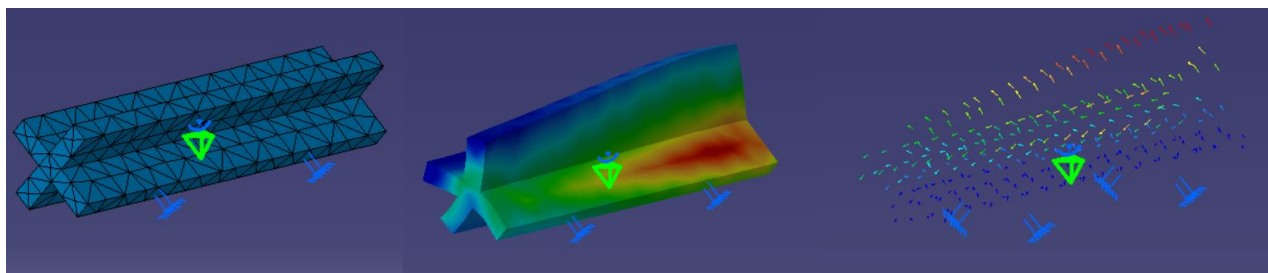


Рисунок 5

Если использовать инструмент «Animate» можно увидеть, как деталь деформируется в режиме реального времени и контролировать анимацию с помощью меню похожего на плеер. При желании можно переключаться через пункты меню и просматривать результаты других экспериментов.

2) Анализ свободной частоты. Главное отличие от обычного частотного анализа состоит в том, что наша деталь находится внутри некоего механизма и движется как бы внутри него, поэтому нам не нужно фиксировать деталь и можно сразу приступить к расчетам. Для этого используем инструмент «Compute», после генерации отчета воспользуемся меню анимации и получим следующий результат (см. рис. 6).

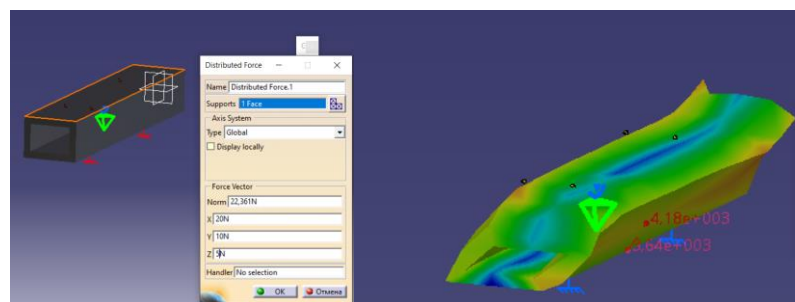


Рисунок 9

Пункт Acceleration позволяет моделировать силы тяжести на деталь. Для использования команды необходимо выбрать деталь, и назначить ускорения свободного падения по различным осям. Будем предполагать что деталь летит вниз без отклонений в горизонтальном положении и установим соответствующие значение . Получим вид деформированной детали после расчета (см. рис. 10).

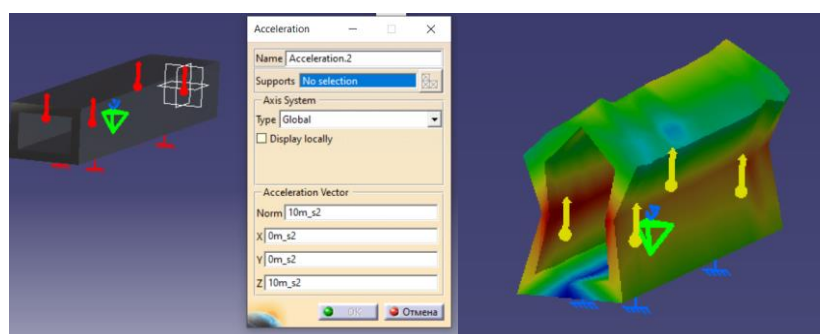


Рисунок 10

Пункт Line Force Density обеспечивает распределенное давление по краям элемента. После выбора команды необходимо выбрать ребра приложения силы и значения ее по разным осям использовать инструмент «Compute» (см.рис.11).

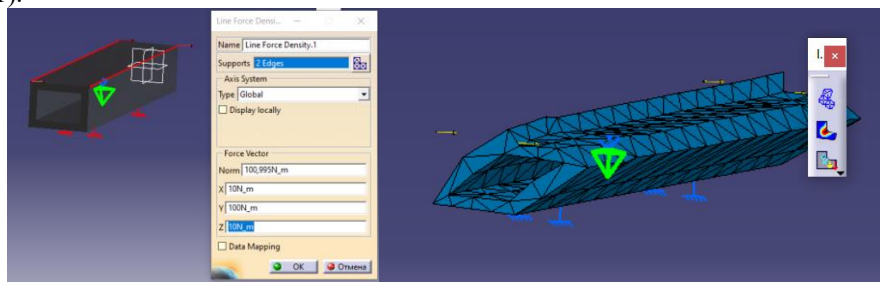


Рисунок 11

Пункт Enforced Displacement показывает, как реагирует деталь в момент удара. После выбора команды необходимо выбрать тиски, которые фиксируют деталь, указать значения и рассчитать. Внешне деталь никак не меняется , показывается только напряжение на участок детали в момент удара (см. рис. 12).

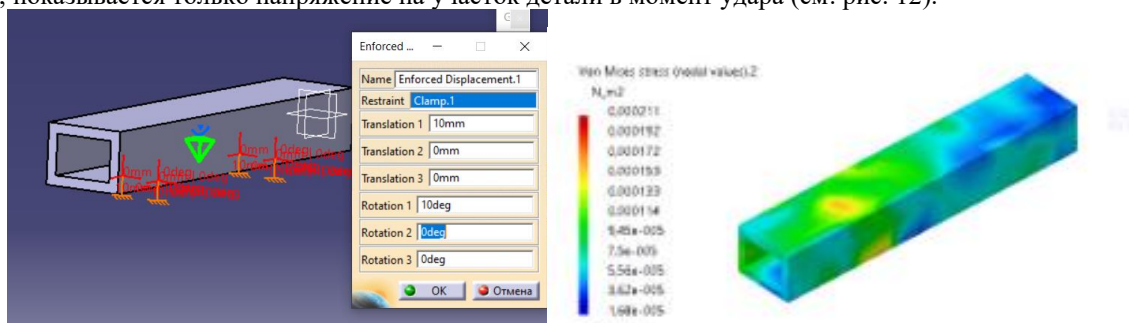


Рисунок 12

Пункт Temperature Field позволяет рассчитать температурные деформации тела. После появления меню настройки необходимо выбрать тело, указать температуру в Кельвинах и использовать инструмент «Compute», после расчетов получим следующий результат (см. рис. 13).

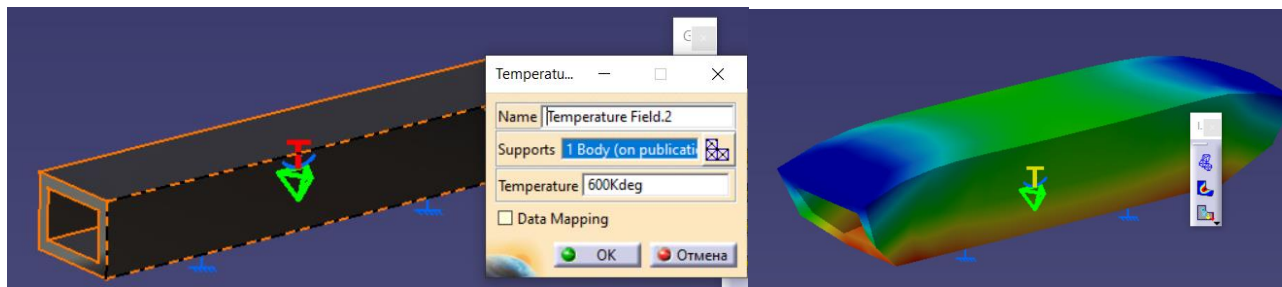


Рисунок 13

Пункт Combined loads. Если необходимо провести эксперимент с несколькими типами воздействий одновременно в CATIA это можно сделать. Для этого на нашу деталь окажем воздействие прессом и давлением применив соответствующие инструменты. Используем инструмент «Compute», мы можем увидеть, что при деформации учитываются оба воздействия (см. рис. 14).

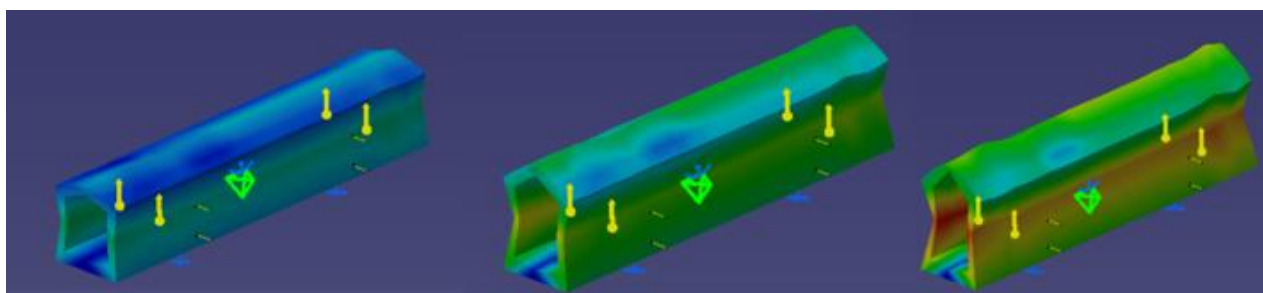


Рисунок 14

Стоит отметить, что метод конечных элементов не ограничен только этими видами анализа. В будущем с развитием ПО станет возможным применения данного метода для любых деформаций элементов с учетом всех условий окружающей среды.

Такие факторы как коррозия, циклические нагрузки, засоры, усталость металла и т.д. станут доступны к изучению непосредственно на стадии проектирования, что позволит снизить затраты на ремонт и расходы связанные с эксплуатацией оборудования, повысить производительность и качество готовых изделий.

Заключение

В настоящее время метод конечных элементов (МКЭ) прочно занимает лидирующее положение в практике инженерных расчетов строительных конструкций, зданий и сооружений, является мощным инструментом при проведении научных исследований.

Огромной популярности метода способствовали следующие преимущества метода: общность подхода при решении различных задач расчета конструкций (в том числе сложных, состоящих из различных конструктивных элементов различной мерности), относительная простота учета взаимодействия конструкции с окружающей средой (механические, температурные, коррозионные воздействия, граничные условия и т.д.), высокая степень приспособляемости к автоматизации всех этапов расчета, естественная механическая трактовка и возможность построения моделей на основе физического, а не математического подхода, возможности рассмотрения неоднородных тел произвольной природы и т.д.

В статье выполнены следующие задачи: изложен метод конечных элементов, подробно описаны варианты решения задач возникающих в ходе работы метода на каждом шаге алгоритма, выделены преимущества применения, выделены основные причины популярности и перспективы развития метода.

Литература

1) Основные формулировки метода конечных элементов в задачах строительной механики. [Электронный ресурс] // КОД - Режим доступа <https://cyberleninka.ru/article/n/osnovnye-formulirovki-metoda-konechnykh-elementov-v-zadachah-stroitelnoy-mehaniki-chast-1/viewer>

2) Метод конечных элементов [Электронный ресурс] // КОД - Режим доступа <https://studfile.net/preview/16408030/page:11/>

3) Решение дифференциальных задач методом конечных элементов [Электронный ресурс] // КОД - Режим доступа https://irbis.amursu.ru/DigitalLibrary/AmurSU_Edition/11280.pdf

4) Математика метода конечных элементов [Электронный ресурс] // КОД - Режим доступа <https://fea.ru/docs/FEM/Rozin.pdf>

Горбунов Н.А., Григорьев А.В. Динамичный расчет твердотельной детали с применением метода конечных элементов. В данной статье рассмотрены порядок практического применения метода конечных элементов, описаны преимущества метода и его особенности, рассмотрен метод структурного анализа методом конечных элементов. Были затронуты метод прямого равновесия, методы работы или энергии, принцип виртуальной работы, методы взвешенных остатков, компьютерные методы решения задач. Сделаны выводы.

Ключевые слова: МКЭ, уравнение, узел, точка, тело, элемент, анализ, функция, матрица, метод, деформация, напряжение, задача.

Gorbunov N.A., Grigoriev A.V. Dynamic calculation of a solid-state part using the finite element method. In this article, the order of practical application of the finite element method is considered, the advantages of the method and its features are described, the method of structural analysis by the finite element method is considered. The method of direct equilibrium, methods of work or energy, the principle of virtual work, methods of weighted balances, computer methods of solving problems were touched upon. Conclusions are drawn.

Keywords: FEM, equation, node, point, body, element, analysis, function, matrix, method, deformation, stress, problem.

Анализ программного обеспечения логистики в разрезе модификации и синтеза требований для создания универсального оптимизированного программного продукта

А.В. Гончаров^{*1}, А.В. Григорьев^{*2}

^{*1} бакалавр, Донецкий национальный технический университет,
artemvalerevich2704@gmail.com

^{*2} к.т.н, доцент, Донецкий национальный технический университет,
grigorievalvl@gmail.com

Гончаров А.В., Григорьев А.В. Анализ программного обеспечения логистики в разрезе пересмотра старых и синтеза новых требований для создания универсального оптимизированного программного продукта. В статье проведен анализ программного продукта для обеспечения потребностей транспортной логистики, выполнена оценка функционала и отдельных критических модулей системы, выявлены основные векторы в практической плоскости, выдвинуты новые требования для создания универсального оптимизированного продукта.

Ключевые слова: системы управления производством, системы принятия решений в реальном времени, логистика, оценка, синтез, функционал, модули, требования к программному обеспечению.

Введение

Обеспечение логистической стабильности импортируемой и экспортируемой продукции в рамках производства - ключевая задача, без которой не может существовать предприятие. Система управления транспортными перевозками (Transportation Management System) позволяет получить контроль в реальном времени с момента предварительного просчета и оценки стоимости перевозки и, собственно, поиска необходимого транспорта под груз до момента выгрузки как по времени и во времени (сроки прибытия на склад в соответствии с меняющимися исходными данными). Информационная система так же позволяет контролировать перевозку с точки зрения гео-данных о местоположении зафрахтованного транспорта, данных о соответствии наименований искомого товара товаро-транспортным накладным международного образца (CMR), а так же данным о готовности сопроводительной документации. Товаро-транспортная накладная международного образца необходима только в случаях перемещения груза в другое государство, что сопровождается таможенными процедурами установленного порядка в рамках законодательства.

Существует несколько наиболее популярных на данный момент TMS-систем, среди которых очевидными лидерами являются "1С: Предприятие 8. TMS Логистика. Управление перевозками" и "Axelot". Далее в ходе работы мы проанализируем "1С: Предприятие 8. TMS Логистика. Управление перевозками". TMS-система от компании "1С: Предприятие" имеет больше перспектив для ее развития и доработки по причине более простой логической организации в отличие от продукта "Axelot", а так же ориентированности на наемный транспорт, в то время как в "Axelot" выделено более половины компонентов-модулей для собственного автопарка. Так же немаловажным фактором в пользу TMS-системы от компании "1С: Предприятие" является более лояльная ценовая политика на варианты приобретения лицензии.

Цели работы:

- сформировать понимание базовых задач и механизмов работы отдела логистики;
- провести анализ критических механизмов и модулей программного продукта в разрезе бизнес процессов;
- выполнить оценку функционала модулей программного обеспечения;
- модифицировать базовые требования путем внесения качественно новых задач, продиктованных рассмотренными в ходе анализа бизнес процессами;
- определить какие из модулей должны быть подвергнуты модификации и степень таковой;
- дать прогнозируемый результат внесения изменений в модули системы в соответствии с обновленными требованиями.

Структура и задачи типичных программных пакетов логистики

Основная задача логистического отдела предприятия может быть сформулирована следующим образом: доставить груз в кратчайшие сроки в соответствии с требованиями собственного производства, конечных потребителей - клиентов, законодательства, международных стандартов в сфере грузоперевозок, экспедирования грузов и выполнения таких операций как логистическое сопровождение, подготовка транспортной документации и путевых листов [1].

Основная задача программного комплекса логистики состоит в том, чтобы предоставить комфортное место работы для сотрудника с максимальной информативностью, гибкостью настройки под текущие задачи, онлайн мониторингом отправных данных и параметров. Таким образом, система позволяет минимизировать время принятия решений, реагирования на меняющиеся во времени факторы, предотвращая тем самым, сводя к минимуму или полностью исключая финансовые издержки на этапе перевозки грузов по многим аспектам. Такими аспектами могут быть: штрафные санкции от органов правопорядка и надзора за движением на дорогах и магистралях всех уровней значения, компенсации материального и иного характера по заключенным договорам вследствие порчи груза или задержки доставки и т.д. [2].

Структура пакета программного обеспечения отдела логистики (см. рис. 1).

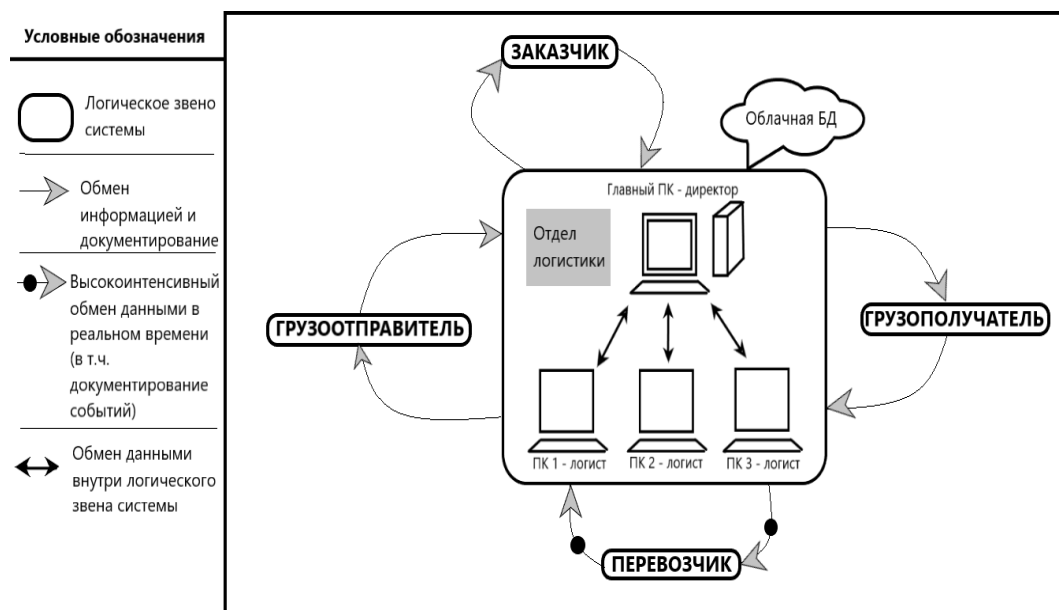


Рисунок 1 - Структура работы пакета

Выявление и формирование требований к программному комплексу на основе существующей реализации на информационном рынке - 1С:Предприятие 8. TMS Логистика. Управление перевозками

Этап первый - это первичный просчет километражей маршрута и альтернативных маршрутов, просчет себестоимости маршрута пустым и груженым, просчет стоимости пути от средней цены по рынку услуг грузоперевозок.

Требования на данном этапе к соответствующему программному модулю:

- калькулятор просчета километража по параметрам с возможностью запоминания и структурирования сохраненных вычислений,
- информационная панель для возможности анализа факторов влияющих на ценовую политику в расчете на километр,
- интеграция в форме отдельной информационной панели с биржей грузоперевозок для возможности оценки анализа цен километра по данному маршруту в данный момент времени.

Проанализировав функционал в рамках существующих программных решений обратим внимание на наиболее популярный и наиболее интуитивно простой комплекс - “1С: Предприятие 8. TMS Логистика. Управление перевозками” [3].

В подсистеме “1С:Предприятие 8. TMS Логистика. Управление перевозками” реализованы следующие функции: фиксирование показателей расчета в виде списка, зависящих от одного или двух параметров, формирование правил расчета, установка тарифов услуг и затрат в разрезе видов транспорта и тарификации (см. рис. 2).

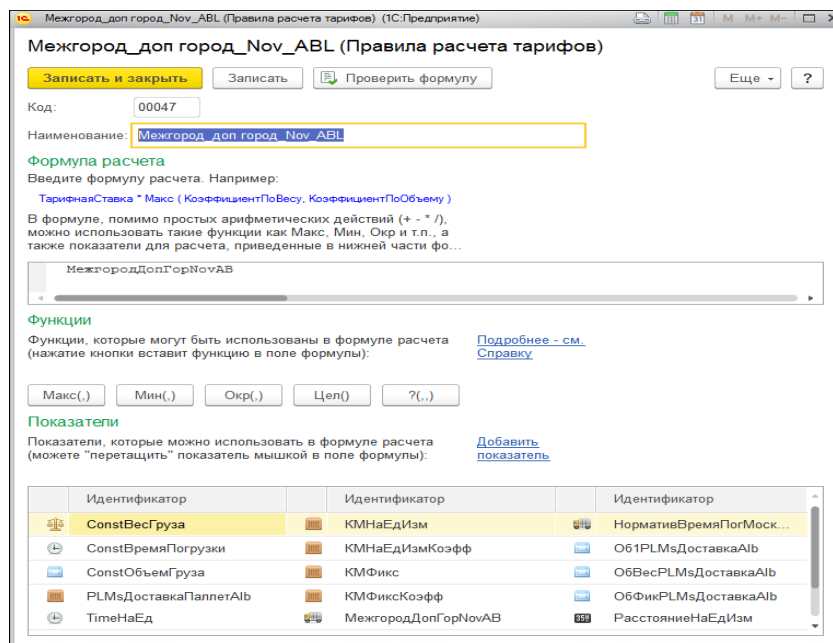


Рисунок 2 - Управление ценообразованием и тарифами [3]

Просчет тарификации по параметрам и установленным правилам имеет высокий коэффициент эффективности, однако реализация сохранения ранее проведенных расчетов не реализована, не говоря уже о структурированном подходе к реализации хранения таких данных. Нагруженность интерфейса модуля расчета не позволяет быстро и эффективно ориентироваться в режимах работы и опциях, что замедляет работу и повышает риск ошибки сотрудника в процессе работы. В “1С:Предприятие 8. TMS Логистика. Управление перевозками” отсутствует интеграция с биржей грузоперевозок, к примеру, с биржей АТІ или иными подобными ресурсами, что делает невозможным анализ цены выведенной по правилам и прежним тарифам компании. Отсутствие описанного ранее функционала может в условиях найма транспорта сделать неконкурентоспособной заявленную ставку на рынке искомых услуг [4].

Определим перспективы решения обозначенных изъянов на практике. Во-первых, дизайн окна рабочего модуля должен быть лаконичным и лишенным информационной нагруженности. Все дополнительные параметры и режимы должны быть реализованы в качестве выпадающего списка для параметров и правил расчета, отдельное меню - для режимов и переключения между группами тарификации. Во-вторых, в рамках рабочей области модуля должен быть реализован доступ к ранее проведенным вычислениям в табличной форме, которая удобна для восприятия и делает материал пригодным для анализа и поиска необходимых данных [5].

Следующий этап в работе отдела логистики - это планирование рейса и поиск исполнителя. На данном этапе формируется потребность в перевозке, происходит постановка задач на перевозку, выполняется анализ существующих задач и их сроков с целью покрытия всех запросов на перевозку во время. Следующим шагом является само нахождение исполнителя которым могут являться: водитель с грузовым авто содержащемся на балансе предприятия, предприятия с которыми уже заключены договора на перевозку и были ими оказаны такого рода услуги ранее или поиск, в случае отсутствия возможности использования ранее упомянутых вариантов, нового перевозчика, который в срок закроет заказы.

Ручная обработка каждого искомого фактора условия выставленного заказчиком снижает эффективность параллельной обработки сотрудниками растущего количества заявок, что требует повышения расходов на штат сотрудников вследствие пропорционального роста количества сотрудников - объема заявок. Таким образом снижается чистый доход предприятия, а так же повышается риск возникновения ошибок связанных с человеческим фактором на любом этапе, в т.ч. там, где это критически не допустимо [6].

В ИС логистики предприятия автоматизирован выбор базовых параметров, полный список которых всегда перед глазами, что снижает риск упустить важное. В случае необходимости можно закрепить при формировании заявки файл присланный заказчиком или комментарий, если информация была передана устно. Таким образом, информация при дальнейшей обработке не будет утеряна, а наоборот - будет всегда доступна для дальнейшего анализа, что снижает время на поиски и риск утраты таковой.

При обработке потребности заказчика все инструменты для таких процедур находятся в рабочей зоне логиста на мониторе, что повышает удобство и снижает расход времени на каждую процедуру, в том числе благодаря отсутствию необходимости использования бумажных носителей, которые считаются не надежными при потоке информации высокой интенсивности (см. рис. 3).

Рисунок 3 - Формирование задания на перевозку в 1С:Предприятие 8. TMS Логистика. Управление перевозками [3]

Организация контроля и мониторинга существующих заданий и потребностей заслуживает отдельного внимания по причине критической важности данного модуля и, в частности, доступного функционала. Проводя анализ модуля контроля и мониторинга, будет актуальным делать это в разрезе двух других модулей: модуля поиска искомого транспорта под существующее задание и модуля контроля и оповещения.

В 1С:Предприятие 8. TMS Логистика. Управление перевозками контроль и мониторинг существующих заданий реализован в рамках информационно-ознакомительной среды (см. рис. 4).

← → ☆ Рабочее место: Контроль прохождения рейсов

Рейсы

Создать Найти... Отменить поиск Создать на основании ... Отчеты

Статус	Дата	Номер	Организация	Перевозчик	Вид ТС	Время начала (план)	Время окончания ...	Вес	Объем	Количество мест
Новый	21.11.2013 10:59:35	000000168	Транссервис	Трасса (Пер...	ЖД транспорт	20.04.2010 12:00:00	21.04.2010 9:00:00	4 500,000	5,748	15,000
Новый	21.11.2013 11:17:26	000000169	Транссервис	Трасса (Пер...	Автотранспорт					
Новый	21.11.2013 11:45:10	000000170	Транссервис	Трасса (Пер...	Автотранспорт	22.07.2010 9:00:00	23.07.2010 18:00:00	3 300,000	1,200	9,000
Выполняется	31.05.2017 18:01:45	0000-0003	Транссервис	Транссервис	Авиа	01.06.2017 0:00:00	02.06.2017 0:00:00	5 000,000		

Маршрут

Записать прохождение точек маршрута Отменить точки

Номер стр...	Состояние	Задание на перевозку груза	Номер ...	Операция	Адрес	Время прибытия	Время убытия	Вес
1	✓ Пройдена	Задание на перевозку груза 0000-000589 от 31.05.2017 ...	1	Погрузка	109386, ...	01.06.2017 0:00:00	01.06.2017 0:00:00	5 000,000
2	Отменена	Задание на перевозку груза 0000-000589 от 31.05.2017 17:56:35	1	Разгрузка	143350, М...	02.06.2017 0:00:00	02.06.2017 0:00:00	5 000,000
3		Задание на перевозку груза 0000-000589 от 31.05.2017 ...	1	Разгрузка	109386, ...	02.06.2017 0:00:00	02.06.2017 0:00:00	5 000,000

Рисунок 4 - Контроль прохождения рейсов и мониторинг существующих заданий [3]

Информационно-ознакомительный подход хорошо подходит для дальнейшего интеллектуального анализа данных сотрудником в течение рабочего дня, однако, существует такая проблема, как внезапные события, которые должны быть отмечены в системе и сотрудник должен быть проинформирован вне зависимости от времени суток. Контроль над такими ситуациями не проработан в должной степени ни в одном программном комплексе среди лидеров рынка этой ниши информационных систем.

Следует отметить, что обработка заданий полностью возлагается на сотрудника и, таким образом, затрачивается больше времени на обработку каждой позиции. По данным, предоставляем фирмой 1С о своем продукте "1С:Предприятие 8. TMS Логистика. Управление перевозками" имеются исключительно два параметра поиска: по существующим в базе данных предприятия фирмам, оказывающим услуги перевозки грузов и с помощью собственного транспорта. Явно прослеживается невозможность оперативно решать поставленные задачи при большой нагрузке на логистический отдел. Решением данной проблемной ситуации может стать интеграция с биржей грузоперевозок АТІ, которая не только содержит базу данных потенциальных исполнителей, но и дает возможность страховать груз, а так же дает полный доступ к оценке каждого

исполнителя - рейтинг перевозчика, складывающийся с момента открытия искомого предприятия до текущего дня [4].

Немаловажный аспект контроля исполнения при ситуации, когда уже найден транспорт, заключается в получении точных координат нахождения транспортного средства на всех этапах. Для этого существует отдельный опциональный модуль GPS/ГЛОНАСС мониторинга за гео-данными. Данный функционал реализован во всех продуктах-лидерах на рынке информационных систем данного назначения.

Завершающий этап перевозки - документирование и оборот документации между заказчиком перевозки - отделом логистики и перевозчиком унифицирован в соответствии со стандартами и не требует необходимости подробного анализа с целью выявления недостатков в практической реализации данного модуля.

Общее видение проблематики гибкости программного продукта

Готовые программные продукты, лидирующие на рынке, обладают тремя факторами, препятствующими интеграции таких информационных систем в малый и средний бизнес:

1. Сложность настройки, требующей привлечения IT-специалистов или даже полного сопровождения таким специалистом деятельности программного комплекса.

2. Часть функционала избыточна для предприятий или компаний с небольшим производством и/или крупным производством, но импорт/экспорт продукции которых не выходит за рамки ограниченной территории (город, район или если предприятие является звеном цепи по производству готовой продукции и остальные звенья находятся в рамках некоторой агломерации или района).

3. Недостаточные привилегии администратора-менеджера в системе.

Фактор первый и третий непосредственно связаны между собой, так как менеджер среднего звена не обладает соответствующей квалификацией и знаниями для доступа к структуре программного комплекса, но это и не требуется. Данную проблему можно решить путем внедрения онлайн технической поддержки в рамках возникновения проблем с работой системы. Таким образом, нет необходимости держать в штате IT-специалиста, что, в рамках малых и средних предприятий делает искомый продукт более выгодным и привлекательным для внедрения. Менеджеру, управляющему отделом, следует передать более широкий спектр возможностей: принудительное резервное копирование данных, мониторинг активности сотрудников подключенных к системе, доступ к данным и ПК каждого сотрудника. Данные возможности должны быть унифицированы и реализованы в рамках подсистем администрирования с отсутствием ввода параметров или данных для использования каждой отдельно взятой подсистемы.

Пункт второй является ключевым в рамках финансовой выгоды решения для фирм, которые в ближайшее время своей работы не планируют использовать весь функционал программного продукта. Данная проблема может быть решена путем индивидуальной сборки модулей под требования определенного заказчика или продукт может поставляться в полном объеме с отключенными модулями, необходимости в которых у заказчика нет. Во втором случае добавление функционала может быть выполнено путем включения соответствующих модулей администратором системы - менеджером без привлечения IT-специалиста [7].

Так же следует отметить необходимость функциональных возможностей от имени администратора системы отключать модули, которые не являются критически важными для работы системы. Данное решение приведет к повышению КПД сотрудников вследствие уменьшения нагруженности интерфейса, если, конечно, надобности в отключенных модулях нет.

Заключение

Согласно полученным данным при анализе и предложенным в ходе анализа решениям необходимо отметить важность внесения оговоренных модулей и возможностей работы программного комплекса при разработке программного решения следующего поколения. Новые модули в рамках существующих практик позволят не только оптимизировать работу в условиях растущего объема перевозок, увеличить прибыль предприятия использующего программный комплекс нового поколения, но и существенно снизить риски, свести к минимуму человеческий фактор в тех информационных потоках и задачах, где ответственность возлагается исключительно на логиста. Новые решения в рамках создания продукта нового поколения позволят открыть новую нишу сбыта для программного комплекса - малый и средний бизнес, которой практически не коснулись лидеры рынка вследствие дороговизны приобретения искомой ИС, ее обслуживания и квалификации персонала. Из этого следует, что стоит поставить акцент при разработке больших программных комплексов на их гибкость в рамках запроса рынка - необходимо поставлять продукт под задачи конкретного конечного пользователя, которые могут значительно отличаться, будь то только открывшееся ИП или логистический отдел порта в рамках крупного холдинга.

Литература

1. Остроумов, А. А. Совершенствование процесса управления перевозками на основе современных информационных систем / А. А. Остроумов, Е. Ф. Мельникова, В. Л. Бордюг // Международный научно-исследовательский журнал. - 2014. - №11-3 (30) - С. 60-63.
2. Левина, Т.В. Системы управления транспортировкой. Февраль 2012. [Электронный ресурс]. Режим доступа: <http://lscm.ru/index.php/ru/po-godam/item/790/> (дата обращения: 06.11.2022).
3. 1С:Предприятие 8. TMS Логистика. Управление перевозками. Возможности продукта. [Электронный ресурс]. Режим доступа: <https://solutions.1c.ru/catalog/tms/features> (дата обращения: 06.11.2022).
4. Биржа грузоперевозок и крупнейшая экосистема сервисов для транспортной логистики в России и СНГ. [Электронный ресурс]. Режим доступа: <https://ati.su/> (дата обращения: 06.11.2022).
5. Человеко-машинный интерфейс: понятие, подходы, принципы. [Электронный ресурс]. Режим доступа: <https://www.4stud.info/user-interfaces/ui-design-intro.html> (дата обращения: 06.11.2022).
6. Бунова, Е. В. Оценка эффективности внедрения информационных систем / Е. В. Бунова, Буслаева О.С. // Вестник АГТУ. Серия: Управление, вычислительная техника и информатика. - 2012. [Электронный ресурс]. Режим доступа: <https://cyberleninka.ru/article/n/otsenka-effektivnosti-vnedreniya-informatsionnyh-sistem> (дата обращения: 08.11.2022).
7. Методы упрощения систем. [Электронный ресурс]. Режим доступа: https://studopedia.ru/8_131665_metodi-uproshcheniya-sistem.html?ysclid=la7bnq9ax1204640841 (дата обращения: 06.11.2022).

Гончаров А.В., Григорьев А.В. Анализ программного обеспечения логистики в разрезе пересмотра старых и синтеза новых требований для создания универсального оптимизированного программного продукта. В статье проведен анализ программного продукта для обеспечения потребностей транспортной логистики, выполнена оценка функционала и отдельных критических модулей системы, выявлены основные векторы в практической плоскости, выдвинуты новые требования для создания универсального оптимизированного продукта.

Ключевые слова: системы управления производством, системы принятия решений в реальном времени, логистика, оценка, синтез, функционал, модули, требования к программному обеспечению.

Goncharov A.V., Grigoriev A.V. Analysis of logistics software in terms of revision of old and synthesis of new requirements for creating a universal optimised software product. The article analyses the software to meet the needs of transport logistics, evaluates the functionality and individual critical modules of the system, identifies the main vectors in the practical plane, and puts forward new requirements for the creation of a universally optimised software product.

Keywords: production management systems, real-time decision-making systems, transport logistics, evaluation, synthesis, functionality, modules, software requirements.

Проектирование компьютерной системы распознавания лиц

В. Г. Шапалин^{*1}, Д. В. Николаенко^{*2}, И. А. Янковский^{*3}

^{*1} магистрант, Донецкий национальный технический университет,
shapalinv@gmail.com

^{*2} к.т.н, доцент, Донецкий национальный технический университет,
dv.nikolaenko@yandex.ru, SPIN-код: 9819-4763

^{*3} к.э.н, доцент, Полесский государственный университет.

Шапалин В. Г., Николаенко Д. В., Янковский И. А. Проектирование компьютерной системы распознавания лиц. В статье представлен краткий обзор современных методов распознавания лиц, включая отдельные алгоритмы для нахождения, нормализации и непосредственно распознавания человека в нормализованном изображении. Также рассмотрены способы распознавания с помощью машинного обучения и виды нейронных сетей, наиболее подходящие для решения данной задачи. Описано проектирование детектора, нормализатора и экстрактора сети.

***Ключевые слова:** распознавание лиц, матрица Габона, перспетрон, свёрточные нейронные сети, гистограмма направленных градиентов, антропометрические точки.*

Введение

Процесс обнаружения и распознавания человеческих лиц в настоящее время имеет важное значение. Её актуальность растет также в связи с ростом потребности реализации данных процедур в различных сферах жизни и деятельности общества. В задачу биометрических методов распознавания лиц входит автоматическое нахождение лица на изображении, а также, при необходимости, идентификация человека. Интерес к сфере обнаружения распознавания лиц также становится значителен благодаря разнообразию их практического применения в таких областях, как охранные системы, криминалистическая экспертиза, верификация, телеконференции, в фототехнике для автоматической фокусировки на лице человека и прочие.

Технология идентификации личности на основе изображения лица, в отличие от использования других биометрических методов, не требует физического контакта с устройством, а также с учетом стремительного развития цифровой техники является наиболее приемлемой для массового применения.

При проектировании системы распознавания лиц необходимо, прежде всего, определиться с методом. Рассмотрим некоторые из них.

Метод гибкого сравнения на графах для распознавания лиц

Суть метода сводится к эластичному сопоставлению графов, описывающих изображения лиц. Лица представлены в виде графов со взвешенными вершинами и ребрами. На этапе распознавания один из графов – эталонный – остается неизменным, в то время как другой деформируется с целью наилучшей подгонки к первому. В подобных системах распознавания графы могут представлять собой как прямоугольную решетку, так и структуру, образованную характерными (антропометрическими) точками лица.

В вершинах графа вычисляются значения признаков, чаще всего используют комплексные значения фильтров Габора или их упорядоченных наборов – Габоровских вейвлет (строи Габора), которые вычисляются в некоторой локальной области вершины графа локально путем свертки значений яркости пикселей с фильтрами Габора.

Ребра графа взвешиваются расстояниями между смежными вершинами. Различие между двумя графами вычисляется при помощи некоторой ценовой функции деформации, учитывающей как различие между значениями признаков, вычисленными в вершинах, так и степень деформации ребер графа.

Деформация графа происходит путем смещения каждой из его вершин на некоторое расстояние в определенных направлениях относительно ее исходного местоположения и выбора такой ее позиции, при которой разница между значениями признаков (откликов фильтров Габора) в вершине деформируемого графа и соответствующей ей вершине эталонного графа будет минимальной. Данная операция выполняется поочередно для всех вершин графа до тех пор, пока не будет достигнуто наименьшее суммарное различие между признаками деформируемого и эталонного графов. Значение ценовой функции деформации при таком положении деформируемого графа и будет являться мерой различия между входным изображением лица и эталонным графом. Данная «релаксационная» процедура деформации должна выполняться для всех эталонных лиц,

заложенных в базу данных системы. Результат распознавания системы – эталон с наилучшим значением ценовой функции деформации.

Эффективность распознавания при наличии различных эмоциональных выражений и изменении ракурса лица до 15 градусов составляет от 90 до 95 %.

Достоинствами метода гибкого сравнения на графах являются:

- высокая точность распознавания;
- устойчивость к смене ракурса, мимике.

Однако, есть и недостатки:

- высокая вычислительная сложность процедуры распознавания;
- низкая технологичность при запоминании новых эталонов;
- линейная зависимость времени работы от размера базы данных [1, 2].

Распознавание лиц с помощью машинного обучения

Нейронная сеть представляет собой математическую абстракцию, содержащую нейроны, соединенные друг с другом синапсами. Структура нейронной сети была вдохновлена подобной структурой работы нашего мозга. Благодаря такой структуре, машина может самостоятельно подбирать подходящие признаки, анализировать их, и выдавать результат, а в некоторых случаях, даже запоминать информацию.

Нейронные сети используются для решения сложных задач, к которым невозможно подобрать универсальный алгоритм. Смысл нейронной сети заключается в самостоятельном подборе подходящих признаков или параметров, которые выбирает сам компьютер в процессе обучения. Наиболее часто нейронные сети применяют в задачах: классификации; предсказания и распознавания.

Общий принцип работы нейрона сводится к передаче данных с выхода одних нейронов на вход других нейронов, умножая значение выхода на коэффициент, который называется весом связи. Все входные значения суммируются, нормализуются с помощью функции активации, и идут на выходной сигнал.

Функция активации — это способ нормализации входных данных. Если на входе большое число, то пропустив его через функцию активации, получим выход в нужном вам диапазоне. Самые часто встречающиеся функции активации – линейная, сигмоид (логистическая) и гиперболический тангенс.

В настоящее время существует около десятка разновидностей нейронных сетей (НС). Одним из самых широко используемых вариантов является сеть, построенная на многослойном перцептроне (рис. 1), в котором нейроны каждого слоя связаны с выходами предыдущего слоя. Такая сеть может гибко подстраиваться под нужный алгоритм, меняя значения весов. Обычно, для корректировки весов используют метод обратного распространения ошибки [3, 4].

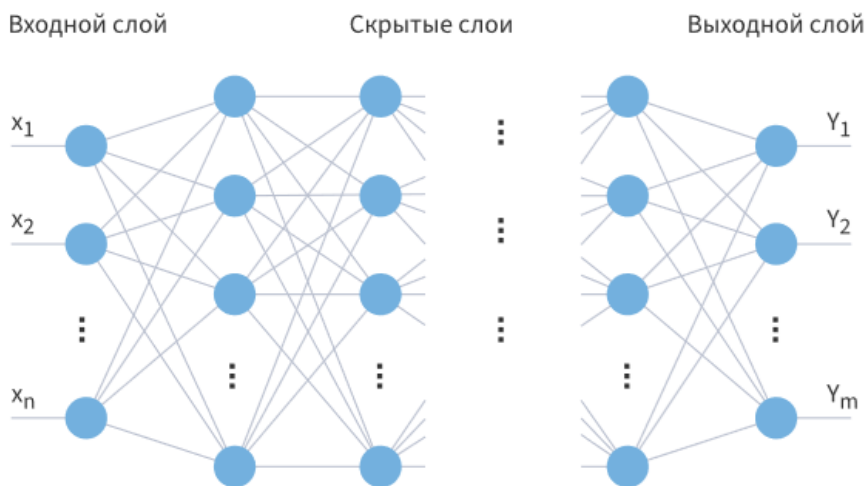


Рисунок 1 – Схематическое изображение многослойного перцептрона

Свёрточные нейронные сети.

Для распознавания лиц и общего анализа изображений лучше всего подходит Convolutional Neural Network или сверточная нейронная сеть (далее – СНС). Успех обусловлен возможностью учета двумерной топологии изображения, в отличие от многослойного перцептрона. Сверточные нейронные сети более устойчивы к изменениям масштаба, смещениям, поворотам, смене ракурса и прочим искажениям.

В СНС имеется несколько видов слоёв: сверточные (convolutional) слои, субдискретизирующие (subsampling, подвыборка) слои и слои «обычной» нейронной сети – перцептрона.

Первые два типа слоев (convolutional, subsampling), чередуясь между собой, который далее подаётся на полносвязный слой, фактически являющийся многослойным персептроном.

Свёрточной сетью называется по названию операции свёртки, суть которой будет объяснена далее.

Топология нейронной сети берется исходя из условий задачи, топологии похожих нейронных сетей и личного опыта, пример возможной архитектуры с тремя входными матрицами, представляющими цвета, представлен на рис. 2.

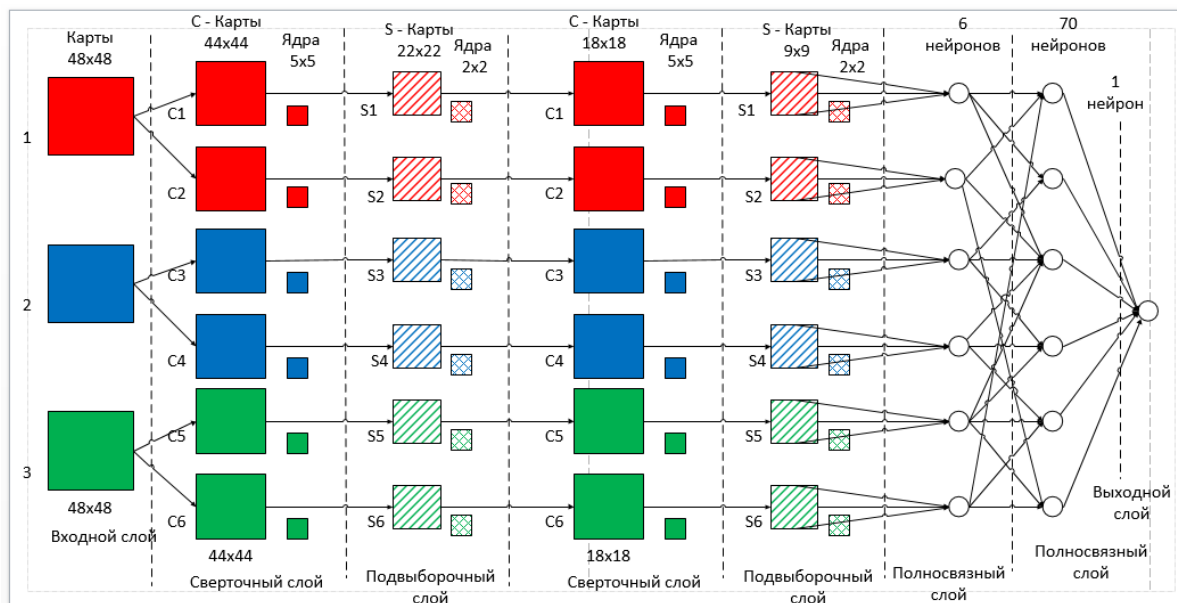


Рисунок 2 – Топология сверточной нейронной сети

Входные данные представляют собой двумерную матрицу определённого размера. Сверточный слой представляет из себя набор карт (другое название – карты признаков, по своей сути являющиеся матрицами), у каждой карты есть синаптическое ядро (в разных источниках он называется по-разному: сканирующее ядро или фильтр). Количество карт определяется требованиями к задаче, если взять большое количество карт, то повысится качество распознавания, но увеличится вычислительная сложность. Исходя из других работ по данной теме и данных из похожих статей, в большинстве случаев рекомендуется брать соотношение один к двум.

Ядро представляет из себя фильтр или окно, которое скользит по всей области предыдущей карты и находит определенные признаки объектов. Изначально значения каждой карты сверточного слоя равны 0. Значения весов ядер задаются случайным образом в области от -0.5 до 0.5. Ядро скользит по предыдущей карте и производит операцию свертки, которая часто используется для обработки изображений, формула для операции свёртки представлена ниже:

$$(f * g)[m, n] = \sum_{k, l} f[m - k, n - l] * g[k, l], \quad (1)$$

где f – исходная матрица изображения; g – ядро свёртки.

Подвыборочный слой также, как и сверточный имеет карты, но их количество совпадает с предыдущим (сверточным) слоем. Цель слоя – уменьшение размерности карт предыдущего слоя. Уменьшение изображения используется для уменьшения деталей в изображении, в результате, нейронная сеть работает с более общими признаками.

В процессе сканирования ядром подвыборочного слоя (фильтром) карты предыдущего слоя, сканирующее ядро не пересекается в отличие от сверточного слоя. Обычно, каждая карта имеет ядро размером 2x2, что позволяет уменьшить предыдущие карты сверточного слоя в 2 раза. Вся карта признаков разделяется на ячейки 2x2 элемента, из которых максимальное, минимальное, или среднее значения, в зависимости от решаемой задачи.

Последний из типов слоев это слой обычного многослойного персептрона. Цель слоя – классификация, моделирует сложную нелинейную функцию, оптимизируя которую, улучшается качество распознавания [5].

Проектирование системы распознавания лиц

При грубом рассмотрении, создание системы распознавания лиц сводится к созданию трёх отдельных элементов системы, каждая со своей функцией: детектор, нормализатор и экстрактор. Система принимает на вход изображение, а на выходе возвращает некоторый набор вещественных чисел, который «кодирует» лицо. Этот набор часто ещё называют «вектором признаков» или «биометрическим шаблоном».

Детектор.

Детектор – система для обнаружения лиц. Разные алгоритмы обнаружения начали набирать популярность с 2000-х годов. В качестве готового алгоритма часто используется гистограмма направленных градиентов, или HOG. Для работы метода, используется черно-белое изображение. Каждый пиксель изображения мы сравниваем с окружающими его, и проводим вектор по направлению затемнения изображения.

Сохранение градиента для каждого отдельного пикселя даёт нам способ, несущий слишком много подробностей, которые могут быть неправильно рассмотрены, поэтому необходимо разбить изображения на блоки 16x16. В каждом квадрате следует подсчитать, сколько градиентных стрелок показывает в каждом главном направлении. Затем рассматриваемый квадрат на изображении заменяют стрелкой с преобладающим направлением в этом квадрате.

Чтобы обнаружить лица на обработанном HOG-изображении, всё, что требуется, это найти такой участок изображения, который наиболее похож на известную HOG-структуру, полученную из группы лиц, использованной для обучения.

Тем не менее, нейронные сети вытеснили все классические алгоритмы распознавания лиц практически полностью. Они менее чувствительны к ракурсу съёмки (наклонам, поворотам) и куда более стабильны в предсказаниях, чем классические методы. Даже скорость работы сегодня перестала быть проблемой для нейросетей: с тем огромным объёмом доступных данных для обучения и развитием вычислительных ресурсов можно без проблем подобрать такой размер нейросети, который удовлетворит нашим запросам.

Нормализатор.

Теперь мы имеем выделенное лицо из нашего изображения. Но теперь появляется проблема: одни и те же лица с разных ракурсов для компьютера выглядят по-разному. Чтобы учесть это, попробуем преобразовывать каждое изображение так, чтобы глаза и губы всегда находились примерно на одном и том же месте изображения.

Для этого используем алгоритм, называемый «**оценка антропометрических точек**». Основная идея заключается в выделении 68 специфических точек (меток), имеющих на каждом лице. Затем происходит настройка алгоритма на поиск этих 68 специфических точек на каждом лице:

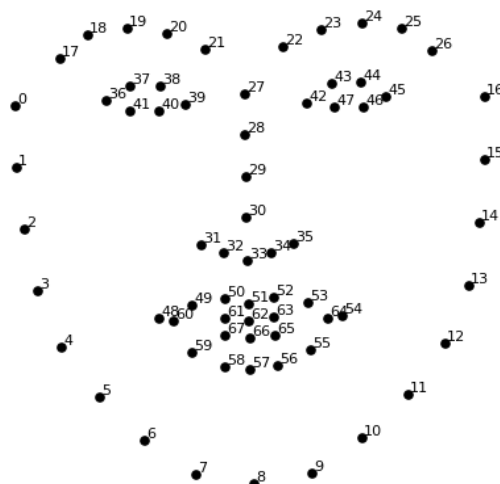


Рисунок 3– Антропометрические точки, располагающиеся на каждом лице

После того, как определено, где находятся глаза и рот, необходимо изменить размер изображения для его отцентрования с помощью аффинных преобразований.

Способ довольно простой, однако он сильно зависит от качества детектирования ключевых точек.

Альтернативой является нейросеть. С её помощью мы можем предсказать итоговую матрицу преобразования напрямую, не отыскивая ключевые точки и не делая каких-либо предположений о расположении носа и глаз.

Экстрактор.

Теперь, когда получено нормализованное лицо, необходимо построить вектор признаков — этим занимается экстрактор. На вход подаётся изображение фиксированного разрешения — обычно 90–130 пикселей, такой размер позволяет соблюсти баланс между точностью работы алгоритма и его скоростью.

Какие же характеристики лица необходимо учитывать? Характеристики, представляющиеся очевидными для человека (размеры уха, длина носа, цвет глаз и т.д.), не имеют смысла для компьютера, анализирующего отдельные пиксели на изображении. Наиболее адекватным подходом является дать возможность компьютеру самому определить характеристики, которые надо собрать. Сверточные нейронные сети позволяют лучше, чем это могут сделать люди, определить части лица, важные для его распознавания. Задача состоит в том, чтобы обучить глубокую свёрточную нейронную сеть и научить её получать собственные характеристики для каждого

лица. Например, при распознавании двух разных людей, можно научить нейросеть на выборке из трёх лиц. Обучение сводится к следующим шагам:

- загрузка обучающего изображения лица известного человека;
- загрузка другое изображения лица того же человека;
- загрузка изображения лица какого-то другого человека.

Затем алгоритм рассматривает характеристики, которые он в данный момент создаёт для каждого из указанных трёх изображений. Он слегка корректирует нейронную сеть так, чтобы характеристики, созданные ею для изображений 1 и 2, оказались немного ближе друг к другу, а для изображений 2 и 3 — немного дальше.

При повторении шагов миллионы раз для тысячи разных людей, нейросеть оказывается в состоянии надёжно создавать 128 характеристик для каждого человека. Любые десять различных изображений одного и того же человека дадут примерно одинаковые характеристики.

Чем большее количество людей будет использовано в обучении, тем лучше и точнее будет работать алгоритм. На каждого человека достаточно пятидесяти фотографий для достаточно точного его распознавания.

При формировании обучающей выборки следует учесть, что экстрактор всегда будет лучше работать на данных, похожих на те, на которых он учился. Если в датасете будут только представители европеоидной расы, то распознавание лиц азиатской внешности, либо черных людей может работать некорректно, поэтому нужно постараться сделать обучающую выборку как можно более разнообразной.

Нахождение изображения. На последнем шаге остается найти человека в базе данных известных лиц, который имеет характеристики, наиболее близкие к характеристикам тестового изображения. Это можно сделать с помощью любого базового алгоритма классификации обучения машин.

Вывод

В результате проделанной работы были исследованы принципы работы как классических алгоритмов для распознавания лиц, так и методов машинного обучения, а также были рассмотрены виды нейронных сетей, а именно, многослойные перцептроны и сверточные нейронные сети. Описано проектирование детектора, нормализатора и экстрактора сети.

Литература

1. Исследование характеристик алгоритмов распознавания лиц [Электронный ресурс] – Режим доступа: https://icmmg.nsc.ru/sites/default/files/pubs/levchuk_yakimenko_vestnik_ngtu2018.pdf. – Загл. с экрана.
2. О некоторых аспектах технологии распознавания лиц [Электронный ресурс] – Режим доступа: https://humancapital.su/wp-content/uploads/2020/04/202004-5_p142-152.pdf. - Загл. с экрана.
3. Многослойных перцептрон [Электронный ресурс] – Режим доступа: <https://wiki.loginom.ru/articles/multilayered-perceptron.html>. - Загл. с экрана.
4. Введение в нейросети на примере распознавания цифр [Электронный ресурс] – Режим доступа: <https://proglab.io/p/neural-network-course>. - Загл. с экрана.
5. Сверточная нейронная сеть [Электронный ресурс] – Режим доступа: <https://habr.com/ru/post/348000>. - Загл. с экрана.

Шапалин В. Г., Николаенко Д. В., Янковский И. А. Проектирование компьютерной системы распознавания лиц. В статье представлен краткий обзор современных методов распознавания лиц, включая отдельные алгоритмы для нахождения, нормализации и непосредственно распознавания человека в нормализованном изображении. Также рассмотрены способы распознавания с помощью машинного обучения и виды нейронных сетей, наиболее подходящие под данную задачу. Описано проектирование детектора, нормализатора и экстрактора сети.

Ключевые слова: распознавание лиц, матрица Габона, перспетрон, свёрточные нейронные сети, гистограмма направленных градиентов, антропометрические точки.

Shabalin V. G., Nikolaenko D. V., Yankovsky I. A. Designing a computer facial recognition system. The article provides a brief overview of modern methods of face recognition, including separate algorithms for finding, normalizing and directly recognizing a person in a normalized image. The methods of recognition using machine learning and the types of neural networks that are most suitable for this task are also considered. The design of the detector, normalizer and network extractor is described.

Keywords: face recognition, Gabon matrix, perceptron, convolutional neural networks, histogram of directional gradients, anthropometric points.

Выбор средств реализации кибербезопасности системы «Умный дом»

Р. В. Мальчева^{*1}, А. И. Кобыляцкий^{*2}, Л. П. Володько^{*3}

^{*1} к.т.н, доцент, Донецкий национальный технический университет,
raisa.malcheva@yandex.ru, OrcID: 0000-0003-4983-8878, SPIN-код: 7421-7049

^{*2} магистрант, Донецкий национальный технический университет,

^{*3} к.т.н, доцент, Полесский государственный университет, Республика Беларусь,
anton4ik1232@gmail.com

Мальчева Р. В., Кобыляцкий А. И., Володько Л. П. Выбор средств реализации кибербезопасности системы «Умный дом». В статье описано назначение современных многофункциональных комплексов «Умный дом». Представлен краткий обзор проблем безопасности систем и стратегий, обеспечивающих их защиту. Выполнен анализ способов и готовых системных решений для реализации защиты системы, а также сформулированы требования к реализации защиты. Выявлены наиболее важные уязвимые элементы защиты системы, а также возможные варианты их устранения. Предоставлен выбор реализаций защиты системы «Умный дом».

Ключевые слова: проектирование, Умный дом, кибербезопасность, защита, аутентификация.

Введение

Понятие Интернета вещей стало в настоящее время одним из основных направлений в ведущих странах мира. В подмножество данного направления так же входит инфраструктура «Умный дом» [1]. Стимулом развития данной инфраструктуры является улучшение жизнедеятельности человека. Всесторонний анализ улучшения эффективности, снижения финансовых затрат, а также обеспечение кибербезопасности являются важным аспектом при разработке такой инфраструктуры. На сегодняшний день, практически у каждого более-менее крупного производителя умных устройств имеется своя, так называемая, экосистема (рис. 1), внутри которой происходит управление элементами и передача информации между компонентами и центральным контроллером.

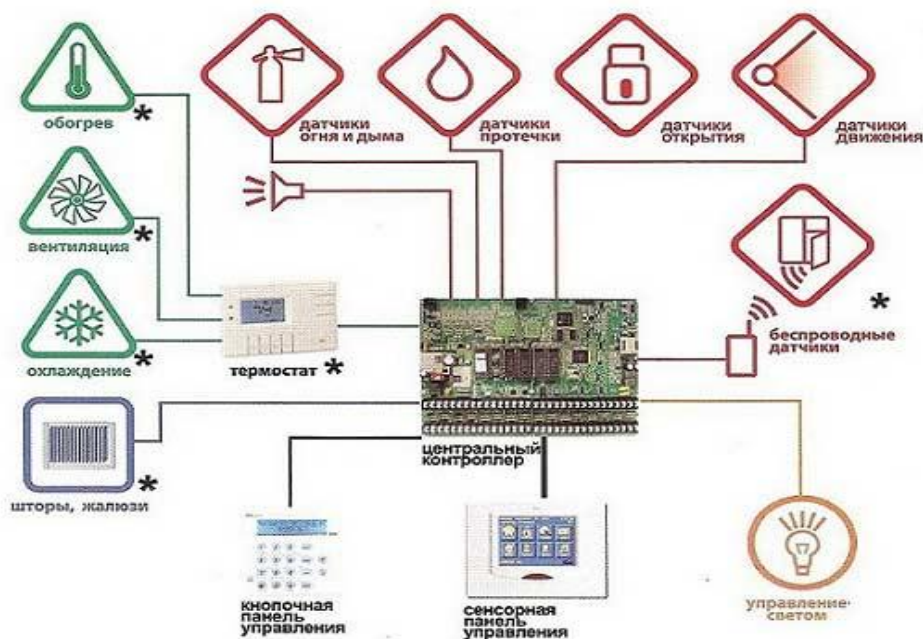


Рисунок 1 – Пример системы «Умный дом»

«Умные» дома и их многочисленные элементы делают жизнь комфортнее, упрощают рутину, повышают безопасность и энергоэффективность зданий. Однако, и сама система нуждается в защите. Несмотря на продолжающийся прогресс в технологиях умного дома, проблемы кибербезопасности остаются неизменными. И дело не только в вопросе конфиденциальности. Цифровая экономика постепенно вторгается в повседневную жизнь людей с помощью умных устройств, создавая подходящие условия для хакеров. Взлом умного дома считается серьезной проблемой, в частности потому, что все больше людей работают сегодня удаленно. Обеспечение кибербезопасности должно быть приоритетом для производителей систем безопасности и видеонаблюдения, поскольку даже минимальная уязвимость может подорвать доверие клиентов, которое компании создают с помощью времени, денег и усилий.

Поэтому задача выбора эффективных средств реализации кибербезопасности системы «Умный дом» остается актуальной.

Стратегии защиты «Умного дома»

Эксперты индустрии безопасности отмечают ряд стратегий, которые помогут пользователям обеспечить надежную киберзащиту датчиков умного дома [5].

1. Реализация автономной системы сети. К одной из главнейших уязвимостей «Умного дома» можно отнести отсутствие главного элемента – источника питания системы. Зачастую пользователи не воспринимают данную проблему со всей серьезностью, ведь производители периферии устанавливают автономные источники питания, которых зачастую не хватает для стабильной работы всех элементов.

2. Двухфакторная аутентификация системы. Двухфакторная аутентификация — это дополнительный уровень безопасности помимо применения простого пароля. При использовании режима двухфакторной аутентификации после того, как пользователи введут свой пароль, им придется дополнительно удостоверить свою личность согласованным ранее способом. Как правило, процесс подтверждения подразумевает получение индивидуального конфиденциального кода, который генерируется случайным образом приложением для телефона, отправляется пользователям посредством текстового сообщения или телефонного звонка.

К сожалению, режим безопасности с использованием двухфакторной аутентификации не очень распространен на устройствах «Умного дома», но постепенно процесс внедрения дополнительного инструмента защиты становится востребованным и приобретает статус обязательного присутствия. Торговые марки «Nest» и «Wyze», используемые для продвижения продуктов для «Умного дома», теперь предлагают двухфакторную аутентификацию. Камеры безопасности также, скорее всего, будут иметь двухфакторную аутентификацию, и пользователи не должны пренебрегать возможностью ее использовать.

По результатам независимого исследования было определено, что часто злоумышленникам гораздо быстрее и проще использовать украденные регистрационные данные для входа в учетные записи, связанные с пользовательскими устройствами «Умного дома», чем пытаться взломать защиту маршрутизатора. И двухэтапная аутентификация может помочь предотвратить такой вариант удаленного проникновения.

Разработчики систем рекомендуют дополнительно проверять приложения, связанные с умными устройствами, и включать, где только возможно, режим подтверждения личности.

Также рекомендуется объединить двухфакторную аутентификацию с приложением для аутентификации, таким как «Google Authenticator» для мобильных операционных систем «iOS» и «Android», позволяющим генерировать коды для сторонних приложений и сервисов [5].

3. Обновление официального ПО. Большинство умных домашних устройств будут иметь функцию автоматического обновления, которую пользователь может включить. Это поможет производителям обновлять и вносить исправления в контроль безопасности устройств при каждом обнаружении уязвимости. Если автоматическое обновление программного обеспечения не включено, пользователю нужно будет отслеживать обновления от компании и следить за тем, чтобы они были загружены и установлены.

4. Регулярное изменение SSID (сервисный идентификатор сети) «Умного дома». Т.к. SSID по умолчанию показывает тип используемого пользователем устройства, это может помочь хакерам найти уязвимости для данного устройства.

Когда на ноутбуке или телефоне открывается список локальных сетей Wi-Fi, отображаются идентификаторы SSID (идентификатор набора услуг), некоторые из которых могут поступать из сетей соседей. Беспроводные маршрутизаторы или точки доступа передают идентификаторы SSID для того, чтобы расположенные поблизости устройства могли находить и отображать все доступные сети Wi-Fi.

Для дополнительного уровня кибербезопасности пользователь должен изменить SSID своего Wi-Fi, чтобы сразу не было известно, какой тип оборудования Wi-Fi используется [6].

5. Лицензированный фаерволл. Не менее важный аспект защиты «Умного дома». Он также является последней линией защиты персонального жилья.

Зачастую выбор брандмауэра зависит от конфигурации системы, а также реализации одного из протоколов, внедренного в «Умный дом». Лучшим выбором в данном сегменте – это консультация с производителями систем, такими как Xiaomi, Rubetek, IKEA и другие.

Дополнительные требования к защите «Умного дома»

Важным критерием к эксплуатации «Умного дома» является возможная замена устаревших устройств и комплектующих системы. Данную возможность стоит учесть еще на начальном этапе проектирования дома, т.к. цифровые технологии постоянно развиваются, и злоумышленники всё чаще находят уязвимости в старых системах, программное обновление которого не всегда успевает обеспечить защиту от новых угроз. Поэтому лучшим способом будет инвестирование в новейшие технологии. Чем чаще будут обновляться устройства, их программное обеспечение и системы мониторинга и сигнализации, тем безопаснее будет.

Также рекомендуется брать устройства известных брендов, потому что за счет опытных специалистов и большего охвата аудитории, а, следовательно, и большего количества потенциальных тестировщиков, они быстрее обнаруживают проблемы в своих устройствах и, не мешкая, разрабатывают обновления для их устранения. Таким образом, пользователи могут наиболее удачно защитить свою сеть от взлома.

Также стоит учесть наличие защищенного автономного источника питания системы. В этом плане есть широкий выбор возможных реализаций, начиная от автоматических электрогенераторов и заканчивая энергоёмкими бесперебойными батареями, которых вполне должно хватить для работы важных элементов системы, таких как электрозамки и системы сигнализации системы.

Немаловажным фактором является дополнительное устройство управления умного дома, которое в случае утери основного устройства, такого как смартфон владельца, позволит не потерять доступ к важным защитным свойствам дома. Также рекомендуется хранить дополнительный пульт управления в защищенном месте, вдали от злоумышленника, а также предусмотреть, что бы доступ к нему не зависел от самой системы «Умный дом». Для этого подойдет личный сейф либо его альтернатива.

Программную политику авторизации при входе в систему рекомендуется менять через определенные интервалы времени во избежание утечек в руки потенциального злоумышленника. Так же рекомендуется использовать несколько степеней защиты, например, двухфакторную аутентификацию.

Анализ существующих готовых решений

Одним из лучших и надежных систем реализации всего комплекса «Умного дома» и его защиты является выбор готовых типовых проектов. В наше время данный сегмент рынка активно развивается. Причем, уже нет необходимости использовать импортные решения или их аналоги для реализации данной системы в современном доме. В российском сегменте так же присутствуют корпоративные гиганты, занимающиеся проектировкой сложнейших систем структуры «Умного дома», например, Rubetek (рис. 2).

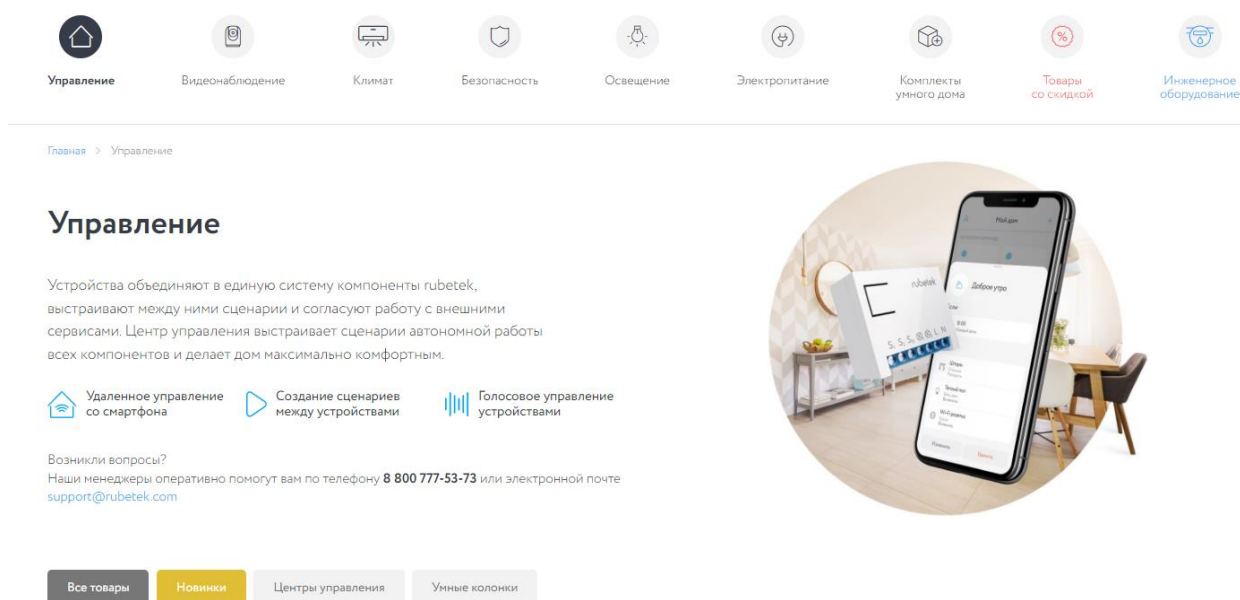


Рисунок 2 – «Умный дом» Rubetek

Rubetek— это один из лидеров российского рынка оборудования для «Умного дома». «Умный дом» Rubetek интересен тем, что решает большое число задач пользователей при относительно небольшой цене на устройства с возможностью внедрения новейших передовых технологий, существующий на сегодняшнее время, таких как удалённый доступ и голосовое управление всеми функциями дома. Данная компания предоставляет широкий спектр возможностей реализации «Умного дома» с огромным выбором, исходя из финансовых возможностей заказчика. Также присутствует полная техническая поддержка со стороны фирмы с

последующими возможными решениями технических проблем, с которыми может столкнуться неопытный пользователь [7].

Вторым интересным примером является российский аналог производителя **MiMiSmart** (рис. 2). Компания позволяет выстраивать умный дом, исходя из потребностей пользователя, т.к. предоставляет в его распоряжение компоненты, покрывающие любые сценарии [8]. Управление может осуществляться как со смартфона, так и с использованием умных часов.

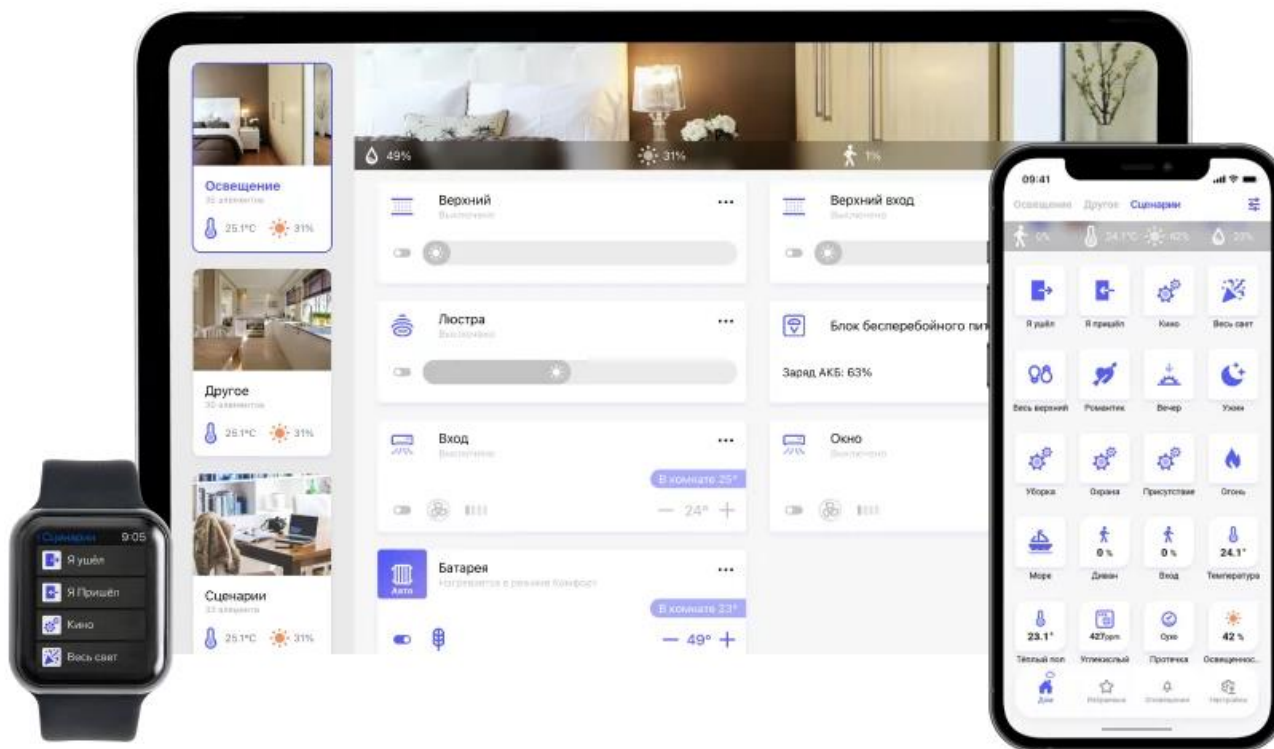


Рисунок 2 – «Умный дом» MiMiSmart

Компания предоставляет все доступные на рынке услуги в сфере «Умный дом», такие как:

1. Оценка условий работы. При разработке технического задания наши проектировщики выезжают на объект. Они документально фиксируют особенности имеющихся коммуникаций, инженерной инфраструктуры. Составление схемы, планов размещения оборудования, а также определение предварительной стоимости проектирования и монтажа Умного Дома возможно только после этого этапа.

2. Создание документации. На данном этапе проектирование Умного Дома сводится к разработке функциональных и структурных схем, осуществлению трассировки и подготовке технической спецификации, чтобы все дальнейшие работы велись строго в соответствии с нормативами, принятыми в Российской Федерации. Наши специалисты учитывают, что умный дом или квартира должны быть красивыми и удобными. Поэтому инженерные коммуникации прокладываются таким образом, чтобы доступ к ним был простым, а их внешний вид гармонировал с интерьером помещений. Кроме того, вы сможете ознакомиться с примерами проектов квартир и домов, монтаж которых проводила наша компания.

3. Расчет стоимости работ и материалов. На данном этапе проектирования мы разрабатываем и согласуем с клиентом проектное (концептуальное) решение для каждой инженерной системы и комплекса технологий в целом. Эта стадия дает представление о стоимости реализации всех ваших желаний. Благодаря этому вы сможете внести необходимые коррективы в проект систем Умного Дома исходя из предполагаемой цены.

4. Согласование пакета документов. После создания рабочей документации проектирование системы Умный Дом и ее составляющих вступает на финишную прямую. Приходит этап согласования планируемых изменений в уполномоченных инстанциях. Благодаря этому система будет полностью отвечать всем нормативным требованиям и у хозяев здания не возникнет сложностей с соответствующими органами. Особенно важно это для владельцев квартир.

5. Сдача проекта. Когда все детали проекта по технологии Умный Дом согласованы, наши специалисты презентуют систему клиенту, а затем подписывают акт. Следом наступает этап монтажа под авторским надзором со стороны наших сотрудников. Следует отметить, что по завершении работ MiMismart предлагает регулярное обслуживание установленных систем [8].

Выводы

В современных информационных реалиях возможность иметь «Умный дом» присуща не только крупным компаниям, но всё больше интегрируется в среду организации быта обычных граждан. Т.к. данный сегмент на рынке активно развивается, существуют и способы взлома систем всевозможными доступными способами.

Выбор эффективной защиты системы «Умный дом» с возможностью её постоянного обновления остается актуальной инженерной задачей.

Литература

1. Кобыляцкий, А. И. Анализ систем мониторинга инфраструктуры «Умный дом» / А. И. Кобыляцкий, Р. В. Мальчева // Информационное пространство Донбасса: проблемы и перспективы : материалы IV Респ. с междунар. участием науч.-практ. конф., 28 окт. 2021 г. – Донецк : ГО ВПО «ДонНУЭТ», 2021. – С. 152-154.
2. Погорелов, А. А. Управление энергопотреблением в системе «Умный дом» / А. А. Погорелов, Р. В. Мальчева, Л. П. Володько // Информатика, управляющие системы, математическое и компьютерное моделирование (ИУСМКМ-2021). Материалы XII Международной научно-технической конференции в рамках VII Международного Научного форума Донецкой Народной Республики к 100-летию ДонНТУ. – Донецк: ДОННТУ, 2021. – С. 141-144.
3. Погорелов, А. А. Выбор способа передачи данных между веб-приложением и микроконтроллером / А. А. Погорелов, Р. В. Мальчева, А. М. Долженко // Современные информационные технологии в образовании и научных исследованиях (СИТОНИ-2021) : Материалы VII международной научно-технической конференции. Под общей редакцией В. Н. Павлыша. – Донецк: ДОННТУ, 2021. – С.180-184.
4. Парфенов, Д. А. Разработка системы контроля приемом посетителей предприятия / Д. А. Парфенов, Р. В. Мальчева, Л. П. Володько // Информатика, управляющие системы, математическое и компьютерное моделирование (ИУСМКМ-2018). Материалы IX Международной научно-технической конференции в рамках IV Международного Научного форума Донецкой Народной Республики. – Донецк: ДОННТУ, 2018. – С. 219-222.
5. Как защитить «умный дом» от взлома? [Электронный ресурс]. – Режим доступа: <https://worldvision.com.ua/ru/kak-zashchitit-umnyy-dom-ot-vzloma/>. - Загл. с экрана.
6. Как защитить свой «Умный дом от атаки? [Электронный ресурс]. – Режим доступа: <https://spark.ru/startup/hetmansoftware/blog/69238/kak-zaschitit-svoj-umnij-dom-ot-ataki/> - Загл. с экрана.
7. Система умный дом Rubetek [Электронный ресурс]. – Режим доступа <https://rubetek.com/> - Загл. с экрана.
8. Проектирование умного дома [Электронный ресурс]. – Режим доступа: <https://mimismart.ru/czenyi/proektirovanie-umnogo-doma/>. - Загл. с экрана.

Мальчева Р.В., Кобыляцкий А.И., Володько Л.П. Выбор средств реализации кибербезопасности системы «Умный дом». В статье представлен краткий обзор современных проблем при организации защиты многофункциональных комплексов «Умный дом». Выполнен анализ способов и готовых системных решений для реализации защиты системы, а также сформулированы требования к реализации защиты. Выявлены наиболее важные уязвимые элементы защиты системы, а также возможные варианты их устранения. Предоставлен выбор реализаций защиты системы «Умный дом».

Ключевые слова: кибербезопасность, защита, аутентификация, проектирование, Умный дом.

Malcheva R.V., Kobylatsky A.I., Volodko L. P. The choice of means of implementing cybersecurity of the "Smart Home" system. The article describes the purpose of modern multifunctional complexes "Smart home". A brief overview of the security problems of systems and strategies to ensure their protection is presented. The analysis of methods and ready-made system solutions for the implementation of system protection is carried out, as well as the requirements for the implementation of protection are formulated. The most important vulnerable elements of the system protection have been identified, as well as possible options for their elimination. A choice of implementations of protection of the "Smart Home" system is provided.

Keywords: design, Smart home, cybersecurity, protection, authentication.

Компьютерная система психофизиологического диагностирования, интеллектуально-духовной реабилитации и безмедикаментозной терапии

С. И. Изосимова, В. Н. Пигуз

Государственное учреждение «Институт проблем искусственного интеллекта»
г. Донецк, Izosimova.snezhana@mail.ru

Изосимова С.А., Пигуз В.Н. Компьютерная система психофизиологического диагностирования, интеллектуально-духовной реабилитации и безмедикаментозной терапии. В статье анализируется возможность создания и продуктивной работы компьютерной системы психофизиологического диагностирования, интеллектуально-духовной реабилитации и безмедикаментозной терапии в данный период. Данная система предназначена для использования как в коллективном (например, на предприятиях, учреждениях, организациях и т.п.), так и в индивидуальном порядках (самодиагностика и саморегуляция ПФС личности). Общая доступность в использовании способствует расширению сферы использования компьютерных систем среди широкого круга людей, не имеющих специального образования и подготовки в данной отрасли.

Ключевые слова: психофизиологическое состояние; интеллектуально-духовная терапия; саморегуляция; компьютерная система; эмоциональный искусственный интеллект.

Человек и искусственный интеллект – две стороны одной медали. Взаимодополняющие, совершенствующие друг друга, но не взаимозаменяемые. Без постоянного поиска себя, своего места в этом мире, без разработок нового, человека нет.

А. Азимов

Введение

Глобальное развитие технологий искусственного интеллекта, переработки, восприятия и усвоения информации стали причиной установки четких параллелей между психофизиологическим состоянием человека в момент его работы со средствами информационных технологий.

Сегодня остро звучит вопрос о необходимости валеологического подхода к проблеме психического здоровья. То есть изначальная (первичная) направленность на воспитание личности, сохранение творческого долголетия и работоспособности, а также особого психического статуса личности.

На данный момент различают более 85 различных психофизиологических состояний человека [1], находящегося в нормальном эмоциональном состоянии: радость, печаль, апатия и проч. Физическое состояние человека обладает четкой взаимосвязью с интеллектуальным. Это две половины единого целого – человеческой личности. Что дает возможность не только диагностировать состояние человека, но и анализировать на данный момент его способности к творчеству, учебе и т.п.

Актуальность исследования

Разумеется, пребывать постоянно в одном и том же состоянии человек не может. Поэтому возникает острая необходимость в обладании средствами, позволяющими оперативно диагностировать интеллектуально-психофизиологические состояния (ИПФС) [2], при этом за минимальный временной промежуток определяя эффективные безмедикаментозные средства для нормализации состояния субъекта, т.е. компьютерной терапии.

Анализ литературных источников

Наиболее полно программно-техническая характеристика данной научной разработки представлена в статье Р.И. Сальникова и соавторов «Разработка и построение комплексной компьютерной системы психофизиологического диагностирования, интеллектуально-духовной реабилитации и безмедикаментозной терапии». Методический аспект раскрывается в работе И.С. Сальникова, Р.И. Сальникова «Современные методы и методики изучения и диагностирования интеллектуально-психофизиологических состояний человека и способы их компьютерной аудиовизуальной терапии». Кроме того, использованы труды ученых международного уровня: Е.П. Ильин «Психофизиология состояний человека»; Л.Ф. Бурлачук «Психодиагностика личности»; Е.Д. Хомская «Нейропсихология» и проч.

Программно-техническая характеристика

В настоящее время в Государственном учреждении «Институт проблем искусственного интеллекта» идет разработка компьютерной системы интеллектуально-духовной реабилитации и безмедикаментозной компьютерной терапии. В основе которой – использование прослушивания произносимых вслух текстов в сопровождении соответствующих иллюстраций. При этом иллюстративный материал должен обладать разнообразной цветовой гаммой с анимационными динамическими объектами точечного типа. Данный синкретический доминант предъявляется тестируемому, либо пациенту, согласно с выявленным по определенным методикам его эмоциональным состоянием.

Таким образом будет формироваться духовная доминанта личности, которая защищает организм от негативного воздействия внутренней (в организме) и внешней (вне организма) среды, при этом оказывая терапевтическое воздействие на человека, блокируя поступление отрицательных эмоций.

Относительно технической и программной характеристики, разрабатываемая в ГУ ИПИИ система психофизиологического диагностирования, интеллектуально-духовной реабилитации и безмедикаментозной терапии (СИДРТ) базируется на 3-х группах прикладных программных продуктов и реализует 14 различных методик диагностирования и изучения психоэмоциональных состояний личности человека:

- программы, обеспечивающие изучение ПФС человека по известным методикам изучения его состояний;
- программы, обеспечивающие изучение индивидуальных психологических особенностей личности;
- программы, обеспечивающие восстановление нормального состояния функционирования организма человека без применения медикаментозных средств [3, с. 10].

Методическая база

В основу предлагаемой разработки положены результаты исследований отечественных и зарубежных ученых в области методик изучения психоэмоциональных состояний, которые и использованы в качестве различного рода тестов психофизиологического диагностирования. Было выполнено ряд исследований и конкретных разработок компьютерных систем на базе персональных компьютеров, подтвердивших возможности широкого использования машинных способов диагностирования. По результатам этих исследований было опубликовано две статьи [2, 3].

Для проектирования методической базы разработки использовались следующие методики диагностирования и распознавания психофизиологического состояния человека:

- методика САН (самочувствие, активность, настроение) – в 1973 г. предложена В.А. Доскиным и соавторами и состоит из 30 биполярных шкал, которые группируются в 3 категории: настроение, активность, самочувствие [2, с. 551]. Тестируемый самооценивает свое состояние в баллах в соответствии с имеющейся шкалой самооценки. Норма находится в пределах 50 – 55 баллов;
- методика «Градусник» для самооценки эмоциональных состояний – (разработчик Н.П. Фетискин) предназначена для экспресс-диагностики эмоциональных состояний, так как фиксирует состояния в их динамическом развитии [2, с. 551];
- опросник ПНПН (признаки нервно-психического напряжения) [2, с. 551];
- методика диагностирования профессионального выгорания (сгорания) [2, с. 551] – текст опросника состоит из 84 суждений-утверждений, по каждому из которых испытуемый должен выразить свое мнение;
- методика «Накопление эмоционально-энергетических зарядов, направленных на самого себя» (разработчик В.В. Бойко) – формирует картину эмоционально-энергетического напряжения, которое накапливается внутри организма человека и не получает выхода вовне [2, с. 550];
- модифицированная шкала личностной соревновательной тревожности (МШЛСТ) – адаптирована к отечественным реалиям и менталитету Ю.Л. Ханиным в 1983 году. Позволяет оценить индивидуальную эмоциональную реакцию человека перед предстоящим событием: соревнованием, стрессовой ситуацией и т.п.

Влияние музыки и цвета на ПФС личности

Более 80% информации человек воспринимает посредством зрения. Влиянию цвета на психоэмоциональное состояние уделялось много внимания еще со времен Египта и Древней Греции. В современности методы и способы цветотерапии совершенствовались и приобрели более лаконичные формы. В системе СИДРТ программно реализован «8-цветовой тест Люшера, диагностирующий психологические особенности человека с осознанными и неосознанными мотивами и побуждениями к деятельности» (рис. 1) [3, с. 10].

Учеными уже давно доказано, что музыка и использование цвета являются лучшими терапевтами и блокираторами негативных эмоций. В частности, соответствующая музыка ускоряет частоту сердечных сокращений, а столь любимая молодым поколением рок-музыка воздействует на организм не только слышимыми звуками, но и инфра-, а также ультразвуками. То есть при длительном воздействии способна разрушить мозговые рецепторы.



Рисунок 1 – Главное окно программы «Цветовой тест Люшера» [3, с. 13]

Анализ проблемы исследования, общие требования к разработке

За рубежом существует большое количество различного рода тестов психологического характера и систем психофизиологической диагностики. Отличительной особенностью их является профессиональная ориентация и чрезмерная детализация тестируемых свойств и особенностей личности в отличие от предлагаемой системы диагностирования, которая носит достаточно общий характер и определяет состояние личности, отдаленное от медицинских диагнозов. В Донецкой Народной Республике исследования и разработки в области психических состояний личности ведутся многими специалистами и учеными и на кафедрах в различных институтах профессионального образования, а также в научно-консультативных психологических центрах. В России также широко развиты научные исследования и разработки в области психофизиологической диагностики. Лидирующее положение в мире в этой области занимает США, однако этот опыт достаточно сложно применять в наших условиях развития производства и общественной жизни. По нашему мнению, предлагаемая разработка в наибольшей мере будет соответствовать как состоянию граждан Республики, так и нашим сегодняшним возможностям применения и использования результатов рассматриваемого проекта.

Предварительные исследования позволили выбрать для дальнейшего использования при построении системы авторские методики изучения психофизиологических состояний, тесты, шкалы и опросники, предложенные В.А. Доскиным, Н.П. Фетискиным, В. Зунгом – Т.Н. Балашовой, Т.А. Немчиным, Ч.Д. Спилбергером – Ю.Л. Ханиным, В.В. Бойко, Л.И. Вассерманом, Р. Мартенсом, К. Маслач – С. Джексоном, Говардом Гарднером как наиболее современные и отвечающие поставленной задаче.

Общетеоретической основой предлагаемого проекта исследований и разработки являются идеи русских ученых В.М. Бехтерева, И.М. Сеченова, И.П. Павлова, П.К. Анохина и их многочисленных последователей и продолжателей как в России, так и в других зарубежных странах.

Из последних исследований, наиболее близких к рассматриваемой проблематике, является постановка задачи и результаты исследований, полученные в докторской диссертации «Генезис полифункциональной регуляции психических состояний личности эмоциогенными средствами» Волженцевой И.В., выполненной в рамках научно-исследовательской темы кафедры психологии Макеевского экономико-гуманитарного института «Психологические особенности личности в условиях трансформации общества». Настоящий проект развивает выдвинутые в этой работе проблемы с использованием компьютерных средств и технологий выявления психофизиологических состояний и их регуляции новыми средствами в рамках безмедикаментозной терапии.

Также к разработке выдвигались следующие общие требования:

- простота диагностирования и самодиагностирования тестируемого;
- обеспечение максимально понятного и удобного интерфейса пользователя с использованием современных средств воспроизведения информации, звуков и изображений;
- сопроводительная анимация и звуковые средства должны нести положительный эффект и обладать разгрузочным характером.

Разработка компьютерных систем контроля психо-эмоциональных состояний человека как личности является одной из актуальных задач сферы искусственного интеллекта [1]. В настоящей работе разработан программный продукт как средство для более точного и более детализированного тестирования и диагностирования

эмоциональных состояний личности по заданному перечню фундаментальных эмоций с целью выделения доминирующего эмоционального состояния и определения настроения клиента в момент тестирования. Для этих целей был выбран тест и шкала дифференциальных эмоций (ШДЭ) американского учёного Кэрола Изарда с соответствующей методикой и алгоритмами выполнения расчётов [2].

В методике Изарда ШДЭ на момент тестирования из девяноста возможных уже выбраны десять фундаментальных эмоций и тестируемому остаётся выбрать только одну доминирующую эмоцию и соответственно состояние и не прямо, а косвенно, отвечая на вопросы, в какой степени каждое из представленных по каждой эмоции понятий описывает его самочувствие в данный момент, оценивая всё это по четырёхбалльной шкале: «совсем не подходит» – 1 балл, «пожалуй, верно» – 2 балла, «верно» – 3 балла, «совершенно верно» – 4 балла. Из десяти рядов заполненных оценками понятий выбирается ряд с максимальным значением суммы, формирующей эмоции тестируемого, которая представляет этот ряд.

Алгоритмы методики Изарда программно реализованы на языке Delphi – структурированном, объектно-ориентированном высокоуровневом языке программирования со средой разработки программного обеспечения с использованием системы программирования Delphi версии 7 фирмы Enterprise (Borland), что предоставляет наиболее широкие возможности для программирования приложений ОС Windows.

Основным требованиям при разработке программного комплекса, а также требованию простоты реализации алгоритмов и ускорения разработки удовлетворяло использование web-интерфейсов, которые имеют множество преимуществ, такие как визуализация и концепция навигации через гиперссылки.

Программа реализована в виде дополнительного пятнадцатого модуля диагностического тестирования, предназначенного для реализации функций обработки запроса по шкале методики Изарда: алгоритмы тестирования и обработки получаемой при тестировании информации, к четырнадцати существовавшим общего комплекса программных средств для диагностирования других различных психоэмоциональных состояний личности и интеллектуально-духовной реабилитации посредством предлагаемого набора мультимедийных средств.

На главной странице общей программы комплекса шкала Изарда занимает последнее место в меню и может выбираться пользователем в пункте «Диагностические тесты».

Методика этой шкалы представляет собой таблицу или бланк, состоящий из вопросов-понятий и оценок-ответов на них.

Пользователь осуществляет выбор варианта нажатием левой кнопки «мыши» на соответствующем поле таблицы (рис. 2). В конце теста при нажатии кнопки «Результат» на экран монитора выводятся результаты тестирования (рис. 3).

Результаты тестирования по модулю запоминаются в глобальные массивы для последующего формирования их и ввода в персональные (рис. 4) и групповые протокольные заключения общего диагностического комплекса.

Дифференциальная шкала эмоций по К. Изарду

Отметьте один из четырех вариантов ответа в зависимости от того, в какой степени каждое понятие описывает Ваше состояние в настоящий момент.

№ п/п	Эмоциональное состояние	Нет, не верно	Пожалуй, верно	Верно	Совершенно верно
2	Концентрированный		+		
3	Собранный		+		
4	Наслаждающийся		+		
5	Счастливый		+		
6	Радостный			+	
7	Удивленный			+	
8	Изумленный			+	
9	Пораженный		+		
10	Унылый		+		
11	Печальный	+			
12	Сломленный		+		
13	Взбешенный			+	
14	Гневный				

Результаты

Рисунок 2 – Тестирование по методике Изарда

Дифференциальная шкала эмоций по К. Изарду

Отметьте один из четырех вариантов ответа в зависимости от того, в какой степени каждое понятие описывает Ваше состояние в настоящий момент.

№ п/п	Эмоциональное состояние	Нет, не верно	Пожалуй, верно	Верно	Совершенно верно
19	Презрительный	+			
20	Пренебрегающий	+			
21	Надменный		+		
22	Пугающий		+		
23	Страшный		+		
24	Сеющий панику		+		
25	Застенчивый		+		
26	Робкий	+			
27	Стыдливый	+			
28	Сожалеющий	+			
29	Виноватый	+			
30	Раскаивающийся	+			

Ваша эмоция в настоящий момент: радость; самочувствие в целом - положительное

ПЕРЕЙТИ К ТЕРАПИИ

Рисунок 3 – Результаты тестирования по методике Изарда

ПРОТОКОЛЬНОЕ ЗАКЛЮЧЕНИЕ

о результатах выполнения комплексного компьютерного психофизиологического
диагностирования эмоционально-психологического состояния личности

Дата проведения диагностирования: 07.05.2021 г.

Диагностируемое лицо

Пол: женщина

Возраст

15. Дифференциальная шкала эмоций по К. Изарду

Баллы: 6

Заключение: Ваша эмоция в настоящий момент: радость; самочувствие в целом - положительное

Справочная информация:

Баллы в тесте начисляются по 4-х балльной шкале (1 – совсем не верно, 2 – пожалуй, верно, 3 – верно, 4 – совершенно верно). Наибольшее количество баллов соответствует вашей эмоции в данный момент, которая определяется по ключу: ряд 1 в тестовом материале (внимательный, концентрированный, собранный) соответствует эмоции интерес и т.д).

Кроме того, полезно дополнительно сравнить результаты сложения сумм отдельных эмоций, а именно: $K = (C1 + C2 + C3 + C9 + C10) / (C4 + C5 + C6 + C7 + C8)$, где К – самочувствие, C1, C2, C3... – эмоция под №1, №2, №3, №... соответственно.

Если показатель К будет больше 1, то самочувствие в целом более отвечает положительному (с повышенным настроением) типу акцентуации человека. Если К получится меньше 1 – самочувствие можно охарактеризовать как отрицательное (с пониженным настроением) типу акцентуации характера человека.

Рисунок 4 – Форма протокольного заключения

Выводы

Из вышесказанного следует вывод: основной целью исследований, проводимых в ГУ «Институт проблем искусственного интеллекта», является разработка компьютерной системы психофизиологического диагностирования, интеллектуально-духовной реабилитации и безмедикаментозной терапии психоэмоционального здоровья человека, исключить бумажные технологии тестирования и обработки его

результатов, использовать немедикаментозные средства коррекции психоэмоциональных состояний. Главными методами и средствами данного проекта являются комплексы средств изучения психоэмоциональных состояний: методики, опросники, шкалы, алгоритмы преобразования входной информации, поступающей от диагностируемых, способы ее машинной обработки и представления результатов. Общий план исследований предполагает поиск и представление для программирования комплекса методик, опросников и шкал, отражающих основные эмоциональные состояния диагностируемых с последующим их экспериментальным апробированием в качестве составляющих общей системы диагностирования. Разрабатываемая система позволяет получать объективизированные заключения по состоянию эмоционального здоровья диагностируемых на основании полученной от их запрашиваемой информации в форме ответов на задаваемые вопросы.

Неоспоримым достоинством системы СИДРТ является то, что она предназначена для индивидуального пользования. Позволяет любому пользователю, заинтересованному в нормализации своего психофизиологического состояния, осуществить данную коррекцию без особых усилий и напряжения. Позволяет разгрузить мозг и психику личности от негативных воздействий окружающей среды, полагаясь при этом на естественные возможности человеческого организма и нервной системы. Кроме того, использование комплекса СИДРТ способствует использованию интеллектуальных компьютерных технологий в более глобальных, индивидуальных, личностных масштабах.

Психофизиология и медицина, благодаря развитию IT-технологий, выходят на новую степень эволюции, взаимодействуют и взаимодополняют друг друга и являются предметом исследования для будущих поколений психологов и IT-специалистов.

Литература

1. Ильин, Е. П. Психофизиология состояний человека [Текст] / Ильин Е. П. – СПб. : Питер, 2005. – 412 с.
2. Сальников, И. С., Современные методы и методики изучения и диагностирования интеллектуально-психофизических состояний человека и способы их компьютерной аудиовизуальной терапии [Текст] / Сальников И. С., Сальников Р. И. // Искусственный интеллект. – 2006, № 4. – С. 548–554.
3. Сальников, Р. И. Разработка и построение комплексной компьютерной системы психофизиологического диагностирования, интеллектуально-духовной реабилитации и безмедикаментозной терапии [Текст] / Сальников Р.И., Шевченко А. И., Дьяченко А. В. // Искусственный интеллект. – 2007, № 3. – С. 6–16.
4. Бурлачук, Л. Ф. Психодиагностика личности [Текст] / Бурлачук, Л. Ф. – К.: Здоровье, 1989. – 168 с.
5. Хомская, Е. Д. Нейропсихология [Текст] / Хомская Е. Д. – СПб.: ПИТЕР, 2005. – 496 с.

Изосимова С.А., Пигуз В.Н. Компьютерная система психофизиологического диагностирования, интеллектуально-духовной реабилитации и безмедикаментозной терапии. В статье анализируется возможность создания и продуктивной работы компьютерной системы психофизиологического диагностирования, интеллектуально-духовной реабилитации и безмедикаментозной терапии в данный период. Данная система предназначена для использования как в коллективном (например, на предприятиях, учреждениях, организациях и т.п.), так и в индивидуальном порядках (самодиагностика и саморегуляция ПФС личности). Общая доступность в использовании способствует расширению сферы использования компьютерных систем среди широкого круга людей, не имеющих специального образования и подготовки в данной отрасли.

Ключевые слова: психофизиологическое состояние; интеллектуально-духовная терапия; саморегуляция; компьютерная система; эмоциональный искусственный интеллект.

Izosimova S.A., Piguz V.N. Computer system of psychophysiological diagnostics, intellectual and spiritual rehabilitation and drug-free therapy. The article analyzes the possibility of creating and operating a computer system for psychophysiological diagnosis, intellectual and spiritual rehabilitation and drug-free therapy in this period. This system is intended for use both in collective (for example, at enterprises, institutions, organizations, etc.) and in individual procedures (self-diagnosis and self-regulation of the personal self-defense System). General accessibility in use contributes to the expansion of the use of computer systems among a wide range of people who do not have special education and training in this industry.

Key words: psychophysiological state; intellectual-spiritual therapy; self-regulation; computer system; emotional artificial intelligence.

Практическое использование методов поиска и корректировки аномалий для построения точных прогнозов

О.В. Рычка

к.т.н., Донецкий национальный технический университет,
olga_rychka@mail.ru

Рычка О.В. Практическое использование методов поиска и корректировки аномалий для построения точных прогнозов. В данной статье описаны особенности и основные функциональные возможности разработанной автоматизированной системы поиска и корректировки аномальных измерений. Проанализирована работа предложенных автором методов и доказана их эффективность на реальных статистических данных.

Ключевые слова: аномальные измерения, автоматизированная система, поиск аномалий, корректировка аномалий, эффективность, прогноз.

Rychka O.V. Practical use of methods for finding and correcting anomalies to build accurate forecasts. This article describes the features and main functionality of the developed automated system for searching and correcting anomalous measurements. The work of the methods proposed by the author is analyzed and their effectiveness is proved on real statistical data.

Key words: anomalous measurements, automated system, anomaly search, anomaly correction, efficiency, forecast.

***Полный текст статьи опубликован в выпуске № 3 (29), 2022 г.
научного журнала «Информатика и кибернетика»***

Обзор инструментов разработки диалоговых агентов с базами знаний

Воробьёв Л.О.¹, Григорьев А.В.²

Донецкий национальный технический университет, г. Донецк

¹lev.vorobyov@rambler.ru, ²grigorievalvl@gmail.com

Воробьёв Л.О., Григорьев А.В. Обзор инструментов разработки диалоговых агентов с базами знаний Обзор современного состояния разработок в области автоматизации программирования с помощью баз знаний на примере создания диалоговых агентов. Диалоговые агенты -- это программы для ЭВМ, которые общаются с пользователем на естественном языке. Проанализированы современные разработки в области синтеза диалоговых агентов, найдено несколько патентов посвящённых созданию программ с помощью баз знаний. Составлена таблица оценки найденных патентов и программ для ЭВМ. Результаты обзора позволят применить синтез программ с помощью баз знаний для создания диалоговых агентов.

Ключевые слова: патент, платформа, база знаний, диалоговый агент, автоматизация.

Vorobyov L.O., Grigoriev A.V. Overview of tools for developing dialog agents with knowledge bases Review of the current state of developments in the field of programming automation using knowledge bases on the example of creating dialog agents. Dialog agents are computer programs that communicate with the user in natural language. Modern developments in the field of synthesis of dialog agents are analyzed, several patents dedicated to the creation of programs using knowledge bases are found. A table of evaluation of found patents and computer programs has been compiled. The results of the review will allow you to apply the synthesis of programs using knowledge bases to create dialog agents.

Keywords: patent, platform, knowledge base, agent, automation.

**Полный текст статьи опубликован в выпуске № 3 (29), 2022 г.
научного журнала «Информатика и кибернетика»**