

ISSN 2072-0297

МОЛОДОЙ УЧЁНЫЙ

МЕЖДУНАРОДНЫЙ НАУЧНЫЙ ЖУРНАЛ



27 2020
ЧАСТЬ I

16+

Молодой ученый

Международный научный журнал

№ 27 (317) / 2020

Издается с декабря 2008 г.

Выходит еженедельно

Главный редактор: Ахметов Ильдар Геннадьевич, кандидат технических наук

Редакционная коллегия:

Ахметова Мария Николаевна, доктор педагогических наук
Жураев Хусниддин Олгинбоевич, доктор педагогических наук (Узбекистан)
Иванова Юлия Валентиновна, доктор философских наук
Каленский Александр Васильевич, доктор физико-математических наук
Кошербаева Айгерим Нуралиевна, доктор педагогических наук, профессор (Казахстан)
Куташов Вячеслав Анатольевич, доктор медицинских наук
Лактионов Константин Станиславович, доктор биологических наук
Сараева Надежда Михайловна, доктор психологических наук
Абдрасилов Турганбай Курманбаевич, доктор философии (PhD) по философским наукам (Казахстан)
Авдеюк Оксана Алексеевна, кандидат технических наук
Айдаров Оразхан Турсункожаевич, кандидат географических наук (Казахстан)
Алиева Тарана Ибрагим кызы, кандидат химических наук (Азербайджан)
Ахметова Валерия Валерьевна, кандидат медицинских наук
Бердиев Эргаш Абдуллаевич, кандидат медицинских наук (Узбекистан)
Брезгин Вячеслав Сергеевич, кандидат экономических наук
Данилов Олег Евгеньевич, кандидат педагогических наук
Дёмин Александр Викторович, кандидат биологических наук
Дядюн Кристина Владимировна, кандидат юридических наук
Желнова Кристина Владимировна, кандидат экономических наук
Жуйкова Тамара Павловна, кандидат педагогических наук
Игнатова Мария Александровна, кандидат искусствоведения
Искаков Руслан Маратбекович, кандидат технических наук (Казахстан)
Кайгородов Иван Борисович, кандидат физико-математических наук (Бразилия)
Калдыбай Кайнар Калдыбайулы, доктор философии (PhD) по философским наукам (Казахстан)
Кенесов Асхат Алмасович, кандидат политических наук
Коварда Владимир Васильевич, кандидат физико-математических наук
Комогорцев Максим Геннадьевич, кандидат технических наук
Котляров Алексей Васильевич, кандидат геолого-минералогических наук
Кузьмина Виолетта Михайловна, кандидат исторических наук, кандидат психологических наук
Курпаяниди Константин Иванович, доктор философии (PhD) по экономическим наукам (Узбекистан)
Кучерявенко Светлана Алексеевна, кандидат экономических наук
Лескова Екатерина Викторовна, кандидат физико-математических наук
Макеева Ирина Александровна, кандидат педагогических наук
Матвиенко Евгений Владимирович, кандидат биологических наук
Матроскина Татьяна Викторовна, кандидат экономических наук
Матусевич Марина Степановна, кандидат педагогических наук
Мусаева Ума Алиевна, кандидат технических наук
Насимов Мурат Орленбаевич, кандидат политических наук (Казахстан)
Паридинова Ботагоз Жаппаровна, магистр философии (Казахстан)
Прончев Геннадий Борисович, кандидат физико-математических наук
Рахронов Азиз Боситович, доктор философии (PhD) по педагогическим наукам (Узбекистан)
Семахин Андрей Михайлович, кандидат технических наук
Сенцов Аркадий Эдуардович, кандидат политических наук
Сенюшкин Николай Сергеевич, кандидат технических наук
Султанова Дилшоода Намозовна, кандидат архитектурных наук (Узбекистан)
Титова Елена Ивановна, кандидат педагогических наук
Ткаченко Ирина Георгиевна, кандидат филологических наук
Федорова Мария Сергеевна, кандидат архитектуры
Фозилов Садриддин Файзуллаевич, кандидат химических наук (Узбекистан)
Яхина Асия Сергеевна, кандидат технических наук
Ячинова Светлана Николаевна, кандидат педагогических наук

Международный редакционный совет:

Айрян Заруи Геворковна, кандидат филологических наук, доцент (Армения)
Арошидзе Паата Леонидович, доктор экономических наук, ассоциированный профессор (Грузия)
Атаев Загир Вагитович, кандидат географических наук, профессор (Россия)
Ахмеденов Кажмурат Максutowич, кандидат географических наук, ассоциированный профессор (Казахстан)
Бидова Бэла Бертовна, доктор юридических наук, доцент (Россия)
Борисов Вячеслав Викторович, доктор педагогических наук, профессор (Украина)
Буриев Хасан Чутбаевич, доктор биологических наук, профессор (Узбекистан)
Велковска Гена Цветкова, доктор экономических наук, доцент (Болгария)
Гайич Тамара, доктор экономических наук (Сербия)
Данатаров Агахан, кандидат технических наук (Туркменистан)
Данилов Александр Максимович, доктор технических наук, профессор (Россия)
Демидов Алексей Александрович, доктор медицинских наук, профессор (Россия)
Досманбетова Зейнегуль Рамазановна, доктор философии (PhD) по филологическим наукам (Казахстан)
Ешиев Абдыракман Молдоалиевич, доктор медицинских наук, доцент, зав. отделением (Кыргызстан)
Жолдошев Сапарбай Тезекбаевич, доктор медицинских наук, профессор (Кыргызстан)
Игисинов Нурбек Сагинбекович, доктор медицинских наук, профессор (Казахстан)
Кадыров Кутлуг-Бек Бекмурадович, кандидат педагогических наук, декан (Узбекистан)
Кайгородов Иван Борисович, кандидат физико-математических наук (Бразилия)
Каленский Александр Васильевич, доктор физико-математических наук, профессор (Россия)
Козырева Ольга Анатольевна, кандидат педагогических наук, доцент (Россия)
Колпак Евгений Петрович, доктор физико-математических наук, профессор (Россия)
Кошербаева Айгерим Нуралиевна, доктор педагогических наук, профессор (Казахстан)
Курпаяниди Константин Иванович, доктор философии (PhD) по экономическим наукам (Узбекистан)
Куташов Вячеслав Анатольевич, доктор медицинских наук, профессор (Россия)
Кыят Эмине Лейла, доктор экономических наук (Турция)
Лю Цзюань, доктор филологических наук, профессор (Китай)
Малес Людмила Владимировна, доктор социологических наук, доцент (Украина)
Нагервадзе Марина Алиевна, доктор биологических наук, профессор (Грузия)
Нурмамедли Фазиль Алигусейн оглы, кандидат геолого-минералогических наук (Азербайджан)
Прокопьев Николай Яковлевич, доктор медицинских наук, профессор (Россия)
Прокофьева Марина Анатольевна, кандидат педагогических наук, доцент (Казахстан)
Рахматуллин Рафаэль Юсупович, доктор философских наук, профессор (Россия)
Ребезов Максим Борисович, доктор сельскохозяйственных наук, профессор (Россия)
Сорока Юлия Георгиевна, доктор социологических наук, доцент (Украина)
Узаков Гулом Норбоевич, доктор технических наук, доцент (Узбекистан)
Федорова Мария Сергеевна, кандидат архитектуры (Россия)
Хоналиев Назарали Хоналиевич, доктор экономических наук, старший научный сотрудник (Таджикистан)
Хоссейни Амир, доктор филологических наук (Иран)
Шарипов Аскар Калиевич, доктор экономических наук, доцент (Казахстан)
Шуклина Зинаида Николаевна, доктор экономических наук (Россия)

На обложке изображен *Василий Валентинович Асмус (1952)*, директор ФГБУ «НИЦ «Планета», председатель правления Российского гидрометеорологического общества, член коллегии Росгидромета, доктор физико-математических наук, профессор, заслуженный деятель науки РФ, лауреат премий Правительства РФ и Правительства Москвы, академик Российской академии космонавтики им. К. Э. Циолковского.

Василий Валентинович родился в Москве, в семье известного русского философа и видного деятеля советской культуры Валентина Фердинандовича Асмуса. Он закончил Московский институт электронного машиностроения (факультет прикладной математики), а затем аспирантуру Вычислительного центра СО РАН. С 1976 года он начал свою деятельность в НИЦ «Планета», с 1997 года — в должности директора.

Асмус широко известен в нашей стране и за рубежом как специалист в области математического моделирования и информатики. Он активно участвует в разработке государственных программ и реализации планов создания космических комплексов наблюдения Земли, а также руководит работами по созданию космических информационных систем, выполняющих стратегические задачи в интересах народного хозяйства и обороны. Под руководством Асмуса спроектирована, разработана и сдана в эксплуатацию крупнейшая в России и одна из самых крупных в мире государственная территориально-распределенная система космического мониторинга окружающей среды. Василий Валентинович внес значительный вклад в разработку технологий дистанционного исследования атмосферы, суши, криосферы, акваторий Мирового океана. Созданные им программные системы обработки спутниковых данных внедрены и широко используются в России и за рубежом. Василий Валентинович принимал активное участие в испытаниях и эксплуатации косми-

ческих систем серий «Метеор», «Метеор-Природа», «Ресурс», «Океан», «Электро», «Канопус» и др.

Василий Асмус — автор более 150 научных работ, включая монографии и патенты, в отечественных и зарубежных изданиях. Он является членом оргкомитетов и программных комитетов международных и отечественных конференций, а также редколлегий ряда периодических научно-технических изданий, главным редактором журнала «Метеорология и гидрология».

Асмус — член государственных комиссий по летным испытаниям космических комплексов социально-экономического и оборонного назначения, а также средств выведения, член Межведомственной экспертной комиссии по космосу, заместитель председателя Совета главных конструкторов по космическим комплексам, член Совета по космосу РАН, член научно-технических советов Военно-промышленной комиссии Российской Федерации, Росгидромета и Роскосмоса, председатель ученого совета ФГБУ «НИЦ «Планета». В качестве эксперта по космическим технологиям он представляет Россию в международных организациях: WMO, CEOS, CGMS, GEOSS, EUMETSAT.

Василий Валентинович Асмус удостоен государственных (орден Дружбы, орден «За заслуги перед Отечеством» II степени, медаль «В память 850-летия Москвы», грамота Президента РФ) и ведомственных наград (Росгидромет, Роскосмос, Минприроды России, РАН, Минобороны России, МЧС России), наград WMO и ICSU, почетных званий «Заслуженный создатель космической техники», «Заслуженный испытатель космической техники», «Ветеран космонавтики России», «Почетный геодезист», «Почетный работник Гидрометеослужбы России», «Ветеран гидрометеослужбы Вооруженных сил России», «Лучший руководитель России».

Екатерина Осянина, ответственный редактор

СОДЕРЖАНИЕ

ПОБЕДИТЕЛИ КОНКУРСА УМНИК В РАМКАХ НАЦИ- ОНАЛЬНОЙ ПРОГРАММЫ «ЦИФРОВАЯ ЭКОНОМИКА РОССИЙСКОЙ ФЕДЕРАЦИИ»

Морозов А. П.

Разработка информационно-сервисной системы мониторинга пунктов сбора и переработки мусора1

Мухаметгалиев Р. Д.

Разработка программы-тренажера для наклонно-направленного бурения 4

МАТЕМАТИКА

Абдиреймов А.

Описание гранево симметричных пространств малых размерностей7

**Иванова А. А., Скородумова Ю. В.,
Кулагина С. А., Тарасов А. А., Павлов Д. Л.**

Применение мультиномиальной логистической регрессии для анализа смертности от коронавируса в регионах РФ 9

**Кулагина С. А., Скородумова Ю. В.,
Иванова А. А.**

Задача адресной перевозки с временными окнами 12

**Мелков Н. А., Ржавитина М. А., Пак А. С.,
Спиридонов А. Ю.**

Определение предпочтительного числа кластеров. Момент остановки метода одиночной связи 16

**Пак А. С., Кашеева О. А., Мелков Н. А.,
Спиридонов А. Ю., Ржавитина М. А.,
Кулагина С. А.**

Анализ влияния вычислительной погрешности в явных методах Рунге — Кутты 18

**Скородумова Ю. В., Иванова А. А.,
Кулагина С. А., Тарасов А. А., Павлов Д. Л.**

Анализ больших деформаций решетчатой пластины глаза 23

ИНФОРМАЦИОННЫЕ ТЕХНО- ЛОГИИ

**Балавнев Д. А., Киндулов М. Л., Горелов Б. Р.,
Шергин Т. О.**

Использование нейронных сетей в задаче прогнозирования закупок товаров30

**Горелов Б. Р., Балавнев Д. А., Шергин Т. О.,
Киндулов М. Л.**

Разработка системы удаленной компиляции..... 32

**Киндулов М. Л., Балавнев Д. А., Горелов Б. Р.,
Шергин Т. О.**

Методы детектирования состязательных атак ... 34

Кириллов П. А.

Искусственный интеллект для образования. Адаптивная система обучения.....39

Круглов Р. М.

Перспективы обучения студентов основам автоматического управления с помощью средств разработки на языке CFD 43

**Можайский Г. В., Мищенко А. Д., Ежов Ф. В.,
Мартынюк Р. А., Кононыхин И. А.**

Разработка алгоритма определения топологии биологической ткани животных и человека по начальному срезу.....45

Морозов А. Д., Мартынов П. А., Чернышова А. П., Сиркин Т. В. Децентрализованная система управления программным обеспечением для IoT-устройств	49
Мартынов П. А., Морозов А. Д., Сиркин Т. В., Чернышова А. П. Разработка программного обеспечения для децентрализованной идентификации IoT-устройств в сетях	52
Павлов Д. Л., Кащеева О. А., Иванова А. А., Скородумова Ю. В., Тарасов А. А. Разработка в среде C# программной реализации алгоритма выделения структурных особенностей	54
Порошин И. Е. Большие данные (Big Data) и наука о данных (Data Science): прогрессивные тренды современности.....	59
Ракинцев Н. А. Инциденты информационной безопасности 1-го полугодия 2020 года в транспортной отрасли. Введение мер по их предотвращению	61
Ржавитина М. А., Мелков Н. А., Пак А. С., Спиридонов А. Ю. Обзор платформ для разработки смарт-контрактов	63
Сиркин Т. В., Чернышова А. П., Мартынов П. А., Морозов А. Д. Разработка автоматизированной системы составления и оптимизации расписания занятий	65
Спиридонов А. Ю., Пак А. С., Ржавитина М. А., Мелков Н. А. Проблема оптического распознавания субтитров, встроенных в видеопоток	71
Чернышова А. П., Сиркин Т. В., Морозов А. Д., Мартынов П. А. Анализ распространения высказываний с негативной тональностью в пользовательских дискуссиях в социальных сетях	73
Шергин Т. О., Горелов Б. Р., Балавнев Д. А., Киндулов М. Л. Методы детектирования искусственных новостей	77

ПОБЕДИТЕЛИ КОНКУРСА УМНИК В РАМКАХ НАЦИОНАЛЬНОЙ ПРОГРАММЫ «ЦИФРОВАЯ ЭКОНОМИКА РОССИЙСКОЙ ФЕДЕРАЦИИ»

Разработка информационно-сервисной системы мониторинга пунктов сбора и переработки мусора

Морозов Андрей Павлович

Красноярский колледж радиоэлектроники и информационных технологий



В современном мире с ростом населения и промышленного производства проблема утилизации мусора становится все актуальнее. Огромные территории отводятся под свалки. Мусор долго разлагается, производит токсины, портит природу и ее ресурсы, которыми мы пользуемся: воздух, воду, еду (рис. 1). При сжигании

мусора образуются токсичные вещества — тяжелые металлы, диоксины, которые имеют свойство накапливаться в воде, почве, в телах животных и человека. Они вызывают серьезные заболевания. Самый безопасный способ обращения с отходами — отдельный сбор и переработка.



а б
Рис. 1. Скопления мусора на земле (а) и в воде (б)

В России существует около 1000 мусороперерабатывающих предприятий, причем они недозагружены и используются не на полную мощь. Раздельный сбор отходов не только даст дышать полной грудью уже функционирующим предприятиям, но и позволит строить новые. Во многих городах нашей страны в больших количествах расположены стационарные и мобильные пункты приема вторсырья, так что задача раздельного сбора отходов легко выполнима.

В настоящее время на территории России отсутствуют сервисы, предоставляющие полную информацию о преимуществах раздельного сбора отходов, а также о нахождении пунктов сбора отходов того или иного типа. Кроме того, у населения отсутствует мотивация заниматься раздельным сбором отходов. Здесь показателен европейский опыт. Например, в Норвегии в ценнике продукта, продаваемого в металлической банке, отдельно обозначена сто-

имость этой банки. При этом практически все торговые центры страны оборудованы специальными устройствами, где человек может сдать пустую банку и возместить себе ее стоимость.

Цель данного проекта заключается в разработке информационно-сервисной системы мониторинга пунктов сбора и переработки мусора. Система должна информировать население о пунктах раздельного приема мусора и преимуществах применения данного подхода на практике. В рамках выполнения этой цели были поставлены следующие задачи:

1. Исследование текущей ситуации с раздельным сбором мусора в городах России с целью выявления препятствий для раздельного сбора мусора и разработки наиболее оптимальной модели раздельного сбора мусора.
2. Разработка геоинформационной системы с внесенными в нее пунктами раздельного сбора мусора (рис. 2).

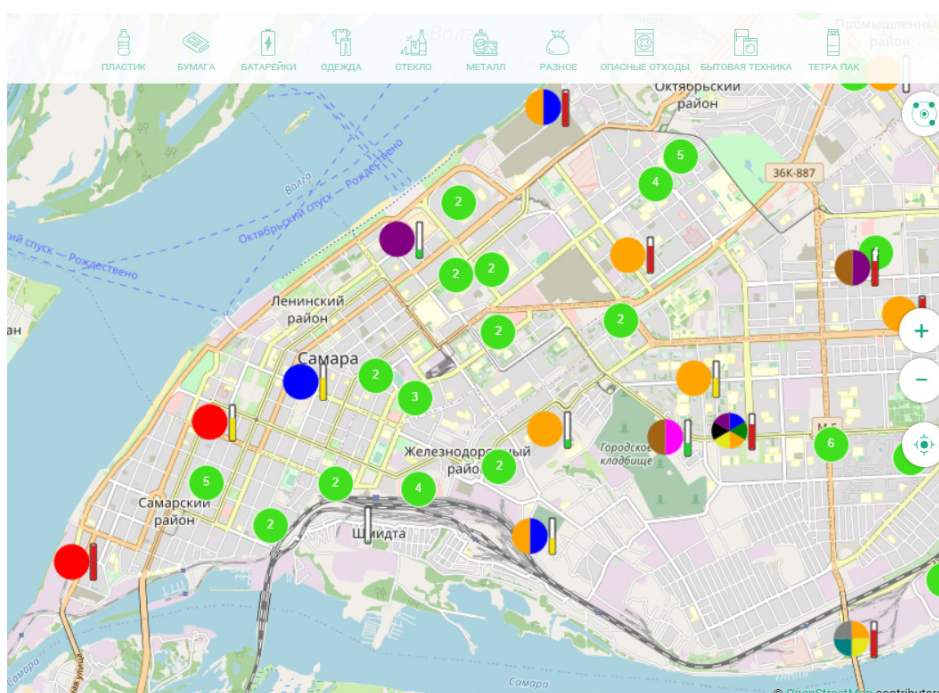


Рис. 2. Геоинформационная система с внесенными в нее пунктами раздельного сбора мусора

3. Разработка функционала построения маршрутов в созданной системе и определения ближайших к данному маршруту пунктов сбора мусора (рис. 3). Данный функци-

онал позволит пользователю заехать на ближайший пункт сбора мусора, например, по дороге на работу без существенного отклонения от маршрута.



Рис. 3. Построение маршрута до выбранной цели и отображение ближайших к этому маршруту пунктов сбора мусора

4. Разработка нейронной сети для распознавания основных типов отходов. Данный функционал позволит пользователю быстро найти ближайший пункт приема отходов, тип которых будет автоматически определен, например, по сделанной на мобильный телефон фотографии.

5. Исследование проблемы мотивации граждан на раздельный сбор мусора с целью разработки оптимальной мотивирующей бонусной программы. В рамках этого пункта предполагается использовать систему достижений (рис. 4) и систему накопления баллов с последующей геймификацией (рис. 5, 6).

Достижения






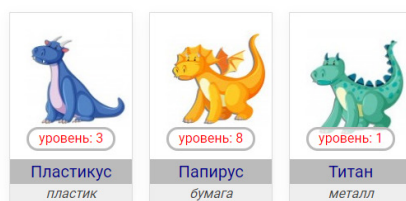
-  Выкинуть мусор в первый раз
-  Выкинуть первые батарейки
-  Выкидывать мусор каждую неделю на протяжении месяца
-  Лучший в своем районе по раздельному мусору за месяц
-  Лучший в своем доме по раздельному мусору за месяц

Рис. 4. Достижения пользователя

Геймификация

Мои питомцы



Демо бой питомцев

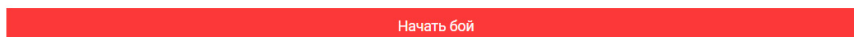


Рис. 5. Система накопления баллов в виде повышения уровня питомцев

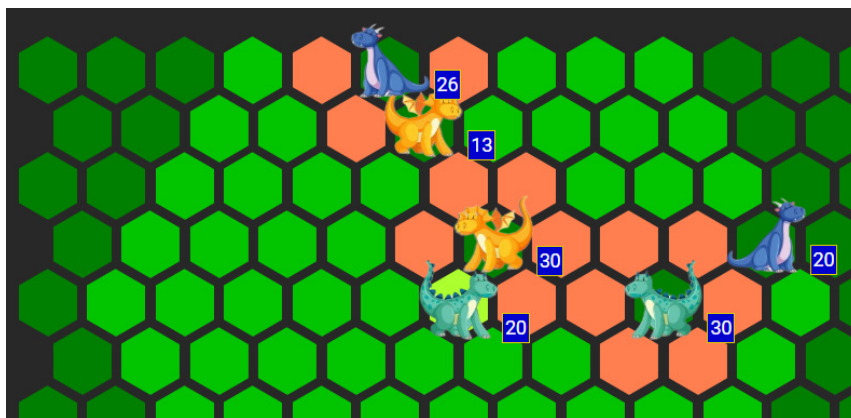


Рис. 6. Геймификация: проведение боев между питомцами разных пользователей

В качестве конечного результата предполагается создание web-приложения, а также его мобильной версии. Приложение будет объяснять пользователям преимущества раздельной сдачи отходов на специализированных пунктах, предоставлять конкретную информацию о местонахождении

пунктов приема отходов определенного типа, строить оптимальный маршрут до выбранного пункта, а также с помощью специальных систем достижений и накопления баллов мотивировать пользователей на раздельный сбор отходов и, как следствие, очищение нашей планеты.

Разработка программы-тренажера для наклонно-направленного бурения

Мухаметгалиев Радмир Дамирович, студент
Уфимский государственный нефтяной технический университет

Development of the simulator program for directional drilling

Muhametgaliev Radmir Damirovich, student
Ufa State Oil Technical University



Зачастую процесс бурения наклонно-направленных скважин (ННБ) может сопровождаться различного рода технико-технологическими осложнениями [1]. При этом непрерывный и эффективный процесс бурения наклонно-направленных и горизонтальных скважин (ННГС) имеет высокую зависимость от компетенции инженерно-технического персонала, который обязан оперативно принимать сложные решения в условиях производственной неопределенности.

Разрабатываемая для решения задачи повышения компетенции инженеров ННБ программа-тренажер состоит из сложных математических и имитационных моделей [2], программный алгоритм представлен разветвленно-циклическим типом, а человеко-машинный интерфейс (ЧМИ) адаптирован под пользователя путем оптимизации размеров управляющих графических объектов диалоговых окон (Рис. 1), состава и расположения блоков ЧМИ для информирования и управления имитируемым процессом (Рис. 2).

Функциональными преимуществами тренажера выступают не столько возможности дублирования производственного программного обеспечения для обработки параметров кривизны скважины, сколько имитация про-

изводственных процессов, которые благодаря личной научно-исследовательской работе автора имеют высокие показатели подобия с реальным производством.

Несмотря на то что тренажер находится на активной стадии завершения разработки, с начала разработки 2017 года было опубликовано

несколько статей в научно-популярных и социально-массовых издательских ресурсах [2,3], в том числе на официальной странице социальной сети Минобрнауки РФ, получено два авторских свидетельства, апробирован в рамках научно-технической конференции [4].

Ориентированный на повышение качества обучения в учебно-образовательных учреждениях и отработку действий инженера ННБ тренажер не остался без внимания колледжей, вузов, производственных компаний. К маю 2020 года тренажер успешно коммерциализирован и применяется в Пермском колледже и в двух производственных предприятиях. Покупатели отметили яркий интерес среди их пользователей, ведь благодаря работе в режиме конкурсного соревнования проявляется азарт не только по отношению к тренажеру, но и производственной деятельности в целом.



Рис. 1. Человеко-машинный интерфейс диалоговых окон тренажера: настройки параметров тренировки (слева), имитации процесса бурения направленной скважины (справа)

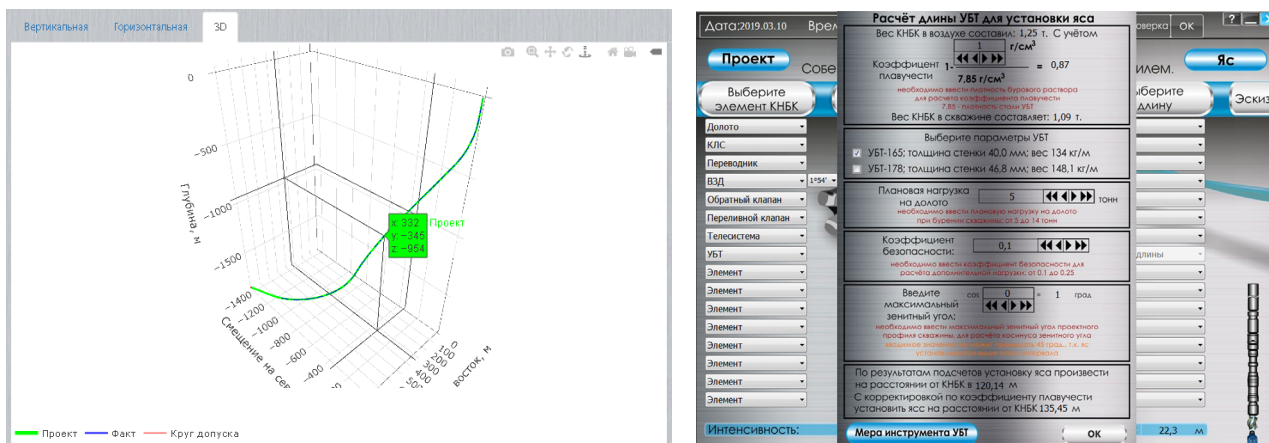


Рис. 2. Человеко-машинный интерфейс диалоговых окон тренажера: трехмерная модель траектории направленной скважины (слева), имитации расчета скважинного оборудования (справа)

Несмотря на то что релиз модернизированных версий тренажера запланирован на несколько лет вперед, уже сейчас ведется работа по непосредственному по расширению функционала с уклоном на геймификацию, что позволит проводить обучение в игровой форме, а свидетельства о достигнутых результатах в тренажере послужат подтверждением компетенции соискателя на собеседовании.

В сфере HR тренажер может успешно помочь решать кадровые вопросы в компаниях, оказывающих сервис

по технико-технологическому сопровождению бурения нефтяных и газовых скважин. Тренажер значительно облегчает процесс оценки квалификации и компетенций кандидатов при приеме инженеров на работу, моделируя производственную среду в специализированном виде. Потенциальные сотрудники при помощи тренажера могут продемонстрировать свои навыки проводки горизонтальной скважины, строительства бокового ствола, бурения с аномальными условиями, а также при возникновении множества возможных рисков ситуаций.

Литература:

1. Исмаков, Р. А., Рахматуллин Д. В., Мухаметгалиев И. Д. Проектирование профиля наклонно-направленной скважины с применением ЭВМ // Электронное учебно-методическое пособие. Уфа. 2014. с. 18.
2. Агзамов, З. В., Мухаметгалиев И. Д. Экспертная система формирования компоновки низа бурильной колонны для управления направленным бурением скважин // Проблемы автоматизации технологических процессов добычи, транспорта и переработки нефти и газа. — 2017. — с. 134-140.
3. Исмаков, Р. А., Хафизов А. Р., Мухаметгалиев И. Д., Гуменников С. Г., Галлямов М. Р. Анализ работы имитационных тренажерных комплексов для обучения практическим навыкам бурения // Нефтегазовое дело. 2016. № 4 (14). с. 9-13.
4. Мухаметгалиев, И. Д., Исмаков Р. А., Чиглинцев С. С. Аккредитация кадров и рационализация набора полевого персонала объектов бурения. Материалы II научно-технической конференции «Сервисные услуги в добыче нефти // Уфимский государственный нефтяной технический университет. 2015. с. 148-154.

МАТЕМАТИКА

Описание гранево симметричных пространств малых размерностей

Абдиреймов Арысланбай, студент
 Каракалпакский государственный университет имени Бердаха (г. Нукус, Узбекистан)

В этой работе дается описание единичных шаров конечномерных нейтральных сильно гранево симметричных пространств малых размерностей.

Исследования гранево симметричных пространств связаны с геометрической характеристикой предопределенных пространств JBW^* -троек, допускающих алгебраическую структуру, и восходят к работам Я. Фридмана и Б. Руссо [1]. Аксиомы, требуемые в этих характеристиках, являются естественными предположениями для пространств состояний физических систем. Такие пространства рассматриваются как геометрические модели для состояний квантовой механики. Естественно, что предопределенные пространства для комплексных алгебр фон Неймана и более общих JBW^* -троек являются нейтральными сильно гранево симметричными пространствами [2].

В работе [3] были даны геометрическая характеристика комплексных гильбертовых пространств и комплексных спин-факторов, а также дано описание JBW^* -троек ранга 1 и 2, факторов Картана типа 1 и 4. М. Нейл и Б. Руссо в [4] нашли геометрические условия, при которых гранево симметричное пространство является изометричным предопределенному пространству JBW^* -тройки. Условие, при котором слабо гранево симметричное пространство является сильно гранево симметричным пространством, было найдено Н. Ядгоровым в [5].

Сначала приведем необходимые сведения о гранево симметричных пространствах.

Пусть Z — вещественное или комплексное нормированное пространство. Элементы $f, g \in Z$ называются ортогональными, обозначение $f \diamond g$, если

$$\|f + g\| = \|f - g\| = \|f\| + \|g\|.$$

Грань F единичного шара $Z_1 = \{f \in Z : \|f\| \leq 1\}$ называется выставленной по норме, если $F = F_u = \{f \in Z_1 : f(u) = 1\}$ для некоторого $u \in Z^*$ с $\|u\| = 1$. Элемент $u \in Z^*$ называется проективной единицей, если $\|u\| = 1$ и $g(u) = 0$ для всех $g \in F_u^\diamond$.

Выставленная по норме грань F_u из Z_1 называется симметричной гранью, если существует линейная изометрия S_u из Z на Z такая что $S_u^2 = I$, и множество всех неподвижных точек которой в точности совпадает с топологической прямой суммой замыкания $\overline{sp F_u}$ линейной оболочки грани F_u и ее ортогонального дополнения F_u^\diamond , т. е. совпадает с $(\overline{sp F_u}) \oplus F_u^\diamond$.

Пространство Z называется слабо гранево симметричным пространством (WFS-пространством), если каждая выставленная по норме грань F_u из Z_1 симметрична (см. [1]).

Для каждой симметричной грани F_u определяются сжимающие проекторы $P_k(F_u)$, $k = 0, 1, 2$ на Z следующим образом: Во первых $P_1(F_u) = (I - S_u)/2$ является проектором на собственное подпространство соответствующий собственному значению -1 симметрии S_u , Далее определим $P_0(F_u)$ и $P_2(F_u)$ как проекторы из Z на F_u^\diamond

и $\overline{sp} F_u$ соответственно, т. е. $P_2(F_u) + P_0(F_u) = (I + S_u)/2$. Проекторы $P_k(F_u)$ называются геометрическими Пирсовскими проекторами.

Проективная единица $u \in Z^*$ называется геометрическим трипотентом, если F_u является симметричной гранью и $S_u^* u = u$ для симметрии S_u , соответствующей F_u .

WFS-пространство Z называется сильно гранево симметричным пространством (SFS-пространством), если для каждой симметричной грани F_u из Z_1 и каждого $y \in Z^*$ с $\|y\| = 1$ и $F_u \subset F_y$ мы имеем $S_u^* y = y$, где S_u — симметрия, соответствующая F_u (см. [1]).

Сжимающий проектор Q на Z называется нейтральным, если для каждого $f \in Z$ равенство $\|Qf\| = \|f\|$ влечет $Qf = f$. Пространство Z называется нейтральным, если для каждой симметричной грани F_u , проектор $P_2(u)$, соответствующей F_u , является нейтральным.

Определение [3]. Сильно гранево симметричное пространство Z называется пространством ранга n , если всякое семейство взаимно ортогональных геометрических трипотентов имеет мощность не более n , и существует по крайней мере одно семейство взаимно ортогональных геометрических трипотентов содержащее ровно n элементов (обозначение $rank Z = n$).

Имеет место следующая теорема.

Теорема 1. Пусть Z — вещественное нейтральное сильно гранево симметричное пространство.

1) Если $\dim Z = 2$, то Z изометрически изоморфно пространству \mathbb{R}^2 с нормой либо $\|x\| = \sqrt{x_1^2 + x_2^2}$, либо $\|x\| = |x_1| + |x_2|$, где $x = (x_1, x_2) \in \mathbb{R}^2$.

2) Если $\dim Z = 3$, то Z изометрически изоморфно пространству \mathbb{R}^3 с нормой либо $\|x\| = \sqrt{x_1^2 + x_2^2 + x_3^2}$, либо $\|x\| = |x_1| + \sqrt{x_2^2 + x_3^2}$, либо $\|x\| = |x_1| + |x_2| + |x_3|$, где $x = (x_1, x_2, x_3) \in \mathbb{R}^3$.

Следующая теорема, дает описание единичных шаров четырехмерных нейтральных сильно гранево симметричных пространств.

Теорема 2. Пусть Z — четырехмерное вещественное нейтральное сильно гранево симметричное пространство.

1) Если $rank Z = 1$, то Z изометрически изоморфно пространству \mathbb{R}^4 с нормой $\|x\| = \sqrt{x_1^2 + x_2^2 + x_3^2 + x_4^2}$;

2) Если $rank Z = 2$, то Z изометрически изоморфно пространству \mathbb{R}^4 с нормой либо $\|x\| = |x_1| + \sqrt{x_2^2 + x_3^2 + x_4^2}$, либо $\|x\| = \sqrt{x_1^2 + x_2^2} + \sqrt{x_3^2 + x_4^2}$;

3) Если $rank Z = 3$, то Z изометрически изоморфно пространству \mathbb{R}^4 с нормой $\|x\| = |x_1| + |x_2| + \sqrt{x_3^2 + x_4^2}$;

4) Если $rank Z = 4$, то Z изометрически изоморфно пространству \mathbb{R}^4 с нормой $\|x\| = |x_1| + |x_2| + |x_3| + |x_4|$, где $x = (x_1, x_2, x_3, x_4) \in \mathbb{R}^4$.

Геометрический трипотент u называется *максимальным*, если

$$P_0(u) = 0.$$

Обозначим

$$D_1(u) := \{ y \in Z^* : \exists a > 0, \|u + ay\| = \|u - ay\| = 1 \}.$$

Имеет место следующая

Теорема 3. Пусть Z — сильно гранево симметричное пространство и $u \in Z^*$, $\|u\| = 1$. Тогда следующие соотношения эквивалентны:

- 1) u — максимальный геометрический трипотент;
- 2) u — экстремальная точка Z_1^* ;
- 3) $D_1(u) = \{0\}$.

Литература:

1. Friedman, Y., Russo B. Affine structure of facially symmetric spaces // Math. Proc. Camb. Philos. Soc. Vol. 106. 1989. № 1. P. 107-124.
2. Friedman, Y., Russo B. Some affine geometric aspects of operator algebras // Pac. J. Math. Vol. 137. 1989. № 1. P. 123-144.
3. Friedman, Y. and Russo B. Geometry of the dual ball of the spin factor // Proc. Lon. Math. Soc. 1992. Vol. 65. № 3. P. 142-174.
4. Neal, M., Russo B. State space of JB*-triples // Math. Ann. 2004. Vol. 328. № 4. P. 585-624.
5. Ядгоров, Н.Ж. Слабо и сильно гранево симметричные пространства // Докл. АН РУз. 1996. № 5. с. 6-8.
6. <https://cyberleninka.ru/article/n/geometricheskaya-harakterizatsiya-veschestvennyh-jbw-faktorov>

Применение мультиномиальной логистической регрессии для анализа смертности от коронавируса в регионах РФ

Иванова Анастасия Анатольевна, студент;
 Скородумова Юлия Владимировна, студент;
 Кулагина Софья Александровна, студент;
 Тарасов Антон Алексеевич, студент;
 Павлов Дмитрий Львович, студент
 Санкт-Петербургский государственный университет

В исследовании рассмотрено применение мультиномиальной логистической регрессии при анализе уровня смертности от коронавируса. В работе приведены результаты построения модели, проведена интерпретация и оценка качества моделей. По результатам работы выявлены факторы, оказывающие влияние на целевую переменную.

Ключевые слова: регрессия, коронавирус, уровень смертности.

Мультиномиальные модели — это модели множественного выбора. Данный тип моделей применяется на наборах данных, содержащих некоторое количество объясняющих независимых переменных x и зависимую переменную y . Целевая переменная y является категориальной, она может являться номинальной или порядковой и имеет 3 и более альтернатив.

Допустим, в ходе эксперимента было проведено n наблюдений, в которых целевая переменная y_i может принимать m различных значений. Вероятность того, что i -е наблюдение имеет класс j описывается формулой:

$$p_{ij} = P\{y_i = j\} = F_j(\beta^T x_i), \quad j = 1, \dots, m, \quad i = 1, \dots, n,$$

где F_j — некоторая функция распределения, отвечающая условиям того, что вероятность p_{ij} при фиксированном i принадлежит отрезку от 0 до 1, а в сумме такие вероятности дают 1.

В случае семейства логистических моделей, вместо функции F_j используется логистическая функция распределения:

$$\Lambda(z) = \frac{e^z}{1 + e^z}$$

Можно выделить несколько типов мультиномиальных логистических моделей. Есть ряд особенностей, связанных с природой объясняющих переменных. В зависимости от них используют разные подвиды модели.

Если значение переменной x_i не зависит от значения y_i в рамках одного наблюдения, т. е. x_i имеет одни и те же значения для каждого класса y_p , то используется мультиномиальная логит модель:

$$p_{ij} = \frac{e^{\beta_j^T x_i}}{\sum_{i=1}^m e^{\beta_i^T x_i}}, \quad j = 1, \dots, m.$$

Поскольку сумма вероятностей p_{ij} равна единице, то для определения модели накладывается условие $\beta_i = 0$, если первая переменная выбрана базовой.

Данные для анализа уровня смертности от коронавируса собирались по всем 85 субъектам России из открытых источников, таких как Росстат, Роспотребнадзор и содержат такие показатели как: общее количество коек в больницах, количество коек в инфекционных палатах, врачей, общая смертность региона, население региона, его плотность, показатель валового регионального продукта, доля пенсионеров среди населения региона, количество аппаратов ИВЛ. Коэффициент смертности от коронавируса находится как отношение смертей от коронавируса к общей смертности субъекта РФ. Регионы были разбиты на 3 категории, с низким, средним и высоким уровнем смертности от коронавируса на равные группы.

region	covid_coef_death	all_bed	infective_bed	doctors	population	density	vrp	ivl	pensioner
Алтайский край	0.009204148	23195	1200	9957	2317153	13.79	550.0	869	32.71087
Амурская область	0.009006575	8254	447	4366	790044	2.18	301.1	336	29.76151
Архангельская область	0.006748094	11476	595	6403	1092424	2.64	514.0	419	38.00621
Астраханская область	0.052511815	9572	655	6209	1005782	20.52	553.4	240	26.03815
Белгородская область	0.024199013	11952	621	6263	1549151	57.09	866.0	260	32.94805
Брянская область	0.068123758	9408	326	4861	1192491	34.21	328.8	163	34.96152
Владимирская область	0.056649200	12017	556	4769	1358416	46.71	440.5	426	33.95587
Волгоградская область	0.036626683	22779	1155	11304	2491036	22.07	852.0	700	30.75764
Вологодская область	0.006110229	10013	498	4127	1160445	8.02	582.6	257	32.06287
Воронежская область	0.036494131	19987	1136	12164	2324205	44.51	943.6	348	32.14695

Одним из важных шагов при подготовке данных к построению регрессионных моделей, является проверка параметров на мультиколлинеарность. Для этого сначала построим корреляционную матрицу признаков. При абсолютных значениях коэффициента корреляции больше 0.8 связь между переменными считается сильной и использование таких параметров вместе может привести к плохому качеству модели, поэтому необходимо либо удалить такие переменные, либо преобразовать так, чтобы уменьшить корреляцию.

	all_bed	infective_bed	doctors	population	density	vrp	ivl	pensioner
all_bed	1.00	0.92	0.26	0.97	0.62	0.78	0.90	-0.24
infective_bed	0.92	1.00	0.25	0.86	0.46	0.58	0.72	-0.28
doctors	0.26	0.25	1.00	0.23	0.00	0.06	0.16	-0.15
population	0.97	0.86	0.23	1.00	0.68	0.85	0.93	-0.28
density	0.62	0.46	0.00	0.68	1.00	0.82	0.79	-0.26
vrp	0.78	0.58	0.06	0.85	0.82	1.00	0.93	-0.31
ivl	0.90	0.72	0.16	0.93	0.79	0.93	1.00	-0.27
pensioner	-0.24	-0.28	-0.15	-0.28	-0.26	-0.31	-0.27	1.00

Из корреляционной матрицы можно сделать вывод, что в данных присутствуют коррелирующие величины.

Общее количество коек и количество коек в инфекционных палатах имеют сильную прямую зависимость, а значит не могут одновременно использоваться при построении моделей. Переменные ВРП региона и количество аппаратов ИВЛ также имеют сильную прямую зависимость с плотностью населения. Имеет смысл взять относительные показатели этих величин количество аппаратов ИВЛ на 100 тысяч населения и аналогичный параметр для ВРП. Переменная *population* имеет сильную корреляцию со многими другими параметрами, поэтому исключаем ее.

Проверку на мультиколлинеарность можно произвести с помощью коэффициента VIF (Variance Inflation Factor). Он показывает, во сколько раз возрастает дисперсия коэффициента регрессии из-за того, что переменные коррелируют по сравнению с тем, когда они не коррелируют. Если значение VIF > 5, то имеет место мультиколлинеарность. В данных не обнаружено мультиколлинеарности, значит, можно строить модели.

переменные	VIF
infective_bed	1.4453
doctors	1.1109
density	1.3785
pensioner	1.1829
ivl100	1.1244
vrp100	1.1095

Чтобы можно было оценить предсказательную мощьность модели, разделим выборку на 2 части в пропорции 80/20: тренировочную, на которой модель будет обучаться, и тестовую, на которой будет выполняться проверка ее точности.

Было построено множество моделей, а с помощью метода обратного исключения последовательно удаляя из уравнения регрессии наименее значащие предикторы, получена наиболее оптимальная модель. Она также оказалась наилучшей по информационному критерию AIC и предсказательной способности на тестовых данных.

За базовый уровень был принят низкий уровень смертности. Для него все коэффициенты модели равняются нулю. А другие 2 уровня сравниваются с ним. Коэффициенты для высокого и среднего уровня смертности модели представлены в таблице ниже.

переменные	high	middle
свободный член	1.748946	1.495921
infective_bed	-0.001229	-0.000832
density	0.016465	0.015003
ivl100	-0.060092	-0.051951

Зададим уровень значимости 0.05. Тогда при p-value < 0.05 параметр модели значим. Согласно критерию отношения правдоподобия модель значима в целом, т.к. p-value = 0.017474, а значит такую модель можно использовать для предсказаний на тестовой выборке. Модель статистически значима в целом, ее коэффициенты значимы по критерию Вальда и проверка на тестовых данных дает хороший прогноз — 78.57% правильных предсказаний.

Логарифмы отношения шансов для модели выглядят как:

для низкого уровня смертности

$$\ln(g_0(x)) = \ln \frac{P(low)}{P(low)} = \ln(1) = 0$$

для среднего уровня смертности

$$\ln(g_1(x)) = \ln \frac{P(middle)}{P(low)} = b_{10} + b_{11} \text{infective}_{bed} + b_{12} \text{density} + b_{13} \text{ivl100}$$

для высокого уровня смертности

$$\ln(g_2(x)) = \ln \frac{P(high)}{P(low)} = b_{20} + b_{21} \text{infective}_{bed} + b_{22} \text{density} + b_{23} \text{ivl100}$$

где b_{ij} — коэффициенты модели, индекс i отвечает за класс (1 — middle, 2 — high), j — за переменную, перед которой стоит коэффициент.

Чтобы получить значение вероятности выбора класса $j = \{low, middle, high\}$, необходимо перейти от логарифмов отношения шансов к выражению:

$$P(y = j) = \frac{g_j(x)}{1 + g_1(x) + g_2(x)}$$

Интерпретация коэффициентов b_{ij} моделей логистической регрессии такова: при изменении значения переменной x_j на 1 единицу, при неизменных остальных значениях переменных, логарифм шансов изменяется в $e^{b_{ij}}$ раз.

В таблице представлены значения $e^{b_{ij}}$ для переменных.

	своб. член	infective_bed	density	ivl100
middle	4.4634	0.9991	1.1511	0.9493
high	5.7485	0.9987	1.0166	0.9416

Коэффициенты данной модели можно рассмотреть с точки зрения отношения шансов. При увеличении количества коек в инфекционных палатах на 1, при неизменных других факторах, шансы, что в регионе будет высокий уровень смертности по сравнению с низким, понижаются в 0.99 раз, т.е. значительное увеличение количества инфекционных коек может привести к снижению уровня смертности. Аналогично, увеличение коэффициента оснащения аппаратами ИВЛ также может способствовать небольшому снижению смертности в 0.94 раза.

Количество коек в инфекционных палатах является важным параметром при исследовании уровня смертности, т.к. вирус очень быстро распространяется и необходимо оперативно изолировать зараженных людей от здоровых, чтобы как можно меньше людей заболели, получили квалифицированную помощь и в случае осложнений, когда часто необходим аппарат ИВЛ, не скончались. Также можно отметить, что плотность населения играет немаловажную роль, наша страна обладает большой территорией и на разные регионы ложится разная нагрузка.

По итогам исследования выявлено, что наиболее сильное влияние оказывают такие параметры, как количество инфекционных палат, плотность населения, количество аппаратов искусственной вентиляции легких на 100 тысяч населения региона. В будущем можно увеличить количество факторов и выявить новые зависимости, влияющие на уровень смертности от коронавируса на территории Российской Федерации.

Литература:

1. Long, J. S. Regression models for categorical and limited dependent variables, Thousand Oaks: Sage Publ., 1997
2. Cameron, A. C., Pravin K. T. Microeconometrics: Methods and Applications, Cambridge University Press, 2005
3. Fei, Yu Ting, Du Ronghui, Clinical course and risk factors for mortality of adult inpatients with COVID-19 in Wuhan, China: a retrospective cohort study/Journal Title: The Lancet/2020
4. Федеральная служба государственной статистики — URL: <http://www.gks.ru/> — Текст: электронный.
5. Роспотребнадзор — URL: <https://rosпотребнадзор.ru/> — Текст: электронный.

Задача адресной перевозки с временными окнами

Кулагина Софья Александровна, студент;
Скородумова Юлия Владимировна, студент;
Иванова Анастасия Анатольевна, студент;
Санкт-Петербургский государственный университет

Задача адресной перевозки состоит в составлении маршрутов и расписании транспортных средств для перевозки людей «от двери до двери». С такой задачей сталкиваются службы такси, школьные автобусы и социальные службы специального транспортного обслуживания. В данной работе рассматривается статическая задача адресной перевозки с временными окнами и одним транспортным средством. Для решения поставленной задачи использовались различные вариации алгоритма муравьиной колонии. Проведено сравнение работы алгоритмов на тестовых данных.

Ключевые слова: задача адресной перевозки, муравьиный алгоритм.

Введение. Темой данной работы является задача адресной перевозки (Dail-a-Ride Problem или DARP) с временными окнами. DARP относится к классу задач вывоза и доставки, однако в отличие от остальных задач данного класса, она связана с перевозкой людей, а не товаров. С задачей адресной перевозки сталкиваются службы такси, школьные автобусы и социальные службы специального транспортного обслуживания, занимающиеся перевозкой пожилых и маломобильных людей.

Постановка задачи. В данной работе рассматривается задача с одним транспортным средством, которое начинает и заканчивает свой маршрут в депо. Транспортное средство не может выполнять несколько заказов одновременно, то есть забирать людей «по дороге». Перевозка осуществляется по заранее известным запросам, включающим следующие данные: 1) координаты отправления и прибытия, 2) временные окна на вывоз или доставку (то есть максимальное и минимальное время начала и окончания поездки), 3) время на обслуживание клиента до и после поездки (если клиенту требуется дополнительное время, чтобы сесть в машину или загрузить багаж).

Математическая постановка. Задачу адресной перевозки можно задать на графе $G=(N, A)$, где N — множество вершин: $N = P \cup D \cup \{0\}$, $P=\{1, \dots, n\}$ и $D=\{n+1, \dots, 2n\}$ — вершины отправления и прибытия соответственно, $\{0\}$ — вершина, обозначающая депо, в котором изначально находятся все транспортные средства, а A — множество ребер. Каждый маршрут клиента i начинается в вершине $i \in P$ и заканчивается в вершине $i + n \in D$. Кроме того, для каждой вершины $i \in N$ известно время c_i на обслуживание клиента перед или после поездки и временное окно $[e_i, l_i]$. Для каждой дуги $(i, j) \in A$ известно время t_{ij} пути по ней.

$$\min \sum_{i \in N} \sum_{j \in N} t_{ij} x_{ij}, \tag{1}$$

$$\sum_{i \in P} x_{0i} = 1, \tag{2}$$

$$\sum_{j \in D} x_{j0} = 1, \tag{3}$$

$$\sum_{j \in N} x_{ji} - \sum_{j \in N} x_{ij} = 0, \quad \forall i \in N, \tag{4}$$

$$\sum_{j \in N} x_{ij} = 1, \quad \forall i \in P, \tag{5}$$

$$\sum_{j \in N} x_{ij} - \sum_{j \in N} x_{n+i, j} = 0, \quad \forall i \in P, \tag{6}$$

$$B_j \geq (B_i + c_i + t_{ij})x_{ij}, \quad \forall i, j \in N, \tag{7}$$

$$e_i \leq B_i \leq l_i, \quad \forall i \in N, \tag{8}$$

$$x_{ij} \in \{0,1\}, \quad \forall i, j \in N. \tag{9}$$

Целевая функция (1) минимизирует общую стоимость маршрута, которая в данном случае является временем. Ограничения (2), (3) и (4) гарантируют, что все маршруты транспортного средства начинаются и заканчиваются в депо. Условия (5) и (6) устанавливают, что каждый заказ будет выполнен один раз, и транспортное средство переместится из вершины отправления в соответствующую ей вершину доставки. Ограничения на время прибытия в следующую вершину маршрута B_j задано в неравенстве (7). Условие (8) ограничивает время прибытия минимальным и максимальным временем начала обслуживания вершины. Наконец, в условии (9) полагаем, что $x_{ij}=1$, если транспортное средство переместилось из вершины i в вершину j , и $x_{ij}=0$ в ином случае.

Алгоритм решения. Для решения задачи адресной перевозки, поставленной в данной работе, было решено использовать метаэвристический алгоритм муравьиной колонии. В общем случае алгоритм муравьиной колонии для решения задачи адресной перевозки выглядит следующим образом.

1. Инициализация параметров.
2. Расстановка муравьев по вершинам в случайном порядке.
3. **for** $t = 1$ **to** t_{max} **do**
4. **for** $k = 1$ **to** m **do**
5. **for** $i = 1$ **to** $n - 1$ **do**
6. применение стратегии выбора пути;
7. обновление феромонов;
8. поиск лучшего маршрута;
9. Вывод глобально лучшего маршрута.

Здесь t_{max} — заранее заданное число повторов основной части алгоритма (или жизненный цикл муравьиной колонии), m — количество муравьев в колонии, n — количество вершин в графе.

Есть несколько стратегий выбора пути — Ant System (AS) [1], Ant Colony System (ACS) [2] и Modified Ant Colony System (M — ACS). Применяя данные стратегии, муравьи опираются на следующую информацию — количество феромона ij на ребре (i, j) , видимость ij вершины j и список посещенных вершин $tabu_list$.

$$\text{AS: } P_{ij} = \frac{[\tau_{ij}^\alpha][\eta_{ij}^\beta]}{\sum_{j=1}^n [\tau_{ij}^\alpha][\eta_{ij}^\beta]}$$

$$\text{ACS: } j = \begin{cases} \arg \max \{[\tau_{ij}^\alpha][\eta_{ij}^\beta]\}, j \notin tabu_list, q \leq q_0, \\ J, q \geq q_0, \end{cases}$$

$$\text{M-ACS: } j = \begin{cases} \arg \max \{[\tau_{ij}^\alpha][\eta_{ij}^\beta]\}, j \notin tabu_list, q \leq q_0, \\ random\ j \notin tabu_list, q \geq q_0. \end{cases}$$

Также существуют различные способы вычисления видимости вершины. В данной работе рассмотрены несколько подходов, учитывающих временные окна. Для удобства обозначим их как ANT_TIME 1 [3], ANT_TIME 2 [4] и ANT_TIME 3 [5].

ANT-TIME 1

$$\eta_{ij} = \frac{1}{\max\{1, t_j - e_j\} d_{ij}}$$

ANT-TIME 2

$$\eta_{ij} = \frac{1}{g_{ij}} \cdot \frac{1}{t_{ij}}$$

$$g_{ij} = \max\{1, \Delta t_{ij}(l_j - t_j)\}, \Delta t_{ij} = \max\{t_i + t_{ij}, e_j\} - t_i$$

ANT-TIME 3

$$\eta_{ij} = f_{ij}^\beta \cdot w_{ij}^\gamma$$

$$f_{ij}^\beta = \begin{cases} \frac{1}{1 + e^{\sigma(F_{ij}-\mu)}}, & F_{ij} \geq 0, \\ 0, & F_{ij} < 0. \end{cases} \quad F_{ij} = b_j - t_j, \quad w_{ij}^\beta = \begin{cases} \frac{1}{1 + e^{\lambda(W_{ij}-\nu)}}, & W_{ij} > 0, \\ 1, & W_{ij} \leq 0. \end{cases} \quad W_{ij} = a_j - t_j,$$

Для решения поставленной задачи использовались алгоритмы ANT_TIME 1, ANT_TIME 2 и ANT_TIME 3 в связке с каждой из стратегий выбора пути AS, ACS и M-ACS. Стоит отметить, что подходы ANT_TIME 2 и ANT_TIME 3 впервые использовались для решения задачи адресной перевозки.

Алгоритмы были реализованы на языке Python. Данные о вершинах и разметка вершин на точки отправления и доставки брались из тестовых примеров, размещенных в сети Интернет [6]. Временные окна генерировались случайным образом.

Алгоритмы были реализованы с использованием следующих значений параметров: $n = 41$ - количество вершин графа, $m = 20$ — количество муравьев в колонии, $\alpha = 1, \beta = 1, \gamma = 1$ — параметры, отвечающие за влияние концентрации феромона и видимости на вероятность перехода, $Q = 1$ — количество феромонов, $\omega = 0.5$ — параметр локального испарения феромона, $\rho = 0.6$ - параметр глобального испарения, $\sigma = 0.01, \lambda = 0.01$ — параметры для вычисления эвристик, $q_0 = 0.7$ — параметр для выбора правила для перехода в следующую вершину в стратегиях ACS и M-ACS, $\nu = 60$ — скорость, $t_{max} = 20$ — жизненный цикл муравьиной колонии.

Для получения средних значений каждый алгоритм запускался 100 раз. В таблице 1 представлены результаты работы алгоритмов для одной из тестовых выборок.

Таблица 1. Результаты работы алгоритмов на графе с 41 вершиной

Алгоритм	Стратегия	Лучшее время маршрута	Среднее время маршрута	Среднее время вычислений	Среднее кол-во заказов	Кол-во успешных туров
ANT — TIME 1	AS	27.36762871	30.00670987	0.05981349	11.64	0
	ACS	24.33553884	27.66034498	0.08169099	13.6	0
	M-ACS	24.66516612	27.85776373	0.05200895	12.79	0
ANT — TIME 2	AS	24.58951634	25.70720050	0.32875761	15.8556	1
	ACS	23.86681285	24.98715482	0.35313985	15.15	3
	M-ACS	25.77217115	25.35720826	0.20443104	14.55	1
ANT — TIME 3	AS	26.23591265	26.89348178	0.43719965	10.54	0
	ACS	24.65265292	27.00833381	0.48515794	14.18	5
	M-ACS	23.92085032	27.54231113	0.44881124	17.57	39

Ни в одном из запусков алгоритма ANT-TIME 1 не было найдено маршрутов, построенных с соблюдением всех временных окон, поэтому лучшее время приводится для максимального количества обслуженных заказов. Данный алгоритм показал худшие результаты относительно количества успешных туров, однако время работы алгоритма ANT-TIME 1 значительно меньше времени работы остальных алгоритмов. Это связано с малым количеством операций для вычисления эвристик.

С помощью алгоритмов ANT-TIME 2 и ANT-TIME 3 удалось построить маршруты, удовлетворяющие всем ограничениям. Лучшее время маршрута, построенного алгоритмом ANT-TIME 2, было найдено с использованием стратегии ACS. То же можно сказать и о количестве успешных туров.

В алгоритме ANT-TIME 3 использовались эвристики, направленные на строгое соблюдение временных окон, этим объясняется достаточно большое количество успешных туров. При этом использование стратегии M-ACS позволило достичь лучших результатов как по времени лучшего маршрута, так и по количеству успешных маршрутов. В то же время ANT-TIME 3 требует наибольшего времени вычисления.

Таким образом, наилучшие результаты по значению целевой функции и времени работы показал алгоритм ANT-TIME 2 с использованием стратегии ACS, а количество успешных туров — алгоритм ANT-TIME 3 в связке со стратегиями M-ACS.

Алгоритмы ANT-TIME 2 ACS и ANT-TIME 3 M-ACS, показавшие наилучшие результаты, были дополнительно рассмотрены на графе со 101 вершиной (50 запросов на перевозку). Результаты представлены в таблице 2.

Таблица 2. Результаты работы алгоритмов на графе со 101 вершиной

Имя файла данных	Алгоритм	Лучшее время маршрута	Макс. кол-во заказов	Среднее время маршрута	Среднее кол-во заказов	Среднее время вычислений
1P1	ANT-TIME 2	52.29623731	49	52.29623731	33.43	4.81587873
	ANT-TIME 3	61.00058275	50	67.09935076	14.18	11.08235990
2P1	ANT-TIME 2	49.90624517	49	52.12220228	35.02	5.99185033
	ANT-TIME 3	64.44667793	50	67.22705415	32.74	10.53928622
3P1	ANT-TIME 2	52.03960973	49	52.61987721	34.46	7.99570506
	ANT-TIME 3	58.85601463	50	67.25229198	31.1	10.55909965
4P1	ANT-TIME 2	50.46140168	50	51.57021183	38.78	4.52485980
	ANT-TIME 3	59.75183647	50	63.02709233	28.88	7.97703070
5P1	ANT-TIME 2	52.82012369	50	51.76323587	38.79	4.57656121
	ANT-TIME 3	58.03609214	50	62.79899191	28.49	7.01111531

Для трех тестовых наборов ANT-TIME 2 ACS не смог построить маршруты с соблюдением всех ограничений, в то время как ANT-TIME 3

M-ACS построил 1-3 успешных тура для каждого из пяти наборов данных. Однако, для наборов 4P1 и 5P1, в которых удалось обслужить всех клиентов, ANT-TIME 2 ACS показал лучшее время маршрута. Кроме того, ANT-TIME 2 ACS позволил включить в маршрут в среднем большее количество клиентов. Стоит также отметить, что время работы алгоритма ANT-TIME 2 ACS на используемых данных в среднем в 1.7 раз меньше времени работы ANT-TIME 3 M-ACS.

Заключение. В данной работе была рассмотрена статическая задача адресной перевозки с временными окнами и одним транспортным средством. В качестве основного метода решения был выбран алгоритм муравьиной колонии. По результатам исследования выявлены алгоритмы, строящие маршрут с наибольшим количеством обслуженных заказов — ANT-TIME 2 и ANT-TIME 3. Кроме того, для каждого из алгоритмов найдена стратегия перехода, с помощью которой можно получить наилучшее значение целевой функции. Сравнение этих алгоритмов на тестовых данных большей размерности позволило определить метод, позволяющий получить лучшее время маршрута. Таким алгоритмом является ANT-TIME 2 со стратегией ACS.

Литература:

1. Colorni, A., Dorigo M., Maniezzo V. Distributed optimization by ant colonies // European Conference on Artificial Life, 1991.
2. Dorigo, M. Gambardella L. M. Ant colony system: a cooperative learning approach to the traveling salesman problem // IEEE Transactions on Evolutionary Computation, 1997. Vol. 1, No 1, P. 53–66.
3. Tripathy, T., Nagavarapu S. C., Azizian K., Pandi R. R., Dauwels J. Solving dial-a-ride problems using multiple ant colony system with fleet size minimisation // Advances in Computational Intelligence Systems, 2017. P. 325–336.
4. Baran, B. and Schaerer M. A multiobjective ant colony system for vehicle routing problem with time windows // The 21st IASTED International Multi-Conference on Applied Informatics, 2003. P. 97–102.
5. Cheng, C. B., Mao C. P. A modified ant colony system for solving the traveling salesman problem with time windows // Mathematical and Computer Modeling, 2007. Vol. 46. P. 1225–1235.
6. The VRP Web [Электронный ресурс]. URL: <http://www.bernabe.dorronsoro.es/vrp/> (дата обращения: 10.01.2020).

Определение предпочтительного числа кластеров. Момент остановки метода одиночной связи

Мелков Никита Александрович, студент;
Ржавитина Мария Андреевна, студент;
Пак Алексей Станиславович, студент;
Спиридонов Александр Юрьевич, студент
Санкт-Петербургский государственный университет

Кластерный анализ является одним из основных методов предварительной классификации большого количества информации. Актуальной задачей остаётся определение момента остановки процесса кластеризации. Можно рассмотреть кластерный анализ данных методом «одиночной связи» в виде дискретного случайного процесса и определить момент остановки с помощью квадратичных форм погрешности приближения «минимальных расстояний» функцией. В статье даётся определение «аппроксимационно-оценочным критериям». Применён один из критериев для кластеризации множества из 33 точек.

Ключевые слова: кластер, метод одиночной связи, аппроксимация, квадратичная форма, кластерный анализ.

Метод одиночной связи

Метод одиночной связи — иерархический агломеративный алгоритм, который выполняется в двумерном евклидовом пространстве со стандартной метрикой. За кластер мы берём центроиду всех точек, входящих в кластер, то есть изначально каждая точка является отдельным кластером, затем при увеличении кластера — средним арифметическим координат всех точек, входящих в кластер.

Итерации алгоритма можно описать следующим образом: на первом шаге необходимо определить расстояния между всеми n кластерами, построить множество минимальных расстояний, затем найти минимальное. Два кластера, расстояние между которыми минимально, объединяются, их центроида перевычисляется, получаем $n-1$ кластер [2]. Без остановки алгоритм в конечном итоге объединит все точки в один кластер, что является неудовлетворительным результатом.

Остановка процесса кластеризации

Множество минимальных расстояний является монотонно неубывающим, и при объединении больших кластеров испытывает «скачок». Например, для множества из 33 точек [2]

$X = \{(0; 0); (2; 4); (3; 3); (1; 2); (3; 0); (3; 1); (1; 1); (12; 18); (13; 17); (11; 15); (13; 14); (14; 16); (11; 16); (12; 15); (13; 18); (12; 5); (13; 2); (14; 4); (12; 3); (13; 1); (14; 2); (24; 19); (22; 22); (21; 24); (23; 21); (24; 20); (22; 39); (23; 38); (24; 39); (21; 37); (2; 26); (24; 6); (10; 36)\}$ график множества минимальных расстояний указан на рис. 1.

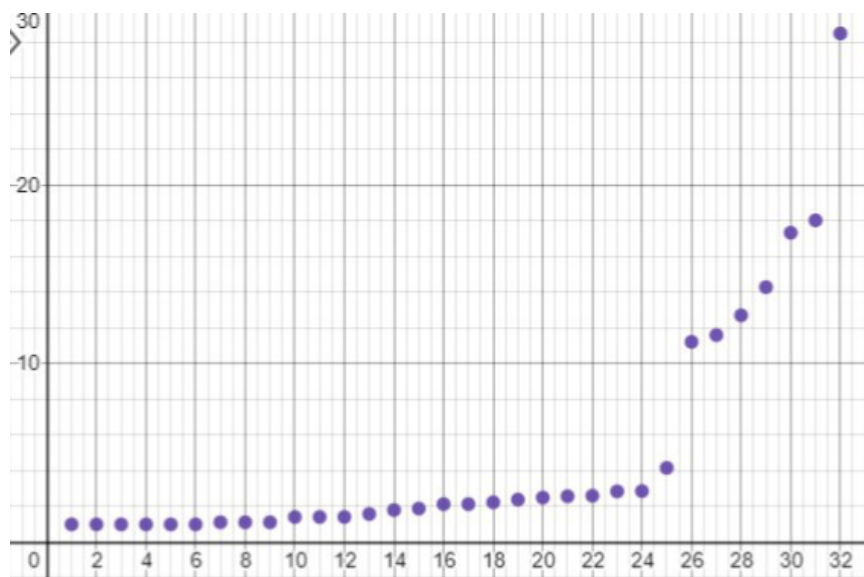


Рис. 1. Множество минимальных расстояний. На оси абсцисс указаны номера итераций, на оси ординат — расстояние между кластерами

Для того чтобы избежать проблемы объединения точек в один большой кластер, необходимо ввести критерий остановки. Будем приближать множество минимальных расстояний функцией для того, чтобы зафиксировать скачок (объединение кластеров). Хорошие результаты даёт приближение не всех точек сразу, а последовательно по 3 или 4 точкам. «Аппроксимационно-оценочные критерии» [1] являются квадратичными формами, оценивающими разность между квадратичной погрешностью линейной аппроксимации и квадратичной погрешностью одного из видов нелинейной аппроксимации (например, биквадратной — четвёртой степени). С вычислением критериев можно ознакомиться в работе [1]. Биквадратный аппроксимационно-оценочный критерий по четырём точкам имеет вид:

$$\delta = \frac{1657y_1^2 + 4206y_2y_1 - 6652y_3y_1 - 11428y_2y_3 - 743y_2^2 + 6023y_3^2}{22085}$$

Здесь три переменные, но точек четыре. Это связано с введением упрощения: y_0 — первая точка, по которой проводится аппроксимация, y_1 — вторая, y_2 — третья, y_3 — четвёртая; мы обнуляем первую точку и от значений остальных отнимаем y_0 — по сути, мы переносим начало координат в y_0 . Подставляем вычисленные минимальные расстояния и считаем квадратичную форму — получаем число. Если это число больше нуля, то процесс кластеризации следует остановить, если меньше нуля — то следует продолжить. Данная квадратичная форма — разность погрешности аппроксимации линейной и биквадратной функциями по четырём точкам, и если линейная точнее, то вычисленное значение будет отрицательным. Это означает отсутствие скачка. Если приближение биквадратной функцией точнее, то функция испытывает скачок, и значение становится положительным. [1]

Результаты использования

Чувствительность критерия можно задать с помощью коэффициента q при линейном преобразовании $y_i = F_i + qi$, где i — номер итерации, F_i — значение минимального расстояния на i итерации, q — чувствительность критерия.

Для рассмотренного множества X данный способ показывает хорошие результаты — при q от 0.18 до 0.63 процесс кластеризации методом одиночной связи заканчивается после 25 итерации (рис. 2)

Выводы

Необходимо отметить, что рассмотренный критерий не единственный — можно вычислить экспоненциальный, квадратичный, кубический критерии. Они достаточно просты в использовании (но иногда сложны в вычислении), показывают хорошие и устойчивые результаты.

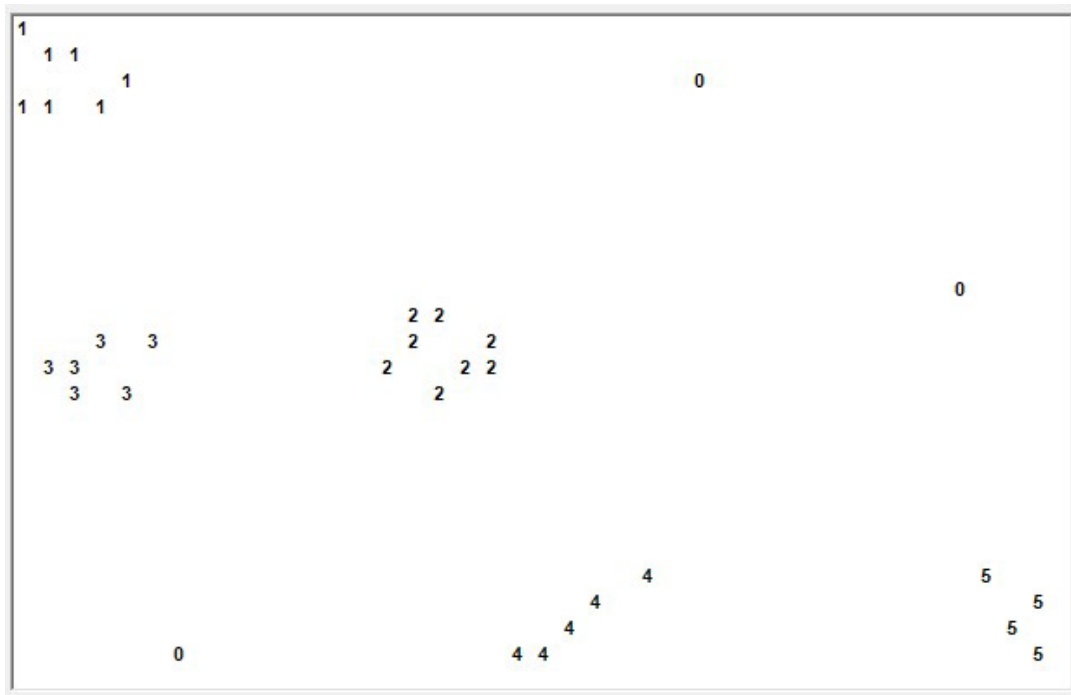


Рис. 2. Множество X , разбитое на кластеры, после 25 итерации. Начало координат находится в левом верхнем углу

Литература:

1. Орехов, А. В. Аппроксимационно-оценочные критерии напряженно-деформируемого состояния твердого тела // Вестник Санкт-Петербургского университета. Прикладная математика. Информатика. Процессы управления. 2018. Т. 14. Вып. 3. с. 230-242. <https://doi.org/10.21638/11702/spbu10.2018.304>
2. Орехов, А. В. Марковский момент остановки агломеративного процесса кластеризации в евклидовом пространстве // Вестник Санкт-Петербургского университета. Прикладная математика. Информатика. Процессы управления. 2019. Т. 15. Вып. 1. с. 00-00.

Анализ влияния вычислительной погрешности в явных методах Рунге — Кутты

Пак Алексей Станиславович, студент;
 Кашеева Ольга Александровна, студент;
 Мелков Никита Александрович, студент;
 Спиридонов Александр Юрьевич, студент;
 Ржавитина Мария Андреевна, студент;
 Кулагина Софья Александровна, студент
 Санкт-Петербургский государственный университет

Статья посвящена нахождению приемов и способов улучшения и оптимизации известных методов интегрирования систем обыкновенных дифференциальных уравнений (СОДУ). Задача уменьшения вычислительной погрешности при меньших затратах является наиболее актуальной для всех численных методов. В статье подробно рассматривается применение метода компенсированного суммирования для явного метода Рунге — Кутты четвертого порядка.

Ключевые слова: система обыкновенных дифференциальных уравнений, компенсированное суммирование, алгоритм Гилла — Мёллера, методы Рунге — Кутты.

Двадцать первый век можно охарактеризовать как период, в течение которого классические методы решения обыкновенных дифференциальных уравнений (ОДУ), таких как методы Рунге — Кутты, разработанные для ручного счета, развивались исходя из требований и цифровых возможностей. Однако в практических вычислениях природа компьютерной арифметики может играть значительную и, возможно, подавляющую роль.

Алгоритм Гилла — Мёллера

В книге Бутчера [1] представлен анализ влияния ошибки округления в явном методе Эйлера, а также приведен алгоритм Гилла — Мёллера (Гилл [2], Мёллер [4], [5]), который называют «компенсированным суммированием». В данном разделе будет проведен аналогичный численный эксперимент для преодоления последствий накопления ошибок округления в явном методе Рунге — Кутты 4-го порядка.

Пусть a_n — общая погрешность, вычисленная на шаге n , b_n — соответствующая погрешность вычисления производной на этом же шаге, тогда

$$\begin{aligned} a_n &= y(x_n) - y_n, \\ b_n &= f(x_n, y(x_n)) - f(x_n, y_n), \end{aligned}$$

последовательности точных и приближённых значений связаны между собой следующим образом

$$\begin{aligned} y_n &= y_{n-1} + hf(x_{n-1}, y_{n-1}) - r_n, \\ y(x_n) &= y(x_{n-1}) + hf(x_{n-1}, y(x_{n-1})) + l_n, \end{aligned}$$

где r_n — ошибка округления, совершенная на этом шаге, а l_n — методическая погрешность. Данная операция приводит к следующему разностному уравнению

$$a_n = a_{n-1} + hb_{n-1} + l_n + r_n,$$

Так как, детальный анализ ошибки округления в расчетах данной задачи представлен в работе Хенричи [3], то вместо того, чтобы пытаться провести анализ r_n и l_n , а также, поскольку шаги вычисляются последовательно, заметим, что можно преодолеть наихудшие последствия накопления ошибок округления. Это делается путем оценки значения r_n на любом конкретном шаге, а затем добавление его к значению $hf(x_n, y_n)$, прежде чем оно будет добавлено на следующем шаге.

Этот усовершенствованный метод, который может быть использован для многих ситуаций, связанных с суммированием большого количества малых величин, и называют алгоритмом Гилла — Мёллера или «компенсированным суммированием».

«Компенсированное суммирование» в явном методе Рунге — Кутты 4-го порядка

Общая схема явного метода Рунге — Кутты (ЯМРК) численного интегрирования СОДУ

$$\begin{aligned} \frac{dy_s}{dx} &= f_s(x, y_0, \dots, y_n), s = 0, 1, \dots, n, \\ y_s(X) &= y_{s0}, s = 0, 1, \dots, n, \\ x &\in [X_0, X_k] \subset R, y_s: [X_0, X_k] \rightarrow R, \\ f_s: [X_0, X_k] \times R^{n+1} &\rightarrow R, s = 0, 1, \dots, n, \end{aligned}$$

выглядит следующим образом:

$$y_s(x+h) \approx z_s = y_s(x) + \sum_{g=1}^{w-1} b_{sw} k_{sw}(h), s = 0, 1, \dots, n,$$

где функции $k_{sw} = k_{sw}(h)$ вычисляются по схеме

$$\begin{aligned} k_{sw} &= hf_s(x + c_{sw}h, y_0(x) + \sum_{g=1}^{w-1} a_{sw0g} k_{0g}, \dots, y_n(x) + \sum_{g=1}^{w-1} a_{swng} k_{ng}) \\ c_{s1} &= 0, a_{swt1} = 0, t = 0, 1, \dots, n. \end{aligned}$$

Здесь $y_s(x+h), z_s$ — точное и приближенное значение s -й компоненты в точке $x+h \in [X_0, X_k]$ соответственно, $y_s(x)$ — точное значение s -й компоненты в точке $x \in [X_0, X_k]$, h — шаг метода.

Традиционным считается символически представлять данный метод посредством таблиц Бутчера. Приведем пример таблицы Бутчера для ЯМРК 4-го порядка.

0				
$c_2 = \frac{1}{2}$	$a_{w1} = \frac{1}{2}$			
$c_3 = \frac{1}{2}$	$a_{w1} = 0$	$a_{w2} = \frac{1}{2}$		
$c_4 = 1$	$a_{w1} = 0$	$a_{w2} = 0$	$a_{w3} = 1$	
	$b_1 = \frac{1}{6}$	$b_1 = \frac{2}{6}$	$b_1 = \frac{2}{6}$	$b_1 = \frac{1}{6}$

Далее будет показано как применить «компенсированное суммирование» для уменьшения ошибки округления в ЯМРК четвертого порядка на примере системы дифференциальных уравнений

$$\begin{aligned}y_1' &= 2xy_2^{1/5}y_4 \\y_2' &= 10x\exp(5(y_3 - 1))y_4 \\y_3' &= 2xy_4\end{aligned}$$

$$y_4' = -2x\ln(y_1),$$

с начальным приближением

$$y_i(0) = 1, i = 1,2,3,4,$$

которое имеет аналитическое решение

$$\begin{aligned}y_1 &= \exp(\sin(x^2)) \\y_2 &= \exp(5\sin(x^2)) \\y_3 &= \sin(x^2) + 1 \\y_4 &= \cos(x^2).\end{aligned}$$

Реализация данного метода была выполнена в системе научных вычислений MATLAB. (ЯМРК4)

```

for i = 1:n
    x0 = xStart + i*h;
    X = x0;
    Y = y0;
    for j=1:N
        K1 = f{i}(X,Y);
    end
    X = x0 + c2*h;
    Y = y0 + h*a21*K1;
    for j=1:N
        K2 = f{i}(X,Y);
    end
    X = x0 + c3*h;
    Y = y0 + h*(a31*K1 + a32*K2);
    for j=1:N
        K3 = f{i}(X,Y);
    end
    X = x0 + c4*h;
    Y = y0 + h*(a41*K1 + a42*K2 + a43*K3);
    for j=1:n
        K4 = f{i}(X,Y);
    end

    y1 = y0 + h*(b1*K1 + b2*K2 + b3*K3 + b4*K4);

    y0=y1; % for next step
end

```

ЯМРК4 с компенсированным суммированием)

```

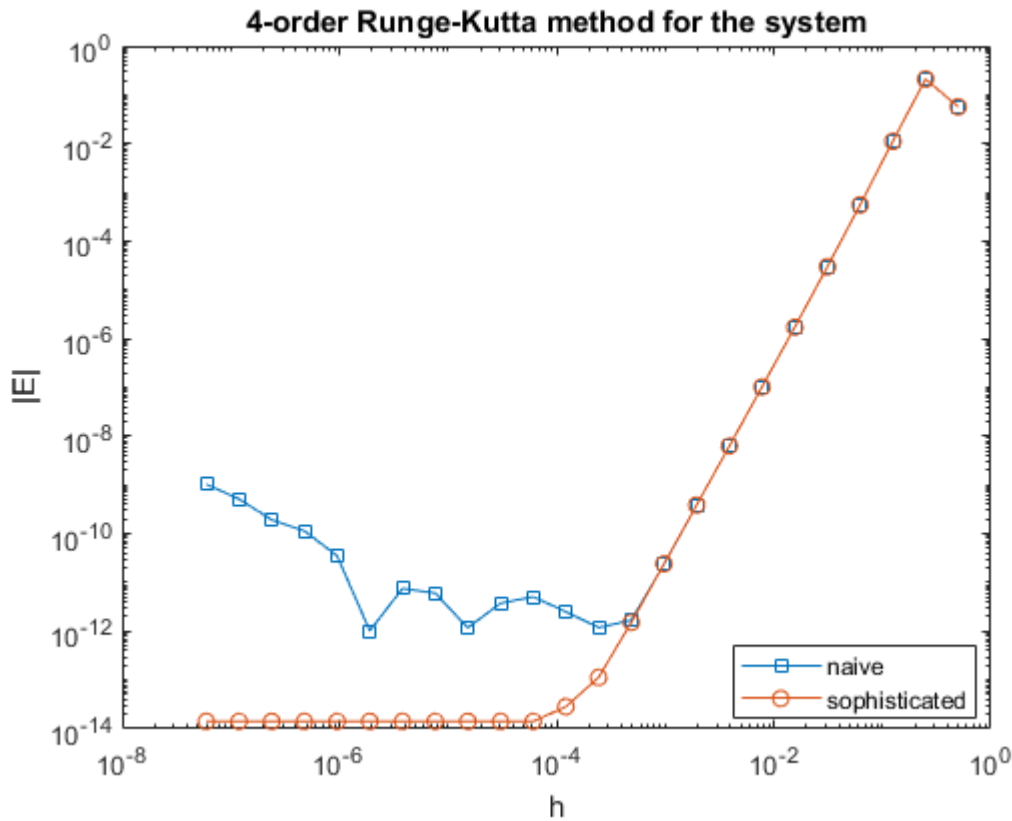
z = zeros(N,1);
term = zeros(N,1);
newy = zeros(N,1);
for i = 1:n
    x0 = xStart + i*h;
    X = x0;
    Y = y0;
    for j=1:N
        K1 = f{i}(X,Y);
    end
    X = x0 + c2*h;
    Y = y0 + h*a21*K1;
    for j=1:N
        K2 = f{i}(X,Y);
    end
    X = x0 + c3*h;
    Y = y0 + h*(a31*K1 + a32*K2);
    for j=1:N
        K3 = f{i}(X,Y);
    end
    X = x0 + c4*h;
    Y = y0 + h*(a41*K1 + a42*K2 + a43*K3);
    for j=1:n
        K4 = f{i}(X,Y);
    end
    term = h*(b1*K1 + b2*K2 + b3*K3 + b4*K4) + z;
    newy = y0 + term;
    z = term - (newy - y0)
    y1 = newy;
    y0=y1; % for next step
end

```

В данном примере была взята последовательность шагов $h = 2^{-2}, h = 2^{-3}, \dots, h = 2^{-24}$.

Результаты

Нанесем глобальные погрешности этих двух алгоритмов на один график в логарифмическом масштабе.



Видно, что явная форма дает результаты, которые сильнее ухудшаются при определенном шаге, в то время как компенсированная версия значительно улучшает результат.

Выводы

Как видно из проведенного исследования, метод «компенсированного суммирования» значительно улучшает ЯМРК четвертого порядка, а также его можно распространить и на другие численные методы интегрирования СОДУ.

Литература:

1. Butcher, J. C. «Numerical Methods for ordinary Differential Equations, Second Edition». The University of Auckland, New Zealand 2008, 463 p.
2. Gill, S. «A process for the step-by step integration of differential equations in an automatic computing machine». Proc. Cambridge Philos. Soc 1951.
3. Henrici, P. «Discrete Variable Methods in Ordinary Differential Equations». John Wiley & Sons Inc, New York 1962.
4. Møller, O. «Quasi double-precision in floating point addition». BIT 1965.
5. Møller, O. «Note on quasi double-precision». BIT 1965.

Анализ больших деформаций решетчатой пластины глаза

Скородумова Юлия Владимировна, студент;
 Иванова Анастасия Анатольевна, студент;
 Кулагина Софья Александровна, студент;
 Тарасов Антон Алексеевич, студент;
 Павлов Дмитрий Львович, студент
 Санкт-Петербургский государственный университет

В работе рассмотрена устойчивость осесимметричных форм равновесия однородных изотропных пластин, моделирующих решетчатую пластинку диска зрительного нерва человека. В предположении, что несимметричная составляющая решения носит периодический характер, численным методом определено наименьшее значение нагрузки, при которой появляются волны в окружном направлении. Исследовано влияние различных условий закрепления пластины, наличия и размеров отверстия в центре пластины на величину критической нагрузки и форму потери устойчивости.

Ключевые слова: биомеханика глаза, круглая пластина, потеря устойчивости.

Введение. В данной работе была рассмотрена задача о потере устойчивости симметричной формы равновесия однородной круглой пластины. Такая пластина может быть простейшей моделью решетчатой пластины диска зрительного нерва человека [6]. Были поставлены две задачи: исследовать влияние условий закрепления края на величину критической нагрузки; рассмотреть задачу о возникновении неосесимметричных состояний у кольцевой пластины.

Впервые вопрос о существовании несимметричных решений у симметрично нагруженной круглой пластины был рассмотрен в [1]. Исследуя большие прогибы пластины, нагруженной постоянным давлением, авторы получают решение, соответствующее несимметричным формам равновесия.

Строгое доказательство существования несимметричного решения для симметрично нагруженной пластины было проведено в [2], а его единственность доказана в работе [3].

В работах [4], [5] для пологой сферической оболочки и круглой пластины при различных условиях закрепления и нагружения определены значения критической нагрузки, при которой происходит переход от симметричной формы равновесия к неосесимметричной.

Постановка задачи. Рассмотрим круглую изотропную пластину, защемленную по краю и находящуюся под действием равномерно распределенного давления. Запишем систему уравнений в виде [7]:

$$D\Delta\Delta w = p + L(w, F), \quad \frac{\Delta\Delta F}{E} = -\frac{h}{2}L(w, w),$$

$$\left(\cdot \right)' = \frac{\partial(\cdot)}{\partial r}, \quad \left(\cdot \right) \dot{\cdot} = \frac{\partial(\cdot)}{\partial \theta} \tag{1}$$

где $w(r, \theta), F(r, \theta)$ — неизвестные функция нормального прогиба и функция усилий; $D = Eh^3/12(1-\nu^2)$ — цилиндрическая жесткость пластины, E — модуль упругости пластины, R — радиус пластины, h — толщина пластины, ν — коэффициент Пуассона материала пластины, p — нормальное давление, действующее на пластину; r и θ — полярные координаты срединной поверхности пластины: $0 \leq r \leq R, 0 \leq \theta \leq 2\pi$; оператор Лапласа Δ и дифференциальный оператор L имеют вид:

$$\Delta = \left(\cdot \right)'' + \frac{\left(\cdot \right)'}{r} + \frac{\left(\cdot \right)}{r^2}, \quad L(y, z) = y'' \left(\frac{z'}{r} + \frac{\ddot{z}}{r^2} \right) + z'' \left(\frac{y'}{r} + \frac{\ddot{y}}{r^2} \right) - 2 \left(\frac{\dot{y}}{r} - \frac{\dot{y}}{r^2} \right) \left(\frac{\dot{z}}{r} - \frac{\dot{z}}{r^2} \right).$$

Функция $F(r, \theta)$ связана с возникающими при деформации усилиями в пластине, по следующим формулам [7]:

$$T_r = \frac{1}{r}F' + \frac{1}{r^2}\ddot{F}, \quad T_\theta = F'', \quad S = -\left(\frac{1}{r}\dot{F} \right) = 0.$$

Предположим, что точки края $r = R$ закреплены от поворотов, но свободно смещаются в радиальном и окружном направлениях. В этом случае растягивающее (T_r) и сдвигающее (S) усилия на контуре полагаем равными 0:

$$w = w' = 0, \quad T_r = \frac{1}{r}F' + \frac{1}{r^2}\ddot{F} = 0, \quad S = -\left(\frac{1}{r}\dot{F} \right) = 0. \tag{2}$$

Введем безразмерные переменные

$$r^* = \frac{r}{R}, \quad w^* = \beta \frac{w}{h}, \quad p^* = \beta^3 \frac{pR^4}{Eh^4}, \quad F^* = \beta^2 \frac{F}{Eh^3}, \quad \beta^2 = 12(1-\nu^2).$$

Тогда система уравнений (1) примет следующий вид (знак * в дальнейшем опускаем):

$$\begin{aligned} \Delta\Delta w &= p + L(w, F), \\ \Delta\Delta F &= -\frac{L(w, w)}{2}. \end{aligned} \tag{3}$$

Граничные условия сохраняют вид (2).

Для сплошной пластины граничные условия (2) должны быть дополнены условиями ограниченности решений в центре $w'(0) = F'(0) = 0$.

Метод поиска несимметричного решения. Нам нужно найти критическое значение нагрузки $p = p_{cr}$, при котором возможна бифуркация пластины в неосесимметричное состояние. При значениях нагрузки p меньше критического значения система (3), дополненная необходимыми граничными условиями, имеет только симметричное решение. Несимметричное решение этой системы появляется при возрастании нагрузки.

Будем искать решение в виде

$$w(r, \theta) = w_s(r) + w_n(r) \cos(n\theta), \quad F(r, \theta) = F_s(r) + F_n(r) \cos(n\theta), \tag{4}$$

где функции w_s, F_s описывают докритическое симметричное решение, а функции $w_n(r, \theta) = w_n(r) \cos(n\theta), F_n(r, \theta) = F_n(r) \cos(n\theta)$ — закритическое состояние пластины (здесь n — число волн в окружном направлении, образовавшихся после потери устойчивости).

После разделения переменных (4) исходная система (1)–(2) распадается на нелинейную для нахождения симметричного решения $w_s(r), F_s(r)$ и линейную относительно $w_n(r), F_n(r)$, так как функции w_n, F_n полагаются малыми сразу после перехода пластины в неосесимметричное состояние.

Подставляя (4) в (3), с учетом обозначений $\omega_0 = w'_s, \phi_0 = F'_s$ получаем систему уравнений для симметричного решения задачи:

$$\begin{aligned} \omega_0'' + \frac{\omega_0'}{r} - \frac{\omega_0}{r^2} &= \frac{pr}{2} + \frac{\omega_0\phi_0}{r}, \\ \phi_0'' + \frac{\phi_0'}{r} - \frac{\phi_0}{r^2} &= \frac{pr}{2} + \frac{\omega_0\phi_0}{r}. \end{aligned} \tag{5}$$

Граничные условия (2) с учетом ограниченности решений в центре пластины примут вид

$$\omega_0(0) = \omega_0(1) = \phi_0(0) = \phi_0(1) = 0. \tag{6}$$

Для несимметричного решения после разделения переменных получим линейную систему уравнений относительно $w_n(r)$ и $F_n(r)$

$$\begin{aligned} \Delta_n \Delta_n w_n &= \omega_0' \left(\frac{F_n'}{r} - \frac{n^2}{r^2} F_n \right) + \phi_0' \left(\frac{w_n'}{r} - \frac{n^2}{r^2} w_n \right) + \frac{w_n''}{r} \phi_0 + \frac{F_n''}{r} \omega_0, \\ \Delta_n \Delta_n F_n &= -\omega_0' \left(\frac{w_n'}{r} - \frac{n^2}{r^2} w_n \right) - \frac{w_n''}{r} \omega_0, \end{aligned} \tag{7}$$

где $\Delta_n y = y'' + \frac{y'}{r} - \frac{n^2}{r^2} y$, и граничные условия

$$w_n(1) = w_n'(1) = F_n(1) = F_n'(1) = 0. \tag{8}$$

Учитывая ограниченность искомых решений, в центре пластины полагаем

$$w_n'(0) = F_n'(0) = 0. \tag{9}$$

Для каждого числа волн n будем искать такие значения нагрузки p_n , при которых существуют отличное от нуля решение системы (7)-(9) при условии (5)-(6). Под критической нагрузкой p_{cr} будем понимать наименьшее значение нагрузки p_n , т. е. $p_{cr} = \min_n(p_n)$.

Схема численного решения задачи состоит из двух этапов и аналогична методу, описанному в работах [4], [5]. На первом этапе решается симметричная задача (5)-(6). Для заданных p и n вычисляются w , F и их производные. Затем с помощью метода конечных разностей проверяется существование решения несимметричной задачи.

Пластина с упруго-заделанным краем. Нелинейная задача теории пластин и оболочек допускает различные варианты граничных условий на краю пластины. Рассмотрим случай наличия по краю пластины упругой связи (пружины), препятствующей ее свободному смещению в радиальном направлении, тогда проекция вектора перемещения u на ось r и тангенциальное усилие T_r на внешнем крае взаимосвязаны, т. е. $k_u u + T_r = 0$, где k_u — коэффициент упругости заделки.

Таким образом, система уравнений (1) сохраняется, а граничные условия запишутся в виде

$$w = w' = 0, \quad k_u u + T_r = k_u u + \frac{1}{r} F' + \frac{1}{r^2} \dot{F} = 0, \quad S = -\left(\frac{1}{r} \dot{F}\right) = 0. \tag{10}$$

Чтобы выразить компоненту перемещения $u(r, \theta)$ через искомые функции w, F , воспользуемся представлением компонент деформации $\varepsilon_r, \varepsilon_\theta, \omega$ через компоненты вектора перемещения u, v, w , с одной стороны, и усилий T_r, T_θ, S , с другой:

$$\begin{aligned} \frac{\partial \varepsilon_\theta}{\partial r} - \frac{1}{r} \varepsilon_r - \frac{1}{r} \frac{\partial \omega}{\partial \theta} &= \frac{\partial}{\partial r} \left(\frac{T_\theta - \nu T_r}{E(r)h} \right) - \frac{1}{E(r)h} \frac{T_r - \nu T_\theta}{r} - \frac{2(1+\nu)}{E(r)h} \frac{1}{r} \frac{\partial S}{\partial \theta} = \\ &= -\frac{u}{r^2} - \frac{1}{r^2} \frac{\partial^2 u}{\partial \theta^2} - \frac{1}{2r} \left(\frac{\partial w}{\partial r} \right)^2 - \frac{1}{r} \frac{\partial w}{\partial r} \frac{\partial^2 w}{\partial \theta^2} - \frac{1}{r^3} \left(\frac{\partial w}{\partial \theta} \right)^2 \end{aligned} \tag{11}$$

После разделения переменных (4) граничные условия для симметричной задачи (5) запишутся в виде

$$\omega_0(0) = \phi_0(0) = 0, \quad \omega_0(1) = 0, \quad k_u \phi_0'(1) + \phi_0(1)(1 - k_u \nu). \tag{12}$$

Для системы (7), описывающей несимметричное решение, граничные условия примут вид

$$w_n(1) = w_n'(1) = 0, \quad k_u u_n + F_n'(1) - n^2 F_n(1) = 0, \quad F_n(1) - F_n'(1) = 0, \tag{13}$$

где $u_n = \frac{1}{n^2 - 1} (F_n''' - ((2 + \nu)n^2 + 1 - \nu)F_n' + 3n^2 F_n)$

Случай кольцевой пластины. В случае кольцевой пластины вместо условий ограниченности решений в центре пластины необходимо задать граничные условия. Одним из возможных вариантов закрепления внутреннего края ($r = \delta$) может быть свободное перемещение точек края в радиальном и окружном направлении при ограничении поворотов. Тогда необходимо принять равным нулю поперечное реактивное усилие, складывающееся из поперечной силы и производной от крутящего момента. В терминах нормального перемещения w и функции усилий F граничные условия запишутся в следующем виде:

$$r = \delta, \quad w' = -\left(\frac{\dot{F}}{r}\right)' = 0, \quad (\Delta w)' + \frac{1-\nu}{r} \left(\frac{\ddot{w}}{r}\right)' = 0, \quad (\Delta F)' + \frac{\Delta F}{r} - \frac{1+\nu}{r^2} (\ddot{F} + F)' = 0. \tag{14}$$

На внешнем крае пластины $r = 1$ в общем случае можно задать условия упругой заделки (10). Тогда система уравнений, описывающая симметричное равновесное состояние, примет вид

$$\omega_0'' + \frac{\omega_0'}{r} - \frac{\omega_0}{r^2} = \frac{pr}{2} \left(1 - \frac{\delta^2}{r^2}\right) + \frac{\omega_0 \phi_0}{r}, \tag{15}$$

$$\phi_0'' + \frac{\phi_0'}{r} - \frac{\phi_0}{r^2} = \frac{pr}{2} + \frac{\omega_0 \phi_0}{r},$$

с граничными условиями $\omega_0(\delta) = \phi_0'(\delta) - \frac{\nu}{\delta} \phi_0(\delta) = 0, \quad \omega_0(1) = 0, \quad k_u \phi_0'(1) + \phi_0(1)(1 - k_u \nu)$.

Результаты. Были проведены серии расчетов для находящейся под давлением однородной пластины, а также для пластины с упруго-заделанным краем, для кольцевой пластины и для пластины, совмещающей эти свойства.

Для однородной сплошной пластины с краем, который может свободно смещаться в радиальном направлении, наименьшее критическое значение нагрузки p_{cr}^0 и соответствующее ему волновое число n , найденные в результате решения системы (7) — (8), оказались равны $p_{cr}^0 = 67236$ и $n = 13$. Полученные результаты для $n = 10, 11, \dots, 20$ представлены в таблице 1.

Таблица 1. Значения критических нагрузок и соответствующих волновых чисел для однородной сплошной пластины.

n	10	11	12	13	14	15
p_{cr}	114751	77425	69387	67236	67587	69335
n	16	17	18	19	20	
p_{cr}	71998	75331	79197	83512	88223	

В случае упругой связи по краю пластины, с ростом жесткости пружины k_u пластина переходит в неосесимметричное состояние при более высоких величинах критической нагрузки и с образованием большего числа волн в окружном направлении, что можно увидеть в таблице 2. В таблице 3 представлены минимальные критические нагрузки для рассмотренных k_u .

Таблица 2. Значения критических нагрузок и соответствующих волновых чисел для однородной сплошной пластины с упруго-заделанным краем.

$n \backslash k_u$	0.001	0.01	0.05	0.1	0.15	0.2	0.3
10	170170						
11	80908	94485					
12	69107	77457	139087				
13	65527	72240	115036	241263			
14	65088	71111	106897	191116	420604		
15	66296	72026	104650	173081	315936	748190	
16	68538	74184	105413	166377	277496	524978	
17	71514	77205	108021	165256	261119	444182	
18	75058	80883	111909	167447	255179	407551	
19	79074	85098	116770	171866	255211	390838	1177638
20	83501	89776	122424	177927	259124	385084	1007669

Таблица 3. Минимальные значения критических нагрузок и соответствующих волновых чисел для однородной сплошной пластины с упруго-заделанным краем

k_u	0.001	0.01	0.05	0.1	0.15	0.2	0.3
n	14	14	15	17	18	20	24
p_{cr}^0	65088	71111	104650	165256	255179	385084	839714

Значения критических нагрузок и соответствующих им волновых чисел, полученных в результате вычислений для кольцевой пластины при различных значениях радиуса отверстия δ , показаны в таблице 4. Наименьшие значения нагрузки для каждого δ представлены в таблице 5.

Таблица 4. Значения критических нагрузок и соответствующих волновых чисел для однородной кольцевой пластины

$n \backslash \delta$	0.01	0.05	0.1	0.15	0.2
10	132711	100303	87653	84136	85533

11	78075	74328	73353	75429	80280
12	67938	67746	69631	73746	80285
13	64952	66238	69583	74971	82766
14	64827	67024	71400	77817	86740
15	66242	69133	74378	81740	91765
16	68639	72133	78183	86470	97613
17	71743	75800	82636	91857	104156
18	75401	80006	87633	97815	111317
19	79521	84676	93111	104286	119048
20	84047	89761	99027	111238	127318

Таблица 5. Минимальные значения критических нагрузок и соответствующих волновых чисел для однородной кольцевой пластины.

δ	0.01	0.05	0.1	0.15	0.2
n	14	13	13	12	11
p_{cr}^0	64827	66238	69583	73746	80280

В случае наличия у кольцевой пластины упруго-заделанного края, так же, как и в случае сплошной пластины, при увеличении жесткости пружины k_u переход в неосесимметричное состояние происходит при более больших значениях критической нагрузки и с образованием большего числа волн. Результаты расчетов можно увидеть в таблице 6. В таблице 7 показаны наименьшие критические значения нагрузки и соответствующие им волновые числа для каждого рассмотренного δ и $k_u = 0.001, 0.01, 0.05$.

Таблица 6. Значения критических нагрузок и соответствующих волновых чисел для кольцевой пластины с упруго-заделанным краем

$n \backslash k_u$	0.001	0.01	0.05	0.001	0.01	0.05
	$\delta = 0.01$			$\delta = 0.05$		
10	187592			108171		
11	80840			76321	115819	
12	69235	88269		68813	83346	
13	65769	76222		66944	75778	
14	65409	72546		67536	73822	
15	66680	72050		69521	74342	
16	68978	73228		72430	76274	
17	72005	75459	149816	76024	79143	137269
18	75599	78436	127516	80167	82704	126009
19	79663	81988	120868	84780	86817	122550
20	84136	86013	119003	89809	91397	122395
	$\delta = 0.1$			$\delta = 0.15$		
10	91445			86641	154917	
11	74842	97850		76640	93552	
12	70514	81970		74512	84222	
13	70191	77731		75510	82227	
14	71846	77397		78211	83253	
15	74713	79029		82028	85959	
16	78433	81874	154570	86672	89773	149685
17	82813	85574	134873	91984	94408	139630
18	87746	89935	129286	97870	99704	137238
19	93161	94843	128379	104273	105564	138238
20	99017	100227	129883	111156	111927	141206

		$\delta = 0.2$				
10	87448	120305				
11	81323	95204				
12	80970	89635				
13	83250	89427				
14	87086	91755				
15	92003	95611	177220			
16	97760	100533	155426			
17	104220	106282	149991			
18	111300	112720	149921			
19	118951	119761	152622			
20	127138	127351	157069			

Таблица 7. Минимальные значения критических нагрузок и соответствующих волновых чисел для кольцевой пластины с упруго-заделанным краем

$\delta \backslash k_u$	0.001		0.01		0.05	
	n	p_{cr}^0	n	p_{cr}^0	n	p_{cr}^0
0.01	14	65409	15	72050	20	119003
0.05	13	66944	14	73822	20	122395
0.1	13	70191	14	77397	19	128379
0.15	12	74512	13	82227	18	137238
0.2	12	80970	13	89427	18	149921

Для $k_u = 0.1, 0.2, 0.3$, рассмотренных в случае сплошной пластины, значения критических нагрузок оказываются очень большими при большом числе образующихся волн. Пример для случая $k_u = 0.1$ приведен в таблице 8.

Таблица 8. Минимальные значения критических нагрузок и соответствующих волновых чисел для кольцевой пластины с упруго-заделанным краем при $k_u = 0.1$

δ	0.01	0.05	0.1	0.15	0.2
n	31	30	29	29	28
p_{cr}^0	1165458	1153910	1190263	1444323	1488912

Заключение. В данной работе представлены результаты численного исследования потери устойчивости осесимметричных форм равновесия однородных сплошных и кольцевых пластин при различных условиях закрепления внешнего края.

Показано, что при увеличении жесткости пружины, ограничивающей перемещение края пластины в радиальном направлении, как в случае сплошной, так и в случае кольцевой пластины переход в неосесимметричное состояние происходит при значительно больших критических нагрузках и с образованием большего числа волн в окружном направлении. Показана зависимость значений критической нагрузки от размеров отверстия в центре в случае кольцевой пластины.

Литература:

1. Панов, Д. Ю., Феодосьев В. И. О равновесии и потере устойчивости пологих оболочек при больших прогибах // ПММ. Т. XII. 1948. с. 389-406.
2. Морозов, Н. Ф. К вопросу о существовании несимметричного решения в задаче о больших прогибах круглой пластины, нагруженной симметричной нагрузкой // Изв. Высш. Уч. Заведений. Математика. 1961. № 2. с. 126-129.
3. Piechocki, W. On the non-linear theory of thin elastic spherical shells // Arch. Mech. Stos. 1969. N21. P. 81-101.
4. Huang, N. C. Unsymmetrical buckling of thin shallow spherical shells // J. Appl. Mech. 1964. N31. P. 447-457.
5. Cheo, L. S., Reiss E. L. Unsymmetric wrinkling of circular plates // Quart. Appl. Math. 1971. N31. P. 75-91.

6. Bauer, S. M., Voronkova E. B. On the deformation of the Lamina Cribrosa under intraocular pressure // Russian Journal of Biomechanics. 2001. Vol. 5. N1. P. 73-82.
7. Бауэр, С. М., Воронкова Е. Б. О потере устойчивости симметричных форм равновесия круглых пластин под действием нормального давления // Вестн. С.-Петерб. ун-та. Сер. 1. 2012. Вып. 1. с. 80-85.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Использование нейронных сетей в задаче прогнозирования закупок товаров

Балавнев Данил Андреевич, студент;
Киндулов Михаил Львович, студент;
Горелов Борис Романович, студент;
Шергин Тимур Олегович, студент
Санкт-Петербургский государственный университет

В статье представлен подход к прогнозированию временных рядов на заданный промежуток учитывающий сезонность продаж.

Ключевые слова: прогнозирование временных рядов, LSTM-сети, рекуррентные нейронные сети.

В условиях нестабильного рынка деятельность любого бизнеса зависит от точности определения необходимого объема закупок. На плечи закупщика ложится тяжелая задача расчета оптимального количества ликвидного товара. Низкий товарный запас порождает дефицит и негативно сказывается на лояльности клиентов, и напротив, чрезмерное количество влечет издержки хранения. Для определения закономерностей в данных по продажам и построения прогноза, учитывающего колебания спроса и сезонности необходимого количества продукта, отлично подходят нейронные сети.

Нейронная сеть

Центральным моментом при составлении прогноза закупок товаров является то, что существует множество непредсказуемых факторов, приводящих к сложным временным зависимостям. Нейронные сети являются одним из методов, который способен учесть большое количество факторов, сезонность и историю продаж. Они устойчивы в разумных пределах к помехам во входных данных и могут даже поддерживать обучение и прогнозирование при наличии пропущенных значений. Также важным свойством нейронной сети является обобщающая способность — способность выдавать правильные результаты не только для данных участвующих в обучении, но и для новых. Нейронные сети не исходят из априорных предположений о функциональной взаимосвязи ряда и его независимых переменных, что является весьма желательной характеристикой в случае работы с временными рядами.

При использовании сетей прямого распространения подразумевается, что все входы и выходы независимы [1], но для задачи прогнозирования временного ряда это не подходит. Рекуррентные нейронные сети (RNN), называются рекуррентными, потому что они имеют обратные

связи и выполняют одну и ту же задачу для каждого элемента последовательности и выход зависит от предыдущих вычислений [2]. Каждый нейрон может использовать свою внутреннюю память для хранения информации о предыдущем состоянии сети. RNN имеет обратные связи, которые позволяют передавать информацию от одного шага сети к следующему. Однако, фундаментальная проблема использования простых рекуррентных сетей заключается в том, что они способны обнаруживать только краткосрочные зависимости. Причиной этого является проблема отсутствия долгосрочной памяти. В простых RNN нейроны хорошо «помнят» недавно полученную информацию, но не имеют возможности надолго сохранить информацию в памяти.

Нейронные сети с долгой краткосрочной памятью (LSTM) [3] представляют собой особый тип RNN, которые имеют более длинную «память», чем их предшественники, и способны изучать долгосрочные зависимости. В LSTM блок работает как обычный рекуррентный блок, но имеет дополнительную ячейку памяти и различные фильтры, обычно называемые вентилями. Вентили помогают определить долгосрочные зависимости, управляя потоком данных внутри блока. Ячейка дает сети своего рода память и сохраняет долгосрочные зависимости. Процесс обучения и обратного распространения здесь работает так же, как и в обычной RNN, но с добавлением, что вентили также имеют веса. Таким образом, модель становится лучше в определении правильной информации, что приводит к хорошим прогнозам. Поэтому использовалась архитектура на основе LSTM-ячеек.

Обучение и тестирование

Для разработки и тестирования нейронной сети была взята двухгодичная база данных продаж онлайн магазина

содержащая в себе 1400000 записей. Компания специализируется на продаже сувенирной продукции.

Для повышения качества работы нейронной сети были выделены следующие признаки:

- Время совершения продажи.
- Был ли в этот день праздник.
- Количество дней до ближайшего праздника. В предпраздничные дни возможно сильное повышение уровня продаж и, если запасы окажутся недостаточными, это чревато высоким уровнем упущенной прибыли.
- Общий тренд. Изменение средней цены товара в зависимости от времени, цена постепенно повышается или понижается.

При составлении прогноза закупок важно не допустить дефицита товара на складе. Небольшие излишки товара можно назвать страховым запасом, а отсутствие необходимого товара при обращении клиента несет, как минимум — потерю выручки, а как максимум — ущерб имиджу компании. Чтобы учесть этот фактор при обучении, в качестве функции ошибки будем использовать

асимметричную квадратичную ошибку (1), которая обеспечивает более высокое значение ошибки, когда модель предсказывает значение ниже реального [4].

$$\begin{cases} (y_{predict} - y_{target})^2, & y_p > y_t \\ (y_{predict} - y_{target})^4, & else \end{cases} \quad (1)$$

Также, чтобы повысить эффективность обучения, важно уменьшать шаг обучения на каждой эпохе. В данной реализации уменьшение скорости обучения происходит раз в 10 эпох. Чтобы не допустить переобучения нейронной сети, важно следить за ошибкой валидации. Когда она начинает возрастать, необходимо остановить обучение.

Для того, чтобы проверить качество разработанной нейронной сети, была использована отложенная выборка. Выборку разбили на две части, 80% использовались для обучения, а 20% данных были отложены для тестирования. Ниже (табл. 1 и рис. 1) представлены результаты тестирования.

Таблица 1. Результаты тестирования

	train	valid	test
Средняя абсолютная ошибка	33.97396061691954	35.91977171419663	37.123308042252916
Средняя квадратичная ошибка	1154.23	1290.23	1378.14
Ассиметричная квадратичная ошибка	1466.56	1722.0	1879.34

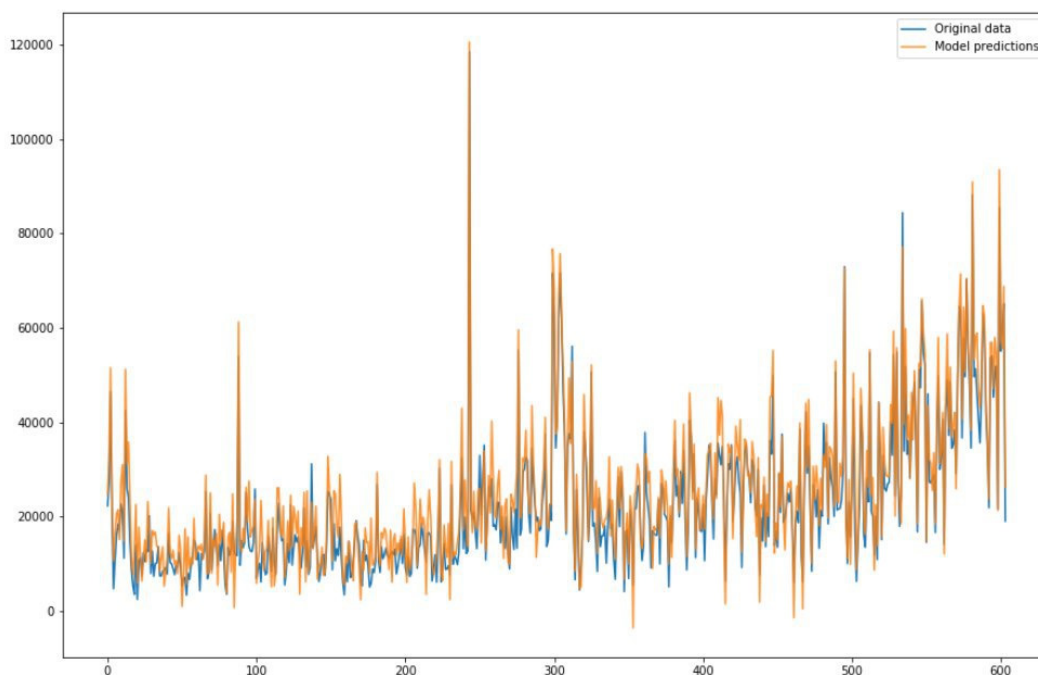


Рис. 1. Предсказание для всех товаров

Литература:

1. С. Хайкин, Нейронные сети: полный курс, 2-е изд., испр.: Пер. с англ. — М.: ООО «И. Д. Вильямс», 2006, 1104 стр.
2. Introduction to Recurrent Neural Network [Электрон. ресурс] — URL: www.geeksforgeeks.org/introduction-to-recurrent-neural-network/
3. Essentials of Deep Learning: Introduction to Long Short Term Memory [Электрон. ресурс] — URL: www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm
4. Kevin Aretz, Söhnke M, «Асимметричные функции потерь и рациональность ожидаемой доходности фонда». Международный журнал по прогнозированию.
5. A Beginner's Guide to LSTMs and Recurrent Neural Networks [Электрон. ресурс] — URL: www.pathmind.com/wiki/lstm
- 6.

Разработка системы удаленной компиляции

Горелов Борис Романович, студент;
Балавнев Данил Андреевич, студент;
Шергин Тимур Олегович, студент;
Киндулов Михаил Львович, студент
Санкт-Петербургский государственный университет

В статье будут рассмотрены и проанализированы существующие подходы к решению задачи удаленной компиляции и предложен альтернативный подход, сочетающий в себе простоту и безопасность использования, высокую скорость работы.

Ключевые слова: удаленная разработка, сокеты, TLS.

Сейчас на рынке можно найти много готовых решений, позволяющих вести удаленную разработку и в том числе производить удаленную компиляцию. Условно решения данной задачи можно разделить на несколько классов. Один из них — это полноценные интегрированные среды разработки (IDE), предоставляющие широкие возможности, в том числе удаленную разработку (например NetBeans, CLion). К сожалению, часто для комфортной работы с большим проектом в таком приложении требуется мощный компьютер.

Другим классом решений являются утилиты, автоматизирующие сборку, с функционалом распределенных вычислений (например FASTBuild, IncrediBuild, Distcc). При наличии большого количества вычислительных мощностей данные утилиты многократно ускоряют процесс компиляции и сборки программного обеспечения, что является их главным плюсом. Среди недостатков можно отметить требование серьезной настройки программного окружения, узкий круг поддерживаемых языков (чаще всего только C/C++).

Для устранения недостатков существующих подходов, а также для обеспечения простоты и безопасности использования была разработана собственная система, представляющая из себя консольное клиент-серверное приложение, разрабатываемое на языке C под ОС Linux. Ее работа состоит из нескольких этапов:

1. Инициализация соединения клиента с сервером

2. Архивация и сжатие выбранных файлов (опционально)
3. Инициализация работы протокола TLS (опционально)
4. Отправка данных на сервер
5. Компиляция файлов/сборка проекта
6. Отправка результата клиенту.

Для всех сетевых взаимодействий был использован зарекомендовавший себя интерфейс сокетов Беркли. Сокет позволяет управлять сетевым адресом, транспортом (то есть выбирать протокол транспортного уровня), а также предоставляет базовые операции ввода вывода (открытие/закрытие соединения, отправка/получение данных). В качестве протокола транспортного уровня при создании сокета было решено использовать TCP, так как в отличие от UDP он обеспечивает надежную доставку данных. Для того, чтобы отслеживать несколько клиентских сокетов, не блокируясь на них, использовался мультиплексор select. Он работает как обработчик прерываний, который активируется, как только на любой из сокетов приходят какие-либо данные. Если запрос пришел на исходный серверный сокет, он трактуется как запрос на новое соединение. Иначе происходит взаимодействие с уже существующим клиентом.

Для удобства пользователя была добавлена возможность архивации данных. Если требуется обработать 1-2

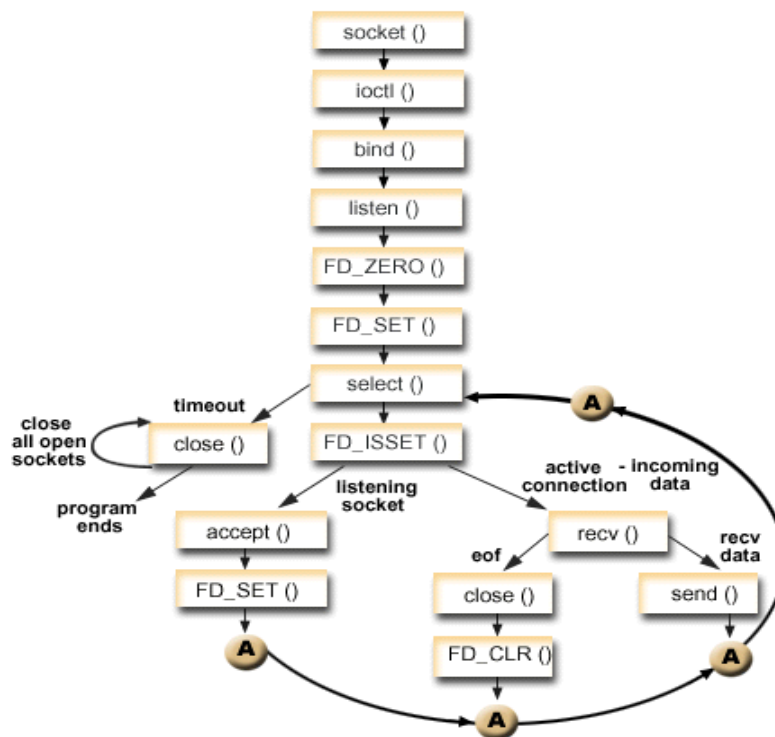


Рис. 1: Интерфейс сокетов Беркли, мультиплексор select

файла, не составляет труда ввести их имена. В противном случае, этот процесс может быть трудоемким, поэтому вместо файла пользователь может указать папку с файлами, содержащими исходный код. В этом случае с помощью архиватора tar создается архив. Далее при помощи утилиты gzip производится сжатие. В полученном виде информацию можно удобно и быстро передать на сервер для последующей работы.

Поскольку компилируемая программа может представлять коммерческий интерес, в разрабатываемую систему была добавлена возможность шифрования данных. Для этих целей использовалась широко известная криптографическая библиотека с открытым исходным кодом OpenSSL. Используемый в системе протокол TLS версии 1.2 является стандартом на рынке и гарантирует что данные, передаваемые между сервером и клиентом, защищены и не повреждены. TLS использует асимметричные алгоритмы шифрования. Рассмотрим процесс установки соединения. После того, как стороны договорились использовать TLS, согласование деталей соединения происходит с помощью представленной на Рис. 2 процедуры рукопожатия. В начале клиент отправляет запрос на установление безопасного соединения и указывает в нем доступный ему набор шифров и хэш функций. Сервер выбирает шифр и хэш-функцию, отправляет подтверждение, а также свой цифровой сертификат, содержащий имя сервера, публичный ключ и название центра сертификации. Затем

в случае, если клиент подтвердил действительность сертификата, он шифрует случайное число публичным ключом сервера и отправляет его. Последним этапом сервер расшифровывает переданное случайное число своим приватным ключом и обе стороны с помощью данного числа генерируют сеансовый ключ для последующего шифрования и дешифрования данных во время сеанса. Таким образом обеспечивается защита данных даже в случае, если они были перехвачены третьей стороной.

Были протестированы базовый и расширенный функционал приложения. Создан каталог с тестами и версиями системы для отслеживания изменений в работе приложения. В базовом режиме на сервер передавались однофайловые и многофайловые проекты. Тесты подразумевали как корректный пользовательский ввод, так и нет (например, содержащий ошибки в названиях файлов). Для тестирования расширенного функционала были выбраны проекты и разной структурой. Тесты показали, что текущая версия системы без ошибок работает с проектами с одноуровневой структурой, обработка многоуровневых проектов требует участия пользователя.

Благодаря тому, что основной функционал приложения написан на языке C, есть возможность реализовать кроссплатформенную версию приложения. Для этого потребуется доработать модуль архивации и добавить аналоги системных вызовов для соответствующих платформ.

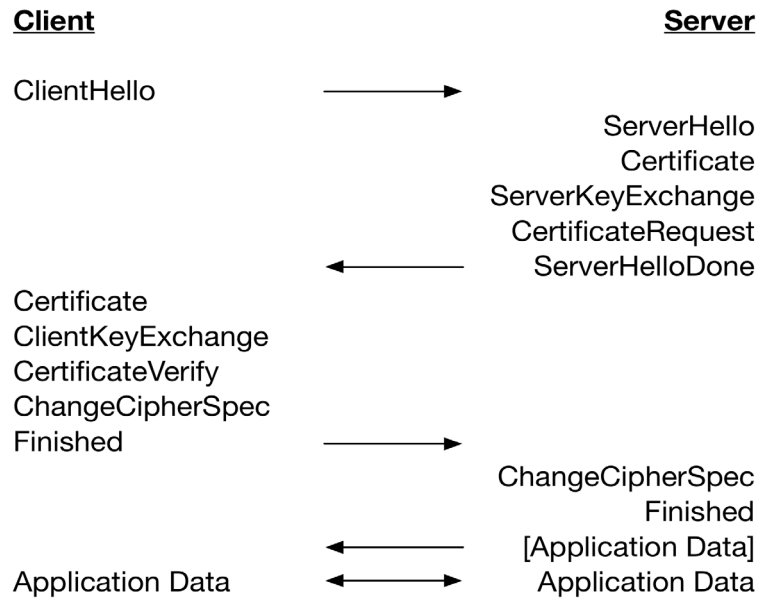


Рис. 2. Процедура рукопожатия TLS

Литература:

1. W. Richard Stevens; Bill Fenner & Andrew M. Rudoff (2003). Unix Network Programming, Volume 1: The Sockets Networking API (3rd Edition). Addison-Wesley Professional.
2. Advanced Programming in the UNIX environment, third edition, W. Richard Stevens and Stephen A. Rago, Addison-Wesley, 2013
3. Postel, Jon, «Transmission Control Protocol — DARPA Internet Program Protocol Specification», RFC 793, DARPA, September 1981.
4. Berkeley sockets — URL: https://en.wikipedia.org/wiki/Berkeley_sockets
5. OpenSSL Cryptography and SSL/TLS toolkit — URL: <https://www.openssl.org>
6. SSL/TLS cryptographic protocols — URL: https://en.wikipedia.org/wiki/Transport_Layer_Security

Методы детектирования состязательных атак

Киндулов Михаил Львович, студент;
 Балавнев Данил Андреевич, студент;
 Горелов Борис Романович, студент;
 Шергин Тимур Олегович, студент
 Санкт-Петербургский государственный университет

В статье рассматриваются подходы к детектированию состязательных атак. Используются такие классические атаки как FSGM, Deer Fool, C&W и PGD, нейронная сеть ResNet-18, датасеты MNIST и CIFAR-10.

Ключевые слова: машинное обучение, глубокое обучение, состязательные атаки, защита от состязательных атак.

В современном мире модели машинного обучения используются повсеместно, но, к сожалению, на многие из них могут быть произведены атаки. Состязательными атаками называются методы, направленные на введение в заблуждение моделей машинного обучения. Для защиты от подобных атак необходимо использовать специальные модификации моделей машинного обучения. Именно такие модификации будут рассмотрены в данной статье.

Достаточно большое число статей, связанных с защитой от состязательных атак, относятся к типу детекции атакованных примеров [1] [2] [3] [4] [5] [6]. Методы этого типа имеют широкое практическое применение, так как просты в реализации.

Алгоритмы защиты, предложенные в статье, также относятся к типу детекции атак. Хочется отметить, что представленные ниже методы могут быть перенесены на боль-

шинство обучаемых с помощью градиентного спуска моделей с дважды дифференцируемой функцией ошибки.

Рассмотрим первый из методов. Для иллюстрации его работы используем нейронную сеть ResNet-18 [7], атаки Deep Fool [8], FSGM [9] и набор данных CIFAR-10 [e10].

Выберем случайным образом небольшое количество изображений из тестового набора набора данных, обозначим их как X , обозначим также через X_i i-ое изобра-

жение. Обозначим через X_{df} изображения, атакованные с помощью Deep Fool, и через X_{fsgm} — изображения, атакованные с помощью FSGM. Пусть также функция $N(x, \mu, \sigma) = \mathcal{N}(\mu, \sigma) + x$ добавляет на изображение гауссовский шум с параметрами μ, σ .

Рассмотрим логиты модели для изображений $X, X_{df}, X_{fsgm}, N(x, \mu, \sigma), N(X_{df}, \mu, \sigma), N(X_{fsgm}, \mu, \sigma)$ (рис. 1).

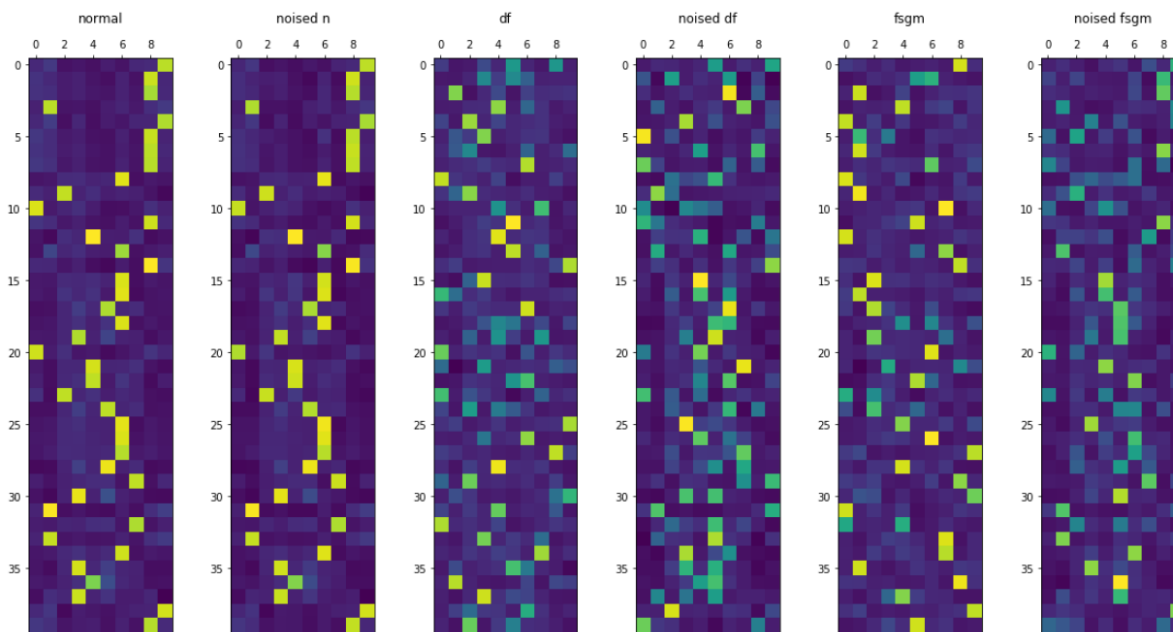


Рис. 1: Логиты модели для каждого из полученных изображений. Слева направо: для обычных изображений, для зашумленных, для атакованных с помощью Deep Fool, для атакованных с помощью Deep Fool и после этого зашумленных, для атакованных с помощью FSGM, для атакованных с помощью FSGM и после этого зашумленных. По оси y отложены различные объекты, по оси x логиты их классов

Интересным здесь является то, что предсказания моделей для атакованных и атакованных, а затем зашумленных изображений значительно различаются (рис. 2, рис. 3).

Основываясь на этих эмпирических наблюдениях, можно построить простой детектор атакованных изображений. Например, можно просто передавать разности логитов (выходной сигнал нейронной сети до применения softmax) основной модели для зашумленного и обычного изображений в логистическую регрессию. Также до передачи в модель, можно попробовать их отсортировать. Последний подход дает ROC AUC равный 0.998 для атак типа FSGM и Deep Fool. Этот метод, однако, плохо работает с таргетированными атаками. Попробуем улучшить полученные результаты.

Рассмотрим второй из методов. Для иллюстрации его работы также используем нейронную сеть ResNet-18 [7], атаки Deep Fool [8], FSGM [9] и набор данных CIFAR-10 [10].

Известно, что некоторые аугментации данных при обучении позволяют получать более стойкие к атакам модели машинного обучения, например MaxUp [11], MixUp [12],

CutMix+MaxUp [11]. Более того, в [11] доказано, что обучение с использованием MaxUp привносит дополнительную гладкость в ландшафт функции потерь, величина которой для обучающих примеров напрямую зависит от параметров аугментации. Иными словами, для небольших заранее заданных возмущений гарантировано отсутствие резких скачков функции потерь в некотором ϵ -шаре от обучающего примера x .

Рассмотрим отклик в ϵ -шаре от атакованного примера X , чтобы понять как мы можем использовать аугментации для детекции атак. До передачи в модель применим $N(x, \mu, \sigma)$ к изображениям, построим гистограммы (рис. 4).

Как видно из распределений на рис. 4, дисперсия отклика параметров для атакованных примеров значительно больше, чем для обычного изображения. Очевидно, что за этим стоит невысокая стабильность функции потерь вокруг атакованных изображений. Это также может быть связано с тем, что в большинстве атак требуется найти минимальное возмущение от оригинального примера X_i и, соответственно, минимальные же возмущения могут перенести объект в область пространства класса, отличного от атакованного.

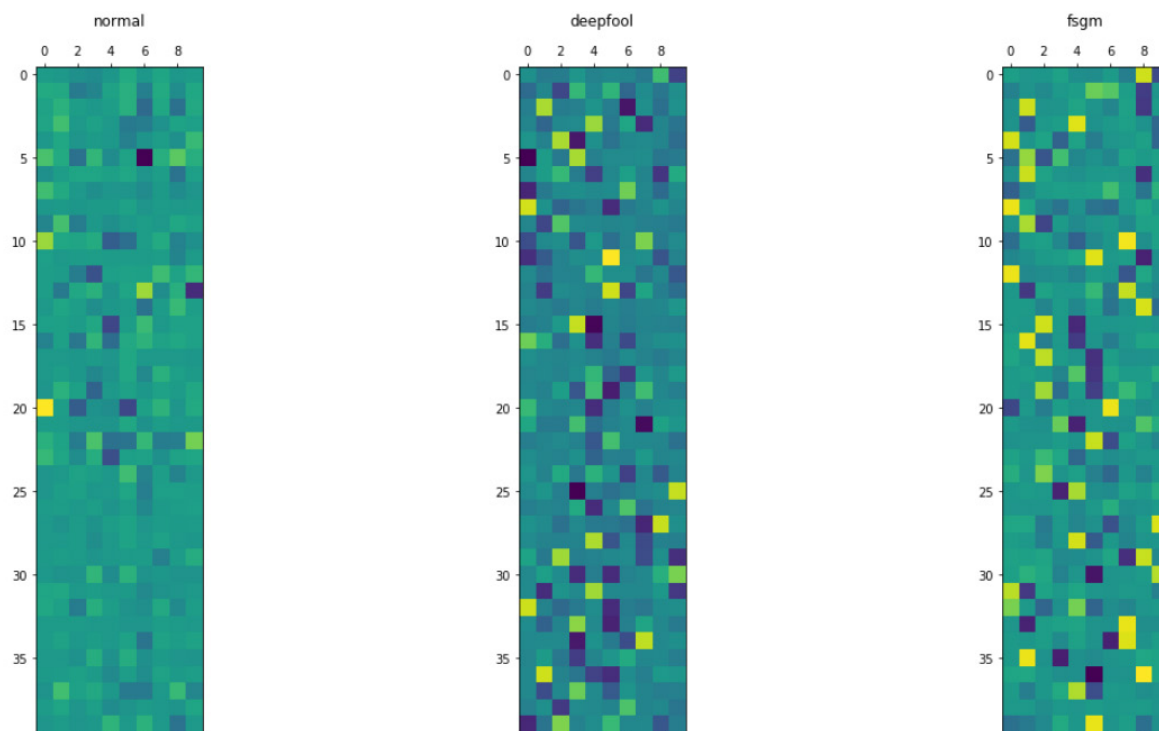


Рис. 2: Разности логитов для обычных изображений и изображений с шумом. По оси y отложены различные объекты, по оси x разности логитов их классов

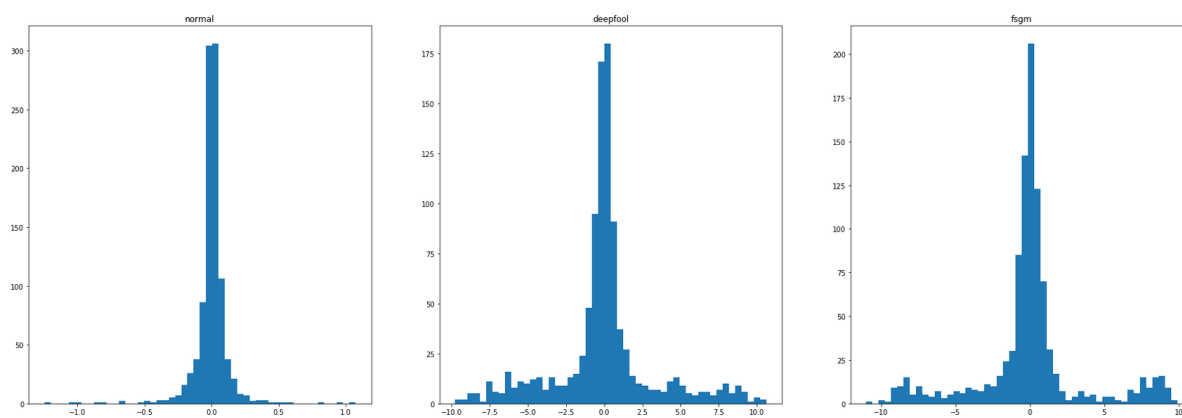


Рис. 3: Распределение разностей логитов

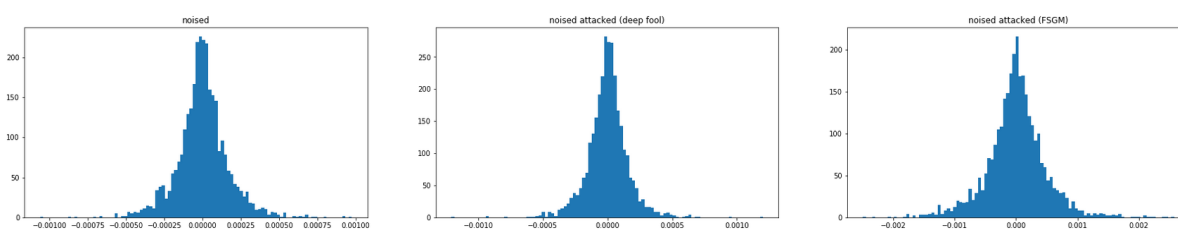


Рис. 4: Распределения откликов параметров

Так как гладкость функции потерь гарантирована в ϵ -шаре, вблизи обучающих примеров небольшие возмущения параметров практически не вызывают изменений отклика параметров. Воспользуемся этим.

Рассмотрим распределения разностей откликов для обычного и зашумленного изображений при обычной модели (рис. 5).

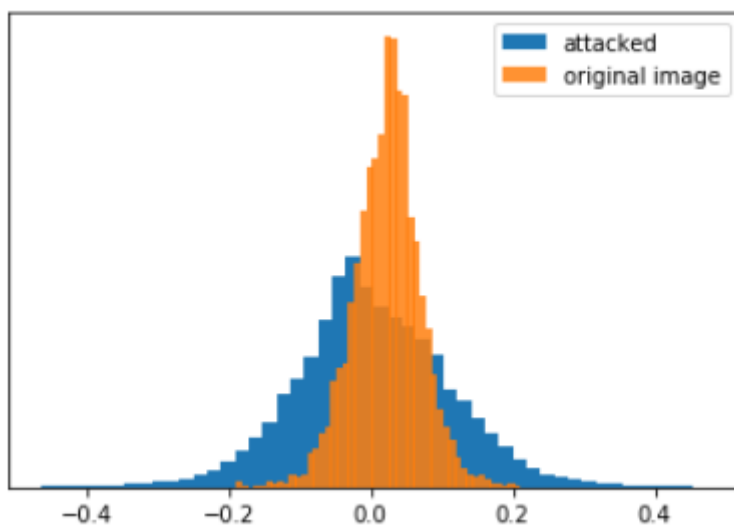


Рис. 5: Разность откликов для обычного и зашумленного изображений при обычной модели (атака Deep Fool)

Теперь рассмотрим распределения разностей откликов для обычного и зашумленного изображений при MaxUp модели (рис. 6).

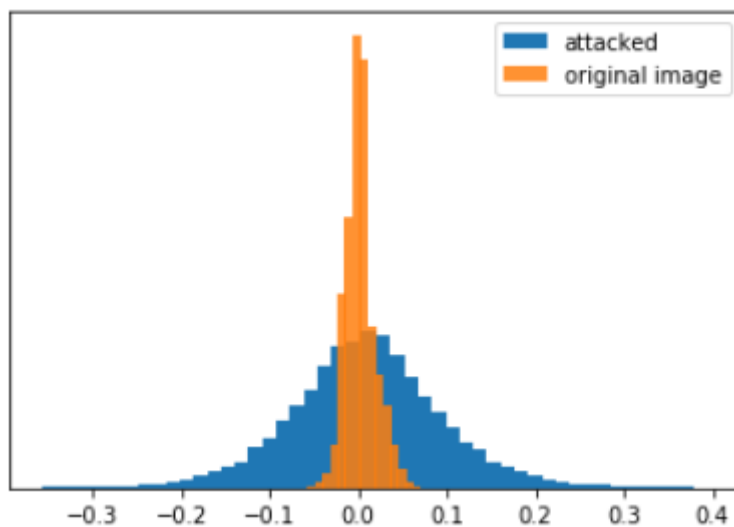


Рис. 6: Разность откликов для обычного и зашумленного изображений при MaxUp модели (атака Deep Fool)

Как видно на рис. 6 и рис. 7, распределения имеют значительные различия для атакованных и не атакованных объектов, в особенности для MaxUp модели. За этим стоит очевидный факт: так как модель научилась игнорировать малые возмущения для объектов выборки, распределение откликов практически не изменяется. С другой стороны, если атакованный пример, до атаки распознавался как класс i , в ϵ -шаре от него не существует негладких областей перехода из одного класса в другой. Это значит, что любое возмущение любого объекта в ϵ -шаре от дан-

ного примера будет либо распознано как класс i , либо иметь большую дисперсию отклика чем объекты класса i .

Таким образом метод поиска атакованных объектов может быть сведен к следующему:

- Дообучение модели с использованием MaxUp.
- Создание набора данных, состоящего из разностей откликов для обычных и зашумленных объектов.
- Подсчет разности откликов для нового примера, сравнение его дисперсии и среднего с дисперсией и средним ранее полученного набора данных.

Для проведения экспериментов были выбраны открытые наборы данных MNIST [13] и CIFAR-10. Тестирование производилось на ResNet-18. Метрика, по которой сравнивались детекторы атак — ROC AUC. Атаки, исследованные во время эксперимента:

— Deep Fool.

— FSGM.

— C&W [14].

— PGD (для $\|\cdot\|_\infty$ -нормы) [15].

Результаты тестирования приведены в сводных таблицах ниже.

Таблица 1. Результаты тестирования для первого подхода.

	MNIST	CIFAR-10
Deep Fool	1.0	0.998
FSGM	0.99	0.998
C&W	0.899	0.987
PGD	0.73	0.89

Таблица 2. Результаты тестирования для второго подхода.

	MNIST	CIFAR-10
Deep Fool	1.0	0.998
FSGM	1.0	0.9936
C&W	1.0	0.989
PGD	1.0	0.997

Стоит также отметить, что при столь высокой точности нахождения атак, минусами решения являются:

— дополнительные затраты на дообучение с использованием MaxUp;

— необходимость сделать предсказания для двух изображений: обычного и зашумленного;

— точность предсказаний основной модели может незначительно уменьшиться (в проведенных экспериментах падение точности составило 0.007).

Литература:

1. Xin Li and Fuxin Li. Adversarial examples detection in deep networks with convolutional filter statistics. In Proceedings of the IEEE International Conference on Computer Vision, pages 5764–5772, 2017.
2. Arjun Nitin Bhagoji, Daniel Cullina, Chawin Sitawarin, and Prateek Mittal. Enhancing robustness of machine learning systems via data transformations. In 2018 52nd Annual Conference on Information Sciences and Systems (CISS), pages 1–5. IEEE, 2018.
3. Dan Hendrycks and Kevin Gimpel. Early methods for detecting adversarial images. In International Conference on Learning Representations, 2017.
4. Kathrin Grosse, Praveen Manoharan, Nicolas Papernot, Michael Backes, and Patrick McDaniel. On the (statistical) detection of adversarial examples. arXiv preprint arXiv:1702.06280, 2017.
5. Zhitao Gong, Wenlu Wang, and Wei-Shinn Ku. Adversarial and clean data are not twins. arXiv preprint arXiv:1704.04960, 2017.
6. Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. On detecting adversarial perturbations. In International Conference on Learning Representations, 2017.
7. He, Kaiming, et al. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
8. Moosavi-Dezfooli, Seyed-Mohsen, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2016.
9. Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. International Conference on Learning Representations, 2015
10. Krizhevsky, Alex, and Geoffrey Hinton. Learning multiple layers of features from tiny images, 2009.
11. Gong, Chengyue, et al. MaxUp: A Simple Way to Improve Generalization of Neural Network Training. arXiv preprint arXiv:2002.09024, 2020.
12. Pang, Tianyu, Kun Xu, and Jun Zhu. «Mixup inference: Better exploiting mixup to defend adversarial attacks». arXiv preprint arXiv:1909.11515, 2019.

13. Y. LeCun, L. Bottou, Y. Bengio and P. Haffner: Gradient-Based Learning Applied to Document Recognition, Proceedings of the IEEE, 86(11):2278–2324, 1998
14. Carlini, Nicholas, and David Wagner. Towards evaluating the robustness of neural networks. 2017 IEEE symposium on security and privacy (sp). IEEE, 2017.
15. Madry, Aleksander, et al. Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083, 2017.

Искусственный интеллект для образования. Адаптивная система обучения

Кириллов Павел Андреевич, курсант

Краснодарское высшее военное училище имени генерала армии С. М. Штеменко

Технологии, совершив огромный скачок в развитии за достаточно короткое время, уже смогли существенно облегчить жизнь человеку.

Создание сильного ИИ для включения его в образовательный процесс является актуальной задачей, поскольку технические разработки в соответствующих сферах науки привели к значительным результатам.

Но развитие ИИ ведет к замене человека в различных сферах деятельности.

Нам же следует создать ИИ, как помощника человека в его развитии. ИИ, способный контактировать с человеком, понимать его и давать решение проблемы.

Для этого ИИ должен видеть, слышать и понимать чувства человека.

В период возрастания объема информации и знаний, обучение должно быть личностно-ориентированным, учитывающим способности, потребности, особенности обучающегося.

Ключевые слова: искусственный интеллект, ИИ, методы и модели исследований, машинное обучение, обработка естественного языка, представление знаний, машинное творчество, техника, адаптивное обучение, адаптивная система обучения, АСО, образование.

Понятийный аппарат:

Интеллект — способность воспринимать информацию и сохранять её в качестве знания для построения адаптивного поведения в среде или контексте [1]

Искусственный интеллект (ИИ) — это наука и технология разработки интеллектуальных машин и компьютерных программ, возможность выполнять творческую деятельность, которая обычно считается возможностью только человека. [2]

Адаптивное обучение — образовательный метод, который использует компьютерные алгоритмы, чтобы организовать взаимодействие с обучаемым и предоставить настроенные ресурсы и учебные мероприятия для удовлетворения уникальных потребностей каждого обучающегося. [3]

Техника — обобщающее наименование сложных устройств, механизмов, систем. [4]

Образование — единый целенаправленный процесс воспитания и обучения, а также совокупность приобретаемых знаний, умений, навыков, ценностных установок, функций, опыта деятельности и компетенций. [5]

Смотря на современное образование, на то, как обучаются дети, а также на проблемы, с которыми столкнулось общество в 2020 году, когда общество не готово было к обучению без прямого контакта с преподавателем, мало кто будет отрицать явные недостатки традиционного способа обучения. Поэтому как никогда важно рассмотреть новые способы получения образования и способы обучения

в целом. Одним из многообещающих, и продуктивных является адаптивное обучение [6]

При адаптивном обучении происходит взаимодействие информационных и педагогических технологий, что обеспечивает интерактивную возможность обучения с учетом индивидуальных способностей обучающегося. С развитием информационных технологий электронные средства обучения имеют все больше возможностей для реализации такого обучения на практике. [7; 8; 9]

Проблемы традиционного обучения, затрагивающие детей: преподаватель не может охватить всю детскую аудиторию, с которой он работает, как и не может подставиться лично под каждого ученика, что вынуждает его работать только с теми, кто проявляет способности к обучению, тем самым отстающие не получают должный охват знаний, что приводит к потере успеваемости детей в целом.

Помимо этого, существует множество других проблем, в основном проявляющихся на занятии:

- при работе с отдельными учащимися остальные ученики в работу почти не включены;
- ученики бездельничают на уроке, притворяясь, что слушают;
- плохая обратная связь ученика с учителем;
- нет возможности адаптироваться к индивидуальным особенностям всех учащихся во время урока;
- оценка учеников субъективна и бывает не связана с реальным уровнем знаний;

— нежелание преподавателей работать с отстающими учениками;

При использовании адаптивного ИИ проблем, выявленных при традиционном обучении, не возникнет, ведь он может подстраивать способ обучения под абсолютно любого человека индивидуально. [10] Пример реализации такого обучения представлен сервисом Knewton. Его методы обучения людей намного продуктивнее обычного образования. [11;12;13;14]

Но у него есть недостатки, которые не позволяют такому способу раскрыться и полностью заменить стандартное образование. Основная проблема Knewton в том, что он не готов реагировать на психологическое состояние ребенка. [15]

Далее следует научно-теоретическая модель успешной реализации адаптивного обучения с использованием искусственного интеллекта.

Решением такой проблемы является загрузка в базу данных ИИ психологические данные о человеке и реакцию на их проявление. ИИ должен уметь определять по манере написания, по ответам, по внешним и речевым признакам состояние ребенка. Если у него возникли сомнения по одному из признаков, то ИИ начинает задавать наводящие вопросы ребенку и наблюдать его реакцию. Таким образом он может подправить психологический баланс ребенка. Например, разный контекст применения эмоций. Улыбка может быть наигранной. Поэтому помимо внешних данных, нужно знать данные организма. Т. е. при выявлении подозрений в пошатнувшемся психологическом состоянии ИИ задает наводящие вопросы, либо проводит тестирование, после которого смотрит на результаты. При выявлении, он предлагает задания, которые смогут улучшить настроение либо вовсе отойти на некоторое время от учебы. Оценивая критичность, такой ИИ либо развлекает ребенка, либо освобождает от занятий. В случае обнаружения симуляции накладывается штраф. Штраф может быть в виде лишения возможных подарков, утрата очков опыта и т. п.

Но при этом невозможно такое обучения без контроля третьего лица, который в случае непредвиденных обстоятельств сможет среагировать на них. Для такой задачи должен быть назначен определенный человек — контроллер. Например, в случае отклонения психологического состояния обучаемого об этом уведомляется контроллер с рекомендациями по устранению.

До начала обучения ИИ должен выявить сильные и слабые стороны ребенка. Узнав уровень обученности, ИИ не выдает ему готовую программу обучения, а создает новую, подстраивая под его способности и корректируя, в зависимости от успеваемости.

Для вовлечения обучаемого в процесс образования, ИИ целесообразнее и эффективнее подавать задания в игровой форме. Дети произвольно запоминают учебный материал, вызывающий у них интерес, преподнесенный в игровой форме, связанный с яркими наглядными пособиями или образами-воспоминаниями.

Следуя современным тенденциям и наблюдениям за интересами детей, прогресс обучения, выражающийся в накоплении опыта, получения новых уровней и открытие достижений также способствует погружению обучаемого в процесс получения новых знаний. ИИ узнает, что любит обучаемый и поощряет его бонусами в том виде, в котором ученик будет счастлив их получить. Необходимо отметить, что сама модель АСО при правильном ее использовании всегда гарантирует успешное формирование коммуникативной компетенции. Принцип рейтинга позволяет отойти от традиционной отметочной системы контроля. Ребенок продвигается к результату, набирая определенный рейтинг. С одной стороны — это элемент игры и соревнования, а с другой — это отличный измеритель всех учебных действий.

Но все обучение не должно превращаться в игру. Т. е. чем взрослее обучаемый, тем больше сложных и неигровых задач. Должна присутствовать подготовка человека к стрессовым ситуациям, реализуемая сложными задачами, вопросами, требующими мгновенного решения, в которых «неудача» приведет к потере какого-либо бонуса.

Человек не должен заикливаться на общении с искусственным интеллектом. В процессе обучения обучающему должны быть приобщены социальные навыки. Для этого нужна работа между группами, при которой решение задачи будет зависеть от взаимодействия групп между собой. В условиях АСО ИИ способен довести обучающихся до продуктивного (творческого) этапа в формировании коммуникативной компетенции. Личностное общение — это всегда творческое общение, которое может пронизывать любое учебное задание.

Принцип коллективного взаимодействия предполагает выполнение таких заданий, которые моделируют реальную коммуникацию в парах, а также в группах более двух человек.

В процессе обучения должна присутствовать возможность человека самому исправлять свои ошибки, для формирования в обучаемом критического мышления.

Для глубокого взаимодействия с человеком ИИ требуется возможность распознавания внешности, речи и чувств человека. И реализация таких возможностей представляна в любой сфере вокруг нас: приложения в Play Market и App Store, «Алиса» от «Яндекс» и другие голосовые помощники, электронные ценники shelfPoint, которые собирают данные об эмоциях покупателей, YouFirst от YouTube, который регистрирует реакцию зрителей на видео, самообучающаяся нейронная сеть Giraffe, которая через 72 часа обучения смогла играть лучше 97,8 процентов игроков турнира. [16;17;18;19]

В настоящее время боязнь уйти от стандартного метода образования только тормозит развитие общества. Гибкость и доступность АСО возможно обеспечить благодаря свободному распространению электронных средств доступа к информации. А программа обучения подбирается в зависимости от цели обучения, поэтому такое адаптивное образование подходит не только для детей школьного возраста, но и людей любого поколения.

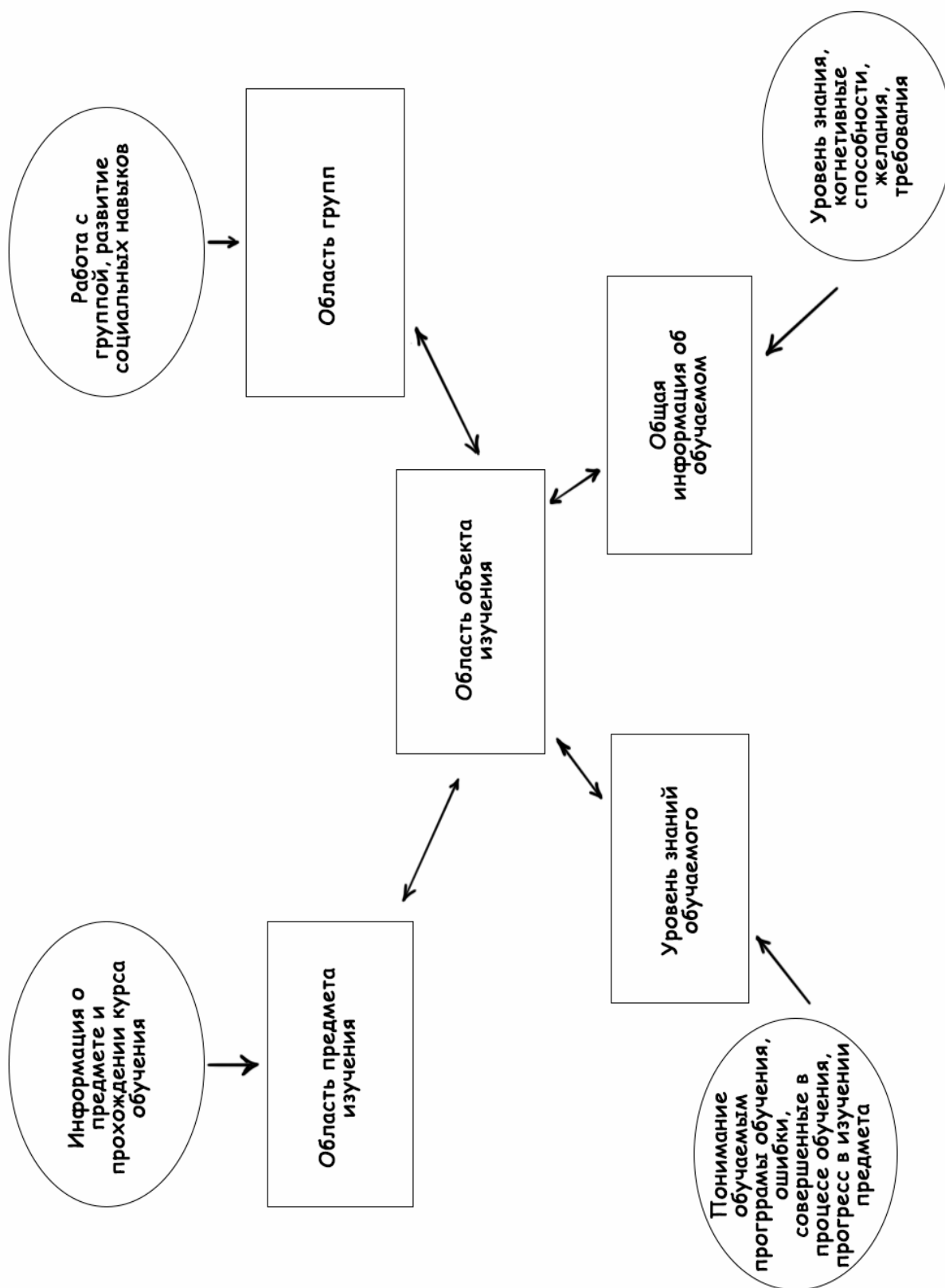


Схема:

Область предмета изучения. Содержит информацию об изучаемом предмете и прохождении курса обучения. Система предоставления данных пользователю должна в реальном времени изменять программу обучения под пользователя, адаптируясь к его требованиям.

Область объекта обучения. Данная область содержит все данные об обучающем и изменяется со временем: прогресс изучения предмета, качество усвоения, поведение, уровень знаний. Область ОО содержит не только общую информацию о студенте, но также отслеживает действия студента в процессе обучения в рамках адаптивного обучения.

В область ОО содержится информация двух типов: конкретный уровень знаний обучаемого (понимание предмета или отдельных его разделов, ошибки, которые объект совершил в процессе изучения, прогресс обучаемого в изучении предмета) и общая информация об обучаемом (включает цели обучения, когнитивные способности обучаемого, например способность рассуждать, выстраивать ассоциации, его мотивация, начальные знания и опыт, предпочтения)

Область групп. Включает работу с группой, развитие социальных навыков.

Литература:

1. Всё, что вам нужно знать об ИИ — за несколько минут // URL: <https://habr.com/ru/post/416889/>
2. Макк, А. А. Способен ли искусственный интеллект превзойти интеллект человека? // URL: <https://elibrary.ru/item.asp?id=29167840/>
3. Адаптивное обучение — Adaptive learning // URL: https://ru.qwe.wiki/wiki/Adaptive_learning/
4. Википедия. Техника. // URL: <https://ru.wikipedia.org/wiki/Техника>
5. Википедия. Образование. // URL: <https://ru.wikipedia.org/wiki/Образование>
6. Ольга Тараканова. Что такое адаптивное образование и почему оно изменит наши школы, университеты и даже онлайн-курсы // НОЖ — 2018 г. // URL: <https://knife.media/adaptive-learning/>
7. Царев, Р. Ю., Тынченко С. В., Гриценко С. Н. Адаптивное обучение с использованием ресурсов информационно-образовательной среды // Современные проблемы науки и образования. — 2016. — № 5.; URL: <http://science-education.ru/ru/article/view?id=25227>
8. Ольга Корбан. Terra incognita: открываем мир адаптивного обучения. // eTutorium — 2018 г. // URL: <https://etutorium.ru/blog/adaptivnoe-obuchenie-tehnologii-modeli>
9. Адаптивное обучение: коротко о главном. // WebSoft — 2018 г. // URL: <http://blog.websoft.ru/2018/07/blog-post.html>
10. Адаптивное обучение. // Fornit — 2018 г. // URL: https://scorcher.ru/adaptologiya/Adaptivnoye_obucheniye/Adaptivnoye_obucheniye.php
11. Адаптивное обучение, или несколько слов о Knewton. // habr — 2014 г. // URL: <https://habr.com/ru/company/newprolab/blog/244539/>
12. ИИ для личных целей: помощь с образованием, работой и планированием. // habr — 2017 г. // URL: <https://habr.com/ru/company/mailru/blog/409173/>
13. Knewton: адаптивное обучение в действии. // Newtonew — 2014 г. // URL: <https://newtonew.com/tech/knewton-adaptivnoe-obuchenie-v-dejstvii>
14. «Подход к образованию не менялся со времен промышленной революции»: Фернандо Родригес-Вилла об проекте Knewton. // Theory&Practice — 2015 г. // URL: <https://theoryandpractice.ru/posts/11857-knewton>
15. Хороший, плохой, злой: возможен ли эмоциональный искусственный интеллект. // Theory&Practice — 2019 г. // URL: <https://theoryandpractice.ru/posts/17440-khoroshiy-plokhoy-zloy-vozmozhen-li-emotsionalnyy-iskusstvennyy-intellekt>
16. «Алису» научили распознавать владельца по голосу // Популярная механика — 2019 г. URL: <https://www.popmech.ru/technologies/news-508532-alisu-nauchili-raspoznavat-vladelca-po-golosu/>
17. Анатолий Ализар. Нейросеть Giraffe за 72 часа научилась играть в шахматы на уровне международного мастера ФИДЕ // Хабр — 2015 г. // URL: <https://habr.com/ru/post/384253/>
18. Технологии будущего. Hanshow // URL: <http://www.hanshow.ru/solution/>
19. YouFirst: отслеживаем реакцию аудитории на ваши видео с помощью ИИ // LPGenerator — 2018 г. // URL: <https://lpgenerator.ru/blog/2018/11/01/youfirst-otslezhivaem-reakciyu-auditorii-na-vashi-video-s-pomoshyu-ii/>

Перспективы обучения студентов основам автоматического управления с помощью средств разработки на языке CFD

Круглов Роман Михайлович, студент
 Российский университет транспорта (МИИТ) (г. Москва)

В статье рассмотрены вопросы, связанные с актуальностью обучения студентов не связанных на прямую со сферой информационных технологий методам и приемам программирования программируемых логических контроллеров с применением языка CFD. В статье предложено применение среды разработки с низким порогом вхождения, рассмотрен пример реализации порогового регулятора, даны рекомендации по дальнейшему внедрению решений на базе языка CFD в учебный процесс.

Ключевые слова: программируемый логический контроллер, язык функциональных блоковых диаграмм, обучение.

Введение. Процесс цифровизации экономики приводит к изменению социально-экономической парадигмы, в связи с чем актуальным становится вопрос о подготовке кадров способных в полной мере воспользоваться преимуществами её изменения [1]. Современный рынок онлайн-обучения воспринял данные тенденции, одним из наиболее крупных сегментов рынка открытых онлайн курсов является именно сегмент учебных программ в области информационных технологий [2]. Однако, зачастую при выборе языка программирования для обучения акцент смещается в пользу его популярности, таким образом зачастую студенту предлагается изучить один из популярных языков, например Java, C#, C++, Python [3] и другие. Вместе с тем цифровая экономика не ограничивается только сервисной сферой, в которой наиболее популярны вышеуказанные языки, не следует забывать о производящих и обрабатывающих отраслях экономики, а также о транспортной отрасли, в которых активное применение находят программируемые логические контроллеры [4].

При этом зачастую программирование логических контроллеров, организовано в более доступной для обучения

форме, как правило при обучении основам программирования логических контроллеров не требуется изучение основ парадигмы объектно-ориентированного или функционального программирования. Существуют графические языки для программирования микроконтроллеров, в частности язык функциональных блоковых диаграмм (FBD) и близкие к нему по парадигме языки.

Применение графических языков программирования на базе функциональных блоковых диаграмм для обучения автоматическому управлению.

На российском рынке представлены различные микроконтроллеры, программируемые на языке FBD [5], данный язык востребован поскольку достаточно нагляден и прост для освоения профессионалами, имеющими опыт работы с микроконтроллерами, однако для лиц только начинающих обучаться нередко сложность освоения и визуальная составляющая интерфейса, могут снизить мотивацию к обучению. Рассмотрим интерфейс (рисунок 1) одной из популярных систем программирования микроконтроллеров в том числе на языке FBD — CoDeSys [6].

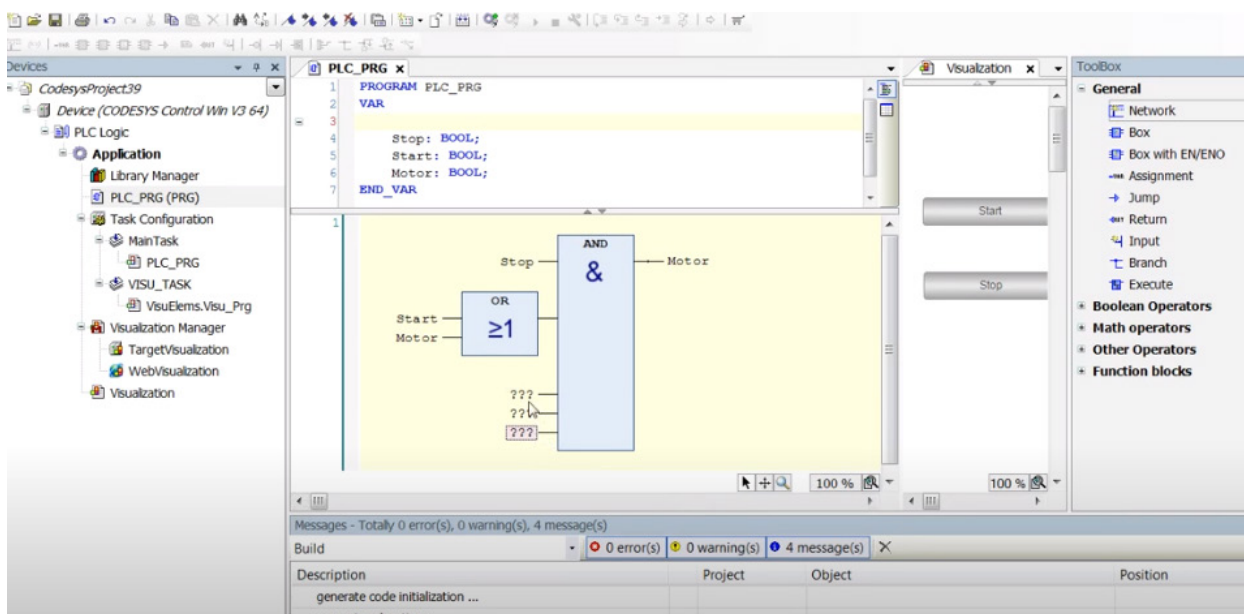


Рис. 1. Фрагмент рабочего окна CoDeSys при работе с программой на языке FBD

Несмотря на свою функциональность интерфейс программы может для неопытного пользователя показаться «перегруженным», поэтому целесообразно перед изучением данного инструмента, ознакомиться с более простой и наглядной средой разработки, основанной на аналогичных принципах работы.

В качестве такой среды разработки в данной статье предлагается рассмотреть программу CannyLab, которая является интегрированной средой разработки на языке Canny Function Diagram (CFD). Язык CFD основан на языке FBD, при его разработке учитывался стандарт МЭК 61131-3 [7]. Внешний вид интерфейса CannyLab показан на рисунке 2.

Несмотря на то, что CannyLab и язык CFD предназначены только для программирования контроллеров CANNY, в программе присутствует режим симулятора, в котором можно провести моделирование распространенных функциональных блоков, достаточных для имитации различных автоматических регуляторов. Также определенным преимуществом данной среды разработки является то, что её не надо устанавливать на компьютер пользователя, достаточно распаковать архив с программой и запустить.exe файл. Данный фактор может быть полезен в условиях ограниченных прав пользователя, в соответствии с различными политиками информационной безопасности учебного заведения.

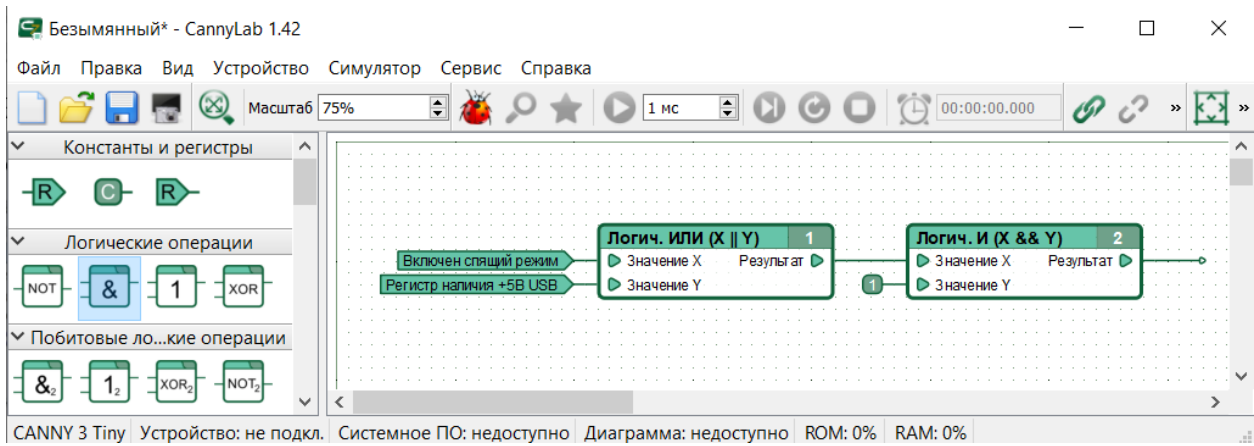


Рис. 2. Фрагмент рабочего окна CannyLab при работе с программой на языке CFD

Рассмотрим пример реализации порогового регулятора на языке CFD в режиме симулятора CannyLab (рисунок 3).

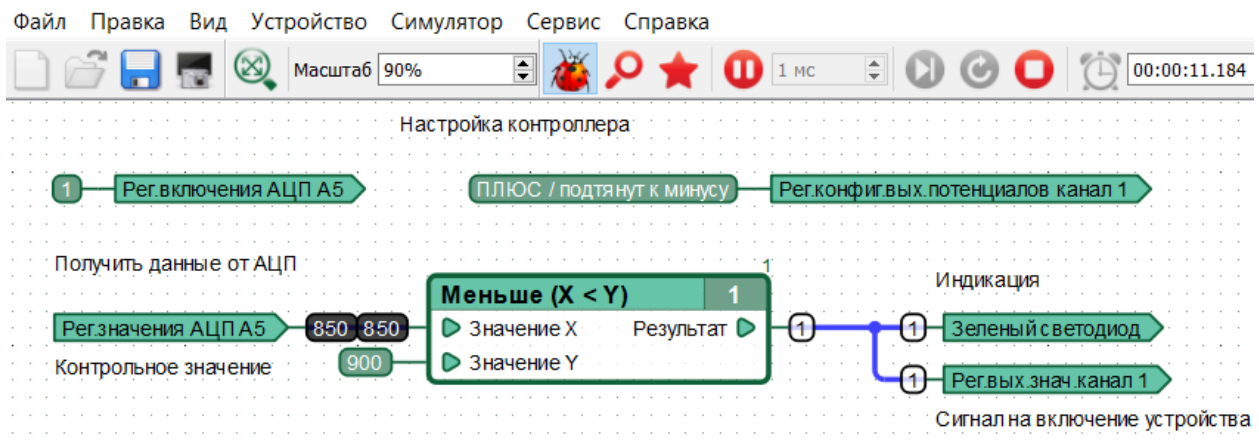


Рис. 3. Пример реализации порогового регулятора в режиме симулятора CannyLab

В данном примере может быть использовано практически любое сочетание датчика и исполнительного устройства, например фоторезистор и светодиодная матрица, включение которой производится при снижении светового потока поступающего на измерительный орган датчика ниже определённого порога.

Дальнейшим развитием практики применения языка CFD в учебном процессе может стать организация различных учебных стендов (макетов) переводящих обучение из исключительно виртуальной плоскости симуляции в физическое воплощение, аналогично примерам учебных стендов для обучения на языке FBD [8]. Учи-

тывая схожесть языка CFD и FBD в дальнейшем возможно будет продолжить обучение именно на языке FBD.

Заключение. В условиях постепенного повышения степени цифровизации экономики РФ актуальной становится задача не только обучения языкам программирования, распространённым в сфере сервисной автома-

тизации, но и языкам программирования, связанным с обеспечением реального сектора экономики, в том числе языку FBD. В случае если студенты имеют низкую способность к освоению языков программирования или низкую мотивацию, целесообразно начать обучение с более наглядного языка CFD.

Литература:

1. Басаев, З. В. Цифровизация экономики: Россия в контексте глобальной трансформации // Мир новой экономики. 2018. № 4. С. 32–38.
2. Семенова, Т. В., Вилкова К. А., Щеглова И. А. Рынок массовых открытых онлайн-курсов: перспективы для России // Вопросы образования. 2018. № 2. с. 173–197.
3. Наумов Родион Владимирович Актуальные языки программирования // Academy. 2016. № 1 (4). с. 49–50
4. Банников Евгений Викторович Использование ПЛК в промышленности // International scientific review. — 2019. № LV. с. 25–28.
5. Лещёв, В. С., Шилин А. А., Светлаков А. А. Современные промышленные контроллеры и их программирование на языке функциональных блочных диаграмм FBD // Доклады ТУСУР. 2007. № 2 (16). с. 62–67
6. Новиков, С. О. Возможности системы программирования CoDeSys при разработке программного обеспечения и проектировании микропроцессорных систем управления // Энергетика. Известия высших учебных заведений и энергетических объединений СНГ. 2009. № 1. с. 27–35
7. Язык CFD. — Текст: электронный // Canny.ru: [сайт]. — URL: https://wiki.canny.ru/index.php?title=%D0%AF%D0%B7%D1%8B%D0%BA_CFD (дата обращения: 21.06.2020).
8. Ежов, В. Д., Дубоделов В. Е., Крышнев Ю. В. Обучающий комплекс для промышленных систем управления // Энергетика. Известия высших учебных заведений и энергетических объединений СНГ. 2013. № 1. с. 17–25.

Разработка алгоритма определения топологии биологической ткани животных и человека по начальному срезу

Можайский Глеб Вадимович, студент;
Мищенко Анастасия Дмитриевна, студент;
Ежов Федор Валерьевич, студент;
Мартынюк Роман Анатольевич, студент;
Кононыхин Иван Александрович, студент
Санкт-Петербургский государственный университет

На сегодняшний день одна из основных причин смертности — рак. Ученые во всем мире борются за прогресс в понимании природы этого заболевания и ищут способы лечения. Диагностика рака производится с помощью гистологического анализа. Анализ основан на методах изучения ткани по ее двумерным срезам. Однако, существует предположение, что причина болезни может быть не в отдельно взятой клетке, а в структуре клеточной ткани. Изучение трёхмерной структуры биологической ткани новая развивающаяся область гистологии. Основная идея — построение множества моделей ткани. В настоящее время не существует удобного программного обеспечения, облегчающего данный процесс. В данной работе в рамках существующего программного обеспечения для трехмерного моделирования тканей животных и человека разработан и реализован алгоритм определения топологии биологической ткани по начальному срезу и создан механизм построения соответствующей ему трёхмерной модели. Разрабатываемое программное обеспечение призвано помочь гистологам в задачах изучения структуры тканей животных и человека.

Ключевые слова: гистология, трехмерное моделирование, клетка, биологическая ткань, топологический примитив.

Введение

По данным Всемирной Организации Здравоохранения рак — одна из основных причин смертности в мире [1]. Ученые борются за прогресс в понимании природы этого заболевания и в поиске его лечения, но единого лекарства,

которое бы подходило для каждого человека все еще не найдено. Точная диагностика данного заболевания производится с помощью гистологического анализа. Процедура заключается в изучении специально подготовленных тон-

чайших срезов тканей под микроскопом, то есть метод основан на анализе двумерной структуры тканей.

Однако существует предположение, что при таких патологиях, как раковые опухоли, видоизменяются не отдельные взятые клетки, а их расположение друг относительно друга в пространстве. Трёхмерный подход к изучению тканей основывается на их экстра- и интерполяции, что позволяет судить о форме и смежности клеток даже на тех уровнях, которые не были исследованы экспериментально. Изучать трёхмерную структуру тканей экспериментальным путем достаточно сложно, поэтому актуальной является задача разработки программного обеспечения, позволяющего моделировать и визуализировать трёхмерное строение тканей по их двумерному срезу.

В данной работе описывается процесс разработки алгоритма определения топологии ткани по начальному срезу путем внедрения новых клеток в имеющуюся топологию клеточного слоя. Данный алгоритм является основой механизма построения трёхмерной модели клеток по начальному двумерному срезу биологической ткани. Разработка производится в рамках существующего программного обеспечения, основанного на теории трёхмерной структуры биологической ткани Савостьянова Г. А. [2], позволяющего строить и исследовать различные модели пространственной организации клеток.

Описание алгоритма

Основной задачей программного обеспечения, основанного на теории трёхмерного строения ткани Савостьянова Г. А., является построение множества допустимых моделей ткани. Существует несколько подходов к этому процессу. В рамках данной работы рассмотрим один из них. Основой данного подхода является преобразование начального среза биологической ткани путем внедрения в его топологию новых клеток. Рассмотрим идею алгоритма внедрения новых клеток в топологию слоя ткани на основе данного подхода.

1. Выбирается в качестве исходной какая-либо двумерная мозаика из 11 вариантов, совпадающих с Платоновыми и Архимедовыми паркетками [3], построенных для однослойных эпителиев (удобно начинать с простых мозаик, то есть из клеток с минимальной смежностью и постепенно переходить к клеткам с большей смежностью).

2. Выбираются точки внедрения новых клеток.

3. Основываясь на принципе сохранения междугранных углов, правиле равномерности изменения клеточных граней и некоторых топологических преобразованиях, постепенно уменьшается площадь клеток какого-либо типа путем изменения размеров и числа их граней.

4. Согласно теории, освобождающееся место будет занято клетками другого типа, при этом площадь пласта не изменяется (такое внедрение может происходить как в вершины, так и в грани клеток).

5. По множеству сечений восстанавливается форма и взаиморасположение клеток в пространстве — строятся

трёхмерные модели, отражающие топологию пласта и геометрию составляющих его клеток.

6. Проводится верификация построенной модели — сравниваются срезы реальных тканей с теоретическими сечениями моделей, то есть ищутся предсказываемые структуры, которые в наибольшей степени соответствуют реальности.

Формализация алгоритма

Каждому срезу ткани можно сопоставить граф, в котором вершины соответствуют клеткам. Если две клетки имеют общее ребро на срезе, то соответствующие вершины соединены ребром. Такой граф — *топологический граф среза* — полностью отражает топологию ткани на данном срезе. При начальном изменении топологии среза, то есть добавлении новых клеток в вершины / ребра, получается новый граф.

Суть алгоритма определения топологии по начальному срезу сводится к поиску в раскрашенном (черные — вершины, которые были изначально, белые — добавлены в процессе изменения топологии) планарном графе подграфов, изоморфных данному (то есть поиск некоторых топологических примитивов) и замены их на другие примитивы.

В данном случае решается не общая задача, потому что имеется конечный набор топологических примитивов (графов), которым надо найти изоморфные подграфы. В каждом конкретном случае заранее известно, какие именно подграфы надо искать (это зависит от изменения топологии — куда именно была проведена вставка новых клеток: в вершины / в щели). Для поиска в графе изоморфных топологическим примитивам подграфов существуют определённые полиномиальные алгоритмы. Рассмотрим один из них.

Алгоритм поиска топологического примитива № 1:

1. Перебираем в графе ребра, соединяющие две черные вершины. Найденное ребро — e .

2. Сортируем инцидентные черным вершинам ребра по углу α .

3. Две черные вершины инцидентные e назовём v_1, v_2 . Находим самое левое и правое ребра относительно e , если: — они инцидентны v_1, v_2 соответственно и при этом инцидентны одной вершине (пусть v_3)

— они инцидентны v_2, v_1 соответственно и при этом инцидентны одной вершине (пусть v_4)

— То подграф, образуемый вершинами v_1, v_2, v_3, v_4 изоморфен графу, соответствующему топологическому примитиву № 1.

4. Ребро (v_1, v_2) может быть заменено в исходном графе на ребро (v_3, v_4) , то есть удаляем первое, вставляем второе.

5. Повторяем шаги 2–4 для всех найденных ребер в шаге 1.

Временная оценка алгоритма:

1. Перебор вершин — $O(N)$.

2. Сортировка вершин $O(1)$, так как в реальных тканях клетка редко соприкасается более чем с 12 клет-

ками (в общем случае для каждого ребра сложность сортировки $O(N^2)$, если предположить, что каждая вершина может быть инцидента N ребрам).

Следовательно, временная сложность алгоритма в общем случае — $O(N^3)$, в нашем — $O(N)$.

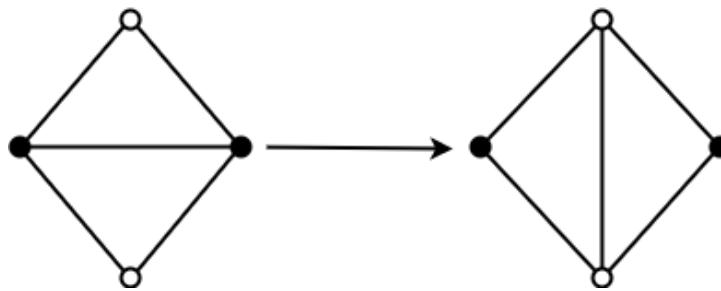


Рис. 1. Изменение топологии согласно алгоритму поиска топологического примитива № 1

Описание имплементации разработанных алгоритмов

В контексте программы связи между клетками — являются линиями, клетки — точками. Начальная и конечная точка связи соответствует какой-либо клетке в зависимости от топологии. Внедрение клеток пользователем построено таким образом, что нет возможности внедрить клетку в место, где её не может быть исходя из топологии начального среза ткани. В практическом смысле это ограничение не позволяет разрывать клетку изнутри. Далее приведены примеры реализации алгоритмов в программе.

Алгоритм поиска топологического примитива № 1:

1. Для всех внедрённых клеток находим порожденные ими новые связи.
2. Из внедрённых клеток выбираются те, которые связаны общим ребром, существовавшим до внесения изменений в топологию слоя. Это и есть топологический примитив № 1.

3. Координаты внедрённых клеток преобразуются в координаты топологии слоя. Это необходимо, поскольку внедрение происходит кликом мыши в желаемую точку.
4. Среди связей, существовавших до внедрения новых клеток, находятся такие, что находятся внутри найденного топологического примитива и удаляются.
5. Добавляется новая связь внутри топологического примитива для внедрённых клеток.

Алгоритм поиска топологического примитива № 3:

1. Из всех связей пласта выбираются такие, у которых одна из точек совпадает с внедрённой клеткой.
2. На основе найденных связей на шаге 1, выбираются клетки, которые имеют 3 связи из найденных и при этом каждая из связей оканчивается во внедрённой клетке.
3. Находятся связи, соответствующие найденным на шаге 2, клеткам и удаляются.

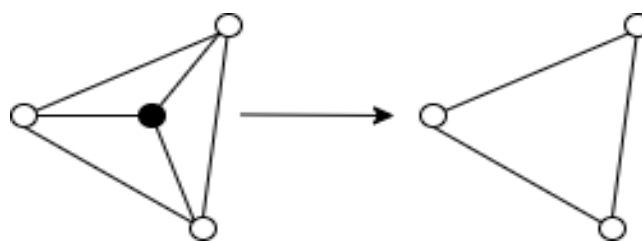


Рис. 2. Изменение топологии согласно алгоритму поиска топологического примитива № 3

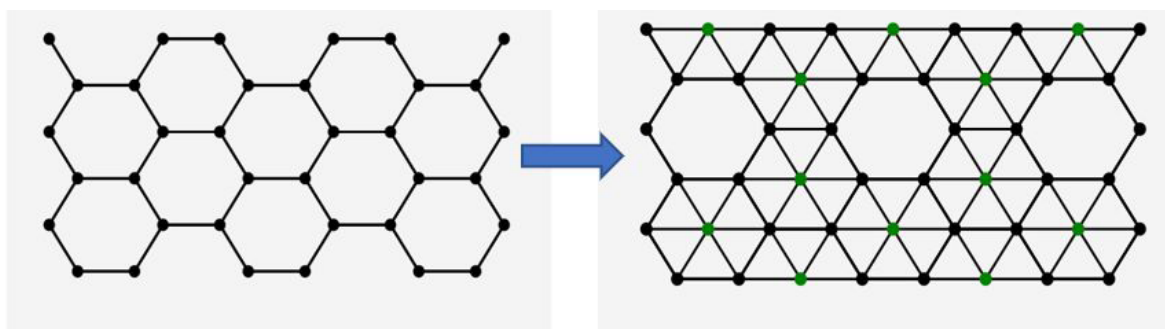


Рис. 3. Процесс внесения точек в начальную топологию пласта (результат работы программы)

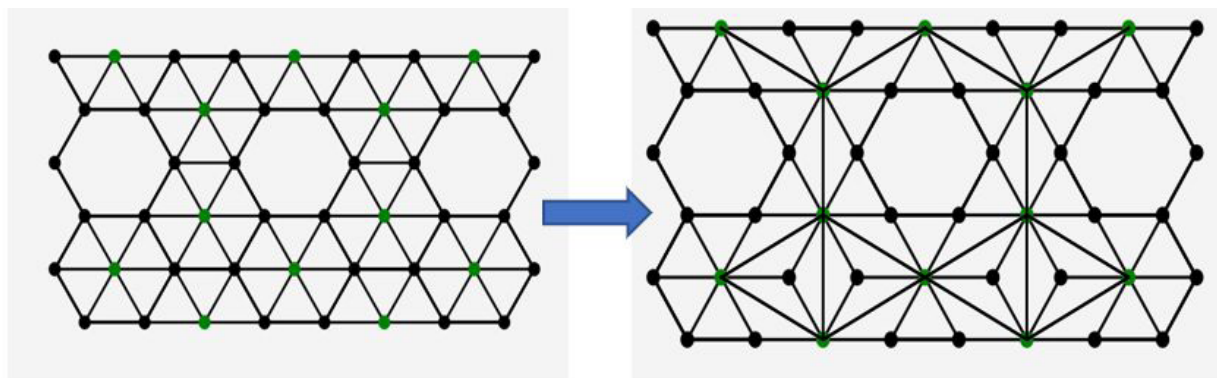


Рис. 4. Преобразование топологического примитива № 1 (результат работы программы)

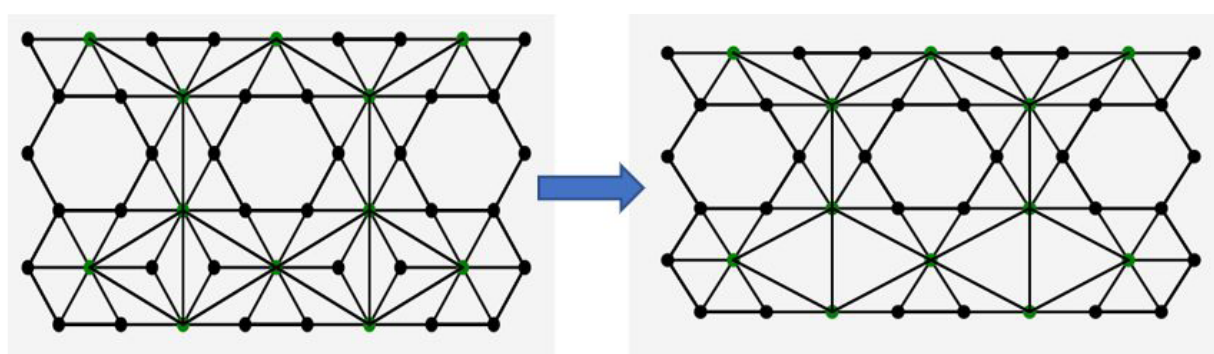


Рис. 5. Преобразование топологического примитива № 3 (результат работы программы)

Выводы

В рамках данной работы разработан алгоритм определения топологии биологической ткани по начальному срезу путем внедрения новых клеток в топологию среза. Данный алгоритм реализован в рамках существующего программного обеспечения, позволяющего строить и ис-

следовать различные модели пространственной организации клеток. Алгоритм является основой для автоматизации процесса трехмерного моделирования структуры ткани путем внедрения новых клеток в топологию начального среза, что является актуальной задачей.

Литература:

1. Cancer. — Текст: электронный // World Health Organization: [сайт]. — URL: <https://www.who.int/news-room/factsheets/detail/cancer> (дата обращения: 28.06.2020).
2. Савостьянов, Г. А. Основы структурной гистологии. Пространственная организация эпителиев / Г. А. Савостьянов. — СПб: Наука, 2005. — 275 с. — Текст: непосредственный.
3. Михайлов, О. В. Одиннадцать правильных паркетов / О. В. Михайлов. — Текст: непосредственный // Квант. — 1979. — № 2. — с. 9–14.

Децентрализованная система управления программным обеспечением для IoT-устройств

Морозов Артемий Дмитриевич, студент;
Мартынов Павел Андреевич, студент;
Чернышова Александра Петровна, студент;
Сиркин Тимур Владимирович, студент
Санкт-Петербургский государственный университет

Ключевые слова: децентрализованная система, управление IoT, IOTA, блокчейн, IPFS, управление программным обеспечением.

Введение

С каждым годом количество IoT устройств растет и как следствия, нужны механизмы для управления этими устройствами. Один из таких важных механизмов — это управление программным обеспечением. Можно выделить централизованный и децентрализованный подход для решения задачи. Особый интерес представляет децентрализованный подход. Он может решить такие проблемы централизованного подхода, как:

1. конфиденциальность
2. безопасность
3. доверие
4. единая точка отказа

Для разработки системы управления программного обеспечения необходимо выбрать распределенную платформу, на которой она будет основана. Все взаимодействия между клиентами и устройствами должны происходить только через нее. Это обеспечит прозрачность всех взаимодействий, а также гарантию того, что никакие данные не могут быть подделаны. Кроме того, необходимо решить проблему распространения обновлений. Совершенно очевидно, что предавать файла внутри распределенного реестра — плохой подход, так как это может увеличить нагрузку на сеть. Поэтому в данной работе предлагается хранить обновления внутри распределенной файловой системы, а в реестр записывать лишь хэш этого файла. Таким образом становится невозможно подменить файл, так как это повлечет изменение хэша, а записи внутри распределенного реестра невозможно изменить.

Компоненты системы

В данной работе была выбрана платформа IOTA по нескольким причинам.

В отличие от блокчейна, в котором транзакции объединяются в блоки, а затем эти блоки верифицируются и добавляются в общую цепочку, в IOTA же используется ориентированный ациклический граф, называемый Tangle. Каждая транзакция должна подтвердить 2 предыдущие. Именно поэтому данная платформа обладает высоким показателем количества транзакция в секунду, так как чем больше транзакций добавляется, тем больше подтверждаются. 4.6 для биткоина, 15 для ethereum, 800 IOTA.

Также важным фактором является то, что транзакции в IOTA бесплатные и как следствие не нужно платить ко-

миссию майнерам, что актуально для сети Интернета Вещей.

Отдельно стоит сказать про важную особенность IOTA — MAM канал. Это надежный и зашифрованный протокол передачи данных, работающий поверх транзакций. MAM канал можно сравнить с каналом в СМИ. Канал может быть публичными, на которой могут подписаться все желающие и ограниченным, для доступа к которому нужен ключ

В качестве распределенной файловой системы был выбран IPFS. В данной файловой системы для каждого файла генерируется уникальный хэш. Благодаря этому возможна адресация на основе контента, а не местоположения. Иными словами, мы можем быть уверенными в том, что по данному хэшу будет доступен именно запрашиваемый файл. Его подмена невозможна. Так как изменение файла повлечет изменение хэша.

Общая архитектура

Задачу управление программным обеспечением можно разбить на 2 пункта:

1. Доставка обновления
2. Применение обновления

Доставка обновлений включает в себя взаимодействие, например, с распределенной платформой, этот пункт отвечает за сетевую часть работы системы.

Применение же является в большей степени локальной задачей каждого устройства.

На рисунке 1 приведена общая архитектура системы управления программным обеспечением.

IoT devices — это устройства Интернета вещей. Каждое такое устройство представляет собой контроллер(например, таким контроллером может быть Raspberry PI) и какое-то количество датчиков. Будем говорить, что все датчики, одной и той же модели принадлежат одному классу.

Loader — это точка взаимодействия пользователя с системой. Компонент Loader написан на NodeJS и предоставляет пользователю интерфейс командной строки. Пользователь может указать 2 основных параметра — это имя класса, для которого будет послано обновление и файл самого обновления.

Как говорилось выше, в IOTA есть защитный механизм передачи зашифрованных сообщений MAM. Для каждого класса существует свой ограниченный MAM канал, пароль

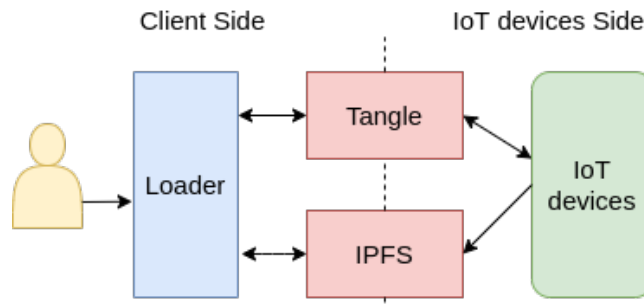


Рис. 1. Общая схема архитектуры

от которого известен устройствам, у которых есть этот класс. Также существует публичный канал, на который подписаны все устройства, в этот канал приходит уведомление о том, что есть обновление для указанного класса.

Отправка обновления

На рисунке 2 приведена схема отправки обновления. Пользователь загружает в IPFS файл обновления, стоит

отметить, что загружается не сам файл, а diff — разница между старой и новой версией файла. Это сделано для того, чтобы передавать файлы меньших размеров. IPFS возвращает хэш этого файла. Далее публикуется сообщение в публичный канал о том, что есть обновления для указанного класса. Затем в ограниченный канал класса отправляется хэш файла.

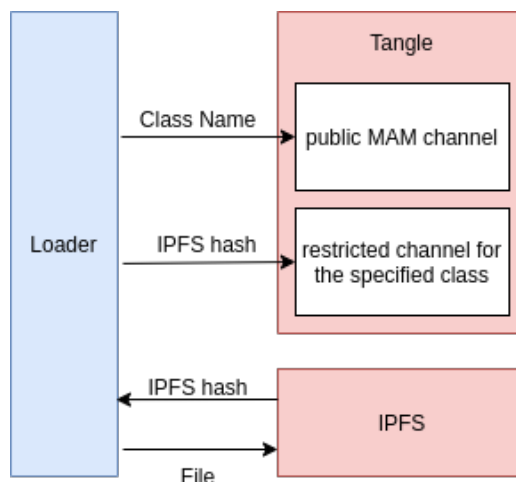


Рис. 2. Схема отправки

Таким образом, мы можем быть уверены в сохранности и надежности передачи, ввиду 2 факторов:

1. Любое изменение файла повлечет за собой изменение хэш значения этого файла в распределенной файловой системе IPFS.
2. Сообщение в MAM канале нельзя изменить

Получение обновления

Как только устройство читает сообщения из публичного канала о том, что есть обновление для класса, который есть у этого устройства, то оно из IPFS получает этот файл. Затем останавливает процесс, который использует этот файл. Обновляет файл, считает хэш сумму получившегося файла и отправляет в Tangle информацию о статусе применения обновления и получившийся хэш. Это нужно для того, чтобы пользователь мог убедиться в корректности обновления. В конце устройство перезапускает процесс.

Стоит отметить, что так как сообщения добавляются в Tangle, то устройство может прочитать его в любой мо-

мент времени, то есть если устройство по каким-то причинам потеряло соединение с сетью, то как только соединение восстановится, устройство сможет прочитать все сообщения, которые были отправлены в момент его отсутствия в сети и тем самым не пропустит обновление, если обновление было.

Тестирование

Применения, как уже говорилось выше, в большей степени является локальной задачей каждого устройства, так как оно может зависеть от характеристик отдельного устройства. А механизм доставки является более общим, именно этот механизм обеспечивает масштабируемость системы.

Тестирование состояло в следующем: были взяты 3, 5 и 10 устройств. На устройства отправлялись обновления. Время измерялось от отправки в public канал сообщения о наличии обновления до принятия файла самым последним устройством, то есть, например, если есть 3

устройства, а какое-то из них получило файл позже всех, то конечным временем считается принятие именно этим устройством. Помимо увеличения числа устройств также увеличивался размер файла обновления

На графике результатов теста (рис. 3) видна тенденция роста времени при увеличении размера файла, что явля-

ется естественным, поскольку увеличивается время на получение файла. Особое внимание стоит уделить тому, что при одинаковых размерах файла, время, потраченное на доставку обновления, не коррелирует с количеством устройств, на которые отправляется обновление. Из этого можно сделать вывод о масштабируемости системы.

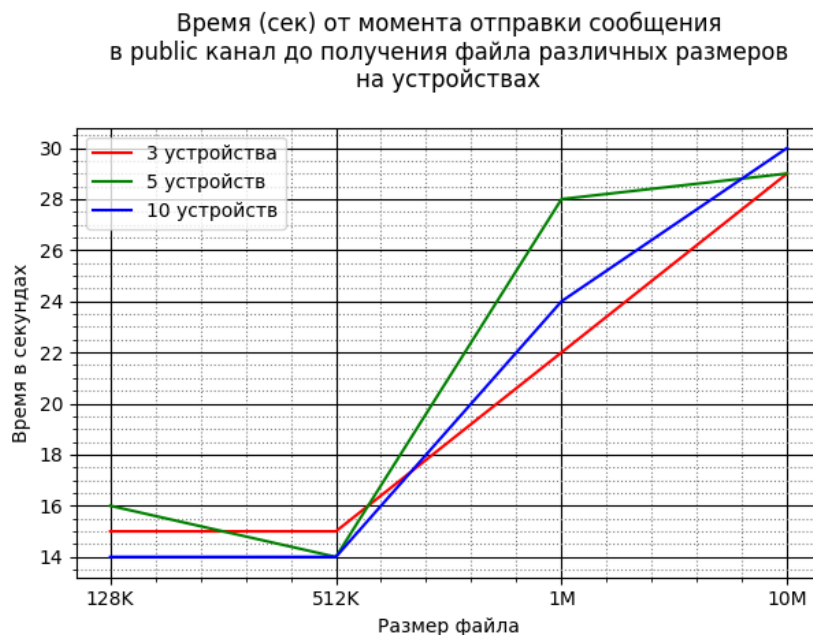


Рис. 3. Результаты тестирования

Литература:

1. Sabrina Sicari, Alessandra Rizzardi, Luigi Alfredo Grieco and Alberto Coen-Porisini Security privacy and trust in Internet of Things: The road ahead. Computer networks, vol. 76, pp. 146–164, 2015.
2. Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, Moussa Ayyash Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications.: IEEE Communications Surveys & Tutorials (Volume: 17, Issue: 4, Fourthquarter 2015)
3. Lee, J. Patch Transporter: Incentivized, Decentralized Software Patch System for WSN and IoT Environments. Sensors 2018, 18, 574–608.
4. Nachiket Tapas, Yechiav Yitzchak, Francesco Longo, Antonio Puliafito, Asaf Shabtai P 4 UIoT: Pay-Per-Piece Patch Update Delivery for IoT Using Gradual Release Sensors (Basel). 2020 Apr 10;20(7):2156.

Разработка программного обеспечения для децентрализованной идентификации IoT-устройств в сетях

Мартынов Павел Андреевич, студент;
Морозов Артемий Дмитриевич, студент;
Сиркин Тимур Владимирович, студент;
Чернышова Александра Петровна, студент
Санкт-Петербургский государственный университет

Ключевые слова: IoT, DID, децентрализованная идентификация, IOTA

Введение

Устройства Интернета Вещей (Internet of Things, IoT) уже не первый год является перспективным направлением. Для повсеместной интеграции IoT технологий в нашу жизнь необходимо внедрить протоколы, стандарты и решить некоторые проблемы, связанные со спецификой этой области. Одной из таких проблем является идентификация устройств. Раз концепция IoT предполагает общение между гаджетами, значит необходимо как то определять, кто посылает данные и уметь заявлять о себе при попытке поделиться с кем то информацией. О идентификации IoT устройств задумались относительно недавно, а решения, предложенные различными стартапами и корпорациями, все еще находятся в разработке.

Особенности IoT устройств

Особенностями устройств Интернета вещей в общем случае являются их мобильность, маломощность, небольшой объем памяти и, вероятно, нестабильные каналы связи. Эти аспекты надо учитывать при разработке ПО, которое будет использоваться в этой сфере.

Топология IoT сети

Так как наличие единой точки отказа сети неприемлемо в 21 веке в IoT, как и в любой современной технологии, наибольшее распространение имеют распределенные сети (системы, где все узлы являются равноправными и решения принимаются совместно). В таких сетях у узлов есть полный контроль над данными, они легко масштабируются, устойчивы к взломам и падениям отдельных узлов. Однако в такой топологии есть свои минусы: много одинаковой работы, проделываемой каждым узлом, и для корректной работы необходима достаточно большая сеть.

SSI. Само-суверенная идентификация

Self-sovereign identity (SSI) — это понятие является ключевым для работы, оно обозначает концепцию, при которой пользователь, создавая свои учетные, данные имеет полный контроль над ними и сам решает кому и в каком объеме раскрывать эти данные.

Распределенные системы идентификации

uPort — система децентрализованной идентификации, основанная на Ethereum, позволяющая создавать представление личности, взаимодействующее посредством смарт-контрактов, и делиться информацией об атрибутах этой личности в сети.

Существует также решение на базе Hyperledger Fabric (проект от Linux Foundation использует модульную структуру для построения блокчейна, сконструированную под отдельные нужды пользователя) fabric-iot. В этой системе сеть заранее настраивается администратором, создаются каналы и смарт-контракты. Далее, это все взаимодействует между собой, позволяя устройствам получать доступ к различным сервисам, согласно установленной политике доступа.

Еще одно решение — это open-source система децентрализованной идентификации Sovrin. Sovrin является публичным блокчейном, но только доверенные институты, такие как банки, университеты, гос. организации, могут управлять узлами. Таким образом создается доверие к выданным сертификатам.

Decentralized Identifier

DID — глобальный уникальный идентификатор, являющийся криптографически безопасным способом идентификации, не требующий централизованного органа, так как используется в децентрализованных сетях. Эта концепция разработана W3C. DID — состоит из самого DID идентификатора и DID документа, связанного с ним.

Документ DID — документ с метаданными, включающий в себя сам DID. Эти данные представлены в виде JSON.

В DID документе могут содержаться следующие данные: timestamp (отметка времени создания), криптографическое подтверждение валидности DID документа, список требуемых публичных ключей, список способов использования DID для аутентификации, список сервисов для которых DID может быть использован

Учетные данные не стоит хранить в распределенном реестре. Это небезопасно. Провайдеры персональных данных должны публиковать в сеть только хэш данных и пересылать. По хэшу легко проверить подлинность данных, не раскрывая их.

Реализация

На основе анализа существующих решений были сформулированы следующие требования к системе.

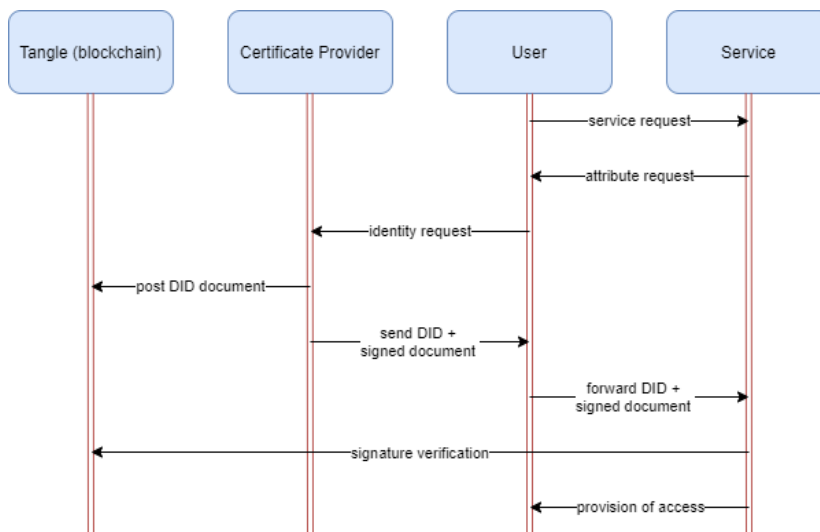
Основные требования к системе идентификации:

1. Полное соблюдение концепции SSI (само-суверенной идентичности).
2. Минимальное раскрытие информации. Сервисы должны запрашивать разглашение только минимального набора данных о пользователе/устройстве.

3. Должна существовать возможность раскрытие информации как публично (всем сервисам в сети), так и группе сервисов или одному конкретному сервису.

В качестве платформы была выбрана IOTA — open-source технология распределенного реестра, ориентированная на IoT устройства. Она отличается от привычной

блокчейн технологии отсутствием самих блоков и цепей этих блоков. Вместо этого все транзакции поступают в сеть, подтверждая 2 другие транзакции, не объединяясь в блок. В ходе этого процесса образуется ациклический граф (DAG), называемый Tangle.



Механизм работы разработанной системы.

1. Устройство обращается к сервису за предоставлением услуги
2. Сервис запрашивает идентификационные данные устройства (пользователя)
3. Пользователь создает заявку у провайдера сертификатов для создания идентификационной информации и соответствующего DID
4. Если CP (провайдер сертификатов) одобряет эту заявку, тогда он отправляет сообщение с подтвержденным утверждением, подписанное своим закрытым ключом, пользователю и публикует соответствующий DID документ в сеть
5. Далее, пользователь, ссылаясь на DID документ, проходит аутентификацию, описанную в нем, и предоставляя подтвержденное утверждение, полученное

от эмитента, которое на целостность может проверить сервис, используя открытый ключ эмитента из DID документа

6. Сервис проверяет всю информацию и делает вывод о соответствии пользователя политике аутентификации и, если все верно, предоставляет пользователю доступ к ресурсам и услугам сервиса.

Для возможности выбора круга участников сети, которым происходит раскрытие информации, использовались МАМ каналы, это настраиваемые каналы транзакций в IOTA, напоминающие мультитевещание в обычных сетях.

Разработанная система получилась безопасной, прозрачной и масштабируемой. В ней есть множество перспектив для расширения. таких как: внедрения систем отзыва лицензий и системы рейтинга пользователей для создания доверия в сети.

Литература:

1. Han Liu, Dezhi Han, Dun Li. Fabric-iot: A Blockchain-Based Access Control System in IoT. IEEE Access (8), 2020, pp. 18207–18218.
2. Xiaoyang Zhu, Youakim Badr. Identity Management Systems for the Internet of Things: A Survey Towards Blockchain Solutions Society of Petroleum Engineers, 2019, pp. 18.
3. Paulo, C. Bartolomeu, Emanuel Vieira, Seyed M. Hosseini, Joaquim Ferreira Self-Sovereign Identity: Use-cases, Technologies, and Challenges for Industrial IoT
4. Nataliia Kulabukhova, Andrei Ivashchenko, Iurii Tipikin, Igor Minin. Self-Sovereign Identity for IoT Devices
5. Kriti Chopra, Kunal Gupta, Annu Lambora Future Internet: The Internet of Things-A Literature Review. 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), 2019
6. Официальный веб-сайт uPort. — URL: <https://www.uport.me>
7. moslah hamza, All you need to know about uPort Identity management
8. Pelle Braendgaard, What is a uPort identity?
9. SovrinTM: A Protocol and Token for Self-Sovereign Identity and Decentralized Trust, A White Paper from the Sovrin Foundation, 2018, p. 42

10. W3C, Decentralized Identifiers (DIDs) v1.0
11. Документация к IOTA — URL: <https://docs.iota.org/>
12. Paul Handy, Introducing Masked Authenticated Messaging

Разработка в среде C# программной реализации алгоритма выделения структурных особенностей

Павлов Дмитрий Львович, студент;
 Кашеева Ольга Александровна, студент;
 Иванова Анастасия Анатольевна, студент;
 Скородумова Юлия Владимировна, студент;
 Тарасов Антон Алексеевич, студент
 Санкт-Петербургский государственный университет

В работе описана реализация алгоритма выделения структурных особенностей матриц на языке C#. Представлена архитектура программы. Были проведены вычислительные эксперименты, результаты которых продемонстрированы в виде графиков.

Ключевые слова: разработка приложения, реализация алгоритма, структурные особенности, C#, архитектура приложения, анализ работы программы.

Структурный метод интегрирования систем m уравнений общего вида [1], [2].

$$z'_k = \varphi(x, z_1, z_2, \dots, z_m), \quad k = 1, \dots, m, \quad (1)$$

предполагает, что существует преобразование, приводящее систему (1) к виду (2)-(4)

$$\frac{dy_0}{dx} = f_0(x, y_0, y_1, \dots, y_n) \quad (2)$$

$$\frac{dy_i}{dx} = f_i(x, y_0, \dots, y_{i-1}, y_{l+1}, \dots, y_n), \quad i = 1, \dots, l, \quad (3)$$

$$\frac{dy_j}{dx} = f_j(x, y_0, \dots, y_{j-1}), \quad j = l + 1, \dots, n, \quad (4)$$

$$l \in \{0\} \cup N, \quad n \in \{0\} \cup N, \quad l \leq n, \quad x \in [X_0, X_k] \subset R,$$

$$y_s : [X_0, X_k] \rightarrow R^{r_s}, \quad s = 0, 1, \dots, n, \quad \sum_{s=0}^n r_s = m,$$

$$f_0 : [X_0, X_k] \times R^m \rightarrow R^{r_0},$$

$$f_i : [X_0, X_k] \times R^{m - \sum_{s=i}^l r_s} \rightarrow R^{r_i}, \quad i = 1, \dots, l,$$

$$f_j : [X_0, X_k] \times R^{m - \sum_{s=j}^n r_s} \rightarrow R^{r_j}, \quad j = l + 1, \dots, n,$$

Группы (3), (4) структурно тождественны. Каждое уравнение одной из этих групп занимает определенное место в последовательности уравнений всей группы. Его правая часть не зависит от искомым функций, поведение которых описывается этим и всеми последующими уравнениями этой же группы. Группа уравнений (1) называется <<общей>> и включает в себя все уравнения, не имеющие структурных особенностей, описанных выше.

Алгоритм выделения структурных особенностей предполагает, что существует преобразование, приводящее случайную структурную матрицу к виду блочной нуль-диагональной матрицы. Задача состоит в том, чтобы найти такую перестановку $\pi=(\pi(1), \pi(2), \dots, \pi(m))$, при которой эффект от структурного подхода будет максимизирован.

В данной работе будет рассматриваться программная реализация алгоритма, выделяющего данные структурные особенности, а также анализ влияния параметров исходной структурной матрицы на быстродействие исполняемого кода.

Архитектура

Следующим шагом является разработка архитектуры программы. В работе активно используются списки (List), так как изначально неизвестен размер множеств и списки позволяют эффективно перемещаться по уровням дерева перебора. В качестве стандартов наименования была принята статья [3].

Логика работы алгоритма описана в исполняемом классе Program. Поля, начиная с omega, представлены в виде массива из двух элементов, каждый элемент массива отвечает за свое дерево перебора. Для удобства, каждый пункт алгоритма вынесен в отдельную функцию, названия которых совпадают с наименованием шага алгоритма. Основные поля и методы приведены в Листинге 0.

Листинг 0. Класс Program

```
public class Program
{
public static int n; //размерность матрицы
public static List<WeightCoefficient> weights; //весовые коэффициенты
public static int [] transition; //перестановка
public static int taskType; //тип задачи
private static Matrix A; //исходная структурная матрица
private static int [,] B; //преобразованная структурная матрица
private static List<List<int>> [] omega; //множества Омега
private static OrderedMultitude recordMultitude; //рекордное множество
private static OrderedMultitude orderedMultitude; //упорядоченное множество
private static List<List<CentralElement>> [] supportingMultitudeF; //первое вспомогательное множество
private static List<List<PossibleExtension>> [] supportingMultitudeQ; //второе вспомогательное множество
private static int [] searchTreeLevel; //уровни дерева перебора
private static CentralElement centralElement; //центральный элемент
private static PossibleExtension knotElement; //узловой элемент
private static List<List<PossibleExtension>> [] quantityOfPossibleExtension; //Множество возможных продолжений для первого множества
public static void Main(string [] args){...}
static void FirstStep(){...}
private static void SecondStep(){...}
private static void SecondStepA(){...}
private static void SecondStepA1(){...}
private static void SecondStepA1b(){...}
private static void SecondStepA1c(){...}
private static void SecondStepB(){...}
private static void ThirdStep(){...} //перебор для задачи 1 начинается с этого пункта
private static void FourthStep(){...}
private static void FourthStepA(){...}
private static void FourthStepA0(){...}
private static void FourthStepA1a(){...}
private static void FourthStepA1b(){...}
private static void FourthStepA1c(){...}
```

```
private static void FourthStepB(){...}
private static void FourthStepB1(){...}
private static void FourthStepB1a(){...}
private static void FourthStepB2(){...}
private static void FifthStep(){...}
private static void SixthStep(){...}
private static void OrderFirstMultitude(){...}
private static void OrderSecondMultitude(){...}
}
```

Сущностью, на основе которой строится работа с входными и выходными данными, является класс Matrix. В данном классе существуют методы, определяющие подготовительную работу алгоритма и выполняющие перестановку матрицы. Объекты данного класса необходимы для работы со структурными матрицами. Поля и методы класса Matrix приведены в листинге 1.

Листинг 1. Класс Matrix

```
public class Matrix
{
public static int rows; //число строк
public static int cols; //число столбцов
public static int [,] A; //структурная матрица
public Matrix(int row, int column){...} // конструктор
public int this [int row, int column]{...} //индексатор
public void Fill(){...} //Заполнение матрицы из файла
public int [,] Displacement(int [] pi) //применение перестановки, pi- вектор перестановки
private static List<List<int>> FindE(){...} //горизонтальное структурное множество
private static List<List<int>> FindH(){...} //вертикальное структурное множество
public List<List<int>> FindOmegaZero(){...} //омега (1,0)
public static D FindD(List<int> omega){...} //на вход омега, исходя из него строятся множества
public void Print(){...} //печать матрицы в консоль
}
```

Следуя принципу ООП — декомпозиции [4], были созданы структуры, отвечающие за основные элементы в работе алгоритма, такие как:

- множество D (D);
- элемент множества возможных продолжений (PossibleExtension);
- центральный элемент (CentralElement);
- весовой коэффициент (WeightCoefficient);
- упорядоченное множество (OrderedMultitude).

В каждой структуре существует конструктор. В большинстве структур представлены методы, связанные с поиском нужных элементов. Структуры и их методы представлены в Листинге 2.

Листинг 2. Класс Matrix

```
public struct D
{
public List<int> i;
public List<int> j;
public List<List<int>> index;
public D(List<int> _i, List<int> _j, List<List<int>> _index){...}
}
```

```

public struct PossibleExtension
{
public int i;
public int j;
public int weight;
public List<int> elements;
public PossibleExtension(int _i, int _j, int _weight, List<int> _elements) {...}
public static List<PossibleExtension> FindPossibleExtensions(List<int> omega){...}

public static PossibleExtension FindKnot(List<PossibleExtension> possibleExtensions){...}
}

public struct CentralElement
{
public int index;
public int weight;
public CentralElement(int _index, int _weight){...}
public static CentralElement FindCentral(List<int> newOmega){...}
}

public struct WeightCoefficient
{
public int index;
public int weight;
public WeightCoefficient(int _index, int _weight){...}
}

public struct OrderedMultitude
{
public int weight;
public List<int> elements;
public OrderedMultitude(int _weight, List<int> _elements){...}
}

```

Анализ результатов работы.

После завершения тестирования алгоритма был проведен анализ работы программы на задаче 2. Анализ производился при помощи встроенных инструментов Visual Studio. Затраты ресурсов памяти проверялись при помощи Visual Studio Diagnostic Tools [5], а времени при помощи System.Diagnostics.Stopwatch [6].

В ходе анализа работы программы была выявлена зависимость ресурсов от размерности при фиксированной плотности и от плотности при фиксированной размерности.

Для тестирования зависимости параметров работы программы от плотности были случайным образом сгенерированы по 10 случайных структурных матриц размерности 10×10 с фиксированной плотностью и брались средние результаты затрат памяти и времени. Таким образом была получена данная зависимость.

Для демонстрации результатов была выбрана логарифмическая шкала, из-за удобства отображения в ней больших диапазонов значений величин. Для этого время (t) переведено в мс., а память (w) в Мб, затем данные значения прологарифмированы по основанию 2. Результаты представлены на рис. (1), (2).

Стоит отметить, что самой ресурсозатратной оказалась плотность 0,2. Это связано с тем, что при такой плотности граф является слабосвязным, работа алгоритма не завершается за один проход алгоритма и существует большое количество перспективных перестановок.

При тестировании зависимости параметров работы программы от размерности, при фиксированной плотности были сгенерированы матрицы размерностями от 10 до 25 включительно. Данный процесс был повторен для плотностей с 0,35 до 0,65 включительно с шагом 0,5. В связи с тем, что разброс результатов оказался невелик, результаты представлены без преобразований, время (t) в секундах и память (w) в Мб. Результаты работы продемонстрированы на рис. (3), (4).

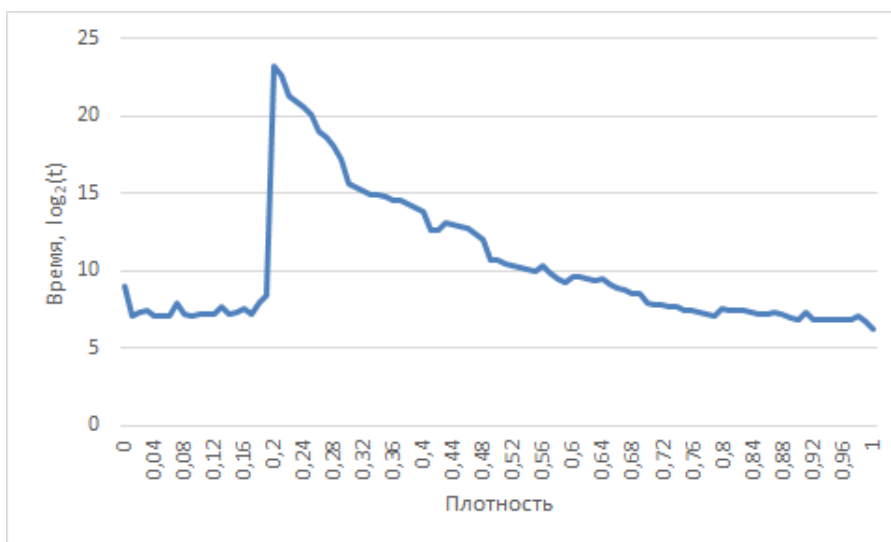


Рис. 1. Зависимость времени от плотности

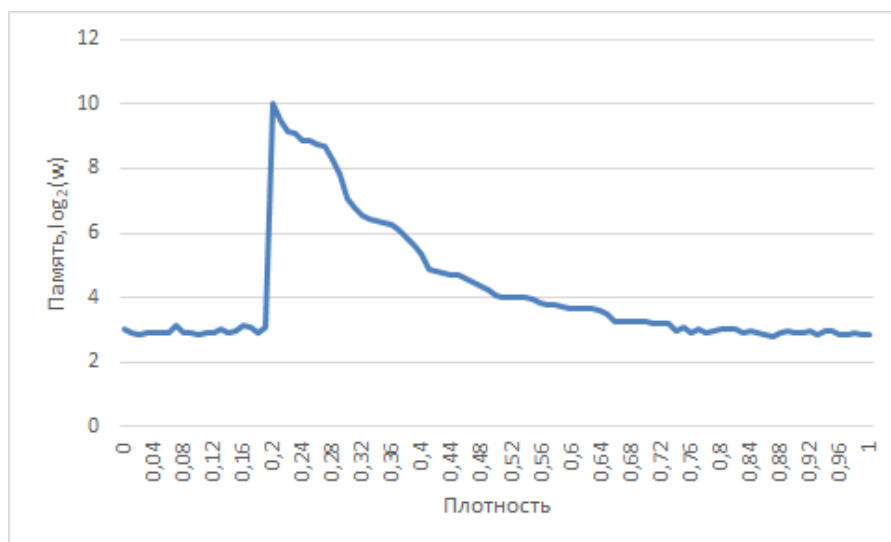


Рис. 2. Зависимость памяти от плотности

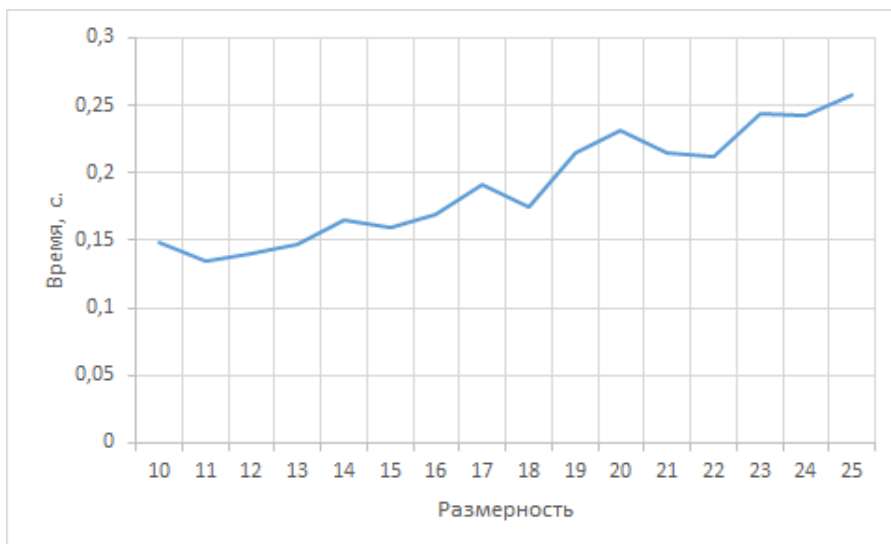


Рис. 3. Зависимость времени от размерности

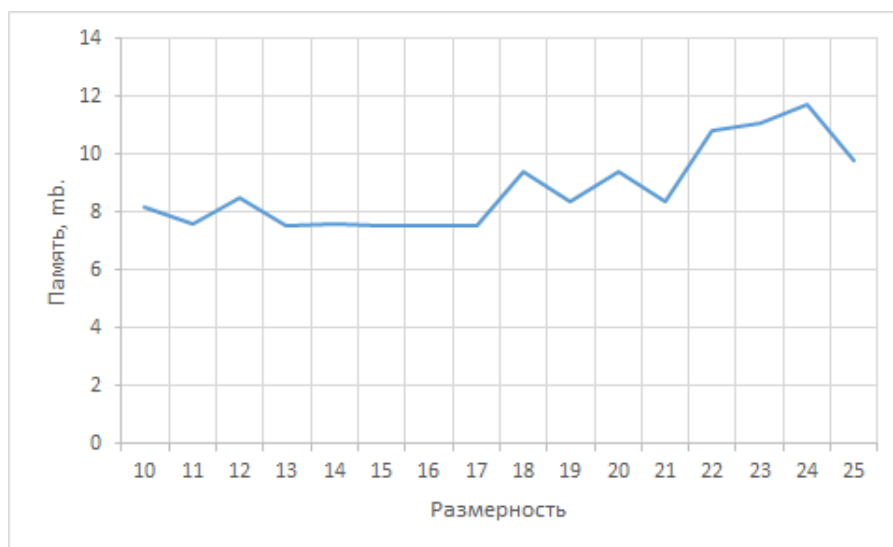


Рис. 4. Зависимость памяти от размерности

Литература:

1. Олемской, И. В. Явный метод типа Рунге — Кутты пятого порядка // ЖВТ. 2005. № 2.
2. Олемской, И. В. Модификация алгоритма выделения структурных особенностей // Вестник СПбГУ. Серия 10. Прикладная математика. Информатика. Процессы управления. 2006. № 2.
3. C#: требования и рекомендации по написанию кода. <https://habr.com/ru/post/26077/>
4. Luca Cardelli, Peter Wegner. On Understanding Types, Data Abstraction, and Polymorphism // ACM Computing Surveys. — New York, USA: ACM, 1985. — Vol. 17, n. 4 — pp 471-522 — ISSN 0360-0300.
5. Measure memory usage in Visual Studio. <https://docs.microsoft.com/en-us/visualstudio/profiling/memory-usage?view=vs-2019>
6. Class Stopwatch. <https://docs.microsoft.com/en-us/dotnet/api/system.diagnostics.stopwatch?view=netcore-3.1>

Большие данные (Big Data) и наука о данных (Data Science): прогрессивные тренды современности

Порошин Иван Евгеньевич, студент магистратуры
Самарский государственный технический университет

В статье рассматриваются основные аспекты, получившего широкое распространение в последнее время, феномена больших данных. Отдельное внимание уделено науке о данных, появление которой явилось следствием геометрического роста информации в современном мире и технологий ее обработки. Также рассмотрены особенности и специфика новой профессии — инженер, работающий с данными. Особый акцент сделан на перспективах и возможностях использования в современном мире машинного обучения.

Ключевые слова: большие данные, наука о данных, машинное обучение, искусственные нейронные сети.

Современный мир в эпоху информатизации и глобализации развивается стремительными темпами, поток данных и их количество увеличивается в геометрической прогрессии. Все это приводит к возникновению нового феномена, получившего в науке и практике название «большие данные».

Большие данные (Big Data) — таким термином принято обозначать совокупность различных методов, технологий

для сбора, обработки неструктурированных и структурированных массивов данных значительного размера. Буквально несколько лет назад Big Data представляло собой инновационное направление, которое использовалось только в сфере высоких технологий. На сегодняшний день большие данные получили широкое развитие во всех сферах и областях человеческой жизни, встречаются и на

ходят свое применение как в коммерческих, так и некоммерческих средах [1, с. 34].

Для сбора и обработки больших данных разработаны специальные программные комплексы и технологии. Более того, создаются и успешно работают компании, которые собирают и продают данные — все это уже стало частью современного бизнеса. Широкое использование Big Data привело к возникновению нового течения — науки о данных (Data Science). В настоящее время многие крупные корпорации применяют Data Science, чтобы произвести хорошее впечатление на своих клиентов, предоставляя персональные предложения. В качестве яркого примера можно привести сервис Google AdSense, который собирает информацию о пользователях для показа контекстной рекламы [2, с. 19].

Очевидно, что наука о данных породила спрос на новые профессии в данной сфере, одной из которых является Data Scientist. Data Scientist — инженер, работающий с данными, обладающий навыками математика, программиста и аналитика. Data Science в последние годы можно назвать профессией, которая находится в мейнстриме науки и техники. Специалисты, которые хорошо владеют математическим аппаратом, ориентируются в сфере программирования — очень востребованы на рынке труда.

Рассматривая более подробно особенности данной профессии, можно отметить следующее. Data Scientist — это специалист, неразрывно работающий с математикой, будь то математическая статистика, теория вероятностей или линейная алгебра. Плюс ко всему, эти знания нужно уметь применять, используя те или иные программные средства — во многом это и отличает математика от специалиста по работе с данными. Data Scientist также может работать в сфере машинного обучения.

Обобщая вышеизложенное, следует отметить, что инженер, работающий с данными, это специалист, который охватывает множество направлений в сфере информационных технологий, в том числе, аналитику, бизнес-аналитику, машинное обучение и многое другое. При этом каждое из этих направлений может представлять собой и отдельную профессиональную сферу Data Science.

В процессе исследования особенностей Big Data и перспектив развития Data Science, особый акцент по мнению автора, необходимо сделать на такой сфере как машинное обучение. Суть машинного обучения сводится к извлечению знаний из данных. Это научная область, которая находится на пересечении математической статистики,

искусственного интеллекта и компьютерных наук, и также известная как прогнозная аналитика или статистическое обучение [3, с. 13].

Машинное обучение с точки зрения программирования — это обучение определенной компьютерной модели каким-либо действиям. В общем смысле это алгоритмы обучения с учителем — системе на вход подается некоторый набор данных и информация о том, что необходимо получить на выходе. Если же в структуре этой модели лежит многослойная искусственная нейронная сеть — речь идет о глубоком машинном обучении. Под искусственной нейронной сетью обычно понимают математическую модель, способную обучаться и решать поставленные перед ней задачи [4, с. 82].

Приведем несколько примеров задач, которые могут быть решены с помощью алгоритмов машинного обучения с учителем:

- определение почтового индекса по рукописным цифрам на конверте;
- нахождение доброкачественности опухоли на основе медицинских изображений;
- обнаружение мошеннической деятельности в сделках по кредитным картам [3, с. 15];
- прогнозирование отказов высокотехнологичного и сложного промышленного оборудования;
- идентификация и распознавание изображений, полученных беспилотными летательными аппаратами.

Подводя итоги проведенного исследования, можно сделать следующие выводы. Появление большого количества данных в цифровом формате, технологий их хранения и вычисления, математических инструментов анализа приводит к трансформации бизнес-процессов. Очевидно, что одновременно со стремительным накоплением информации быстрыми темпами развиваются и технологии их анализа, также возникают новые направления в исследовательской плоскости, такие, например, как наука о данных (Data Science). Data Science представляет собой достаточно обширную сферу, включающую в себя десятки видов деятельности, способную решать огромное количество задач. Именно этот факт делает науку о данных важной для современного мира.

Перспективы дальнейших исследований связаны не только с рисками и сложностями применения Big Data, но и с поиском инсайтов, которые могут создать добавленную ценность для компаний или полезные сервисы для граждан.

Литература:

1. Lee, R. Big Data, Cloud Computing, and Data Science Engineering. — Cham: Springer. — 2020. — 214 p.
2. Силен, Д., Мейсман А., Али М. Основы Data Science и Big Data. Python и наука о данных. — СПб.: Питер. — 2017. — 336 с.
3. Мюллер, А., Гвидо С. Введение в машинное обучение с помощью Python. Руководство для специалистов по работе с данными. — М.: Вильямс. — 2017. — 393 с.
4. Cheng, Q., Li H., Wu Q., Ngan K. Hybrid-Loss Supervision for Deep Neural Network. — Neurocomputing. — 2020. — Vol. 388. — P. 78-89.

Инциденты информационной безопасности 1-го полугодия 2020 года в транспортной отрасли. Введение мер по их предотвращению

Ракинцев Николай Алексеевич, студент
Российский университет транспорта (МИИТ) (г. Москва)

В статье рассматриваются инциденты информационной безопасности, произошедшие в транспортных компаниях, меры по устранению последствий этих инцидентов, а также создание Министерством транспорта РФ единой среды мониторинга защищенности.

Ключевые слова: информационная безопасность, транспортная отрасль, инцидент, утечка данных, информационная система, программное обеспечение.

Мировая транспортная система не стоит на месте и ежегодно увеличивает темпы роста объемов пассажирских и грузовых перевозок. Вместе с этим возрастает объем и ценность информации, обрабатываемой транспортными компаниями, что приводит к необходимости обеспечения достаточно высокого уровня информационной безопасности. Ценность информации, хранящейся на серверах предприятий, побуждает злоумышленников к атакам на информационные системы компаний с целью копирования или блокировки этой информации и получения выкупа за неё. Рассмотрим несколько инцидентов, произошедших в транспортных компаниях, а также введение Министерством транспорта РФ мер по обеспечению защищенности отрасли от угроз информационной безопасности.

Австралийская транспортно-логистическая компания Toll, которая поставляет до 95 миллионов единиц продукции по всему миру каждый год, дважды подверглась кибератакам в первом полугодии 2020 года.

Первый инцидент произошёл 31 января. После обнаружения подозрительной активности на серверах, компания была вынуждена временно отключить свои ИТ-системы в качестве меры предосторожности. Из-за проведения работ по восстановлению атакованной злоумышленниками информационной системы, компания была вынуждена частично перейти на ручное управление процессами. Данная мера существенно снизила скорость обработки, что привело к сбою в работе по отправке и отслеживанию грузов. В результате расследования, проведённого внутри компании совместно с Австралийским центром безопасности, было установлено, что результатом атаки стали утечка данных и заражение серверов компании программой-вымогателем, известной как Malito. Более чем через шесть недель после заражения, компания включила отслеживание поставок и снова вывела свои основные службы в интернет.

Второй инцидент произошёл 4 мая. Аналогичным образом были обнаружены нарушения и отключены информационные системы для предотвращения дальнейшего заражения. Основываясь на опыте, полученном при предыдущей атаке, восстановление было проведено более оперативно, так как был реализован схожий сценарий

действий злоумышленников. В результате данной кибератаки также произошли утечка данных и заражение серверов компании программой-вымогателем, однако на этот раз другой, называемой Nefilim.

Руководство Toll Logistics оба раза отказалось выполнять требования злоумышленников по оплате выкупа. Компания не раскрывает количество пострадавших клиентов и ущерб от произошедших атак. По мнению экспертов, из-за того, что услуги компании используются для транспортировки товаров eBay и поставок мобильных телефонов, а также из-за большого числа обращений от пострадавших клиентов через Twitter, экономический ущерб от кибератак может достигать ста миллионов долларов.

В мае 2020 года одна из крупнейших европейских лоукост авиакомпаний британская EasyJet заявила, что подверглась кибератаке. В заявлении говорилось о том, что хакеры получили доступ к базе данных, содержащей адреса электронной почты, личные данные и детали поездок 9 миллионов клиентов авиакомпании. Также, были скомпрометированы данные кредитных и дебетовых карт 2208 клиентов, в том числе трёхзначные секретные коды CVV на обратной стороне карты. О вторжении EasyJet стало известно в январе. Однако, только в начале апреля, после проведения расследования, компания уведомила клиентов об утечке данных их банковских карт. Кроме этого, EasyJet обязалась оповестить всех клиентов, чьи личные данные были украдены злоумышленниками, об инциденте и принятых мерах безопасности. При проведении расследования, было установлено, что изначально хакеры были нацелены на интеллектуальную собственность компании, а не на кражу личных данных клиентов, а также, что основной угрозой действий злоумышленников, имеющих адреса электронной почты и личные данные миллионов клиентов, является фишинг. В качестве мер предосторожности EasyJet призвала к бдительности клиентов, особенно, при получении непредусмотренных электронных писем и электронных писем, отправитель которых представляется как EasyJet. Данные меры необходимы в связи с тем, что поддельные письма могут содержать вредоносное вложение или ссылаться на сайт-клон, предназначенный для кражи личных данных.

Швейцарский производитель подвижных составов Stadler заявил 7 мая 2020 года, что его IT-сеть была атакована вредоносными программами. Компания немедленно инициировала необходимые меры безопасности, а также выяснила, что атака на сеть компании, скорее всего, привела к утечке данных. Масштабы этой утечки на момент публикации заявления неизвестны. Stadler предполагает, что этот инцидент был вызван профессиональной атакой, преступники вымогали крупную сумму денег и угрожали компании потенциальной публикацией данных, чтобы навредить ей. Резервные данные компании уцелели и функционируют после перезагрузки всех затронутых систем.

В конце первого полугодия 2020 Министерство транспорта РФ анонсировало создание единой среды мониторинга защищенности (ЕСМЗ). Планируется, что данная система в режиме реального времени будет выявлять сбои в работе оборудования, уязвимости информационной системы и попытки хакерских атак. То есть, ЕСМЗ будет работать не только для обеспечения информационной безопасности, но и для обеспечения устойчивой работы транспортной отрасли в целом. Именно своевременно не выявленные сбои и уязвимости в настоящее время чаще всего приводят к задержкам в работе логистических компаний и транспорта. Несмотря на то, что большинство компаний способно самостоятельно отслеживать инциденты и своевременно реагировать на их, создание единой системы упростит мониторинг защищенности и работоспособности как внутри организаций, так и во всей транспортной отрасли. Кроме этого, сбор

и анализ данных позволит создать единый реестр инцидентов, на основе которого Министерство совместно с органами безопасности сможет выявлять потенциальные угрозы и связанные с ними риски, формировать рекомендации по предупреждению инцидентов и принятию мер при том или ином виде состоявшегося происшествия.

Одним из основных преимуществ ЕСМЗ является отсутствие нового единого стандарта программно-технических средств, на основе которых планируется применение этой системы. То есть компании продолжают работать со своим, ранее установленным, сертифицированным оборудованием и программным обеспечением. Это также позволит на основе сравнения результатов выявить преимущества и недостатки программно-технических средств при том или ином виде сбоев или хакерских атак. В перспективе, с использованием наиболее предпочтительного оборудования и программного обеспечения возможно создание на основе ЕСМЗ некоторой системы противодействия вторжениям и восстановления последствий атак специально для всей транспортной отрасли.

Вероятно, основой станет новейшая сертифицированная или специально созданная для ЕСМЗ система обнаружения вторжений. Разработка ЕСМЗ будет производиться в рамках федерального проекта «Информационная безопасность». Назначенное ответственным исполнителем Министерство транспорта проведет в начале 3 квартала тендер на разработку пилотного проекта единой системы мониторинга защищенности транспортной информационной инфраструктуры.

Литература:

1. Crozier, R. Toll Group confirms «targeted» ransomware attack/R. Crozier. — Текст: электронный // itnews: [сайт]. — URL: <https://www.itnews.com.au/news/toll-group-confirms-targeted-ransomware-attack-537494> (дата обращения: 30.06.2020).
2. Crozier, R. Toll Group «returns to normal» after Mailto ransomware attack/R. Crozier. — Текст: электронный // itnews: [сайт]. — URL: <https://www.itnews.com.au/news/toll-group-returns-to-normal-after-mailto-ransomware-attack-539507> (дата обращения: 30.06.2020).
3. Wakefield, J. EasyJet admits data of nine million hacked/J. Wakefield. — Текст: электронный // BBC: [сайт]. — URL: <https://www.bbc.com/news/technology-52722626> (дата обращения: 30.06.2020).
4. Гаврилюк, А.; Волобуев, А. Массированно на атаку: госсистема защитит транспорт от хакеров/А. Гаврилюк, А. Волобуев. — Текст: электронный // Известия: [сайт]. — URL: <https://iz.ru/1028446/anastasiia-gavriliuk-aleksandr-volobuev/massirovanno-na-ataku-gossistema-zashchitit-transport-ot-khakerov> (дата обращения: 30.06.2020).
5. Cyber-attack targeted Stadler's IT network. — Текст: электронный // Global Railway Review: [сайт]. — URL: <https://www.globalrailwayreview.com/news/100586/cyber-attack-stadler/> (дата обращения: 01.07.2020).

Обзор платформ для разработки смарт-контрактов

Ржавитина Мария Андреевна, студент;
Мелков Никита Александрович, студент;
Пак Алексей Станиславович, студент;
Спиридонов Александр Юрьевич, студент
Санкт-Петербургский государственный университет

Для того чтобы разработка смарт-контрактов была эффективной, необходимы знание и понимание разработчиками особенностей блокчейн-платформ. Целью написания статьи являлось проведение обзора текущего состояния платформ смарт-контрактов и их особенностей, влияющих на функциональность смарт-контракта.

Ключевые слова: блокчейн, блокчейн-платформы, смарт-контракты.

Впервые концепция смарт-контракта была изложена в 1997 году Ником Сабо в труде *Formalizing and Securing Relationships on Public Networks*. Смарт-контракт — это набор обещаний, указанных в цифровой форме, включая протоколы, в рамках которых стороны выполняют эти обещания. Ник Сабо, обосновывая целесообразность использования смарт-контрактов, писал, что основная идея интеллектуальных контрактов заключается в том, что в программное обеспечение могут быть встроены многие виды договорных положений, что делает невозможным невыполнение условий контракта. Принципы проектирования умных контрактов вытекают из правовых принципов и экономической теории. Использование технологии блокчейн и смарт-контрактов обеспечивает высокий уровень безопасности данных в сети, исключает необходимость в посредниках, предоставляет возможность установки прав участников сети и ускоряет взаимодействие пользователей системы за счет автоматизации процедуры заключения сделок.

Платформы для разработки смарт-контрактов

В настоящее время существует множество различных платформ для разработки смарт-контрактов. В этой статье платформы умных контрактов рассматриваются на основе нескольких характеристик:

Тип блокчейна. Существует публичный (public) и частный (private) блокчейны. *Public Blockchain* означает, что любой пользователь может участвовать в общедоступной цепочке блоков. Публичные цепочки блоков децентрализованы и они безопасны тем, что данные не могут быть изменены. *Private Blockchain*, наоборот, управляется одним объектом, а участвующим сторонам требуется разрешение на участие в сети.

Количество транзакций в секунду (TPS). Характеризует количество операций передачи данных, которые происходят в каждую секунду.

Алгоритм консенсуса. Представляет собой способ, с помощью которого все участники сети приходят к общему соглашению, чтобы обеспечить единое состояние сети.

Полнота по Тьюрингу. Тьюринг-полные смарт-контракты предоставляют возможности решать любые вычисляемые задачи. Тьюринг-неполные смарт-контракты предлагают только более простые инструменты.

Примечательные особенности нескольких платформ обсуждаются ниже, в то время как сводка более расширенного набора платформ представлена в Таблице 1.

Ethereum — одна из первых платформ, которая представила концепцию умных контрактов в блокчейне. Прелесть платформы заключается в степени стандартизации и поддержки, которую она предлагает. Ethereum опубликовал набор четко определенных правил, которым должны следовать разработчики, для того чтобы сделать разработку интеллектуальных контрактов более простой и менее рискованной. Ethereum разработал собственный язык программирования интеллектуальных контрактов, Solidity, который помогает в стандартизации и значительно упрощает настройку контрактов. Однако у Ethereum есть несколько недостатков. Разработчики обнаружили ряд проблем безопасности с кодом Ethereum, что привело к тому, что многие контракты были открыты для хакеров. Также отсутствие масштабируемости платформы приводит к низкой скорости транзакций. Языку Solidity не хватает гибкости. Он не поддерживает многомерные массивы и допускается только небольшое количество параметров в функции контракта, что значительно сказывается на методах разработки контрактов.

Проект **Hyperledger Fabric** начался в декабре 2015 года и был создан Linux Foundation. Это разрешенная инфраструктура блокчейнов, которая облегчает выполнение умных контрактов. Разработчики Hyperledger создали набор чрезвычайно полезных инструментов на основе Javascript, который позволяет создавать умные контракты намного проще и эффективнее с рядом других распространенных языков программирования, просто установив соответствующие модули. Hyperledger является разрешенной сетью, что означает, что все участники сети имеют известные идентификационные данные. Это позволяет создавать умные контракты с соблюдением закона о защите данных. Но у Hyperledger нет собственной системы токенов, то есть нет возможности разработать умные контракты, включающие необходимость перевода платежей.

EOS — децентрализованная операционная система, созданная Block. one для работы и разработки интеллектуальных контрактов, при этом система чрезвычай-

чайно быстрая при минимальных операционных затратах и платах для пользователей. EOS утверждает, что управляет миллионами транзакций в секунду благодаря своему механизму консенсуса DPOS (делегированное подтверждение ставок). Алгоритм повышает демократию сети и позволяет увеличить скорость транзакций и создания блоков, не ставя под угрозу децентрализованную структуру. EOS использует виртуальную машину Web Assembly, предназначенную для более быстрой загрузки, анализа и выполнения контракта, а также она позволяет создавать смарт-контракты на любом языке программирования, который может быть скомпилирован в WASM.

NEO, основанная Da Hongfei в 2014 году, предлагает возможности для создания децентрализованных приложений и умных контрактов. В отличие от других платформ, Neo использует децентрализованный алгоритм консенсуса византийской отказоустойчивости (dBFT). Этот механизм консенсуса оказался более энергоэффективным и позволяет совершать транзакции намного бы-

стрее. NEO использует облегченную виртуальную машину в качестве интеллектуальной среды исполнения контрактов. Она ускоряет процесс и занимает небольшое количество ресурсов. Интеллектуальный контракт NEO может использоваться напрямую практически любым языком программирования высокого уровня. NEO предоставляет компиляторы и плагины для этих языков, которые используются для компиляции языков высокого уровня в наборы команд, поддерживаемые виртуальными машинами NEO.

Stellar — это платежная сеть, созданная в июле 2014 года Джеймсом Маккалехом. Контракты, предоставленные Stellar, не являются полными по Тьюрингу, что означает, что он предназначен выполнять только простые функции для базовых приложений. Тем не менее, для разработки базовых приложений и токенов Stellar является наиболее простым решением, не требующем много времени. Смарт-контракты могут быть написаны на любом языке, для которого сообщество Stellar предоставит API, например, JavaScript, Python, Golang, PHP.

Таблица 1. Таблица платформ для разработки смарт-контрактов

Платформа	Тип блокчейна	TPS	Алгоритм консенсуса	Тьюринг — полнота	Языки программирования
Cardano	public	1000	PoS	да	Solidity, Plutus
EOS	public	50000	dPos	да	C, C++, а также Go, Rust
Ethereum	public	25	PoW	да	Solidity
Hyperledger Fabric	private	20000	PBFT	да	Go, Java, JavaScript
Lisk	private	10000	DPoS	да	JavaScript
Nem	private	4000	POI	да	JavaScript
NEO	public	10000	dBFT	да	C#, Java, Kotlin, Python
Stellar	public	1000	SCP	нет	JavaScript, Python, Golang
Qtum	public	100	PoS	да	Solidity
Waves	private	500	PoS	нет	Ride
Ripple	public	70000	RPCA	да	Любой

Вывод

Сложно назвать лучшую платформу для данной технологии, так как все платформы используются и выбираются в зависимости от индивидуальных потребностей каждого разработчика. Необходимо обращать внимание и на зрелость платформы, инструменты разработки и на-

личие доступной литературы. Смарт-контракты все еще новое направление, требующее качественного изучения и развития, и возможно, в скором будущем платформы для умных контрактов будут усовершенствованы и смарт-контракты станут неотъемлемой частью повседневной жизни.

Литература:

1. Michael Brenner, Kurt Rohloff, Joseph Bonneau, Andrew Miller, Peter Y. A. Ryan, Vanessa Teague, Andrea Bracciali, Massimiliano Sala, Federico Pintore, Markus Jakobsson. Financial Cryptography and Data Security. // Apress. — 2017
2. Mohammad Ayoub Khan, Mohammad Tabrez Quasim, Fahad Algarni, Abdullah Alharthi. Decentralised Internet of Things: A Blockchain Perspective. // Apress. — 2020
3. Szabo Nick. Formalizing and Securing Relationships on Public Networks // First Monday. — 1997.
4. Smart Contract platforms — Access mode: <https://hashrating.com/>(online; accessed: 2020-07-01).

Разработка автоматизированной системы составления и оптимизации расписания занятий

Сиркин Тимур Владимирович, студент;
 Чернышова Александра Петровна, студент;
 Мартынов Павел Андреевич, студент;
 Морозов Артемий Дмитриевич, студент
 Санкт-Петербургский государственный университет

Ключевые слова: генетический алгоритм, составление школьного расписания, автоматическая генерация, описание ограничений.

Одними из важных задач современного мира являются задачи планирования и оптимального распределения ресурсов.

Большинство из них являются NP-трудными, и алгоритмы, применимые для их решения, реализованные на ЭВМ, зачастую требуют неприемлемо большое время работы для задач «большой размерности».

Область применения решений таких задач крайне обширна. Это может быть, как свой личный график работ, запланированных на какой-то промежуток времени, так и более трудные задачи, например, составление расписания для самолетов или поездов. Такие задачи харак-

теризуются большим количеством параметров, ограничений и критериев оптимальности.

В данной статье рассмотрена задача составления оптимального расписания для школьных учебных заведений. Она является детерминированной многокритериальной задачей распределения и относится к классу задач целочисленного линейного программирования.

При составлении расписания занятий школьные завучи зачастую используют самодельную панель с ячейками для карточек, каждая из которых символизирует занятие по указанному предмету, проводимое указанным учителем. Каждой ячейке также ставится в соответствие кабинет.

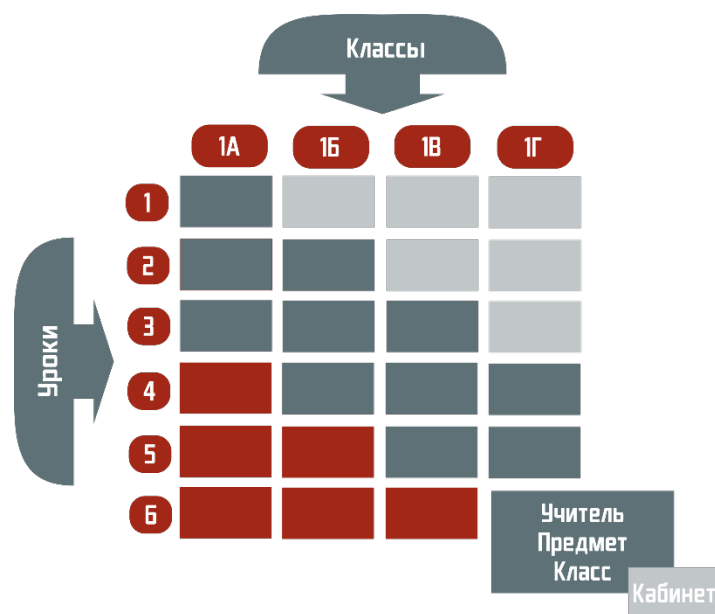


Рис. 1. Схема составления расписания

При этом, для составления расписания, используется метод перебора и экспертной оценки, основанной на опыте составления расписаний предыдущих лет. Такой подход требует больших временных затрат и не может гарантировать какой-либо уровень качества, а оптимальность результата трудно оценить.

Постановка задачи

Таким образом, целью данной работы является создание автоматизированной системы составления и оп-

тимизации расписания занятий. Система должна быть применима для составления и оптимизации расписаний школьных учебных заведений и позволять

1. Описывать ограничения и критерии оптимальности
2. Автоматически генерировать расписание, удовлетворяющее ограничениям и критериям из пункта [1]
3. Составлять расписание вручную
4. Визуализировать и редактировать составленное расписание

Математическая модель

При описании математической модели составления расписания будут использоваться следующие множества объектов.

1. Множество учебных групп $G = \{g_1, \dots, g_{N_G}\}$
2. Множество учителей $T = \{t_1, \dots, t_{N_T}\}$
3. Множество дисциплин $D = \{d_1, \dots, d_{N_D}\}$
4. Множество кабинетов $R = \{r_1, \dots, r_{N_R}\}$
5. Множество уроков $L = \{l_1, \dots, l_{N_L}\}$

Где N_G - количество учебных групп (стоит отметить, что речь идет именно об учебных группах, а не о классах, т.е. класс может быть разбит на несколько учебных групп), N_T - количество учителей, проводящих занятия, N_D — количество учебных дисциплин, по которым проводятся занятия, N_R — количество доступных для использования учебных помещений, N_L - количество уроков в неделю.

Тогда в качестве теоретико-множественной модели будет рассматриваться функция $F(g, t, d, r, l)$ отображения декартового произведения множеств $G \times T \times D \times R \times L$ на множество $\{0,1\}$, где $F(t_{i_t}, g_{i_g}, d_{i_d}, r_{i_r}, l_{i_l}) = 1$ означает, что учитель t_{i_t} проводит занятие у группы g_{i_g} , по дисциплине d_{i_d} в кабинете r_{i_r} во время урока l_{i_l} .

Объединим множества учителей, учебных групп и множество дисциплин в одно композиционное множество $A = \{a_1, \dots, a_{N_A}\}$, где $A = (t_{i_t}, g_{i_g}, d_{i_d})$.

Тогда функцию отображения F можно представить как $F = (A \times R \times L) \rightarrow \{0,1\}$. А каждый конечный вариант расписания можно определить двумя векторами: $\vec{\alpha} = (\alpha_1, \dots, \alpha_{N_A})$ и $\vec{\beta} = (\beta_1, \dots, \beta_{N_A})$, где $\alpha_i \in R$ и $\beta_i \in L$ соответствуют элементу $a_i \in A$.

Ограничения и критерии оптимальности расписания предлагается описывать функциями штрафа $p_i(\vec{\alpha}, \vec{\beta}) \rightarrow \mathbb{R}$.

Тогда задачу нахождения оптимального расписания можно сформулировать как поиск такого варианта расписания $(\vec{\alpha}, \vec{\beta})$, при котором достигается минимум функции $P = \sum p_i(\vec{\alpha}, \vec{\beta})$.

Генетический алгоритм

Для решения данной задачи могут быть использованы как эвристические алгоритмы, так и точные методы решения, например, метод раскраски графа. Но использование точных методов решения повлечет за собой сильный рост временных затрат, при увеличении объема входной информации. В связи с этим, было принято решение использовать эвристический метод, а именно — генетический алгоритм.

Алгоритм позволяет сформировать оптимальную особь (расписание) путем скрещивания различных особей, их мутации и селекции.

При этом каждая особь должна удовлетворять всем строгим ограничениям, которые задаются до начала работы алгоритма. А оптимальная особь должна наиболее хорошо удовлетворять критериям оптимальности, которые также задаются заранее.

Сам алгоритм состоит из нескольких этапов:

1. Формирование начальной популяции
2. Скрещивание особей
3. Селекция особей и формирование новой популяции
4. Мутация особей
5. Проверка условия останова алгоритма

В предложенном алгоритме каждая особь состоит из двух хромосом, которые соответствуют определяющим векторам $\vec{\alpha}$, и $\vec{\beta}$. Каждая из хромосом имеет свой генотип, т.е. состоит из определенного набора генов. В данной реализации каждая из хромосом состоит из N_A генов. В первой хромосоме — это элементы множества кабинетов R , во второй — элементы множества уроков L . При этом генотип каждой хромосомы упорядочен и имеется взаимно однозначное соответствие между i -ой позицией в хромосоме и i -ым элементом в множестве A .

Оптимальность особи оценивается значением целевой функции $P = \sum p_i(\vec{\alpha}, \vec{\beta})$. Чем меньше значение функции, тем оптимальнее особь.

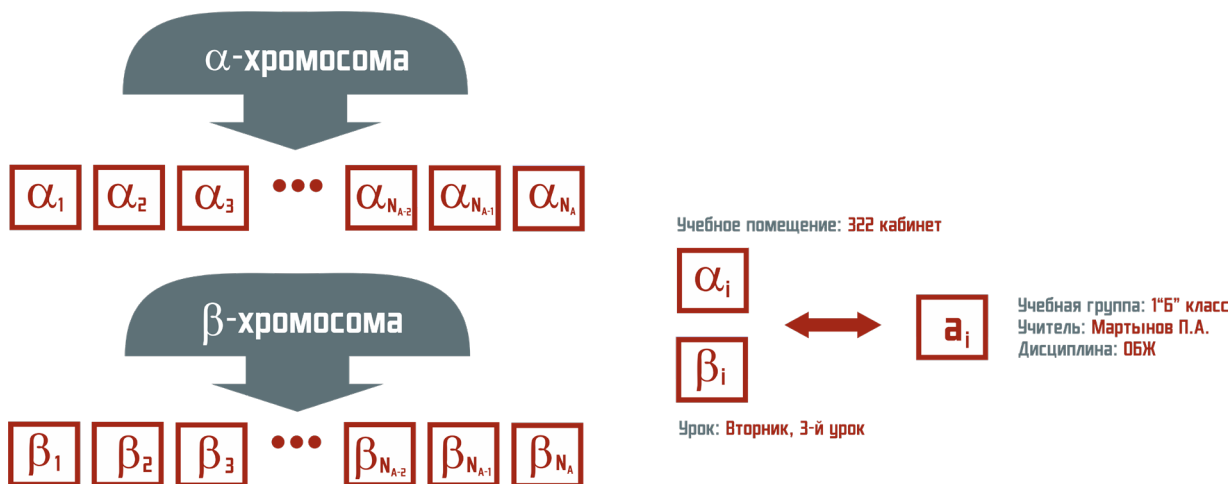


Рис. 2. Особь

Скращивание представляет из себя процесс формирования новых особей путем смешивания генотипов двух расписаний. Так, каждый потомок содержит в себе при-

знаки обоих родителей. Причем более оптимальные особи имеют большую вероятность дать потомство.

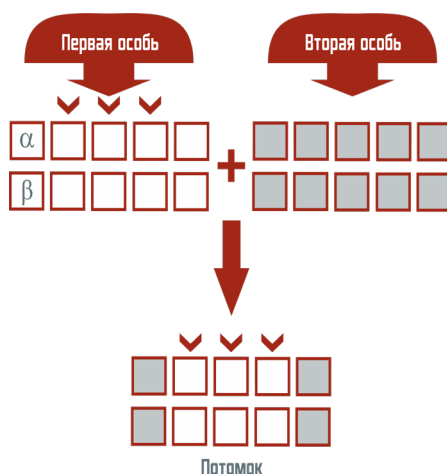


Рис. 3. Скращивание

Селекция является процессом отбора наилучших особей. Таким образом, каждое последующее поколение формируются из наиболее оптимальных вариантов расписания, полученных в ходе формирования начальной популяции и скращивания.

После формирования нового поколения, для избежания преждевременной стабилизации популяций, к некоторым из особей нового поколения применяется оператор мутации, который заменяет в случайно выбранных генах значения на другие допустимые.

В конце каждой итерации проверяется значение функции штрафа наиболее оптимальной особи. Если оно

достигло целевого значения, тогда алгоритм останавливается.

Программная реализация

Система состоит из трех модулей, которые являются независимыми приложениями: модуль описания ограничений, модуль автоматической генерации и модуль управления. Такой подход был выбран для дальнейшего удобства при масштабировании программной реализации. Так, в дальнейшем, эту систему можно будет использовать не только для составления расписаний для школьных учебных заведений, но и для составления расписаний любого назначения.

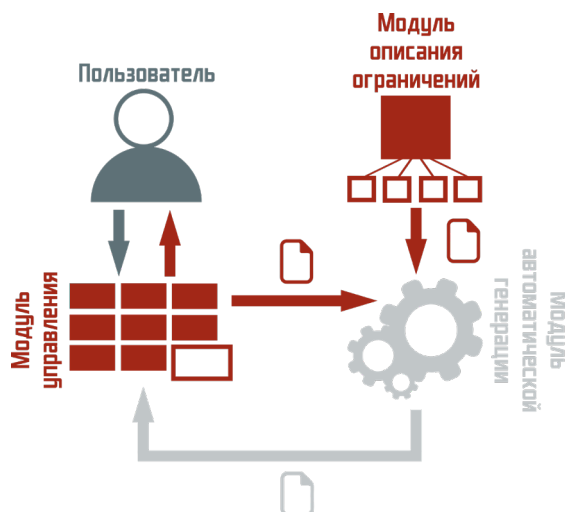


Рис. 4. Схема взаимодействия модулей

Основное взаимодействие конечного пользователя происходит с управляющим модулем, который в свою очередь обращается к модулю автоматической генерации. Предполагается, что модуль описания ограничений используется пользователем единожды для конфигурации критериев оптимальности и ограничений или, в редких

случаях, для их изменения. Эта конфигурация может использоваться модулем автоматической генерации.

Модуль описания ограничений

Данный модуль является настольным приложением для описания критериев оптимальности и ограничений, которые необходимо учитывать, при составлении рас-

писания. Также позволяет описать систему начисления штрафов за нарушение различных ограничений. Основными объектами описания являются сущности: преподаватель, группа, дисциплина, кабинет, урок группы, урок всех групп, учебный день группы, учебная неделя группы. При этом каждая сущность может иметь неограниченное число атрибутов. На выходе пользователь получает конфигурационный файл, который в дальнейшем используется другими модулями.

У каждой сущности есть свое наименование, с помощью которого пользователь может идентифицировать её при описании ограничений и добавлении атрибутов. У некоторых сущностей есть константные атрибуты, которые содержат какое-либо значение или являются ссылками на другие сущности. Например, сущность, представляющая учебную неделю группы, содержит в себе атрибут, показывающий количество дней в учебной неделе.

Таким образом сущности являются представлением реальных объектов, на которые могут накладываться ограничения, при составлении расписания. При этом можно обратиться к любому ее атрибуту, чтобы получить его значение или сравнить с другим значением.

Помимо константных атрибутов, которые нельзя удалить, у каждой сущности можно создавать дополнительные атрибуты, которые можно использовать при опи-

сании ограничений. Или же создать его без привязки какой-либо сущности.

Описание атрибута должно содержать его тип (целочисленное число, текстовое значение, ссылка на элемент и т.д.), может содержать значение по умолчанию и функцию вычисления.

Описание ограничений можно разделить на три условные части: подготовка данных, проверка и вычисление штрафа.

В первой части пользователь имеет возможность вычислить дополнительные значение, которые понадобятся ему для описания.

Во второй части пользователь описывает, как расписание проверяется на нарушение данного ограничения. При этом он может использовать данные из первой части, основные арифметические бинарные операторы, основные логические бинарные операторы и функции из набора. Этими функциями являются: вычисление суммы элементов массива, проверка уникальности элемента в массиве и вычисление количества элементов массива.

Третья часть описывается только для нестрогих ограничений и представляет собой правило, по которому вычисляется функция штрафа за нарушение ограничения. В этой части пользователь может использовать все те же объекты и операции, что и в части проверки ограничения.

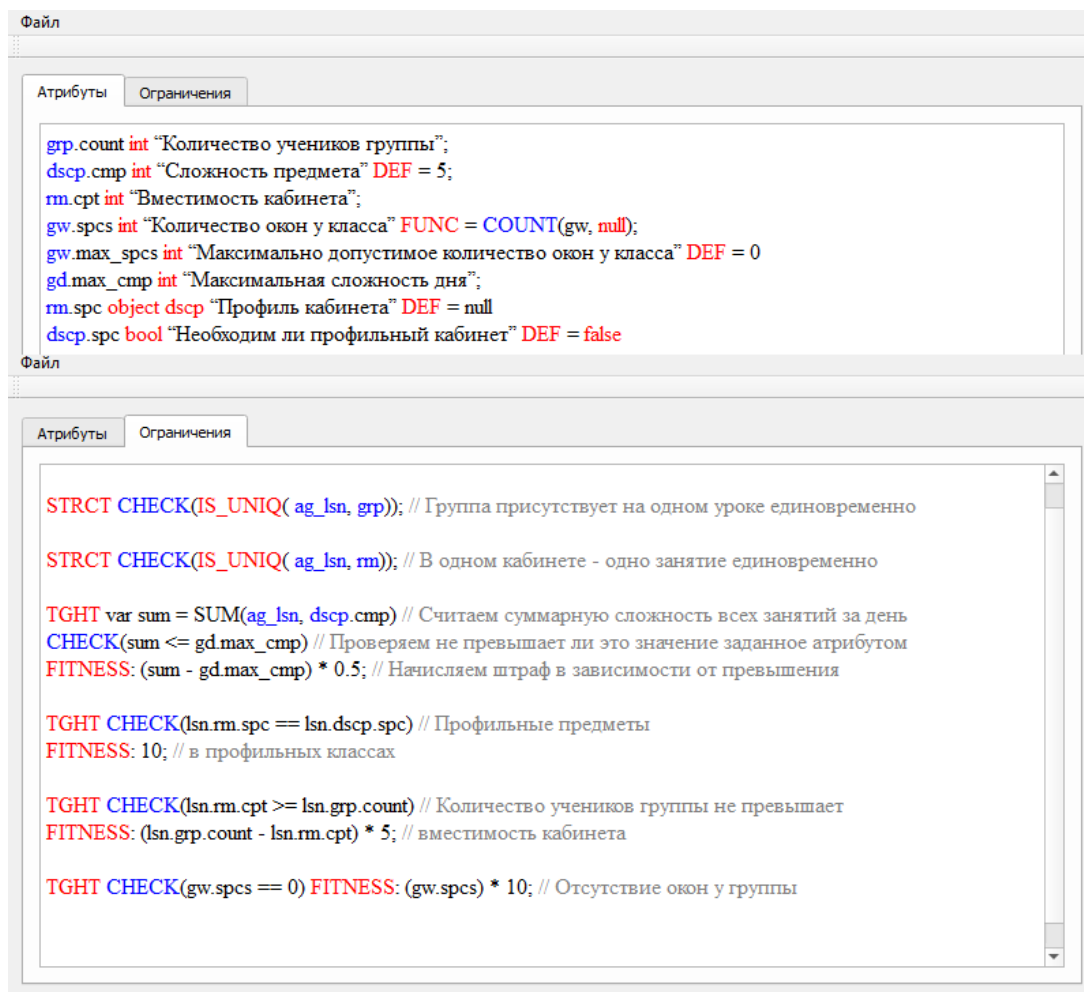


Рис. 5. Вкладки для описания атрибутов и ограничений

Модуль автоматической генерации

Данный модуль является консольным приложением, позволяющим автоматически генерировать расписание, используя генетический алгоритм.

На входе приложение ожидает описание связей между объектами, а именно: между преподавателями, учебными группами, дисциплинами, уроками и кабинетами. Также на вход можно передать конфигурационный файл, полученный при помощи модуля описания ограничений.

При отсутствие конфигурационного файла будут использоваться стандартные ограничения, прописанные в исходном коде программы.

После первичной обработки данных и интерпретации конфигурационного файла запускается генетический алгоритм, описанный выше.

Модуль управления

Данный модуль является настольным приложением и основным для взаимодействия с пользователем. Может быть использован отдельно для ручного и полуавтоматического составления расписания.

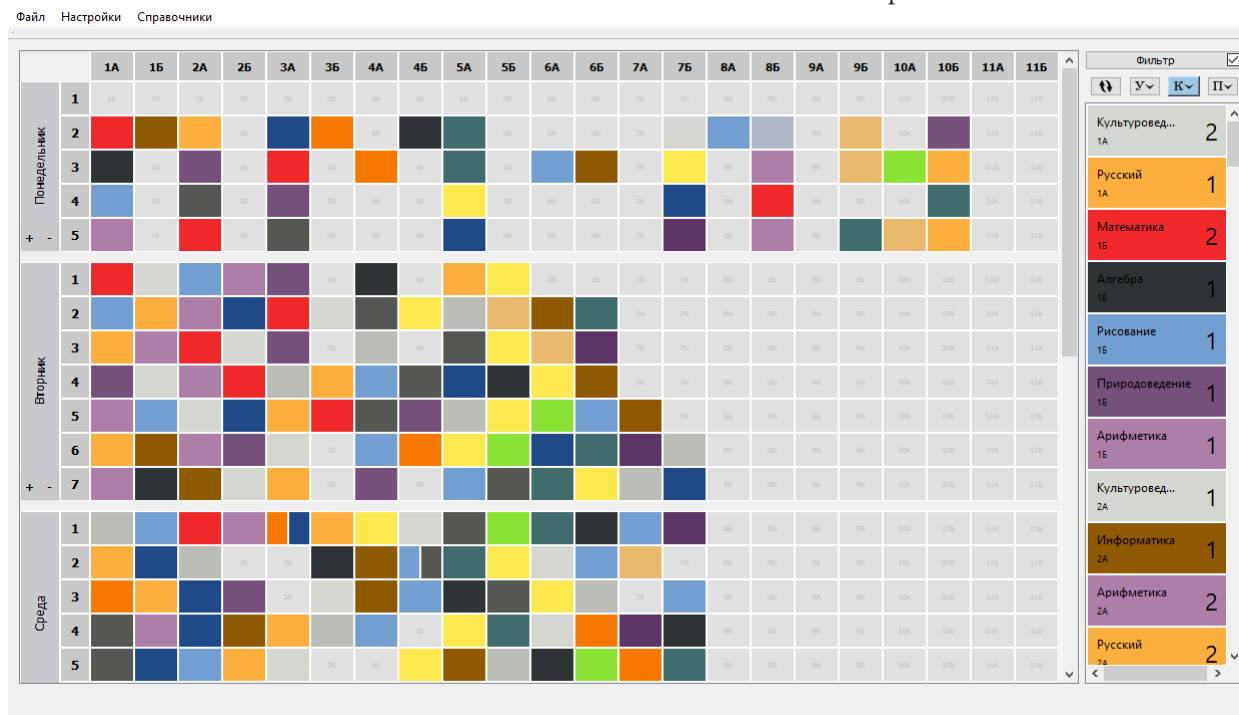


Рис. 6. Модуль управления

Приложение предоставляет следующий функционал

1. Сохранение, загрузка и создание расписания
2. Ведение справочников
3. Ручное составление расписания
4. Проверка ограничений
5. Визуализация в различных форматах
6. Прочие вспомогательные функции

Создать расписание можно как с нуля так и на основе другого. В первом случае будет создана пустая таблица, а справочники будут пустыми. Во втором случае будут импортированы все настройки и справочники, таблица же будет пустой.

Сохраняется расписание в виде текстового файла в формате stt и загружается открытием этих же файлов.

Приложение позволяет вести справочники классов, учителей, кабинетов и предметов. При привязывании класса и предмета учителю, т.е. назначение его проводить

занятие у данного класса по данному предмету, формируется карточка соответствующего цвета — которую в дальнейшем можно будет поместить в таблицу.

Основными объектами при составлении расписания являются таблица и карточки. Пользователь имеет возможность назначить ячейке определенную карточку, поменять карточки местами, или очистить ячейку.

Одной ячейке можно присвоить сразу две карточки. Таким образом реализуется возможность разделять класс на группы для определенных занятий.

При перетаскивании карточки из списка доступных, перетаскивании из таблицы или выборе ячейки, система проверяет установленные ограничения и подсвечивает доступные для назначения ячейки. При попытке назначить карточку недопустимой ячейке система отклонит данное действие и уведомит пользователя о том, какие ограничения нарушаются.

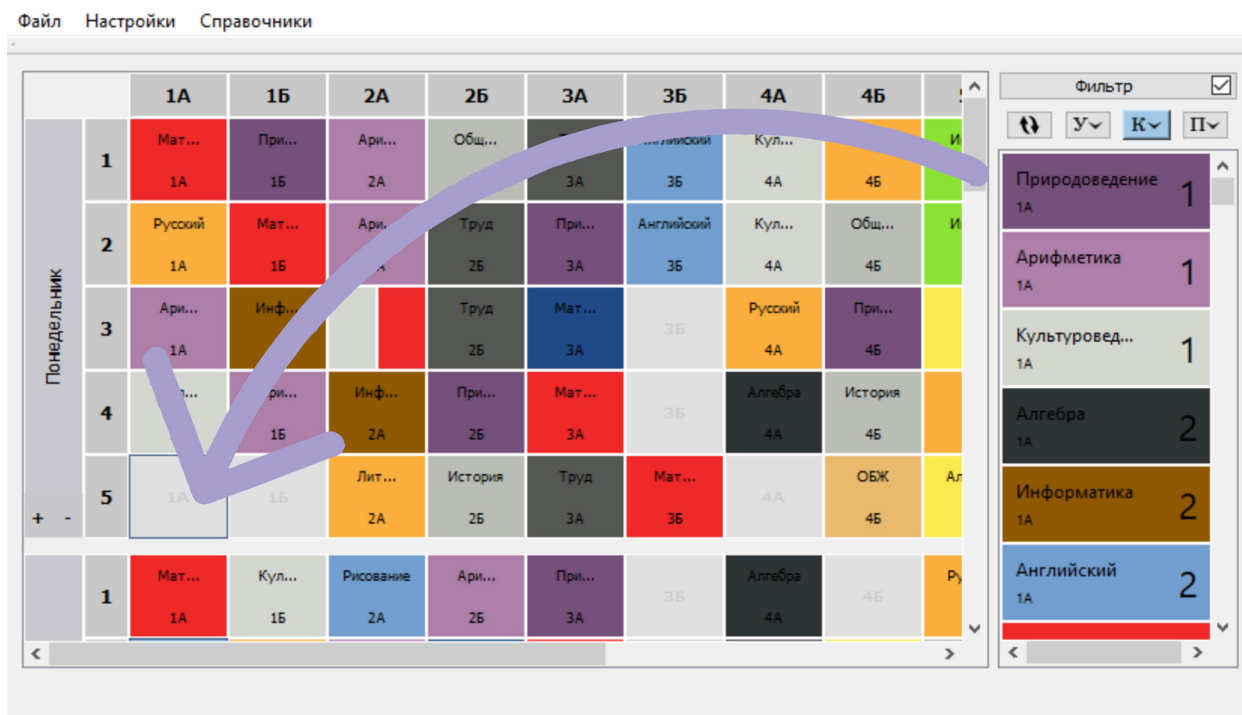


Рис. 7. Подсказка при перетаскивании

Модуль позволяет пользователю настраивать вид таблицы и карточек, масштабировать таблицу, скрывать выбранные классы, фильтровать и сортировать карточки.

Также присутствует возможность экспортировать созданное расписание в формат *xlsx*. Причем экспортирование можно производить в двух самых распространенных форматах: относительно учебных дней и классов, а также относительно учебных дней и учителей. Размеры и вид всех объектов также можно настроить перед экспортированием.

Также реализован ряд других вспомогательных функций: различные подсказки при составлении распи-

сания, заполнение справочников по шаблону и прочие, описание которых опущено.

Заключение

Система уже сейчас может успешно использоваться для ручного, полуавтоматического и автоматического составления расписаний для школьных учебных заведений, хоть и нуждается в дальнейшей доработке и расширении.

В дальнейшем планируется расширить эту систему так, чтобы она была применима не только для составления расписаний для школьных учебных заведений, но и для составления расписаний произвольного назначения.

Литература:

1. Лазарева, А. А., Гафаров Е. Р., Теория расписаний. Задачи и алгоритмы. — М.: МГУ, 2011 г., 224 с.
2. Григорьева, Н. С., Алгоритм ветвей и границ для задачи составления расписания на параллельных процессорах. [Электронный ресурс] — URL: <https://cyberleninka.ru/article/n/algorithm-vetvey-i-granits-dlya-zadachi-sostavleniya-raspisaniya-na-parallelnyh-protessorah/viewer>
3. Кабальнов, Ю. С., Шехтман Л. И., Низамова Г. Ф., Земченкова Н. А., Композиционный генетический алгоритм составления расписания учебных занятий [Электронный ресурс] — URL: <https://cyberleninka.ru/article/n/kompozitsionnyy-geneticheskiy-algoritm-sostavleniya-raspisaniya-uchebnyh-zanyatij>
4. Бабкина, Т. С., Задача составления расписания: решение на основе многоагентного подхода. [Электронный ресурс] — URL: <https://cyberleninka.ru/article/n/zadacha-sostavleniya-raspisaniy-reshenie-na-osnove-mnogoagentnogo-podhoda/viewer>
5. Шлее, М., Qt 5.10. Профессиональное программирование на C++. — СПб.: БХВ-Петербург, 2018 г., 1072 с.
6. Астахова, И. Ф., Фирас А. М., Составление расписаний учебных занятий на основе генетического алгоритма [Электронный ресурс] — URL: <http://www.vestnik.vsu.ru/pdf/analiz/2013/02/2013-02-17.pdf>

Проблема оптического распознавания субтитров, встроенных в видеопоток

Спиридонов Александр Юрьевич, студент;
Пак Алексей Станиславович, студент;
Ржавитина Мария Андреевна, студент;
Мелков Никита Александрович, студент
Санкт-Петербургский государственный университет

В данной работе авторами рассматривается задача создания алгоритма способного распознавать текст субтитров при любых даже наиболее проблемных видеорядах и возвращать его вместе с временными метками пользователю.

Ключевые слова: OCR, видео, субтитры, анализ.

В связи с развитием информационных технологий, методы передачи и получения информации постоянно меняются от использования наиболее эффективных решений, как например текст, к решениям удобным для человеческого восприятия, но более затратным, с точки зрения передачи данных. Одним из возможных решений в наши дни является видео, в связи с массовым распространением и доступностью таких сервисов как YouTube, Twitch и т. п., ставшее одним из главнейших способов передачи информации. Однако видео материалы, как способ фиксирования тех или иных событий, рассчитаны на использование основных чувств человека, а именно зрение и слух. Для людей, которые, в данный момент, или вовсе не имеют возможности слушать звуковую составляющую видео, в наше время существуют субтитры. Текст субтитров является важной частью видеоматериала для понимания информации и её анализа. В данной работе рассматривается решение проблемы, появившейся вследствие существования субтитров, встроенных в видеопоток и соответственно, не имеющих текстовой версии, что не позволяет взаимодействовать с текстом в привычном варианте.

Авторы данной статьи поставили задачу создания алгоритма способного распознавать текст субтитров при любых даже наиболее проблемных видеорядах и возвращать его вместе с временными метками пользователю. Данную задачу можно разбить на несколько этапов: устранение шумов, обнаружение субтитров, определение кадров, содержащих текст, поиск моментов его смены и распознавание текста

Проблема распознавания текста на изображении, в английской литературе называемая Optical Character Recognition (OCR) или оптическое распознавание символов, ставится как задача получения текста, присутствующего на изображении. При попытке применить тот же подход к видео файлу, мы встретимся со следующими трудностями: в отличии от обычного OCR не на каждом кадре видео присутствует текст, что вызывает необходимость разделения кадров на имеющие и не имеющие субтитры и отделения таковых от текста, существующего в видео на фоне. Другое немаловажное отличие от стандартной задачи OCR — временные рамки. Временные рамки — это границы времени, указывающие время на-

чала и конца отображения строки субтитров. Их получение также является одной из задач, поставленных в данной работе.

Выбрав кадры с текстом субтитров, мы переходим к задаче оптического распознавания символов, которая делится на два этапа: предобработка и распознавание.

Цель предобработки — это повышение качества последующего распознавания, посредством максимально возможного «очищения» изображения или в нашем случае кадра от всего, что не является необходимым для распознавания.

Обычно предобработка для OCR состоит из следующих целей [4]:

- выравнивание текста до стандартного горизонтального;
- шумопонижение;
- бинаризация;
- удаление линий;
- анализ структуры, например, таблиц в строках и столбцах которых может находиться искомый текст;
- обнаружение текста;
- распознавание шрифта;
- локализация символов;
- нормализация размера и пропорции изображения.

Учитывая, что цель работы — это получение субтитров, то ввиду их природы, список принимает другой вид. Выравнивание текста в случае с субтитрами не является необходимым, так как они изначально текст и соответственно горизонтальны. Удаление линий не является конкретной задачей, а является частью задачи по отделению субтитров от фона, которому принадлежат горизонтальные линии. В вышеуказанном списке целей она обозначена как шумопонижение. Отсутствие структуры видео устраняет необходимость её анализа. Распознавание шрифта необходимо при смене такового в тексте для подбора соответствующего алгоритма. Однако для удобства восприятия зрителя, шрифт редко меняется по середине видео. При этом шрифт определяется с помощью стандартов [1], соответственно, проблема распознавания шрифта исчезает. Учитывая, что размер изображений одинаковый, так как это кадры одного видео, исчезает задача нормализации размера и пропорций изображений. Список целей предобработки можно представить следующим образом:

- шумопонижение;
- бинаризация;
- обнаружение текста;
- локализация символов.

Последним, но не менее важным этапом является распознавание текста. Распознавание текста может быть сложной задачей в зависимости от работы, проделанной на этапе предобработки, цель которой его упростить. Дополнительным фактором, упрощающим решения данной задачи, является правильный подбор алгоритма, учитывающего особенности искомого текста.

Проблему получения субтитров, встроенных в видеопоток, уже пытались решать при помощи выделения границ текста субтитров [8], алгоритм показал неплохую точность. Недостатками работы алгоритма являются его ограничение на положение субтитров снизу, а также отсутствие способа классификации кадров, имеющих в наличии субтитры. Вместо этого, в данном методе, был выбран сегмент видео на всём промежутке которого присутствуют субтитры. Также у алгоритма имеются проблемы при работе с видео, содержащим контрастные элементы, присутствующие, например, в анимации.

В работе Zarifar B. [9], которая представляет алгоритм для локализации и классификации субтитров в ТВ видео, была попытка показывающая неплохие результаты, но она непосредственно использует особенности телевизионного сигнала и приставки его расшифровывающей для своей работы и поэтому не подходит для работы с обычным видео.

При решении данной проблемы имело место и использование нейронных сетей [7] которое не имеет ограничений в подходе выделения границ [8]. Существующий

алгоритм, как и предыдущий, использовал информацию о соседних кадрах для повышения качества распознавания. В конечном счёте, алгоритм показал неплохие результаты в 86% точности распознавания текста, но кадры не получали никакой предобработки, что позволяет сделать предположение о возможном улучшении результатов.

При рассмотрении существующих программных решений, были обнаружены следующие варианты: SubRip [5], CCEXtractor [3], Burnt-in subtitle extractor [2] и videocr [6]. Большинство данных решений за исключением SubRip используют Tesseract для распознавания текста с кадров видео. SubRip, в свою очередь, требует точной настройки человеком, который выберет параметры предобработки, укажет локацию субтитров, а также исправит ошибки посимвольного распознавания. Следовательно, алгоритм нельзя назвать автоматическим. CCEXtractor и Burnt-in subtitle extractor уже более автоматизированные решения, которым тем не менее, тоже необходим человек для указания цвета субтитров для предобработки в обоих программах и цветов контуров в Burnt-in subtitle extractor. Последнее решение videocr не использует предобработку, а полагается полностью на Tesseract с последующим анализом результатов. Положительное отличие от остальных является максимально автономным с опциональными параметрами настройки распознавания. Однако сам по себе Tesseract не показывает достаточно хороших результатов, так что есть предположение о возможности улучшения алгоритма.

Таким образом на данный момент не существует оптимального решения данной проблемы. Тем не менее результаты указывают на возможность создания такового, что будет сделано в последующих статьях.

Литература:

1. BBC Subtitle Guidelines. [Электронный ресурс]: <https://bbc.github.io/subtitle-guidelines/>
2. Burnt-in subtitle extractor. [Электронный ресурс]: <https://github.com/roybaer/burnt-in-subtitle-extractor>
3. CCEXtractor's home page. [Электронный ресурс]: <https://www.ccextractor.org/start>
4. Optical Character Recognition (OCR) — How it works. [Электронный ресурс]: Nicomsoft.com.
5. SubRip Official site. [Электронный ресурс]: <https://sourceforge.net/projects/subrip/>
6. Videocr. [Электронный ресурс]: <https://github.com/apm1467/videocr>
7. Xiaoou Tang и др. A spatial-temporal approach for video caption detection and recognition // IEEE Transactions on Neural Networks. 2002. Т. 13. № 4. с. 961-971.
8. Yongjiu, L. и др. Video Subtitle Location and Recognition Based on Edge Features // 20196th International Conference on Dependable Systems and Their Applications (DSA): IEEE, 2020.
9. Zafarifar, B., Jingyue Cao, With P.H. N. de. Instantaneously responsive subtitle localization and classification for TV applications // IEEE Transactions on Consumer Electronics. 2011. Т. 57. № 1. с. 274-282.

Анализ распространения высказываний с негативной тональностью в пользовательских дискуссиях в социальных сетях

Чернышова Александра Петровна, студент;
Сиркин Тимур Владимирович, студент;
Морозов Артемий Дмитриевич, студент;
Мартынов Павел Андреевич, студент
Санкт-Петербургский государственный университет

Ключевые слова: анализ тональности текста, анализ данных, векторное представление слов, машинное обучение, глубокое обучение

Введение

Подобного рода исследования в последнее время пользуются большим спросом. Изначально для реализации анализа негативных отзывов необходимо провести классификацию отзывов, что приводит нас в сферу сентиментального анализа, который находится в области NLP. Обработка естественного языка (Natural Language Processing, NLP) изучает проблемы, связанные с синтезом естественных языков.

В последние годы в данной области семантическому анализу уделяется много внимания из-за возможности его применения в различных областях — от бизнеса до политических исследований. Благодаря этому инструменту компании могут отслеживать репутацию своего бренда. Анализируя отзывы в социальных сетях, можно судить об их отношении к интересующему нас продукту.

В современных реалиях одной из популярных платформ пользователей интернета для высказывания своего мнения является Твиттер. В данной социальной сети люди открыто и свободно обсуждают происходящие с ними события, говоря о том, что им нравится или не нравится. Популярность Твиттера, влияние, которое пользователи оказывают своими высказываниями, доступность данных привлекают исследователей и ведет к постановке различных задач.

Постановка задачи

Таким образом, **целью** данной работы является проведение анализа и выявление причин возникновения отрицательной тональности в коротких негативных высказываниях социальной сети Twitter для улучшения качества товара или предоставляемых услуг.

Для достижения цели были поставлены следующие **задачи**:

- Рассмотреть наиболее популярные методы классификации тональности;
- Разработать классификатор тональностей;
- Произвести анализ негативных твитов и представить полученные результаты с помощью графических интерпретаций;

В качестве входных данных использовались общедоступные отзывы пользователей американских авиалиний о качестве полетов [1].

Данные были собраны из социальной сети Twitter за февраль 2015 года. Язык используемых твитов английский. Корпус данных был поделен на обучающую и тестовую выборки в соотношении 70%–30%. Классификация производилась на 3 класса: негативные, нейтральные и положительные.

Полученные тексты отзывов были подвергнуты препроцессингу: были удалены специальные символы, цифры и стоп-слова, было произведено приведение к нижнему регистру.

Для реализации алгоритмов использовался следующий стек технологий: основной язык для реализации Python 3, библиотеки NumPy, Pandas, Keras, Matplotlib. Для разработки использовались Jupyter Notebook и Google Colab.

Наивный Байесовский классификатор

Первым классификатором, который был реализован в данной работе, является — Наивный Байесовский. В его основе лежит одноимённая теорема.

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)}$$

где

$P(c|d)$ — вероятность того, что документ dd принадлежит рассматриваемому классу cc ;

$P(d|c)$ — вероятность обнаружить документ dd среди документов рассматриваемого класса cc ;

$P(c)$ — безусловная вероятность встретить документ, принадлежащий рассматриваемому классу cc , в корпусе документов;

$P(d)$ — безусловная вероятность встретить документ d в корпусе.

Для определения класса с наибольшей вероятностью появления Байесовский классификатор использует оценку апостериорного максимума.

$$c_{map} = \arg \max_{c \in C} \frac{P(d|c)P(c)}{P(d)}$$

Классификатор называется наивным ввиду того факта, что документ представляется как набор условно независимых друг от друга вероятностей. Исходя из этого предположения условная вероятность документа аппроксимируется произведением условных вероятностей всех слов входящих в документ.

$$P(d|c) \approx P(w_1|c)P(w_2|c) \dots P(w_n|c) = \prod_{i=1}^n P(w_i|c)$$

Полученные значения метрик представлены в таблице 1.

Таблица 1

Accuracy	Precision	Recall
0.7521	0.7516	0.9205

LSTM

Long Short-Term Memory это разновидность архитектуры рекуррентных нейронных сетей.

В сети LSTM элементом является не один нейрон, а ячейка или модуль. Модуль состоит из нескольких слоёв,

которые комбинируются посредством поэлементного умножения или сложения. На рисунке 1 продемонстрирован повторяющийся модуль сети.

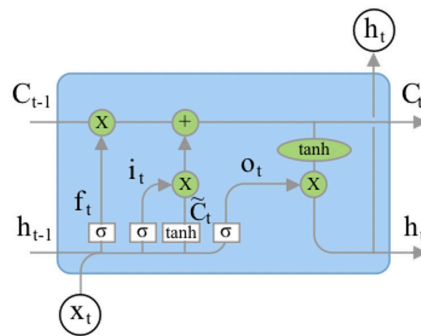


Рис. 1: Пример повторяющегося модуля LSTM

Ключевым для LSTM является состояние ячейки C_t. Именно оно позволяет сохранять данные на длительный промежуток времени. Для того чтобы управлять состоянием ячейки используются так называемые гейты — специальные слои нейронов, отвечающие за интенсивность передаваемого сигнала.

В сетях LSTM используются 3 вида гейтов:

- Гейт забвения (forget gate)
- Входной гейт (input gate)
- Выходной гейт (output gate)

Подробный принцип работы сети описан в работе Хохрейтера [2]. Существуют и более популярные схемы, например описанная Джерсом и Шмидхубером [3], где состояние ячейки также подается на каждый sigma-слой.

В своей работе Чо [4] описал структуру LSTM, в которой слой, использующий гейт забвения и первый этап слоя, использующего входной гейт объединяются. Такая модель в последнее время завоевывает большую популярность.

Была реализована сеть, состоящая из слоев векторных представлений слов, LSTM-слоя, содержащего 10 ячеек, и выходного слоя с функцией активации Softmax.

В модуле в качестве функции активации используется гиперболический тангенс, а в качестве функции, используемой для активации повторного шага — сигмоид.

Количество эпох — 4

Ниже приведены таблицы 2 и 3 значений метрик в зависимости от размера батча обучения и словаря.

Таблица 2

	loss	Accuracy	Recall	Precision
Batch = 32	0.5059	0.7999	0.6302	0.8233
Batch = 25	0.4540	0.8209	0.7135	0.8270
Batch = 15	0.4180	0.8312	0.7437	0.8328
Batch = 5	0.3819	0.8472	0.7611	0.8345

Таблица 3

	loss	Accuracy	Recall	Precision
Voc_size=500	0.4917	0.8049	0.8511	0.8511
Voc_size=750	0.4729	0.8167	0.8454	0.8635
Voc_size=1000	0.4391	0.8317	0.8544	0.8676

Можно заметить, что LSTM дает неплохие результаты, но несильно опережает Наивный Байесовский классификатор.

BERT

BERT состоит из стека энкодеров Трансформера, специальной модели, использующей механизм внимания [5].

BERT — первая нейросеть, которая успешно использует двунаправленный подход для решения задачи представления слов и целых предложений в n-мерном пространстве с учётом контекста [6]. Обычно для решения этой задачи сеть обучают предсказывать слово по заданной последовательности. Двунаправленный подход позволяет учесть влияние каждого слова на контекст, но создаёт дополнительные проблемы, для борьбы с которыми BERT обучают решать задачи Masked Language Modeling.

В ходе решения этой проблемы сеть представляет слова в векторной форме. Это и является основной целью работы нейросети. Эмбединги затем применяются для решения различных задач с использованием техники fine-tuning.

Google предоставляет несколько предобученных моделей.

Для поставленной задачи была использована модель BERT-Base, нечувствительная к регистру, имеющая 12 слоев, 768 узлов, и с 12 attention heads. К выходному слою BERT был добавлен еще один слой с функцией активации — softmax и 3 выходами — для негативных, нейтральных и позитивных твитов. Для того, чтобы избежать переобучения в архитектуре сети также используется dropout-слой. Размер батча равен 32. Количество эпох — 3.

Полученные значения метрик указаны в таблице 4.

Таблица 4

Accuracy	Precision	Recall	F1-мера
0,92739	0.89134	0.92546	0.90808

Как видно из таблицы, BERT является очень мощным инструментом для решения задач NLP. Именно данные тех твитов, которые были классифицированы им, как «отрицательные» будут подвергнуты анализу в дальнейшей главе.

Анализ негативных твитов

После классификация твиты были подвергнуты процедуре анализа.

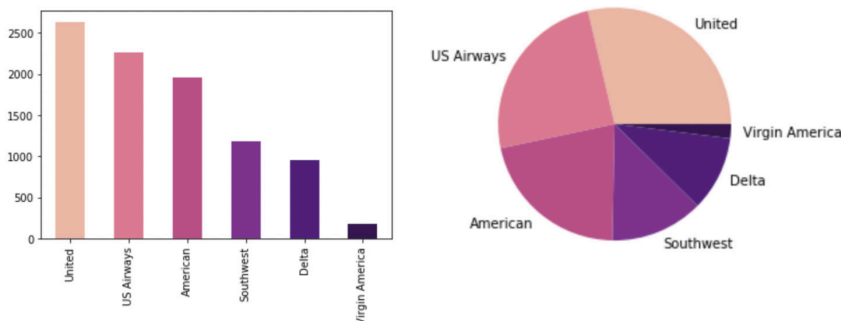


Рис. 2: Гистограмма и Круговая диаграмма, отражающие количество негативных отзывов, которые получили авиакомпании

На рисунке 2 представлены графики, отражающие количество негативных отзывов с разбивкой по авиакомпаниям. Эти данные нерепрезентативны из-за разного количества рейсов, совершаемых компаниями, поэтому было принято решение рассмотреть процентные соотношения твитов разной тональности.

Как видно из приведенных таблицы и гистограммы на рисунке 3 пользователи были больше недовольны перевозками компании «US Airways».

Важным для компании является мониторинг лидеров мнений. Те причины, которые выделяют подобные пользователи могут быть ключевыми негативными качествами, которые следует упразднить. За лидеров мнений приняты те пользователи, отзывы которых получили наибольшее количество ретвитов.

Список лидеров для компании United и твит, получивший наибольший отклик, представлены на рисунке 4.

Одним из самых простых и эффективных способов репрезентации ключевых слов является Облако Слов.

Так выглядит облако слов, в котором использовались слова из отзывов всех авиакомпаний. Можно сделать вывод, что основными проблемами пользователи признают утерю багажа и отмененные рейсы.

Отрицательные твиты были также проанализированы с разбивкой по штатам. Выделив самые проблемные штаты, можно точно решить проблему, что улучшит рейтинг компании. Можно заметить, что такие штаты, как Аляска, Монтана и Вайоминг остались недовольны работой авиакомпаний.

Заключение

В процессе работы были изучены принципы векторного представления слов, популярные модели классификаторов. Проанализированы их возможности, недостатки, потенциальные сферы применения.

В отношении BERT можно с уверенностью сказать, что модель показывает впечатляющие результаты. Особенности архитектуры выделяют её среди существующих нейросетей.

Был проведен анализ негативно классифицированных твитов, результаты которого потенциально могут помочь авиаперевозчикам улучшить качество сервиса.

В дальнейшем планируется представить полученные благодаря анализу данные в виде приложения для компаний. В данном приложении имелась бы, как и ранее приведенная инфографика, так и новые данные, например, процентные соотношения отзывов, полученных от пользователей разных полов, возрастов, отображение графов взаимосвязей между пользователями. Так же хотелось бы добавить возможность компании самой выбирать, твиты с какой тональностью она хочет отслеживать для мониторинга положительных качеств своих конкурентов.

Литература:

1. Датасет Twitter US Airline Sentiment // Kaggle сайт. — URL: <https://www.kaggle.com/crowdflower/twitter-airline-sentiment>
2. S. Hochreiter S., Schmidhuber J Long Short-Term Memory // ResearchGate [Электронный ресурс]. — URL: https://www.researchgate.net/publication/13853244_Long_Short-term_Memory — Текст: электронный.
3. Gers, F.A., Schmidhuber J. Recurrent Nets that Time and Count Difficult // IEEE [Электронный ресурс]. — URL: <https://ieeexplore.ieee.org/document/861302> — Текст: электронный.
4. Cho, K., van Merriënboer B., Bougares F, Shwenk H., Bahdanau D., Bengio Y. Learning Phrase Representations using RNN Encoder — Decoder for Statistical Machine Translation // arXiv [Электронный ресурс]. — URL: <https://arxiv.org/abs/1406.1078> — Текст: электронный.
5. Vaswani, A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A.N., Kaiser L., Polosukhin I., Attention Is All You Need // // arXiv [Электронный ресурс]. — URL: <https://arxiv.org/abs/1706.03762> — Текст: электронный.
6. Devlin, J., Chang M., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding // arXiv [Электронный ресурс]. — URL: <https://arxiv.org/pdf/1810.04805.pdf> — Текст: электронный.

Методы детектирования искусственных новостей

Шергин Тимур Олегович, студент;
Горелов Борис Романович, студент;
Балавнев Данил Андреевич, студент;
Киндулов Михаил Львович, студент
Санкт-Петербургский государственный университет

В статье решается задача детектирования искусственных новостей. Используются классические методы машинного обучения, такие как метод опорных векторов и случайный лес, а также методы глубокого обучения — нейронная сеть LSTM, языковые модели BERT и XLNet.

Ключевые слова: машинное обучение, глубокое обучение, бинарная классификация, фейковые новости.

В последние годы произошел отток пользователей из более традиционных средств массовой информации, таких как газеты, радио и телевидение, в новые форматы: социальные сети, YouTube, подкасты, онлайн-журналы, новостные приложения. Благодаря Интернету появилась возможность мгновенного и бесплатного доступа к широкому спектру источников информации, не говоря уже о многочисленных услугах, позволяющих делиться новостями с миллионами людей по всему миру. Большой объем информации, к которой люди получают доступ, обычно не проверяется и считается достоверным,

что приводит к появлению такой проблемы как дезинформация пользователей посредством публикации фейковых новостей.

Именно поэтому авторы работы поставили перед собой цель выявить метод машинного обучения, который бы наилучшим образом отличал фейковые новости от настоящих.

В процессе обзора существующих решений мы заметили, что в основном решения сосредоточены на анализе отношения пользователей к новости, публикации, посту. Но поскольку характеристики такие как лайки, ко-

личество комментариев и репостов, в наше время могут быть «накрученными», то их использование в детектировании искусственных новостей может приводить к некорректным результатам, поэтому в нашей работе мы сосредоточились только на текстовой составляющей новости.

В данной работе использовался датасет, содержащий 20000 размеченных новостных статей. Была проведена предобработка данных, которая включала в себя удаление стоп-слов, чисел, иноязычных слов, знаков препинания и специальных символов. Это позволило не только уменьшить количество функций для обучения моделей, но и сосредоточиться на более значимом содержании новостей, так как сократилось огромное количество текста, не имеющего важного контекстуального значения.

Поскольку некоторые используемые методы не умеют работать со словами, необходимо использовать модели предварительной подготовки для кодирования текстов в числовые представления. Для этого использовались две модели: TF-IDF Vectorizer и Word2Vec. После получения числовых представлений текстов новостей, мы перешли к непосредственной реализации выбранных методов.

Первый метод — это метод опорных векторов, набор контролируемых алгоритмов машинного обучения, в которых создается гиперплоскость для разделения и классификации функций. Оптимальная гиперплоскость обычно рассчитывается путем создания опорных векторов на обеих сторонах гиперплоскости, в которых каждый вектор должен максимизировать расстояние между собой. Другими словами, чем больше расстояние между каждым вектором вокруг гиперплоскости, тем точнее будет граница принятия решения между категориями объектов. SVM требует определенных параметров, таких как тип ядра, C , максимальные итерации и т. д.. В нашем случае необходимо было определить оптимальное C , а также оптимальное ядро. Мы использовали K -кратную перекрестную проверку с $K = 3$. Был выполнен сеточный поиск типов ядра и C , чтобы дать наиболее точную модель SVM. Параметры, которые использовались для каждого ядра, были линейными и радиальными, в то время как значения, которые использовались для C , были 0,25, 0,5 и 0,75. После того как поиск по сетке для этих гиперпараметров был завершен, модель была оценена с использованием наиболее оптимальных гиперпараметров.

Следующий рассмотренный метод — это случайный лес, контролируемый алгоритм машинного обучения, который является одним из самых гибких и простых в использовании. Основная идея заключается в использовании большого ансамбля решающих деревьев, каждое из которых само по себе даёт очень невысокое качество классификации, но за счёт их большого количества результат получается хорошим. Алгоритм случайного леса требует 4 гиперпараметра для настройки, таких как количество деревьев в лесу (200, 400, 800); максимальная глубина дерева (1, 5, 9); минимальное количество выборок, которое должно быть в ведущем узле (2, 4); минимальное количество выборок, которое должно быть в листовом

узле (5, 10). Минимальное количество выборок в каждом листовом узле имеет эффект сглаживания модели. Все параметры применяются для поиска в сетке, и, в конце концов, лучший набор параметров может быть определен, при помощи кросс-валидации по 3 блокам.

Далее была реализована нейронная сеть LSTM, особый тип рекуррентных нейронных сетей, который способен изучать долгосрочные зависимости. В LSTM блок работает как обычный рекуррентный блок, но имеет дополнительную ячейку памяти и различные вентили. Вентили помогают определить долгосрочные зависимости, управляя потоком данных внутри блока. Ячейка дает сети своего рода память и сохраняет долгосрочные зависимости. Для поиска в сетке применяются следующие параметры: оптимизатор (Adam, SGD), функция активации (ReLU, linear, sigmoid), относительно небольшое количество скрытых слоев (0, 1, 2) и узлов (200, 400, 600) выбрано в качестве основы для поиска в сетке, поскольку это простая задача двоичной классификации и слишком большое количество может привести к переобучению. Из-за ограниченности вычислительных ресурсов поиск в сетке для LSTM делится на четыре последовательных этапа. Вместо того, чтобы выполнять поиск по сетке по всем гиперпараметрам одновременно, поиск по сетке сначала выполняется по количеству скрытых слоев, а все остальные гиперпараметры выбираются случайным образом. Затем выполняется поиск в сетке по количеству узлов в скрытом слое (яч), используя наилучшее количество скрытых слоев, найденных на шаге 1 и т. д.. Поиск по сетке завершается, когда все четыре шага завершены. На каждом этапе использовалась трехкратная перекрестная проверка.

В качестве четвертого метода была использована языковая модель BERT, созданная исследователями в Google основанная на трансформерах, которая решает проблему моделирования двунаправленного контекста, вводя новую задачу в форме моделирования языка масок. Идея проста: вместо предсказания следующего токена в последовательности, BERT заменяет случайные слова во входном предложении специальным токеном [MASK] и пытается предсказать, каким был исходный токен. В дополнение к этому, BERT использовал мощную архитектуру Transformer, чтобы включить информацию из всего входного предложения. Для реализации данного метода использовалась библиотека Simple Transformers которая позволяет быстро обучать и оценивать модели. Из-за ограничений вычислительной мощности, обучение проходило в течение трёх эпох с гиперпараметрами, рекомендованными Google (learningrate = $3e-5$, а также batch_size = 32).

Последний реализованный метод — это языковая модель XLNet, авторегрессионная модель, которая вместо того чтобы полагаться на маскирование входных данных для обучения объединенного представления для левого и правого контекстов, как в BERT, изучает двунаправленный контекст путем максимизации ожидаемой логарифмической вероятности по всем перестановкам порядка факторизации. Поскольку перестановки могут сделать мар-

керы из левого и правого контекстов доступны с одной стороны позиций, XLNet преодолевает ограничения традиционных моделей авторегрессии, которые можно только обучить однонаправленно представлять контекст, а затем объединить их. Это также позволяет избежать потенциального расхождения в предтренировочных настройках, вызванного повреждением данных из-за маскирования токенов в BERT. Несмотря на перестановку, XLNet кодирует позиции, чтобы сохранить информацию о местоположении. XLNet использует механизм рекуррент-

ности и схему относительного позиционного кодирования Transformer-XL для изучения зависимости сверх фиксированной длины и репараметризует ее, чтобы устранить неоднозначность порядка факторизации. Модель реализована с помощью библиотеки Simple Transformers с использованием рекомендованных гиперпараметров (learningrate = 2e-5, batch_size = 32) в течении 3 эпох.

Ниже (табл. 1 и табл. 2) представлены результаты поиска лучших гиперпараметров и точности предсказания рассмотренных методов.

Таблица 1. Результаты поиска лучших гиперпараметров по сетке

	SVM	Random Forest	LSTM
TF-IDF	Kernel = Linear C = 0.75	Max Depth = 9 Min_samples_leaf = 4 Min_samples_split = 5 N_estimators = 400	Activation = sigmoid Optimizer = Adam Hidden_layers = 2 Num_Neurons = 600
W2V	Kernel = Linear C = 0.75	Max Depth = 9 Min_samples_leaf = 2 Min_samples_split = 10 N_estimators = 400	Activation = relu Optimizer = Adam Hidden_layers = 2 Num_Neurons = 600

Таблица 2. Точность предсказания (%)

	SVM	RF	LSTM	BERT	XLNet
TF-IDF	94.58	87.64	93.89	—	—
W2V	91.71	88.6	92.29	—	—
—	—	—	—	97.4	97.7

Среди двух моделей предварительной подготовки TF-IDF Vectorizer достигает в целом наилучшей производительности, а Word2Vec работает относительно плохо. Также можно заметить, что в целом нейронная сеть LSTM работает стабильнее, поскольку нейронные сети служат мощным универсальным аппроксиматором. Комбинация tf-idf vectorizer и SVM является лучшей среди всех ме-

тодов, использующих модели предварительной подготовки, и показывает точность 94.6%. Как и ожидалось, методы, основанные на языковых моделях BERT и XLNet, показали более высокую точность предсказания. Лучшие результаты показала модель XLNet, хотя стоит отметить, что время обучения модели BERT было значительно меньше.

Литература:

- Essentials of Deep Learning: Introduction to Long Short Term Memory, www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm
- Juan Ramos. «Using tf-idf to determine word relevance in document queries», December 2003, <https://www.cs.rutgers.edu/~mlittman/courses/ml03/iCML03/papers/ramos.pdf>
- Jason Brownlee. «How to Prepare Text Data for Machine Learning with scikit-learn», August 2019, <https://machinelearningmastery.com/prepare-text-data-machine-learning-scikit-learn/>
- M. Chang et al J. Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding, 2017.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, Quoc V.Le. XLNet: Generalized Autoregressive Pretraining for Language Understanding, 2019.

Молодой ученый

Международный научный журнал
№ 27 (317) / 2020

Выпускающий редактор Г. А. Кайнова
Ответственные редакторы Е. И. Осянина, О. А. Шульга
Художник Е. А. Шишков
Подготовка оригинал-макета П. Я. Бурьянов, М. В. Голубцов, О. В. Майер

За достоверность сведений, изложенных в статьях, ответственность несут авторы.
Мнение редакции может не совпадать с мнением авторов материалов.
При перепечатке ссылка на журнал обязательна.
Материалы публикуются в авторской редакции.

Журнал размещается и индексируется на портале eLIBRARY.RU, на момент выхода номера в свет журнал не входит в РИНЦ.

Свидетельство о регистрации СМИ ПИ №ФС77-38059 от 11 ноября 2009 г. выдано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций (Роскомнадзор)

ISSN-L 2072-0297

ISSN 2077-8295 (Online)

Учредитель и издатель: ООО «Издательство Молодой ученый»

Номер подписан в печать 15.07.2020. Дата выхода в свет: 22.07.2020.

Формат 60×90/8. Тираж 500 экз. Цена свободная.

Почтовый адрес редакции: 420126, г. Казань, ул. Амирхана, 10а, а/я 231.

Фактический адрес редакции: 420029, г. Казань, ул. Академика Кирпичникова, д. 25.

E-mail: info@moluch.ru; <https://moluch.ru/>

Отпечатано в типографии издательства «Молодой ученый», г. Казань, ул. Академика Кирпичникова, д. 25.