

**ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ
И ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ
В НАУКЕ, ТЕХНИКЕ И ОБРАЗОВАНИИ**

"ИНФОТЕХ – 2019"

**СБОРНИК СТАТЕЙ
всероссийской научно-
технической конференции**



г. Севастополь,

18–20 сентября 2019г.



Министерство науки и высшего образования РФ

Федеральное государственное автономное образовательное учреждение
высшего образования
«Севастопольский государственный университет»



Институт информационных технологий и
управления в технических системах
Кафедра «Информационные технологии и компьютерные системы»

СБОРНИК СТАТЕЙ
всероссийской научно-технической конференции

***ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ
И ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ
В НАУКЕ, ТЕХНИКЕ И ОБРАЗОВАНИИ***

"ИНФОТЕХ – 2019"



г. Севастополь,
18–20 сентября 2019 г.

Научный редактор:

канд. техн. наук, доцент кафедры информационных технологий и компьютерных систем (СевГУ) **Е.Н. Мащенко**

Редакционная коллегия:

Скатков А.В., д-р. техн. наук, профессор, профессор кафедры информационных технологий и компьютерных систем СевГУ;

Апраксин Ю.К., д-р. техн. наук, профессор, профессор кафедры информационных технологий и компьютерных систем СевГУ;

Моисеев Д.В., д-р. техн. наук, доцент, профессор кафедры информационных технологий и компьютерных систем СевГУ;

Лихтциндер Б.Я., д-р. техн. наук, профессор, Вице-президент Академии телекоммуникаций и информатики, профессор Поволжского государственного университета телекоммуникаций и информатики;

Дорогов А.Ю., главный научный сотрудник ПАО «Информационные телекоммуникационные технологии» («Интелтех»), д-р. техн. наук, профессор, г. Санкт-Петербург;

Брюховецкий А.А., канд. техн. наук, заведующий кафедрой информационных технологий и компьютерных систем СевГУ;

Трушин В.А., канд. техн. наук, с.н.с., заведующий кафедрой защиты информации Новосибирского государственного технического университета;

Кабанов А.А., канд. техн. наук, доцент, заведующий кафедрой информатики и управления в технических системах СевГУ;

Шевченко В.И., канд. техн. наук, доцент, доцент кафедры информационных технологий и компьютерных систем СевГУ;

Мащенко Е.Н., канд. техн. наук, доцент, доцент кафедры информационных технологий и компьютерных систем СевГУ;

Ченгарь О.В., канд. техн. наук, доцент, доцент кафедры информационных технологий и компьютерных систем СевГУ;

Ответственный за выпуск:

канд. техн. наук, доцент кафедры информационных технологий и компьютерных систем (СевГУ) **О.В. Ченгарь**

Информационные технологии и информационная безопасность в науке, технике и образовании "ИНФОТЕХ – 2019": Сборник статей всероссийской научно-технической конференции, г.Севастополь, 18–20 сентября 2019 г./ М-во науки и высшего образования РФ, Севастопольский государственный университет; науч. ред. Е.Н. Мащенко – г. Севастополь: СевГУ, 2019. – 193 с. Режим доступа: <http://lib.sevsu.ru:8080/xmlui/handle/123456789/9034> — Загл. с титул. экрана.

В сборник входят статьи по материалам докладов участников всероссийской научно-технической конференции (18–20 сентября 2019 г.), по направлениям: «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей», «Системный анализ, управление и моделирование», «Системы и средства искусственного интеллекта», «Методы и системы защиты информации, информационная и сетевая безопасность», «Компьютерная инженерия систем, сетей и компонентов», «Информационные технологии в образовании», «Информационные технологии в экологии», «Программное и аппаратное обеспечение прикладных областей». Издание рассчитано на ученых, аспирантов и студентов.

*Сборник подготовлен к изданию
кафедрой «Информационные технологии и компьютерные системы»
ФГАОУ ВО «Севастопольский государственный университет».
Научное электронное издание*

ОГЛАВЛЕНИЕ

СЕКЦИЯ 1. МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ВЫЧИСЛИТЕЛЬНЫХ МАШИН, КОМПЛЕКСОВ И КОМПЬЮТЕРНЫХ СЕТЕЙ.....	6
Лихтциндер Б.Я. МИФЫ О САМОПОДОБИИ ТЕЛЕТРАФИКА ПРИ ОБРАБОТКЕ В СМО	7
Скатков А.В. МОДЕЛЬ ВЕКТОРНОЙ ОПТИМИЗАЦИИ СЕТЕЙ КРИТИЧЕСКОГО НАЗНАЧЕНИЯ	16
СЕКЦИЯ 2. СИСТЕМНЫЙ АНАЛИЗ, УПРАВЛЕНИЕ И МОДЕЛИРОВАНИЕ...	20
Грицаенко Н.А., Шевченко В.И. ИССЛЕДОВАНИЕ СТРАТЕГИЙ УПРАВЛЕНИЯ ТРАНСПОРТНЫМИ МАРШРУТАМИ ПРИ ИСПОЛЬЗОВАНИИ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ	21
Скатков А.В., Балакирева И.А. ОПЕРАТИВНОЕ УПРАВЛЕНИЕ МНОГОНОМЕНКЛАТУРНЫМ ПРОИЗВОДСТВОМ В УСЛОВИЯХ НЕОПРЕДЕЛЕННОЙ СРЕДЫ	25
Ткаченко К.С. ОБЕСПЕЧЕНИЕ ФУНКЦИОНИРОВАНИЯ КОМПЬЮТЕРНОГО УЗЛА НА ОСНОВЕ ОЦЕНИВАНИЯ ВЕРОЯТНОСТЕЙ ГИПОТЕЗ В ЗАДАЧЕ МОНИТОРИНГА	30
Шевченко Д.Д., Ченгарь О.В., Шевченко В.И. ИСПОЛЬЗОВАНИЕ ДИСКРЕТНО-СОБЫТИЙНОГО ПОДХОДА ДЛЯ МОДЕЛИРОВАНИЯ ПРОЦЕССОВ ОБРАБОТКИ ДАННЫХ В ОБЛАЧНЫХ КОРПОРАТИВНЫХ СИСТЕМАХ.....	34
Шишкин Ю.Е., Скатков А.В. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ ВИЗУАЛИЗАЦИИ И ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ РАСПРОСТРАНЕНИЯ АНОМАЛИЙ ПРИ ТОЧЕЧНЫХ ДЕГРАДАЦИОННЫХ ВОЗДЕЙСТВИЯХ.....	39
СЕКЦИЯ 3. СИСТЕМЫ И СРЕДСТВА ИСКУССТВЕННОГО ИНТЕЛЛЕКТА...	45
Васяева Т.А., Мартыненко Т.В., Суббота Н.С. ПРОГНОЗИРОВАНИЕ ФИНАНСОВЫХ ВРЕМЕННЫХ РЯДОВ С ПОМОЩЬЮ GRU СЕТЕЙ	46
Дорогов А.Ю. БЫСТРЫЕ НЕЙРОННЫЕ СЕТИ ГЛУБОКОГО ОБУЧЕНИЯ С КОММУТАЦИЕЙ ПЛОСКОСТЕЙ	55
Евсин В.А., Евсина В.А., Широбокова С.Н. ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ МОДУЛЯ ОЦЕНКИ СТОИМОСТИ АРЕНДЫ НЕДВИЖИМОСТИ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ.....	65
Жевакин Д.М., Широбокова С.Н., Перекрестова Т.И. ИСПОЛЬЗОВАНИЕ НЕЙРОСЕТЕВОГО ПОДХОДА И ТЕХНОЛОГИЙ BIG DATA ДЛЯ ПРОГНОЗИРОВАНИЯ ПОКАЗАТЕЛЕЙ РЕАЛЬНЫХ СЕКТОРОВ ЭКОНОМИКИ СУБЪЕКТОВ РФ	70
Ситков Д.А., Шевченко В.И. ИССЛЕДОВАНИЕ МОДЕЛИ СИСТЕМЫ РАСПОЗНАВАНИЯ ЛИЦ НА ОСНОВЕ МЕТОДА ГЛАВНЫХ КОМПОНЕНТ	77
Сколота В.А., Белова И.А., Мартинович М.В., Зиновьев Г.С. ОБЗОР ТЕХНИЧЕСКИХ СРЕДСТВ РЕАЛИЗАЦИИ НЕЙРО-ФАЗЗИ-АЛГОРИТМОВ ПОЛУЧЕНИЯ ПОКАЗАТЕЛЕЙ КАЧЕСТВА ЭЛЕКТРИЧЕСКОЙ ЭНЕРГИИ	82

Шишкин Ю.Е., Скатков А.В. ИНТЕЛЛЕКТУАЛЬНАЯ ТЕХНОЛОГИЯ ОБНАРУЖЕНИЯ АНОМАЛИЙ ПАРАМЕТРОВ СЛОЖНЫХ ОБЪЕКТОВ КАК ЗАДАЧА КЛАССИФИКАЦИИ НА ОСНОВЕ ЭКСПЕРИМЕНТАЛЬНОЙ И ЭКСПЕРТНОЙ ИНФОРМАЦИИ	87
СЕКЦИЯ 4. МЕТОДЫ И СИСТЕМЫ ЗАЩИТЫ ИНФОРМАЦИИ, ИНФОРМАЦИОННАЯ И СЕТЕВАЯ БЕЗОПАСНОСТЬ	90
Бибик А.С., Савкова Е.О. WEB-ОРИЕНТИРОВАННАЯ СИСТЕМА ПРОЦЕССИНГА КРИПТОВАЛЮТНЫХ ОПЕРАЦИЙ	91
Корепанова Н.Л., Лебедева М.А., Ситников Д.А. ПРИМЕНЕНИЕ ХЭШ- ФУНКЦИЙ ДЛЯ ЗАЩИТЫ КОМПЬЮТЕРНЫХ СЕТЕЙ	94
Шумаева Е.А., Джюра Г.С. ИНФОРМАЦИОННОЕ ПРАВО КАК ИНСТРУМЕНТ ОБЕСПЕЧЕНИЯ КИБЕРБЕЗОПАСНОСТИ ГОСУДАРСТВА	98
СЕКЦИЯ 5. КОМПЬЮТЕРНАЯ ИНЖЕНЕРИЯ СИСТЕМ, СЕТЕЙ И КОМПОНЕНТОВ	104
Евсеева В.В., Мащенко Е.Н. ИСПОЛЬЗОВАНИЕ БЕСПРОВОДНЫХ СЕНСОРНЫХ СЕТЕЙ В СИСТЕМАХ БОЛЬШИХ ДАННЫХ	105
Мамошина Ю.С., Росляков А.В. ИССЛЕДОВАНИЕ ЗАДЕРЖЕК В СЕТИ LTE МЕТОДАМИ ТЕОРИИ NETWORK CALCULUS	109
Росляков А.В., Поддубнов И.В. ВИРТУАЛИЗАЦИЯ В СЕТЯХ 5G	114
СЕКЦИЯ 7. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ	118
Евсеева В.В. ОСОБЕННОСТИ ОРГАНИЗАЦИИ ДИСТАНЦИОННОГО ОБУЧЕНИЯ В СИСТЕМЕ MOODLE ДЛЯ ПРОФЕССИОНАЛЬНОГО ОБУЧЕНИЯ ...	119
Козлов П.А., Мохов А.С., Толчеев В.О. ВЫЯВЛЕНИЕ ТЕМАТИК КАФЕДРАЛЬНЫХ ПУБЛИКАЦИЙ СРЕДСТВАМИ TEXT MINING	124
Маслихов С.Р., Мохов А.С., Толчеев В.О. ПРИМЕНЕНИЕ ТЕХНОЛОГИЙ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДЛЯ ОЦЕНКИ СООТВЕТСТВИЯ НАУЧНОГО ПРОФИЛЯ КАФЕДРЫ И ТЕМАТИК ЛЕКЦИОННЫХ КУРСОВ.....	129
Николаева Ю.П. ПРИМЕНЕНИЕ НЕЙРОННОЙ СЕТИ ДЛЯ АНАЛИЗА ПЕРСПЕКТИВ ПОСТУПЛЕНИЯ СТУДЕНТОВ В МАГИСТРАТУРУ	134
Шалимова Е.М. РАЗРАБОТКА ОБУЧАЮЩЕЙ СИСТЕМЫ ПО ДИСЦИПЛИНЕ «ОПЕРАЦИОННЫЕ СИСТЕМЫ»	142
Широбокова С.Н., Диков М.Е., Сериков О.Н. ОБ АВТОМАТИЗИРОВАННОМ ПОДХОДЕ К МОНИТОРИНГУ И АНАЛИЗУ ПРОФОРИЕНТАЦИОННОЙ РАБОТЫ КОНСУЛЬТАНТОВ ВУЗА В СОЦИАЛЬНОЙ СЕТИ «ВКОНТАКТЕ»	146
Широбокова С.Н., Сериков О.Н., Диков М.Е. О ВАРИАНТЕ РЕАЛИЗАЦИИ НА ПЛАТФОРМЕ «1С:ПРЕДПРИЯТИЕ» ИНСТРУМЕНТАРИЯ АВТОМАТИЗИРОВАННОГО ФОРМИРОВАНИЯ УЧЕБНО-МЕТОДИЧЕСКОЙ ДОКУМЕНТАЦИИ НА ОСНОВЕ УЧЕБНОГО ПЛАНА.....	152
СЕКЦИЯ 8. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ЭКОЛОГИИ	158
Вилинский А.А., Мащенко Е.Н. КЛАСТЕРНЫЙ АНАЛИЗ ДАННЫХ ЭКОЛОГИЧЕСКОГО МОНИТОРИНГА ПО ЗАГРЯЗНЕНИЮ АТМОСФЕРНОГО ВОЗДУХА	159

СЕКЦИЯ 9. ПРОГРАММНОЕ И АППАРАТНОЕ ОБЕСПЕЧЕНИЕ ПРИКЛАДНЫХ ОБЛАСТЕЙ.....	164
Бутов Д.В., Ченгарь О.В. КОНЦЕПТУАЛЬНАЯ 3D-МОДЕЛЬ АВТОНОМНОЙ БЕСЕДКИ С АЛЬТЕРНАТИВНЫМ ИСТОЧНИКОМ ЭНЕРГИИ	165
Дзюба С.Н., Фисун С.Н. СРАВНЕНИЕ ЭФФЕКТИВНОСТИ АЛГОРИТМОВ ВОСХОДЯЩЕГО СИНТАКСИЧЕСКОГО АНАЛИЗА	174
Фисун С.Н., Омельченко Д.Ю. ПРОГРАММНАЯ СИСТЕМА ДЛЯ АВТОМАТИЗАЦИИ РАЗРАБОТКИ НИСХОДЯЩИХ СИНТАКСИЧЕСКИХ АНАЛИЗАТОРОВ	177
Цокало Ю.А., Машенко Е.Н., Цокало И.Е. ЭКСПЕРТНАЯ СИСТЕМА ОБРАБОТКИ РЕЗУЛЬТАТОВ ХИМИЧЕСКИХ ЭКСПЕРИМЕНТОВ	182
Шлемко А.Г., Ченгарь О.В. СРАВНИТЕЛЬНЫЙ АНАЛИЗ ЭФФЕКТИВНОСТИ ORM ПРИ ПРОЕКТИРОВАНИИ РЕЛЯЦИОННЫХ БАЗ ДАННЫХ ДЛЯ СЛОЖНЫХ ПРИЛОЖЕНИЙ	187

СЕКЦИЯ 1

Математическое и
программное обеспечение
вычислительных машин,
комплексов и
компьютерных сетей



УДК 621.391: 621 395

Б.Я. Лихтциндер, докт. техн. наук, профессор

Поволжский государственный университет телекоммуникаций и информатики

МИФЫ О САМОПОДОБИИ ТЕЛЕТРАФИКА ПРИ ОБРАБОТКЕ В СМО

Аннотация: В статье приводятся ряд причин возникновения моделей телетрафика, основанных на свойствах самоподобия. Показано, что при заданной загрузке, знание закона распределения интервалов времени между соседними заявками не полностью характеризует размер очередей, получающихся при обработке заявок в системах массового обслуживания (СМО). Показана необходимость учета корреляционных свойств исследуемого потока заявок при анализе размеров очередей и задержек в СМО. Рассмотрены основные положения предложенного автором интервального метода и, полученная на его основе обобщенная формула Хинчина-Поллачека. Рассмотрено влияние отрицательной корреляционной связи на размеры очередей.

Рассмотрены недостатки модели самоподобного трафика, и делается вывод о малой реальности ее применения.

Ключевые слова: телетрафик, самоподобие, системы массового обслуживания, очереди

Annotation: The article gives a number of causes for teletraffic models based on self-similarity properties. It is shown that for a given load, knowledge of the law of the distribution of time intervals between neighboring requests does not fully characterize the queue size resulting from the processing of requests in queuing systems (QS). The main provisions of the interval method proposed by the author and the generalized Khinchin-Pollachek formula obtained on its basis are considered. The effect of negative correlation on queue sizes is considered. The drawbacks of the model of self-similar traffic are considered, and a conclusion is made about the small reality of its use.

Key words: teletraffic, self-similarity, queuing systems, queues.

Введение

Бум вокруг самоподобия телетрафика, имевший место в 80-х годах, в связи с переходом телекоммуникаций на технологии пакетной коммутации, в настоящее время значительно поубавился.

Несмотря на огромное число теоретических работ, посвященных анализу самоподобия трафика, к сожалению, за последние 30 лет имеется весьма небольшое количество практических результатов в этом направлении.

Дело в том, что большинство работ, посвященных анализу самоподобных потоков, проводилось в отрыве от процессов их обработки в системах массового обслуживания (СМО), позволяющих определять значения задержек и размеры очередей.

Причины возникновения моделей телетрафика, основанных на свойствах самоподобия, хорошо сформулированы О. И. Шелухиным в [15] и перечислены ниже.

Поведение пользователей, которое носит циклический характер, заключающийся в чередовании периодов чтения и осмысливания информации, получаемой пользователем, и периодов передачи запросов и ответов в сеть от пользователя [16], [17].

Генерация и поиск данных компьютерными системами. Передача данных в сеть осуществляется определенными порциями, между которыми следуют периоды поиска и обработки данных компьютером.

Объединение трафика. При статистическом мультиплексировании объединенный трафик имеет свойства, существенно отличающиеся от свойств трафика, генерированного отдельными пользователями.

Раньше, при использовании технологии коммутации каналов, потоки заявок от пользователей были достаточно однородны и удовлетворяли характеристикам простейших потоков. Мультиплексирование простейших потоков приводило к возникновению суммарного простейшего потока. При переходе к пакетной коммутации в мультисервисных сетях мультиплексируются разнородные потоки, что приводит к возникновению суммарного потока, обладающего долговременной зависимостью (ДВЗ) [18], [19], [20], [21].

Ограниченность ресурсов, которая приводит к необходимости управления трафиком.

Конфликтные ситуации и перегрузки изменяют свойства трафика и вызывают потери пакетов. Особенно усложняет характер передаваемого трафика управление, основанное на введении обратных связей, вызывающих возникновение флюктуационных процессов.

Все перечисленные факторы приводят к возникновению в трафике долговременных зависимостей (ДВЗ) и установлению длительных корреляционных связей, свойственных самоподобным процессам.

Однако, далеко не всякий процесс, обладающий свойствами ДВЗ, может считаться самоподобным. Многие потоки с независимыми интервалами времени между соседними заявками, также могут иметь пачечный характер, свойственный самоподобным процессам. К их числу, например, могут быть отнесены потоки, удовлетворяющие распределению Парето. В последнее время все большее применение получают модели ВМАР - потоков [13], характеризуемые, в частности, пуассоновским распределением пачек заявок. Размеры пачек определяются соответствующими законами распределения вероятностей.

Учет корреляции

Вся классическая теория СМО основана на предположении о взаимной независимости поступающих заявок. Даже известная классификация Кендалла [2] не предусматривает возможности учета корреляционных свойств исследуемых потоков, а учитывает лишь законы распределения соответствующих вероятностей. Существует множество работ, рассматривающих различные законы распределения вероятностей интервалов между соседними заявками, не пуассоновские распределения вероятностей чисел заявок на некотором временном интервале, и ряд других. Однако, все они предполагают взаимную независимость поступающих заявок.

Если же указанные заявки взаимно коррелированы, то, даже при простейшем экспоненциальном распределении интервалов времени между соседними заявками, очереди таких заявок в СМО, при одинаковых интенсивностях поступления заявок и одинаковых коэффициентах загрузки, могут во много раз превышать очереди, создаваемые пуассоновским потоком.

В качестве примера рассмотрим реализацию простейшего стационарного потока заявок, с экспоненциальным распределением интервалов времени между соседними заявками, на длительном промежутке времени T [4].

Сгруппируем интервалы заявок таким образом, чтобы наиболее короткие интервалы оказались в начальной части промежутка времени T , а наиболее длинные - оказались в его конечной части. Естественно, что распределение временных интервалов между заявками во вновь полученном потоке не изменится и останется экспоненциальным. Однако новый поток уже не будет пуассоновский, поскольку мы ввели в него искусственно дополнительную корреляционную зависимость. Средний размер очереди для такого потока будет во много раз превышать значения, полученные по известной формуле Хинчина-Поллачека для исходного пуассоновского потока.

На рисунке 1 представлены зависимости средних размеров очередей от коэффициента загрузки для пуассоновского потока (нижний график) и того же потока, с внесенной корреляционной зависимостью интервалов времени между соседними заявками (верхний график). Нижняя кривая соответствующая пуассоновскому потоку, на широком участке почти сливается с осью координат. Верхняя кривая, соответствующая потоку с коррелированными интервалами между заявками, показывает весьма существенное возрастание значений очереди, даже при малых коэффициентах загрузки.

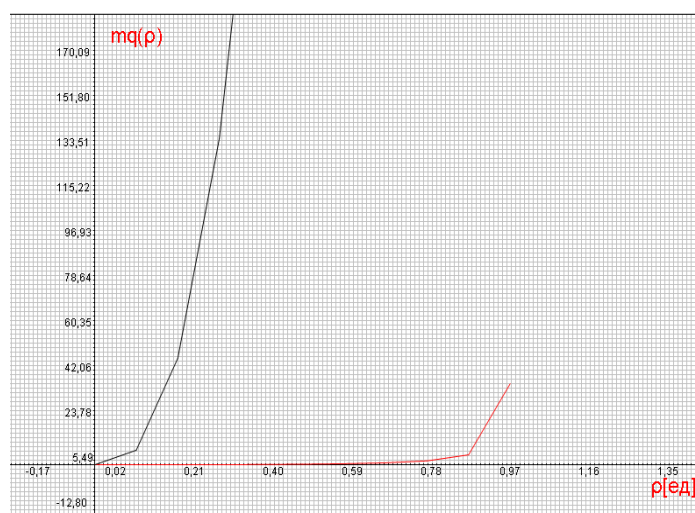


Рисунок 1. Зависимости средних размеров очередей для пуассоновского потока (нижний график) и этого же потока, с коррелированными интервалами между заявками (верхний график).

Этим примером можно показать, что, при заданном времени обслуживания, знание закона распределения вероятностей далеко не полностью определяет размер очередей, получающихся при обработке заявок в СМО. Большое (а часто определяющее) влияние на размеры очередей оказывают корреляционные свойства потоков. Наиболее ранними исследованиями корреляционных свойств потоков заявок в СМО, по-видимому, можно считать работы [5], [6], в которых рассматривается так называемый индекс дисперсии, учитывающий либо корреляционные свойства входных потоков либо корреляционные свойства интервалов обработки заявок и не учитывающий их взаимную корреляцию.

В последние годы появляются работы [7], [8] в которых также не учитываются указанные взаимные корреляционные свойства. Однако, интервалы времени обработки заявок могут сильно зависеть от заявок входного потока, и пренебрегать их взаимной корреляцией нельзя.

Почти одновременно с названными выше работами [5] и [6] была опубликована работа [9] автора, в которой приводится соотношение, обобщающее известную формулу Хинчина Поллачека и учитывающее взаимные корреляционные свойства стационарных потоков заявок общего вида.

Интервальный метод анализа

Одним из перспективных, на наш взгляд, направлений изучения пакетного трафика является интервальный метод [4], позволяющий заменить анализ интервалов времени между соседними заявками и интервалов времени обработки заявок, анализом одной случайной величины - числом заявок, поступающих в течение последовательных интервалов времени обработки заявок. Нами показано, что дисперсия и корреляционные свойства указанной случайной величины, при заданной загрузке, полностью характеризуют средний размер очереди в СМО [10], [11].

Рассмотрим стационарный, ординарный входной поток заявок. Заявки поступают в моменты времени t_i . Разместим на оси времени последовательные интервалы τ_i , соответствующие интервалам времени обработки каждой из заявок. Числа заявок, поступающих в течение каждого из интервалов времени τ_i представляют собою анализируемую случайную переменную $m_i(\tau)$. Обозначим $\overline{m(\tau)}$ - математическое ожидание, $D_m(\tau)$ - дисперсию указанной случайной величины. Если λ - это средняя

интенсивность потока заявок, а τ - математическое ожидание времени обслуживания, то $\rho = \lambda\tau$ - это коэффициент загрузки одноканальной СМО.

Не трудно показать, что для любого потока заявок справедливо равенство $\overline{m(\tau)} = \rho$. Процесс образования очередей в указанной СМО определяется уравнением баланса (1)

$$q_i(\tau) = q_{i-1}(\tau) + m_i(\tau) - \delta_i(\tau). \quad (1)$$

$$\delta_i(\tau) = \begin{cases} 0, & \text{если } q_{i-1}(\tau) = m_i(\tau) = 0; \\ 1 & \text{в противном случае.} \end{cases}$$

Здесь, $q_i(\tau)$ - размер очереди на интервале τ_i . Значения $\delta_i(\tau) = 0$ возникают на тех интервалах времени, когда на текущем интервале заявки не поступают, а на предыдущем интервале отсутствует очередь. В течение интервалов, когда $\delta_i(\tau) = 0$, обработка заявок в процессоре не происходит и система простаивает. Случайная величина $\delta_i(\tau)$ также имеет математическое ожидание, равное коэффициенту загрузки ρ :

$$\overline{\delta_i(\tau)} = \overline{m_i(\tau)} = \rho \quad (2)$$

Следовательно, величина $m_i(\tau) - \delta_i(\tau)$ является центрированной.

На основании (1) для среднего числа заявок, находящихся в очереди $\overline{q(\rho)}$ было получено соотношение, обобщающее известную формулу Хинчина-Поллачека, и справедливое для стационарных потоков общего вида, в том числе и для самоподобного потока.

$$\overline{q(\rho)} = \frac{D_m(\rho) + 2D_m(\rho) \sum_{k=1}^{\infty} r_{m-\delta_i}(k, \rho)}{2(1-\rho)} - \frac{\rho}{2}. \quad (3)$$

Здесь, $r_{m-\delta_i}(k, \rho)$ нормированная корреляционная функция потока заявок $m_i(\rho) - \delta_i(\rho)$ при заданном значении коэффициента загрузки ρ и сдвиге на k интервалов, а $D_m(\rho)$ - дисперсия указанного потока.

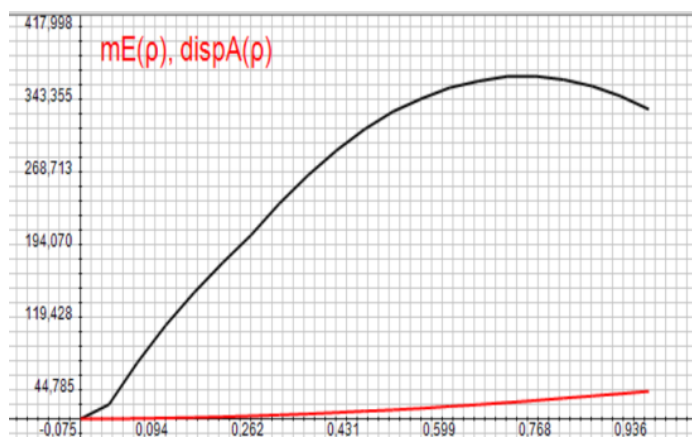


Рисунок 2. Зависимость дисперсии и числителя формулы (3), от коэффициента загрузки для видеотрафика

На Рисунке 2 показана зависимость дисперсии от коэффициента загрузки для видеотрафика (нижний график) и зависимость числителя формулы (3), обусловленная наличием корреляционных связей в потоке видеотрафика (верхний график). Эксперименты показали, что для большинства видов трафика мультисервисных сетей доступа, значения второго члена числителя в формуле (3), обусловленного наличием корреляционных связей в потоке, существенно превышают значения дисперсии, и пренебрегать ими нельзя.

Величину $J_{m-\delta}(k, \rho) = 1 + 2 \sum_{k=1}^{\infty} r_{m-\delta_i}(k, \rho)$ называют индексом дисперсии [6]. Он учитывает влияние корреляционных свойств интервального потока на среднее значение очереди в СМО.

$$\overline{q(\rho)} = \frac{D_m(\rho) \cdot J_{m-\delta}(k, \rho)}{2(1-\rho)} - \frac{\rho}{2}.$$

Для интервальных потоков, в которых корреляционные связи между элементами отсутствуют, индекс дисперсии равен единице, и средний размер очереди, при данной загрузке, определяется исключительно значением дисперсии $D_m(\rho)$ интервального потока.

Величину $D_E(\rho) = D_m(\rho) \cdot J_{m-\delta}(k, \rho)$ назовем эквивалентной дисперсией интервального потока. Она определяет средний размер очереди, при данной загрузке СМО.

Два различных потока имеют одинаковые размеры средних очередей, если для них одинаковы значения эквивалентных дисперсий.

Отрицательная корреляция

Из определения индекса дисперсии следует, что отрицательная корреляционная связь уменьшает значение эквивалентной дисперсии и, следовательно, уменьшает размеры очередей.

На Рисунке 3 показаны графики зависимостей средних размеров очередей для трех потоков, имеющих одинаковые распределения вероятностей интервалов между соседними заявками, удовлетворяющие распределению Парето. Средняя кривая соответствует независимым интервалам между заявками. В поток, соответствующий верхнему графику, была внесена искусственно положительная корреляционная зависимость между интервалами соседних заявок. Она заключалась в том, что короткие интервал сгруппировали в начальной части анализируемого периода T , а длинные - интервалы - сгруппированы в конце. Нижний график соответствует внесению отрицательной корреляции, обусловленной чередованием коротких и длинных интервалов.

Следует подчеркнуть, что распределения вероятностей интервалов между соседними заявками для всех трех потоков одинаковые и соответствуют распределению Парето. При увеличении загрузки наблюдаем существенную разницу между значениями очередей.

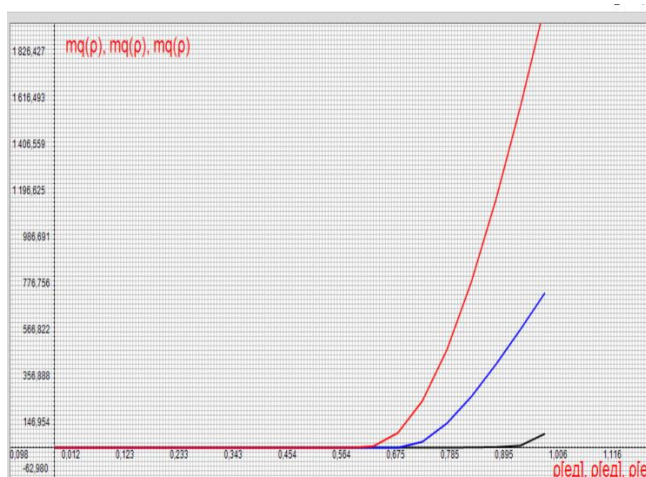


Рисунок 3. Графики зависимостей средних размеров очередей для трех потоков, имеющих одинаковые распределения Парето вероятностей интервалов между соседними заявками

Самоподобный трафик

Измерения на реальных компьютерных сетях и измерения потоков в пакетных сетях передачи данных показали, что передаваемый в них трафик имеет свойства самоподобного случайного процесса. Анализ самоподобных процессов телетрафика посвящено множество публикаций. На качественном уровне самоподобие проявляется в том, что имеется медленно убывающая зависимость между величинами трафика в разные моменты времени, а также в том, что трафик носит пачечный характер. Эти пачки выглядят статистически подобными в различных масштабах интервалов времени [14],

Одной из наиболее фундаментальных отечественных работ в указанном направлении является монография [12], определения которой мы рассмотрим применительно к анализируемому нами случайному процессу.

Рассматриваемый процесс $m_i(\tau)$, $i = \overline{0, N}$ является стационарным и дискретным по времени τ , которое необходимо для обработки одной заявки.

Обозначим через $m_{i-k}(\tau)$ – элемент, сдвинутый на k интервалов τ по отношению к $m_i(\tau)$. Значение корреляционной функции (корреляционный коэффициент) при сдвиге на k интервалов τ обозначим через

$$\mu_m(k, \tau) = \overline{[m_i(\tau) - \overline{m(\tau)}][m_{i-k}(\tau) - \overline{m(\tau)}]}.$$

Очевидно, что дисперсия $D_m(\tau) = \mu_m(0, \tau)$ – это корреляционный коэффициент при нулевом сдвиге ($k = 0$).

Нормированная корреляционная функция:

$$r_m(k, \rho) = \frac{\mu_m(k, \rho)}{D_m(\rho)}.$$

Согласно определениям, принятым в литературе [13], статистический процесс $m_i(\tau)$ называется самоподобным, с параметром самоподобия H , (где $0,5 < H < 1$), если ему соответствует корреляционная функция

$$r_m(k, \tau) = [(k+1)^{2H} - 2k^{2H} + (k-1)^{2H}] \quad (4)$$

Процесс с указанной корреляционной функцией демонстрирует долговременную зависимость, если коэффициент H , называемый параметром Херста, удовлетворяет условию $0,5 < H < 1$. В этом случае

$$\frac{\lim r_m(k, \tau)}{k^{2(1-H)}} = \frac{\lim r_m(k, \tau)}{k^\beta} \rightarrow C, \text{ при } k \rightarrow \infty \quad (5)$$

где $\beta = 2(1-H)$ – коэффициент, меняющийся в пределах $0 < \beta < 1$, а C – некоторая медленно меняющаяся функция. Не трудно показать, что при $H = 0,5$, это отношение убывает, пропорционально первой степени увеличения коэффициента сдвига k .

Для самоподобных процессов убывание корреляционной функции происходит намного медленнее. Поскольку $\mu_m(k, \tau)$ является корреляционной функцией центрированного процесса, значения медленно меняющейся функции C – близки к нулю.

Если выполняется соотношение $0,5 < H < 1$, то самоподобный процесс обладает свойством долговременной зависимости. Из (5) следует, что величина $r_m(k, \tau)$ убывает по степенному закону. Это означает, что корреляционная функция является не суммируемой, и ряд, образованный ее последовательными значениями расходится.

$$\sum_{k=1}^{\infty} r_m(k, \rho) \rightarrow \infty \quad (6)$$

Но, в формуле (3), сумма корреляционных коэффициентов в числителе является конечной, и, следовательно, поток заявок, образующих очередь, не может считаться самоподобным. В чем же – противоречие?

В принятых выше рассуждениях считалось, что случайная величина, образующая самоподобный поток, центрируется с помощью вычитания ее математического ожидания.

В действительности, разностная переменная $m_i(\tau) - \delta_i(\tau)$ центрируется путем вычитания из $m_i(\tau)$ случайной величины $\delta_i(\tau)$. При этом, реальная случайная величина $m_i(\tau)$, представляющая числа заявок, поступающих в течение интервалов обслуживания одной заявки, имеет математическое ожидание $\overline{m_i(\tau)} = \overline{\delta_i(\tau)} = \rho < 1$.

Поскольку величина $\delta_i(\tau)$ может принимать только единичные или нулевые значения, на интервалах, где $\delta_i(\tau) = 0$ заявки и очереди в системе полностью отсутствуют, и ее дальнейшее поведение не зависит от предыдущих значений $m_i(\tau)$. Таким образом, с целью анализа очередей, создаваемых в СМО самоподобными процессами, необходимо учитывать, что корреляционные свойства процесса, с точки зрения образования очередей, определяются разностной переменной $m_i(\tau) - \delta_i(\tau)$, которая не удовлетворяет свойствам самоподобия. Долговременная корреляционная зависимость $m_i(\tau)$ компенсируется корреляционной зависимостью величины $\delta_i(\tau)$. Процесс, соответствующий указанной переменной, имеет временные интервалы, на которых $m_i(\tau) = \delta_i(\tau) = 0$, прерывающие корреляционную связь между предыдущими и последующими частями очереди. Следовательно, ни о какой долговременной зависимости здесь говорить не приходится. Она компенсируется и длится лишь в течение активной части каждого цикла образования очереди. Это еще раз подтверждает, что самоподобный трафик, имеющий долговременную корреляционную зависимость, при обработке в СМО указанную зависимость теряет, соотношение (6) не выполняется и сумма сходится к некоторой постоянной величине C :

$$\sum_{k=1}^{\infty} r_{m-\delta_i}(k, \rho) = C.$$

Именно, значения указанной величины, с учетом того, что $\lambda\tau = \rho$, определяют средний размер очереди по обобщенной формуле (1). На свойства разностной величины, аналогичной $[m_i(\tau) - \delta_i(\tau)]$, обращал внимание еще Л. Клейнрок, в своей монографии [2], указывая, что именно эта случайная величина полностью определяет характер очередей в СМО.

Итак, суммарный поток заявок, образующих очередь в СМО, учитывающий заявки, покидающие очередь, не удовлетворяет требованиям самоподобия, даже в тех случаях, когда поток входных заявок этим требованиям удовлетворяет. Именно указанное отличие является одной из причин недостаточной эффективности применения модели самоподобного трафика при анализе очередей и задержек в СМО.

Сложность модели самоподобного трафика

Второй причиной недостаточной эффективности применения модели самоподобного трафика является ее сложность.

В середине прошлого столетия появилось множество работ, посвященных анализу самоподобных процессов. Однако, в указанных работах такие процессы рассматривались в отрыве от их анализа в составе СМО и не привели к существенным практическим результатам. Причины этого факта изложены в прекрасном обзоре В. М. Вишневого и А. Н. Дудина [13].

Главный недостаток модели самоподобных потоков с точки зрения их использования при аналитическом моделировании процессов передачи информации в телекоммуникационных сетях, состоит в том, что самоподобный поток сам по себе является достаточно сложным объектом.

"Поэтому аналитическое исследование не только самого потока, но и СМО, в которую он поступает, является мало реальным". С этим выводом трудно не согласиться.

Заключение

Отсутствие весомых практических результатов подтверждает малую реальность аналитического исследования СМО с самоподобными входными потоками и ставит под сомнение миф о преимуществах модели самоподобных процессов при анализе трафика телекоммуникационных сетей.

Библиографический список:

1. Степанов С.Н. Теория телетрафика. Концепции, модели, приложения. М.: Горячая линия-Телеком. 2015. 808 с.
2. Клейнрок Л. Вычислительные системы с очередями. Т. 2. Пер. с англ. М.: Мир. 1979.
3. Лихтциндер Б.Я. Интервальный метод анализа мультисервисного трафика сетей доступа // Электросвязь. №12. 2015. С. 52–54.
4. Лихтциндер Б.Я. Трафик мультисервисных сетей доступа (интервальный анализ и проектирование). М.: Горячая линия – Телеком, 2018 г. 290 с.
5. Jagerman D.L., Balcioglu B., Altioek T., Melamed B. Mean Waiting Time Approximations in the G/G/1 Queue // Queueing Systems 46, 481-506, 2004.
6. Balcioglu B., Jagerman D.L., Altioek T. Approximate mean waiting time in a GI/D/1 queue with autocorrelated times to failures // IIE Transactions, 39 10, 985-996, 2007.
7. Карташевский И. В., Сапрыкин А. В. Обработка коррелированного трафика в узле сети типа G/G/1. // Радиотехника, 2017, № 10. – С. 119-125.
8. Карташевский И. В. Модель трафика для программно-конфигурируемых сетей // Радиотехника, №6, 2016, с.124-129.

9. Лихтциндер Б. Я. О некоторых обобщениях формулы Хинчина-Полллячека // ИКТ. Т.5, №4, 2007. – С.253-258.
10. Лихтциндер Б.Я. Корреляционные свойства длин очередей в системах массового обслуживания с потоками общего вида // ИКТ. 2015. Т.13. №3. С. 276–280.
11. Лихтциндер. Б.Я. Корреляционные связи в пачечных потоках систем массового обслуживания // Телекоммуникации. 2015. № 9 С. 8–12.
12. Шелухин О.И., А.М Тенякшев., А.В. Осин, Фрактальные процессы в телекоммуникациях / Под ред. Шелухина О.И.. - М.: Радиотехника, 2003, 480с.
13. Вишневский В. М., Дудин А. Н., Системы массового обслуживания с коррелированными входными потоками и их применение для моделирования телекоммуникационных сетей, *Автомат. и телемех.*, 2017, № 8, С. 3–59; *Autom. Remote Control*, 78:8 (2017), 1361–1403.
14. Цыбаков Б. С., Модель телетрафика на основе самоподобного случайного процесса, *Радиотехника*, 1999, № 5, с.24–31.
15. Шелухин О.И.-Причины самоподобия телетрафика и методы оценки показателя Херста.//*Электротехнические и информационные комплексы и системы* . 2007. №1,т.3 / . 5-13.
16. Jena A.K., PruthiP.,Popesku ., Resource Engineering for Internet Applications, *Proceedings of the 7th IFIP ATM Workshop*. Antverp, Belgium, June 1999.
17. Crovella M. E.,Bestavros A., Explaining World Wide Web Traffic Self-Similarity. Technical Report:TR-95015, Computer Science Department, Boston University, 1995.
18. Erramilli A., Narayan O., Willinger W., Experimental queueing analysis with long-range dependent packet traffic, *IEEE/ACM Transactions on Networking*, 4:209-223, 1996.
19. Li G. L., Dowd W.D., An Analysis of Network Performance Degradation Induced by Workload Fluctuations, *IEEE/ACM Transactions on Networking*, Vol.3, №4, August 1995.
20. Erramilli A., Willinger W., Wang J.L., Modeling and Management of Self-Similar Traffic Flows in High-Speed Networks, *Network Systems Design*, Gordon and Breach Science Publishers, 1999.
21. Park K., Kim G.T., Crovella M.E., On the Relationship Between File Sizes, Transport Protocols, and Self-Similar Network Traffic, preprint, Boston University, 1996.

B.Ya. Likhtsinder Doctor of Technical Sciences, Professor of the Department of NST
Volga State University of Telecommunications and Informatics

MYTHS ABOUT SELF-LIKE TELETRAFFIC IN PROCESSING IN QS

УДК 004.9:519.81

А.В. Скатков, докт. техн. наук, профессор

Севастопольский государственный университет

МОДЕЛЬ ВЕКТОРНОЙ ОПТИМИЗАЦИИ СЕТЕЙ КРИТИЧЕСКОГО НАЗНАЧЕНИЯ

Аннотация: представлен комплекс задач векторной оптимизации в сетях критического назначения проиллюстрированный на примере решения задачи по минимизации стоимости передачи данных, который является основой системы поддержки принятия решений по выбору оптимальных параметров функционирования сложных систем сетей критического назначения

Ключевые слова: мониторинг, математическое моделирование, интеллектуальный анализ данных, сети критического назначения

Annotation: A complex of vector optimization tasks in critical-use networks is presented, illustrated by the example of solving the problem of minimizing the cost of data transmission, which is the basis of the decision support system for choosing the optimal parameters for the operation of complex systems of critical-use networks

Key words: monitoring, mathematical modeling, data mining, critical networks

На сегодняшний день сети критического назначения являются неотъемлемым компонентом городских инфраструктур, обеспечивающих высокое качество жизни социума и в особенности это важно и нужно для информационного обеспечения технологий умного города, использующих беспилотные транспортные средства, решения задач мониторинга городской инфраструктуры, медицинского обслуживания и др [1-3]. Внедрение новых способов коммуникации и связи элементов инфраструктур, а также эволюционная замена традиционных каналов связи сетями новых поколений, помогают использовать имеющиеся ограниченные ресурсы и время более рационально и производительно [4,5].

Высокий динамизм структурных элементов сетей критического назначения и их комбинаций, смена их состояний, широкий диапазон параметрических рядов, множественная вариативность настроек аппаратно-программных компонент делают задачи моделирования, оптимизации и принятия решений чрезвычайно сложными как в системном плане, так и в аспекте конкретной математической аналитики [6,7]. В рамках данной работы предлагается математическая модель для распределительных задач векторной оптимизации по совокупности скалярных критериев: стоимость передачи информации, время доставки сообщений, помехоустойчивость, криптостойкость, которые являются критически важными для сетей критического назначения. Таким образом, выбор управляющих решений по проектированию, настройке, масштабированию и эксплуатации сетей критического назначения, становятся потенциально реализуем в рамках оптимизационных решений, полученных посредством предлагаемой модели. Вне рамок данной работы пока остаются вопросы SWOT-анализа выбранной концепции построения предлагаемых оптимизационных моделей для сетей критического назначения (SWOT: Strengths, Weakness, Opportunities, Threats), но исследования в этих направлениях будут продолжены.

Наиболее общей моделью для задач информационной логистики, по видимому, является двудольные графы $G=\langle A, B \rangle$ с множествами вершин (долями графа) X и Y ($X, Y \neq \emptyset, X \cup Y = V, X \cap Y = \emptyset$), каждое ребро G инцидентно определенной вершине из X и некоторой вершине из Y . В графе G важен основной структурный элемент, подлежащий применению при решении оптимизационных задач, - паросочетание - этот подграф графа G , состоящий из двух смежных вершин и ребра между ними. Множеству паросочетаний далее будет ставиться в соответствии бинарная матрица элементов, подлежащих определению в процессе решения оптимизационной задачи. Известным образом определяется максимальная и совершенное паросочетания, играющие важную роль в построении алгоритмических процедур, ориентированных на программную реализацию. Каждое ребро двудольного графа, определяющее некоторое паросочетание, является элементом гостевого трафика,

обеспечивающего прохождение информации по открытому каналу связи с использованием функционального маршрута.

Базовая задача (1) полимодального комплекса по минимизации стоимости передачи данных C при выполнении ограничений (2) (на время передачи информации, вероятность безотказной работы и уровень помехозащищенности) путем выбора таких элементов четырехмерной матрицы допустимых решений X (3) и выполнения условия (4) которые обеспечивают минимум критериальной функции:

$$C = \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \sum_{l \in L} c_{ijkl} \cdot w_{ijkl} \cdot x_{ijkl} = \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \sum_{l \in L} (cs_{ijkl} + ck_{ijkl} + cf_{ijkl}) \cdot w_{ijkl} \cdot x_{ijkl} \rightarrow \min, \quad (1)$$

где cs_{ijkl} – стоимость подготовки единицы объема информации при передаче от i -ого источника к k -му приемнику посредством j -го канала с использованием кода l ; ck_{ijkl} – стоимость использования выбранного канала при передаче единицы объема информации от i -ого источника к k -му приемнику посредством j -го канала с использованием кода l ; cf_{ijkl} – стоимость обеспечения приема единицы объема информации при передаче от i -ого источника к k -му приемнику посредством j -го канала с использованием кода l .

$$\left\{ \begin{array}{l} T = \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \sum_{l \in L} t_{ijkl} \cdot x_{ijkl} = \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \sum_{l \in L} (ts_{ijkl} + tk_{ijkl} + tf_{ijkl}) \cdot x_{ijkl} < T_0 \\ P = \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \sum_{l \in L} \exp(-\lambda s_{ijkl} - \lambda k_{ijkl} - \lambda f_{ijkl}) \cdot t_{ijkl} > P_0 \\ Q = \sum_{l \in L} \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} (KSL_{ijkl} + KKL_{ijkl} + KfL_{ijkl}) \cdot x_{ijkl} \cdot t_{ijkl} > K_0 \\ \left(\sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \sum_{l \in L} x_{ijkl} \right) \cap (x_{ijkl} \in B) = 1 \end{array} \right. \quad (2)$$

где A – множество узлов – источников (передатчиков) информации $A\{a_i\}$, $i = 1, \dots, |I_0|$ или $i \in I_0$; B – множество доступных каналов передачи информации $B\{b_{ij}\}$, $j = 1, \dots, |J_0|$ или $j \in J_0$; C – множество узлов – потребителей (приемников) информации $C\{c_j\}$, $k = 1, \dots, |K_0|$ или $k \in K_0$; D – множество потенциально доступных средств шифрования $D\{d_l\}$, $l = 1, \dots, |L_0|$ или $l \in L_0$, I_0, J_0, K_0, L_0 – опорные множества допустимых индексов.

Усредненные характеристики комплекса оптимизационных задач: W_{ijkl} – объем передачи данных от i -ого источника к k -му приемнику посредством j -го канала за операционную единицу времени с использованием шифра l ; V_{ijkl} – скорость передачи данных от i -ого источника к k -му приемнику посредством j -го канала с использованием шифра l ; t_{ijkl} – время передачи данных от i -ого источника к k -му приемнику посредством j -го канала с использованием шифра l ; C_{ijkl} – стоимость передачи единицы объема данных от i -ого источника к k -му приемнику посредством j -го канала с использованием шифра l ; λs_{ijkl} – интенсивность отказов (сбоев) при подготовке информации от i -ого источника к k -му приемнику посредством j -го канала с использованием шифра l ; λk_{ijkl} – интенсивность отказов (сбоев) в канале при передаче информации от i -ого источника к k -му приемнику посредством j -го канала с использованием шифра l ; λf_{ijkl} – интенсивность отказов (сбоев) при приеме информации от i -ого источника к k -му приемнику посредством j -го канала с

использованием шифра l ; $p_{ijkl}(t) = p_{s_{ijkl}}(t) \cdot p_{k_{ijkl}}(t) \cdot p_{f_{ijkl}}(t)$ – вероятность безотказной передачи информации от i -ого источника к k -му приемнику посредством j -го канала с использованием шифра l ; $p_{s_{ijkl}}(t)$ – вероятность безотказной работы источника информации; $p_{k_{ijkl}}(t)$ – вероятность безотказной работы канала связи; $p_{f_{ijkl}}(t)$ – вероятность безотказной работы приемника информации.

Матрица допустимых решений X первоначально неизвестных x_{ijkl} , подлежащих определению в результате решения комплекса оптимизационных задач. Элементы матрицы X являются булевыми величинами, т.е. $x_{ijkl} \in B$, где B – множество булевых чисел которые равны единице, если выбрано для реализации допустимое ребро в двудольном графе G рассматриваемой задачи, и равен нулю в противном случае. Структура матрицы X существенно неоднородна, геометрически ее можно интерпретировать как четырехмерный параллелепипед размерности $I_0 \cdot J_0 \cdot K_0 \cdot L_0$, содержащий только координаты целочисленных точек принадлежащих ему. Исходя из содержательного смысла для элементов матрицы X обязательно выполнение одновременно следующих условий равенств:

$$\left\{ \begin{array}{l} \forall l \in L : \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} x_{ijkl} = 1 \\ \forall k \in K : \sum_{i \in I} \sum_{j \in J} \sum_{l \in L} x_{ijkl} = 1 \\ \forall j \in J : \sum_{i \in I} \sum_{l \in L} \sum_{k \in K} x_{ijkl} = 1 \\ \forall i \in I : \sum_{l \in L} \sum_{j \in J} \sum_{k \in K} x_{ijkl} = 1 \end{array} \right. \quad (3)$$

Проинтерпретируем для примера последнее равенство системы (3): для каждого источника информации необходимо осуществить выбор единственных (уникальных для него) приемник информации, канал передачи и способ кодирования. Равенства интерпретируются как единственность использования для каждой пары передатчик–приемник выбранных канала передачи $k \in K_0$ и кода $l \in L_0$ т.е. в каждой размерности четырехмерной матрицы X имеется один и только один единичный элемент $x_{ijkl} \in B$, где B – множество булевых чисел. Для краткости в моделях условие (3) совместно с условием $x_{ijkl} \in B$ будем обозначать как

$$\left(\sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \sum_{l \in L} x_{ijkl} \right) \cap (x_{ijkl} \in B) = 1. \quad (4)$$

Аналогичным образом можно построить следующие целевые критерии T , P , K :

Минимизация времени передачи сообщений T путем выбора таких элементов матрицы X при выполнении ограничений на суммарную стоимость передачи информации, вероятность безотказной работы, уровень криптостойкости и условия (4):

$$T = \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \sum_{l \in L} t_{ijkl} \cdot x_{ijkl} = \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \sum_{l \in L} (ts_{ijkl} + tk_{ijkl} + tf_{ijkl}) \cdot x_{ijkl} \rightarrow \min.$$

Минимизация вероятности потерь сообщений P путем выбора оптимального источника сообщений, канала связи, приемника сообщений и способа шифрования при

выполнении ограничений на суммарную стоимость передачи информации, время передачи информации, уровень криптостойкости и условия (4):

$$P = \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \sum_{l \in L} \exp(-\lambda s_{ijkl} - \lambda k_{ijkl} - \lambda f_{ijkl}) \cdot t_{ijkl} \rightarrow \max.$$

Максимизация криптостойкости каналов связи K путем выбора такого шифра l при выполнении ограничений на суммарную стоимость передачи информации, время передачи информации, вероятность безотказной работы и условия (4):

$$K = \sum_{l \in L} \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} (KSL_{ijkl} + KKLk_{ijkl} + KfL_{ijkl}) \rightarrow \max.$$

Решение задачи векторной оптимизации можно осуществить одним из двух способов: балансировкой критериальных функций или методом построения множеств Парето. Решение этой задачи, если оно существует, позволяет найти план использования заданной сети каналов с таким назначением каналов передачи информации между всеми источниками и приемниками, который минимизирует стоимость телекоммуникации при обеспечении требуемого уровня отказоустойчивости, помехозащищенности, а также времени передачи.

Предлагаемая модель задач векторной оптимизации в сетях критического назначения, проиллюстрированная на примере решения задачи по минимизации стоимости передачи данных, может быть расширена для решения задач улучшения показателей времени доставки сообщений, помехоустойчивости, криптостойкости, безотказности. Представленный комплекс оптимизационных задач является основой для системы поддержки принятия решений по выбору оптимальных параметров систем сетей критического назначения, составляющих основу информационного обеспечения умного города и сетей критического назначения.

Библиографический список:

1. Блануца В.И. Территориальная структура цифровой экономики России: предварительная делимитация "умных" городских агломераций и регионов // Пространственная экономика. 2018. № 2. С. 17–35.
2. Назаров А. Н., Сычев К. И. Модели и методы расчета показателей качества функционирования узлового оборудования и структурно-сетевых параметров сетей связи следующего поколения. – Красноярск: Изд-во ООО «Поликом», 2010. 389 с.
3. Скатков А.В., Шишкин Ю.Е. Модель обнаружения аномалий в наблюдениях параметров полей окружающей среды с использованием систем мониторинга // Системы контроля окружающей среды. 2017. № 10 (30). С. 48–53.
4. Куприяновский В.П., Аленков В.В., Першин А.В. Умные моногорода, как зоны экономического развития цифровой экономики // International Journal of Open Information Technologies. 2018. Т. 6. № 1. С. 46–91.
5. Шишкин Ю.Е., Скатков А.В. Акторная модель мониторинга с использованием мобильных облачных микросервисов // Системы контроля окружающей среды. 2018. № 14 (34). С. 56–62. DOI: 10.33075/2220-5861-2018-4-56-62
6. Линец Г. И. Системные аспекты теории синтеза и практика построения телекоммуникационных сетей. Ставрополь: Альфа-Принт, 2010. 460 с.
7. Голуб Б.В., Кузнецов Е.М., Максимов Р.В. Методика оценки живучести распределенных информационных систем // Вестник Самарского университета. Естественная серия. 2014. № 7 (118). С. 221–232.

СЕКЦИЯ 2

Системный анализ,
управление и
моделирование



УДК 004.048

Н.А. Грицаенко, магистрант**В.И. Шевченко, канд. техн. наук, доцент***Севастопольский государственный университет***ИССЛЕДОВАНИЕ СТРАТЕГИЙ УПРАВЛЕНИЯ ТРАНСПОРТНЫМИ
МАРШРУТАМИ ПРИ ИСПОЛЬЗОВАНИИ ИМИТАЦИОННОГО
МОДЕЛИРОВАНИЯ**

Анотация: Для поиска эффективных стратегий управления транспортными потоками общественного транспорта г. Севастополя, оптимальных решений по проектированию улично-дорожной сети и организации дорожного движения разработана программная модель средствами имитационного моделирования.

Annotation: To search for effective strategies for managing public transport traffic in the city of Sevastopol, optimal solutions for designing a road network and traffic management, a software model has been developed using simulation tools.

В городе Севастополе безопасность дорожного движения, эффективность автомобильных перевозок и качество работы общественного транспорта в большей степени определяется качеством организации дорожного движения, в свою очередь качество организации дорожного движения зависит от управления транспортными и пешеходными потоками, а также от регулирования пропускной способности автодорог города. Незнание природы их и характера ограничивает возможности планирования рациональных мероприятий по организации дорожного движения, их оптимизации и оперативной коррекции в соответствии с изменившимися условиями. В крупных городах данная проблема приобретает особую остроту. Ситуация усложняется такими тенденциями, как постоянно возрастающая мобильность населения, уменьшение перевозок общественным транспортом и увеличение перевозок личным транспортом, нарастающий разрыв между увеличением количества автомобилей и протяженностью улично-дорожной сети, не рассчитанной на современные транспортные потоки [1].

Стремительный рост автомобильного транспорта среди населения, большое количество пробок на автомобильных дорогах разных городов, изменения структуры трафика и его рост ужесточают требования к разрабатываемым вновь и модернизируемым системам автомобильных дорожных сетей городов.

В связи с этим возникают задачи выбора лучшей стратегии для организации дорожного трафика и дорожных сетей города согласно имеющимся потребностям населения и возможностям городских властей. Важную роль играет выбор конфигурации дорожной сети и получение прогноза о качественных показателях при функционировании в конкретных условиях, нахождения новых, более экономичных вариантов организации маршрутных транспортных перевозок, которые характеризовались бы оптимальной избыточностью и универсальностью. Решение всех этих задач невозможно без предварительной оценки принятых решений и сравнения возможных вариантов между собой.

Исследование работы дорожных сетей города очень сложно и дорого осуществлять на опытном уровне, таким образом исследование проще, быстрее и дешевле проводить либо помощи моделирования (т.е. исследования и анализа модели дорожной сети города). Дорожная сеть города является сложным техническим и социальным объектом и создание опытных образцов требует немалых затрат, а иногда этого и достичь невозможно. Моделирование позволяет уменьшить затраты на исследование системы, а также на оценку проектных и коммерческих решений [2].

В рамках представленной работы рассматривается программная система имитационного моделирования фрагмента улично-дорожной сети г. Севастополя. Программная система состоит из 4-х окон интерфейса, каждое окно отвечает за конкретный

участок дорожной сети. Редактирование параметров транспортной сети реализовано в интерфейсе пользователя для адаптивной настройки стратегий управления транспортными маршрутами.

Входными параметрами для моделирования являются: виды транспортных агентов модели (виды транспортных средств, используемых в модели); количество генерируемых транспортных агентов по типам для одного эксперимента; количество генерируемых в модели агентов-пассажиров транспортных средств; интервалы между приходами транспортных агентов на маршрут (по типам агентов); вероятности выбора агентом-пассажиром определенного транспортного маршрута. В качестве результатов работы модели могут быть отображены следующие параметры: среднее время работы транспортного агента на маршруте; среднее время ожидания транспортного агента агентом-пассажиром; среднее число пассажиров, которые использовали определенный маршрут.

Далее, в качестве иллюстрации, рассмотрен маршрут автобуса № 120 г. Севастополя, включающий 21 остановку, протяженность маршрута около 12 км: Урожайная улица - улица Паши Ангелиной - Балаклавское шоссе - проспект Генерала Острякова - улица Олега Кошевого - улица Николая Музыки - улица 4-ая Бастионная - улица Ленина - проспект Нахимова - Большая Морская улица - улица Гоголя.

На рисунке 1 представлен интерфейс основного окна модели. Здесь пользователь задает входные параметры системы и видит перед собой топологию исследуемой транспортной сети. Можно наблюдать за имитационным моделированием маршрутной транспортной сети. Для построения дорожной сети использовались Яндекс.Карты с настройкой реального масштаба. В любой момент времени менять входные параметры, наблюдая за тем, какое влияние они оказывают на систему.

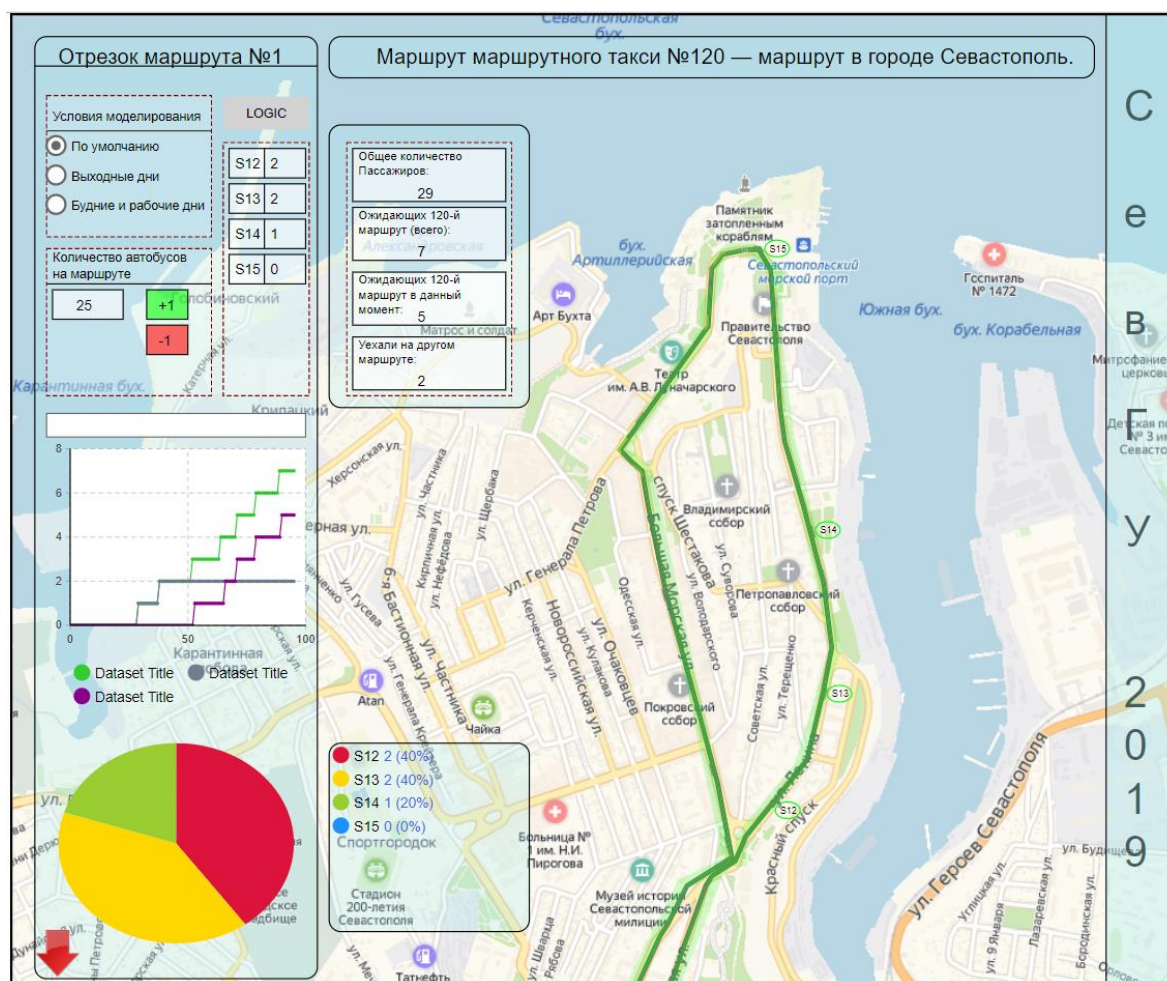


Рисунок 1 – Интерфейс программной системы «Отрезок маршрута №1»

На рисунке 2 показано окно настройки второй части исследуемого маршрута. Переключение между окнами происходит с помощью кнопок, к которым привязаны ссылки переходов к нужным окнам системы.

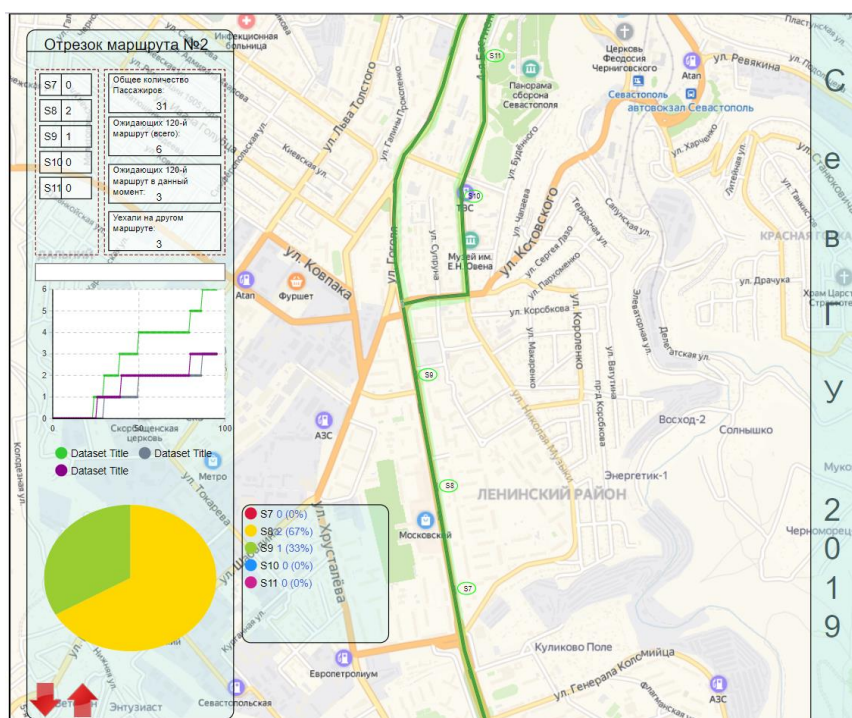


Рисунок 2 – Интерфейс программной системы «Отрезок маршрута №2»

В данном окне пользователь может наблюдать результаты тестового эксперимента по второму отрезку маршрута, но управление входными параметрами осуществляется только в первом окне, при запуске программы. Параметры системы можно корректировать в любой момент работы системы.

На рисунке 3 показано следующее окно программной системы имитационного моделирования, на котором представлен финальный участок маршрута и результаты работы модели.

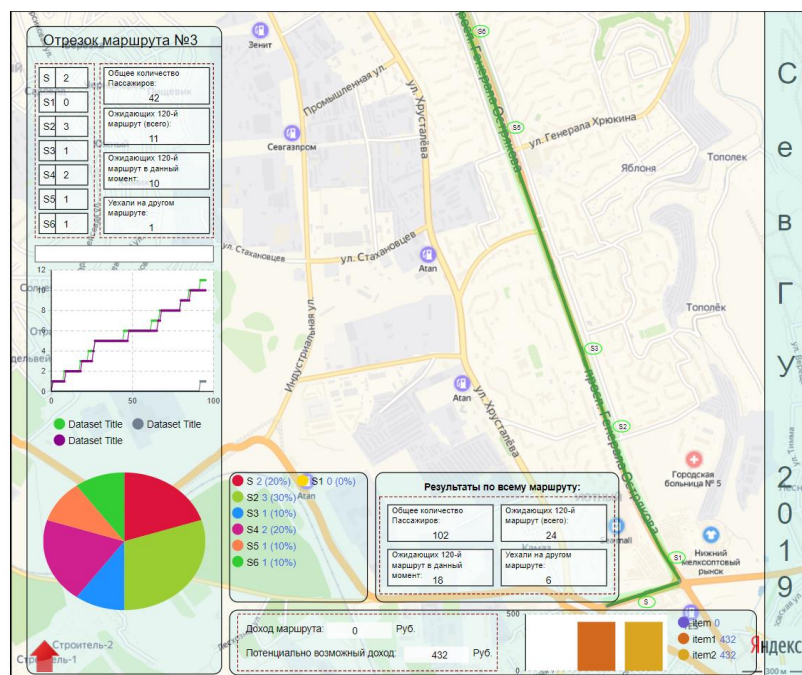


Рисунок 3 – Интерфейс программной системы «Отрезок маршрута №3»

Также в данном окне выводится общая статистика по всем трем отрезкам маршрута и высчитываются финансовые характеристики маршрута, которые предназначены для выбора оптимальной стратегии управления маршрутными средствами во избежание убытков транспортной компании.

На рисунке 4 показана логическая схема, под управлением которой работает модель организации общественного транспорта. Эта схема в режиме реального времени визуализирует интенсивности прибытия транспорта и пассажиров по каждому остановочному пункту исследуемого маршрута. В данном окне так же модно изменять настройки объектов светофорного регулирования по маршруту.

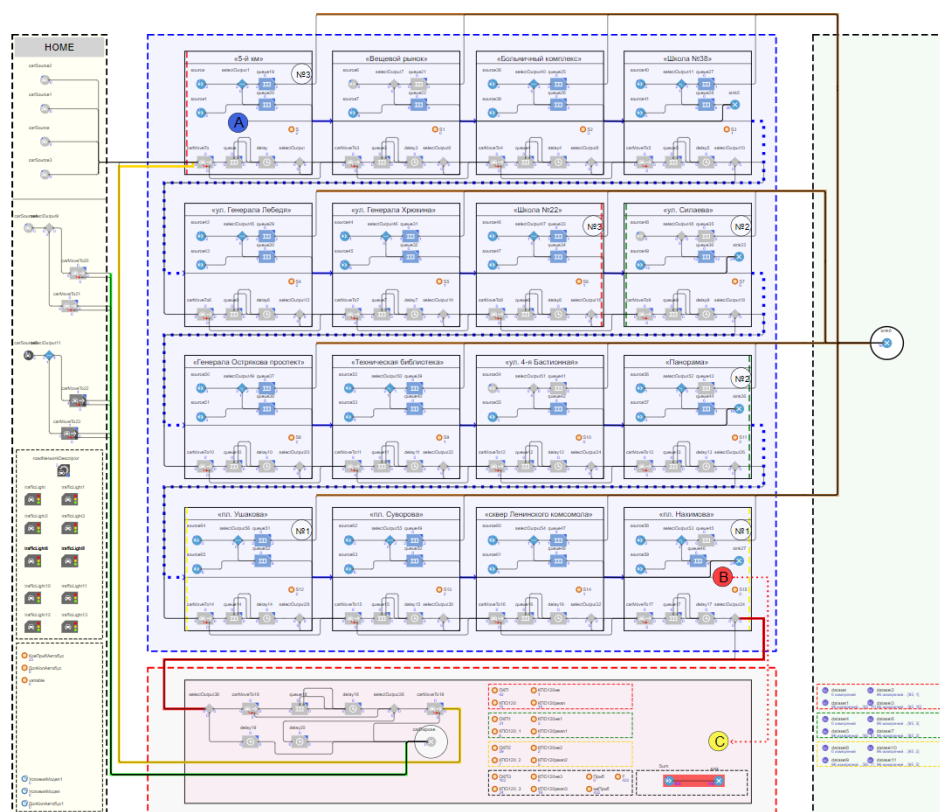


Рисунок 4 – Окно структуры имитационной модели

Имитационное моделирование является единственным практически доступным методом исследования сложной системы, особенно на стадии её проектирования или модернизации. В рамках работы в дальнейшем будут произведены тестовые запуски программы с разными условиями моделирования и детальное изучение полученных результатов.

Перспективами дальнейших исследований будет расширение предложенной модели на другие участки улично-дорожной сети и добавление новых маршрутов общественного транспорта.

Библиографический список:

1. Бараш В.В., Шевченко В.И. Исследование эффективности дорожных коммуникаций в городе Севастополе на примере модели дорожной развязки // Сборник статей всероссийской студенческой НТК "Мир компьютерных технологий". – Севастополь: Федеральное государственное автономное образовательное учреждение высшего образования "Севастопольский государственный университет". 2018. С. 13–17.
2. Советов Б.Я., Яковлев С.А. Моделирование систем: Учебник. – М.: Высшая школа, 2009. – 343с.

УДК 004

А.В. Скатков, докт. техн. наук, профессор**И.А. Балакирева, канд. техн. наук, доцент***Севастопольский государственный университет***ОПЕРАТИВНОЕ УПРАВЛЕНИЕ МНОГОНОМЕНКЛАТУРНЫМ
ПРОИЗВОДСТВОМ В УСЛОВИЯХ НЕОПРЕДЕЛЕННОЙ СРЕДЫ**

Аннотация: рассматривается подход к формированию правил принятия решений при оперативном управлении многономенклатурным производством в условиях неопределенности. Обсуждаются источники неопределенности. Анализируются альтернативные решения.

Ключевые слова: многономенклатурное производство, оперативное управление, источники неопределенности, матрица решений, критерий принятия решения.

Annotation: approach to forming of decision-making rules at operational management of multinomenclature production at the conditions of uncertainty is considered. Uncertainty sources are discussed. Alternative solutions are analyzed.

Keywords: multinomenclature production, operational management, uncertainty sources, matrix of solutions, criterion of decision-making.

Особенность многономенклатурного производства (МП) заключается в том, что при одних и тех же значениях входных параметров происходит разброс выходных параметров выпускаемых изделий под влиянием случайных факторов на технологические процессы (ТП). Эта особенность присуща многим ТП микроэлектроники, приборостроения, химического производства. Во множестве задач, решаемых при оперативном управлении ТП МП, можно выделить класс задач организации направленного выпуска изделий. Эти задачи имеют практическую ценность, связанную как с экономией дорогостоящего сырья и вспомогательных материалов, так и со снижением трудоемкости и объемов незавершенного производства.

Один из распространенных подходов к решению задачи минимизации объемов запуска изделий МП для процессов, имеющих вероятностный характер, основан на методах адаптивной коррекции параметров ТП. Суть этих методов заключается в использовании понятия функции цены, характеризующей вероятность получения продукта заданного качества. Установлено, что, варьированием значений выходных параметров системы в пределах допустимой области, можно изменять положение поверхности их распределения относительно поверхности функции цены. Такой метод получил название метода оптимума номинала [1].

Номенклатурный план заказа на производство представляет собой заранее определенное количество изделий, значения выходных параметров которых принадлежат той или иной номенклатурной группе. Если номенклатурный план выпуска изделий интерпретировать как функцию цены, то, управляя координатами центра настройки ТП, можно преобразовать поверхность распределения классификационных признаков выпускаемых изделий в соответствии с поверхностью номенклатурного плана и, таким образом, влиять на объемы выпуска изделий в ту или иную номенклатурную группу.

Подмножество выходных параметров ТП, по значениям которых определяется принадлежность изделий к той или иной группе, определим как классификационные признаки выпускаемых изделий. Тогда множество изделий, значения классификационных признаков которых принадлежат некоторой области с заранее определенными границами, представляют собой определенную номенклатурную группу. Разбив область изменения классификационных признаков изделий на группы, и, определив границы этих групп, по завершению технологического цикла готовые изделия или полуфабрикаты можно отнести к той или иной номенклатурной группе в зависимости от значений классификационных признаков изделий [2].

При определенных плановых заданиях для соответствия поверхности распределения классификационных признаков номенклатурному плану, необходимо унимодальную поверхность распределения преобразовать в мультимодальную. Подобное преобразование можно осуществить за счет реструктуризации центров настройки, то есть использования нескольких центров настройки в ходе ТП МП.

В случае реструктуризации центров настройки ТП объем выхода изделий в ту или иную номенклатурную группу будет определяться, во-первых, координатами центра настройки ТП, а, во-вторых, интервалом времени, в течение которого этот центр используется. Математическая модель операции классификации изделий в этом случае будет представлена в виде следующей системы уравнений:

$$z_i = \sum_{j=1}^n p_{ij} P_j N, i = \overline{1, m}, \quad (1)$$

где z_i — объем выхода изделий в i -ю номенклатурную группу $i = \overline{1, m}$; вероятность выпуска изделий в i -ю группу при использовании j -го центра настройки; η_j объем выпуска изделий при использовании j -го центра настройки, при этом $\eta_j = P_j N$. Коэффициенты P_j характеризуют долю времени использования j -го центра по отношению ко всей длительности ТП; N — объем запуска изделий на классификацию.

Если в качестве критерия оптимальности многономенклатурного производства принять производственный запуск, обеспечивающий минимальное перекрытие планового задания по номенклатуре, то задачу оптимизации режимов настройки ТП для фиксированного центра можно представить следующим образом.

Определить минимум функции

$$\sum_{j=1}^n P_j N \rightarrow \min \quad (2)$$

при ограничениях

$$\sum_{j=1}^n p_{ij} P_j N \geq q_i \quad (3)$$

В задаче (2), (3) минимизируется линейная форма в общем случае при нелинейных ограничениях. Однако если задано положение центра настройки процесса m_j и известна функция плотности распределения значений классификационных параметров $f(y|m_j)$, то можно определить значения p_{ij} и сформировать матрицу вероятностей. В этом случае условия (3) линейны относительно неизвестных $P_j N$, и задача (2), (3) представляет собой задачу линейного программирования.

Введем величину - коэффициент запуска — $K_{зан}$, характеризующий перекрытие планового распределения изделий по номенклатурным группам общим объемом запуска изделий:

$$K_{зан} = \frac{\sum_{j=1}^n P_j N}{\sum_{i=1}^m q_i} \quad (4)$$

Коэффициент запуска определяется технологическим режимом, то есть зависит от координат центра настройки ТП – $M = (m_1, m_2, \dots, m_n)$. Таким образом, выбрав определенный вектор M и, учитывая (4) можно минимизировать запуск изделий в производство. Итак, задача минимизации коэффициента запуска изделий в многономенклатурных производственных системах с учетом задачи (2), (3) представляется:

$$K_{zan}(M) \rightarrow \min \quad (5)$$

$$\sum_{j=1}^n P_j N \rightarrow \min \quad (6)$$

при ограничениях

$$\sum_{j=1}^n p_{ij} P_j N \geq q_i \quad (7)$$

Задача (5) - (7) является двухуровневой задачей математического программирования, в которой задача (5) является ведущим элементом многономенклатурной производственной системы и относится к классу задач нелинейного математического программирования. В результате ее решения определяется M^* – оптимальный набор центров настройки ТП. Ограничения задачи (7) определяются техническими условиями производственной системы и учитываются в поисковых алгоритмах при определении оптимального режима.

Задача (6) при ограничениях (7) представляет собой ведомый элемент многономенклатурной производственной системы и относится к классу задач линейного математического программирования. Здесь определяется оптимальный технологический режим интенсивности P^* использования центров настройки с координатами M^* [3].

Модель (5) - (7) является детерминированной и может быть применена в системе оперативного управления МП, если известен закон распределения классификационных признаков изделий и заранее определены объемы заказов в каждой из номенклатурных групп. В реальной ситуации существует ряд источников неопределенности, которые данная модель не учитывает, или учитывает не в полной мере.

В условиях изменяющегося рынка заказчик может устанавливать различные объемы заказов изделий в номенклатурных группах, у заказчика может возникнуть необходимость в определенных номиналах изделий, что в свою очередь может привести к изменению границ в номенклатурных группах, а также количеству этих групп. Изменение рыночных цен на изделия приводит к решению компромиссной задачи, в которой, с одной стороны необходимо минимизировать потери, связанные с невыполнением плана заказа, как материальные, так и репутационные, с другой стороны – потери, связанные с перепроизводством.

Не всегда возможно идентифицировать закон распределения классификационных признаков изделий, и задачу (5) - (7) приходится решать при неопределенной функции распределения параметров. Кроме того, неопределенность может возникнуть, если случайные воздействия внешней среды приводят к изменению разброса параметров ТП, поскольку предложенная модель построена при условии, что разброс параметров является постоянным в условиях рассматриваемого производства.

Перечисленные источники неопределенности можно отнести к входным условиям задачи (5) - (7), но даже, если они отсутствуют и рассматриваемая задача является хорошо определенной, существует риск дрейфа оптимальных центров настройки в ходе ТП. Это обстоятельство также может привести к неожиданным результатам, даже при детерминированных входных условиях. Решение задачи компенсации дрейфа центров настройки предложено в [4] за счет управления параметрами контроля МП.

В такой ситуации является целесообразным формирование правил принятия решений при оперативном управлении ТП МП в условиях полной неопределенности. Рассмотрим правила принятия в ситуации, когда неизвестны возможные объемы заказов в номенклатурные группы и нет информации о вероятности получения того или иного заказа. Тем не менее, эта неопределенность не является абсолютной, поскольку известны объемы заказов в каждую из групп, полученные и исполненные ранее. Кроме того, определены оптимальные режимы настройки ТП для исполненных планов. Эта информация является достаточной для построения матрицы решений (таблица 1) [5].

В заголовках столбцов матрицы решений размещены n состояний среды, суть которых есть план заказов на производство. План заказов включает в себя количество номенклатурных групп, которое может произвольным $k_j=1, k_n, \dots$, номенклатурный план в каждую группу q_k^j , $j=1, n$, $k=1, k_j$ и границы номенклатурных групп, представленные вектором G^j .

В заголовках строк матрицы решений размещены допустимые альтернативные решения, которые представляют собой оптимальные режимы ТП для каждого состояния среды. Оптимальный режим ТП представлен двумя векторами M_i^* - координаты центров настройки ТП и P_i^* интенсивности использования этих центров, $i=1, m$. Количество центров настройки, то есть размерность векторов M_i^* и P_i^* , соответствует определенному альтернативному решению.

Таблица 1 – Матрица решений при оперативном управлении ТП МП в условиях неопределенности

		План заказов на производство			
		$s^1 :$ $(q_1^1, q_2^1, \dots, q_{k_1}^1; G^1)$	$s^2 :$ $(q_1^2, q_2^2, \dots, q_{k_2}^2; G^2)$	$\dots$	$s^n :$ $(q_1^n, q_2^n, \dots, q_{k_n}^n; G^n)$
Технологические режимы	$a_1 = (M_1^*, P_1^*)$	$K_{зан}^{11}$	$K_{зан}^{12}$	$\dots$	$K_{зан}^{1n}$
	$a_2 = (M_2^*, P_2^*)$	$K_{зан}^{21}$	$K_{зан}^{22}$	$\dots$	$K_{зан}^{2n}$
	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
	$a_m = (M_m^*, P_m^*)$	$K_{зан}^{m1}$	$K_{зан}^{m2}$	$\dots$	$K_{зан}^{mn}$

Оценка того или иного альтернативного решения для различных состояний среды представлена коэффициентом запуска изделий в производство $K_{зан}^{ij}=(a_i, s^j)$, $i=1, m$, $j=1, n$. Поскольку коэффициент запуска (4) представляет собой уровень перепроизводства, что влечет за собой определенные потери, то матрицу решений, представленную таблицей 1 можно характеризовать как матрицу потерь.

В зависимости от предпочтений лица, принимающего решения (ЛПР), можно воспользоваться существующими критериями для анализа ситуаций, связанных с процессом выбора технологических режимов ТП.

Например, критерий Лапласа использует оптимистическое предположение, что все планы заказов на производство равновероятны, то есть $P(s^j)=1/n$. В этом случае наилучшим является технологический режим, который обеспечивает минимальное значение коэффициента запуска из всех минимальных коэффициентов для каждого из заказов на производство

$$\min_{a_i} \left\{ \frac{1}{n} \sum_{j=1}^n K_{zan}^{ij}(a_i, s^j) \right\}. \quad (8)$$

Консервативное отношение ЛПР выражает минимаксный критерий, который приводит к выбору лучшей альтернативы из наихудших и определяется следующим соотношением

$$\min_{a_i} \left\{ \max_{s^j} K_{zan}^{ij}(a_i, s^j) \right\}. \quad (9)$$

В этой ситуации для каждого из технологических режимов матрицы решений выбирается план заказа на производство, при котором коэффициент запуска будет максимальным, а далее выбирается то решение, которому соответствует минимальный коэффициент запуска изделий в производство.

Критерий Гурвица охватывает ряд различных подходов, в том числе и (8)-(9), к принятию решений – от наиболее оптимистичного до наиболее пессимистичного и представляет собой следующее выражение

$$\min_{a_i} \left\{ \alpha \cdot \min_{s^j} K_{zan}^{ij}(a_i, s^j) + (1 - \alpha) \cdot \max_{s^j} K_{zan}^{ij}(a_i, s^j) \right\}. \quad (10)$$

Коэффициент α называется показателем оптимизма и изменяется в пределах $0 \leq \alpha \leq 1$. Если $\alpha=0$, критерий Гурвица становится консервативным, так как его применение эквивалентно минимаксному критерию (9). При $\alpha=1$ критерий становится слишком оптимистичным, так как рассчитывает на наилучшее из наилучших решений.

При принятии решения о выборе технологического режима в соответствии с матрицей решений (таблица 1) всегда присутствует риск либо перепроизводства, либо невыполнения плана заказа. В зависимости от ситуации риск можно либо принять, либо, в соответствии с моделью (5)-(7) определить оптимальный технологический режим для конкретного производственного заказа. Полученное оптимальное решение можно добавить в матрицу решений, тем самым увеличив список альтернативных решений и состояний среды.

Библиографический список:

1. Горелова Г.В. Метод оптимума номинала и его применение / Г.В. Горелова, В.В. Здор, Д.В. Свечарник. — М.: Энергия, 1970. — 200 с.
2. Скاتков А.В. Технологии системотехнических решений: моногр. / А.В. Скатков, И.А. Балакирева, В.И. Шевченко. — М.: Издательство «Спутник +», 2016. — 267 с.
3. Ширяев, В.И. Исследование операций и численные методы оптимизации / В.И. Ширяев. — М.: Ленанд, 2017. — 224 с.
4. Балакирева И.А. Оптимизация параметров управления КМОП-технологиями / И.А. Балакирева. — «Информационные технологии и информационная безопасность в науке, технике и образовании «Инфотех – 2013»: Материалы междунар. научно-практ. конф., г. Севастополь, 7-11 сент. 2015 г./ М-во образования и науки Российской Федерации, Севастоп. Гос. Ун-т; науч.ред. А.В. Скатков. - Севастополь: СевГУ, 2015. — 109 с. — с.28-29.
5. Черноруцкий И. Г. Методы оптимизации и принятия решений / И.Г. Черноруцкий. — М.: Лань, 2013. — 384 с.

УДК 004.75

К.С.Ткаченко, инженер 1-й кат.

Севастопольский государственный университет

**ОБЕСПЕЧЕНИЕ ФУНКЦИОНИРОВАНИЯ КОМПЬЮТЕРНОГО УЗЛА
НА ОСНОВЕ ОЦЕНИВАНИЯ ВЕРОЯТНОСТЕЙ ГИПОТЕЗ В ЗАДАЧЕ
МОНИТОРИНГА**

Аннотация: Многие современные сложные системы экологического мониторинга состоят из отдельных компьютерных узлов, подключенных к датчикам и объединенных в распределенные системы. Такие компьютерные узлы могут подвергаться искажению входного трафика, который содержит управляющую информацию и мониторируемые данные. Поэтому необходимо управлять компьютерным узлом с учетом состояния его и входного трафика. Для этого управления можно построить оценки гипотез об этом состоянии.

Ключевые слова: мониторинг, компьютерные узлы, оценивание гипотез

Annotation: Many of today's complex environmental monitoring system consists of a separate computer nodes connected to the sensors and integrated into a distributed system. Such computer nodes may be subject to distortion of the input traffic, which contains control information and monitored data. Therefore, you must manage the computer node based on its status and input traffic. For this control it is possible to construct estimates of hypotheses about this state.

Key words: monitoring, computer nodes, hypothesis evaluation

Современные сложные измерительные системы должны иметь возможность использовать различные типы датчиков [1]. Применение датчиков другого типа в таких сложных измерительных системах не приводит к изменению их структуры. Это позволяет системам быть устойчивыми к изменениям окружающей среды, динамически адаптироваться к внешним условиям, что повышает их надежность, отказоустойчивость, масштабируемость. Сложность таких измерительных систем существенно выше существующих за счет возросшего количества датчиков, и, вследствие чего, требований хранения большого количества данных, их передачи и обработки с высоким быстродействием. Поэтому можно построить распределенную систему хранения и обработки данных, в которой каждый из датчиков подключен к единственной компьютерной системе. Эта специализированная компьютерная система должна потреблять малое количество ресурсов и иметь возможность обмена информацией с распределенной измерительной системой.

При передаче данных от специализированных компьютеров в автоматизированные системы мониторинга возникают риски, связанные с искажением информации при ее передаче [2]. Корректная информация может быть заменена на искаженную, неактуальную, недостоверную, в некоторых ситуациях возможно целенаправленное ее повреждение. Поэтому система мониторинга должна оперативно реагировать на изменения этих передаваемых сведений. Для этого системой производится анализ информации, чтобы эффективно оценивать характер работы специализированных компьютеров и отсутствия изменений информации о результатах мониторинга. Контроль состояния этих специализированных компьютеров базируется на программном обеспечении для сбора информации, контроля и настройки датчиков. Такой контроль отражается на временной эффективности функционирования отдельных компьютеров и измерительной системы в целом.

Управляемость компьютерной сети зависит не только от достоверности информации от компьютерных узлов сети, но и от ее полноты и актуальности [3]. По различным причинам, оборудование компьютерных узлов и их каналы связи могут отказаться. Этот отказ отражается на откликах о состоянии удаленных ресурсов. В связи с чем не всегда возможно точно определить причину отказа. Для обеспечения полноты информации о компьютерной сети требуются исчерпывающие данные об изменении пропускной способности каналов связи и компьютерных узлов обработки информации, получаемые непрерывно во времени. Поскольку задания в компьютерной сети обрабатываются на различных компьютерных

узлах, то такая компьютерная сеть считается многозвенной. При резервировании компьютерных узлов и каналов связи за счет дублирования появляются дополнительные расходы на обслуживание компьютерной сети. Поэтому в общем случае наиболее часто требуется для обеспечения управляемости ограничиться существующей инфраструктурой.

Работа с критически важной и конфиденциальной информацией при использовании многозвенных архитектур требует специальных средств контроля доступа [4]. Обработка информации на сервере сопряжена не только с решением вычислительных задач, но и с построением и передачей запросов посредством известных протоколов. Правильное конфигурирование сервера позволяет повысить производительность решения задач, учесть ширину каналов связи, чтобы сервер мог справиться с нагрузкой в зависимости от скорости выполнения программ и канала связи. Бездействующие процессы сервера при этом освобождают использованную ранее оперативную и дисковую память. Если этого не происходит, то в случае пиковой загрузки будут отсутствовать ресурсы на выполнение запросов сервером. Поэтому корректировка функционирования сервера заключается не в минимизации числа запросов к нему, а в уменьшении потребления ресурсов сервером. Для уменьшения потребления ресурсов производится мониторинг, на основании которого проводится анализ по выявлению надежности функционирования и безопасности обмена в сети.

В задачах экологического мониторинга оправдано построение распределенных информационных систем, в основе которых лежат так называемые одноплатные компьютеры [5]. Эти одноплатные компьютеры имеют возможность сопряжения с распределенными системами по проводным и по беспроводным каналам. База одноплатных компьютеров может быть расширена за счет дополнительных аппаратных и программных средств, в том числе модулями спутниковой связи, дополнительными датчиками и контроллерами. Выделенное либо автономное питание позволяет обеспечить функционирование микрокомпьютера в различных условиях.

Сбор данных, их передача и обработка могут выполняться при участии интеллектуальных средств [6]. Автоматизированные рабочие места, построенные на основе таких средств, отличаются возможностью предоставления результатов обработки информации с датчиков в достаточно наглядной и удобной для исследователя форме. Такое представление облегчает дальнейшее накопление и систематизацию, что в перспективе сформирует пригодные проанализированные данные для выработки новых системотехнических решений.

Поэтому необходимо оценивать вероятности гипотез о состоянии компьютерного узла, чтобы впоследствии обеспечить корректное его функционирование. Для задач экологического мониторинга известны способы производства оценок состояния трафика и рисков его изменения в компьютерных узлах [7, 8]. В настоящей работе будет рассматриваться подход к корректировке параметров системы на основе оценки гипотез о состоянии компьютерного узла.

Оценка гипотез о состоянии компьютерного узла может производиться на основе моделей. Применение имитационных моделей позволяет достичь высокого уровня адекватности результатов прогонов к существующему компьютерному узлу. Но в большинстве случаев имитационные модели потребляют большое количество вычислительных ресурсов системы. Поэтому их использование на одноплатных микрокомпьютерах, состыкованных с датчиками, затруднительно либо невозможно. В таком случае, можно расчет оценок гипотез о состоянии узла и трафика выполнять по аналитическим моделям.

Для практических целей управления компьютерным узлом достаточно считать, что его состояние описывается двумя гипотезами: гипотеза H_0 заключается в том, что трафик является невозмущенным и компьютерный узел работает в нормальном режиме, а гипотеза H_1 , в свою очередь, заключается в том, что трафик является возмущенным и компьютерный

узел работает в перегруженном режиме. Наличие двух гипотез приводит к существованию четырех условных вероятностей гипотез: $P(H_0|H_0)$ – это условная вероятность того, что имеется предположение о невозмущенном трафике при его фактическом невозмущении, $P(H_0|H_1)$ – имеется предположение о возмущенном трафике при его фактическом невозмущении, $P(H_1|H_0)$ – имеется предположение о невозмущенном трафике при его фактическом возмущении, $P(H_1|H_1)$ – имеется предположение о возмущенном трафике при его фактическом возмущении.

Компьютерный узел представляется аналитической моделью системы массового обслуживания (СМО) типа М/М/К/Н. В этой модели полагается, что имеется К каналов обслуживания, буфер заявок перед каналами емкости N. Интенсивность входного потока заявок составляет λ ед.вр.⁻¹, производительность обработки заявок – μ ед.вр.⁻¹. Закон входного потока заявок пуассоновский, закон обработки заявок экспоненциальный. На основании кортежа значений $S^{MMKN} = \langle K, N, \lambda, \mu \rangle$ могут быть сформированы важнейшие системные характеристики: вероятность отказа $p_{отк}$, среднее число заявок в очереди L_q , среднее число заявок в системе L_s , среднее время пребывания заявки в очереди T_q , среднее время пребывания заявки в системе T_s .

Оценки величин $P(H_0|H_0)$, $P(H_0|H_1)$, $P(H_1|H_0)$, $P(H_1|H_1)$ получаются на основе аналитического моделирования. Для этого происходит расчет откликов – важнейших системных характеристик, а затем следует их сравнение по простому критерию знаков. Либо, при наличии в подсистемах компьютерного узла достаточных вычислительных ресурсов, с помощью критерия Уилкоксона. Отклики рассчитываются для двух групп. В группу I помещаются результаты расчетов откликов $L_q^I = \langle L_{q1}, L_{q2}, \dots \rangle^I$ для режима нормального функционирования компьютерного узла, характерного для трафика нормальной интенсивности $\Lambda^I = \langle \lambda_1, \lambda_2, \dots \rangle^I$. Используется критерий по отклику: среднее число заявок в очереди L_q . В группу II помещаются результаты расчетов этого же отклика $L_q^{II} = \langle L_{q1}, L_{q2}, \dots \rangle^{II}$ для смещенной интенсивности $\Lambda^{II} = \langle \lambda_1, \lambda_2, \dots \rangle^{II}$ входного потока заявок, которая имеется в случае возмущений входного потока заявок и соответствующей ей перегруженному режиму работы компьютерного узла. Величина различия $||\Lambda^I| - |\Lambda^{II}||$ определяется ЛПР (лицом, принимающим решения).

В модели М/М/К/Н корректировка может производиться различными способами. А именно, изменением производительности обработки заявок на значение $\Delta\mu$, числа каналов – на ΔK , емкости буфера – ΔN . Для случая одноплатных компьютерных узлов, а также узлов на базе микроконтроллеров, наиболее доступной корректировкой является изменение величины производительности $\Delta\mu$. Это корректировка осуществляется путем изменения нагрузочных аппаратных характеристик и доступно для модификации через программное обеспечение. Например, в режиме простоя центральный процессор переходит в режим пониженного энергопотребления либо гибернации. В нормальном режиме энергопотребление среднее и частота работы обычная. В свою очередь, в режиме перегрузки наибольшая частота работы и максимальное энергопотребление. Однако в режиме перегрузки процессорный модуль может работать ограниченное время, как по причине деградации интегральных микросхем, так и по причине отсутствия достаточных резервов источников питания. Этот фактор приводит к необходимости целесообразного управления производительностью.

В таком случае, можно воспользоваться соотношениями корректировки:

$$\mu^{n+1} = \begin{cases} \min\{\mu^{max}; \mu^n + \Delta\mu\}, & P(H_1|H_1)^n \geq p^a, \\ \max\{\mu^{min}; \mu^n - \Delta\mu\}, & P(H_0|H_1)^n + P(H_1|H_0)^n < p^b. \end{cases} \quad (1)$$

В формуле (1) μ^n – текущее значение производительности обработки заявок на n-м шаге функционирования системы, μ^{n+1} – новое значение производительности на (n + 1)-м шаге, $\Delta\mu$ – величина приращения производительности, μ^{min} – наименьшее возможное

значение производительности, μ^{max} – наибольшее возможное значение производительности, p^a – уровень принятия решения об увеличении производительности, p^b – уровень принятия решения о ее уменьшении. $P(H_1|H_1)^n$, равно как $P(H_0|H_1)^n$ и $P(H_1|H_0)^n$, определяются по описанному выше подходу с отличием. В группе II представляется вместо Λ^{II} вектор $\Lambda^{III} = \langle \lambda_1, \lambda_2, \dots \rangle^{III}$, значения отклика $L_q^{III} = \langle L_{q1}, L_{q2}, \dots \rangle^{III}$ которого снимаются с компьютерного узла. Конкретные значения величин p^a и p^b определяются ЛПР. Допускается замена $p^a = p^{a,n}$ и $p^b = p^{b,n}$ на различных шагах n функционирования корректирующих процедур.

Применение (1) на отдельном и выделенном в компьютерной структуре сложной системы экологического мониторинга позволяет организовать обновление производительности обработки заявок на компьютерном узле с учетом текущей информационной ситуации. В случае, когда сильны гипотезы о наличии изменений трафика, происходит увеличение производительности. Это, с одной стороны, повышает износ оборудования и энергозатраты, но с другой стороны, уменьшает вероятность отказа p_{otk} и среднее число заявок в очереди L_q . Когда оказывают влияние дополнительные гипотезы, то снижается возможный износ оборудования и потребление энергии. Работа компьютерного узла на основе гипотез позволяет принимать корректирующие управленческие решения на опережение.

Полученный результат позволяет выполнить построение процесса управления компьютерным узлом на основе построения оценок гипотез о состоянии компьютерного узла и его входного трафика. При применении этого подхода применительно ко всем компьютерным узлам, входящим в состав распределенной сети экологического мониторинга, возможно получение корректного управления всей этой сетью.

Библиографический список:

1. Фролова М.В. Применение веб-технологий при разработке распределенных систем мониторинга / М.В.Фролова // Известия Южного федерального университета. Технические науки, т.118, №5, 2011. С. 41–47.
2. Бубнов В.П. Повышение безопасности передачи данных локального сервера системы автоматизированного геодезического мониторинга / В.П.Бубнов, А.Д.Хомоненко, В.В.Яковлев, С.В.Клименко // Автоматика на транспорте, т.4, №3, 2018. С. 407–420.
3. Дмитриев В.Н. Повышение эффективности системы мониторинга многозвенной сети передачи данных / В.Н.Дмитриев, А.С.Тушнов, Е.В.Сергеева // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика, №2, 2012. С. 112–118.
4. Гибадуллин Р.Ф. Разработка модуля мониторинга сетевой активности при обращении к защищенной картографической базе данных / Р.Ф.Гибадуллин, А.А.Новиков, И.Н.Смирнов, М.Ю.Перухин // Вестник Казанского технологического университета, т.19, №8, 2016. С. 95–98.
5. Слаутин Ю.А. Особенности технической реализации автоматизированной системы мониторинга окружающей среды / Ю.А.Слаутин, И.С.Полевщиков // Символ науки, №7-1, 2015. С. 44–46.
6. Слаутин Ю.А. Концепция интеллектуального автоматизированного мониторинга окружающей среды / Ю.А.Слаутин, И.С.Полевщиков // Символ науки, №7-1, 2015. С. 42-44.
7. Ткаченко К.С. Поточно-структурный подход к построению распределенных сред систем мониторинга / К.С.Ткаченко, И.А.Скатков // Системы контроля окружающей среды. 2017. № 9 (29). С. 41–44.
8. Ткаченко К.С. Имитационная модель оценивания рисков несанкционированных возмущений узлового трафика / А.В.Скатков, К.С.Ткаченко // Системы контроля окружающей среды – 2016. Тезисы докладов Международной научно-технической конференции. 2016. С. 73.

УДК 004.048

Д.Д. Шевченко, аспирант 2-го курса

О.В. Ченгарь, канд. техн. наук, доцент

В.И. Шевченко, канд. техн. наук, доцент

Севастопольский государственный университет

ИСПОЛЬЗОВАНИЕ ДИСКРЕТНО-СОБЫТИЙНОГО ПОДХОДА ДЛЯ МОДЕЛИРОВАНИЯ ПРОЦЕССОВ ОБРАБОТКИ ДАННЫХ В ОБЛАЧНЫХ КОРПОРАТИВНЫХ СИСТЕМАХ

Аннотация: рассмотрена модель и сценарии обработки данных в корпоративной информационной системе на основе дискретно-событийного подхода. Приведена структура имитационной модели и фрагменты экспериментальных исследований.

Ключевые слова: имитационное моделирование, дискретно-событийный подход, корпоративная информационная система, критерий эффективности

Annotation: the model and scenarios of data processing in a corporate information system based on a discrete-event approach are considered. The structure of the simulation model and fragments of experimental studies are presented.

Keywords: simulation modeling, discrete-event approach, corporate information system, performance criterion

В настоящее время облачные вычисления широко применяются во многих отраслях IT-технологий, в том числе и для автоматизации бизнес-процессов. Примером может служить решение 1С:Фреш - облачный сервис, который позволяет работать с привычными программами на платформе 1С через интернет с любого устройства.

Для обеспечения развития облачных сервисов, необходимы не только модернизация структур облачных вычислений в целом, но и разработка средств анализа и оптимизации вычислительных процессов. Одним из инструментов для решения подобных задач является моделирование. При этом, необходимо учитывать, что современные объекты управления обладают высокой степенью сложности (структурной, функционирования, выбора поведения и развития), что значительно затрудняет использование аналитических методов исследования, в этом случае применение имитационного моделирования (ИМ) является оправданным.

В рамках данной статьи рассматриваются результаты экспериментальных исследований над моделью обработки данных в корпоративной информационной системе. Целью проведения исследований является задача, повышения эффективности процессов обработки данных. В качестве критериев эффективности выбраны время ответа на запрос пользователя и процент обработанных запросов пользователей за время, не превышающее параметры, заданные в соглашении об уровне сервисов (*Service Level Agreement, SLA*). Разрабатываемая компьютерная система моделирования базируется на дискретно-событийном подходе и обеспечивает выдачу следующих характеристик функционирования системы обработки данных (СОД):

- среднюю загрузку узлов облачной инфраструктуры;
- средний объем занимаемой данными памяти;
- среднее число запросов, обработанных в СОД в течение рабочего дня;
- среднее число запросов, получивших отказ из-за несоблюдения SLA.

Дискретно-событийное моделирование. В дискретно-событийном моделировании функционирование системы представляется как хронологическая последовательность событий [1]. Событие происходит в определенный момент времени и знаменует собой изменение состояния системы.

Поведение системы с фазовым вектором $x(t)$ в любой момент времени t^* характеризуется парой $(t^*, x(t^*))$, которую принято называть событием. У события две координаты - временная t^* и пространственная $x(t^*)$. Первая указывает на время

наступления события, а вторая говорит о том, что произошло или характеризует явление. В случае если пространственные координаты и время непрерывны - временная траектория системы $(t^*, x(t^*))$ определяет бесконечное множество событий, случившихся за время наблюдения за системой на промежутке $[0, T]$. В непрерывном случае говорят не о всем множестве событий, а лишь об отдельных событиях, называя их особыми, подчеркивая тем самым нечто важное для нас и системы.

Если функция принимает конечное число постоянных значений, то с событием обычно связывают вторую координату, т.е. явление. Это обусловлено еще и тем, что в реальных технических устройствах идеальный скачок (разрыв первого рода) реализовать невозможно. Использование только второй координаты события для кусочно-постоянных функций породило название «дискретные события» и еще один способ описания таких функций. Их изображают в виде графа, приписывая каждому узлу одно значение функции.

Концептуальное описание модели. В рамках исследования рассматривается система обработки данных с многозвенной архитектурой, использующей web-сервера для представления данных, сервера приложений для реализации логики приложений и сервера баз данных для управления базами данных. Концептуальная модель облачной системы с многозвенными приложениями приведена на рисунке 1.

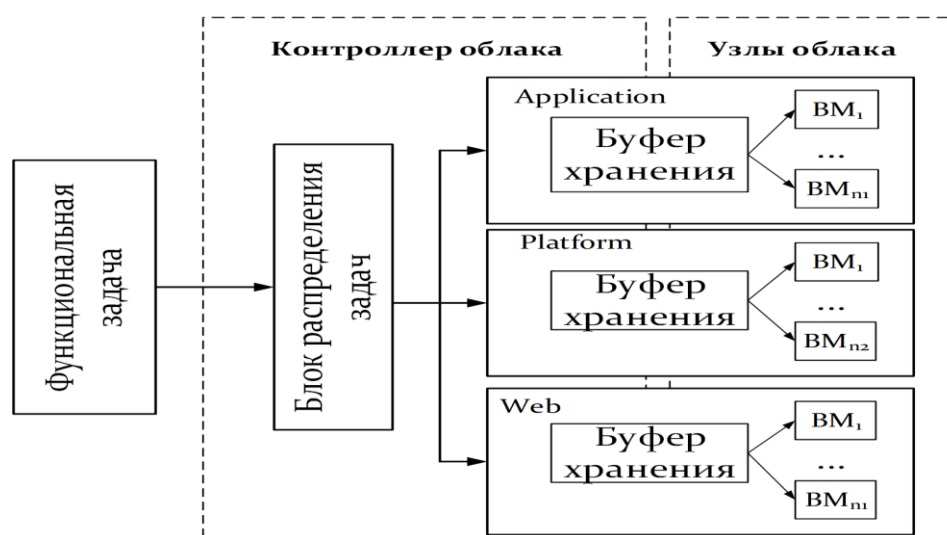


Рисунок 1. Структура облачной СОД с многозвенными приложениями

Контроллер облака, являясь "главным входом" в облако, устанавливает очередь для каждого размещенного в нем приложения. Кроме этого, контроллер облака создает определенное число виртуальных машин на узлах облака. Под «узлом облака» понимается физический сервер, на котором запущен монитор виртуальных машин. Первоначальное число создаваемых виртуальных машин задается в соглашении об уровне сервиса (SLA) или определяется опытным путем.

Число непосредственно работающих виртуальных машин может меняться в зависимости от нагрузки. Все виртуальные машины одного звена web-приложения работают на одном или нескольких узлах облака, и каждая имеет одинаковый объем физических вычислительных ресурсов. Виртуальные машины, на которых установлен один и тот же экземпляр приложения, образуют виртуальный кластер, не зависимо от физического местоположения виртуальных машин.

Когда клиент посылает запрос web-приложению в облаке, то он попадает к контроллеру облака. Распределитель нагрузки, являясь частью контроллера направляет запрос в очередь к нужному приложению.

В модели рассматривается три варианта обслуживания потока заявок (запросов пользователей в корпоративной информационной системе на обработку данных):

- обслуживание заявок происходит по степени срочности ее обслуживания (приоритет) с наличием штрафа за ожидание заявки в очереди, без отказа;
- обслуживание заявок происходит в порядке очереди по принципу FIFO с наличием штрафа за ожидание заявки в очереди, без отказа в обслуживании;
- обслуживание заявок происходит в порядке очереди по дисциплине FIFO с возможностью отказа в обслуживании из-за ограниченности мест в очереди.

Критерий эффективности СОД связан с максимизацией чистого дохода от обработки функциональных задач в облачной системе и имеет вид:

$$F(\lambda, \mu) = e_{\phiз} n_{обр} - e_{отк} n_{отк} - e_{нк} (N - R) - e_{нн} (m - L) - e_{ожс} \bar{t}_{ожс} \rightarrow \max \quad (1)$$

где

$e_{\phiз}$ — ожидаемый доход от обработки функциональных задач;

$n_{обр}$ — среднее число функциональных задач, которые были обработаны в облачной системе за 8 часов (рабочий день);

$e_{отк}$ — штраф за отказ ОВС в обработке функциональной задачи;

$n_{отк}$ — среднее число функциональных задач, которые получили отказ в облачной системе за 8 часов (рабочий день) из-за ограничения на объем буфера хранения;

$e_{нк}$ — штраф за отказ в использовании виртуальных машин;

N — число обслуживающих каналов (виртуальных машин);

R — среднее число занятых каналов (виртуальных машин);

$e_{нн}$ — штраф за неиспользование буфера хранения;

m — ограничение на объем буфера хранения;

L — средняя длина буфера хранения (очереди);

$e_{ожс}$ — штраф за ожидание заявки в очереди;

$\bar{t}_{ожс}$ — среднее время ожидания в очереди.

Имитационная модель. Обобщенная модель системы обработки данных приведена на рисунке 2. Для разработки модели использована среда имитационного моделирования AnyLogic [2].

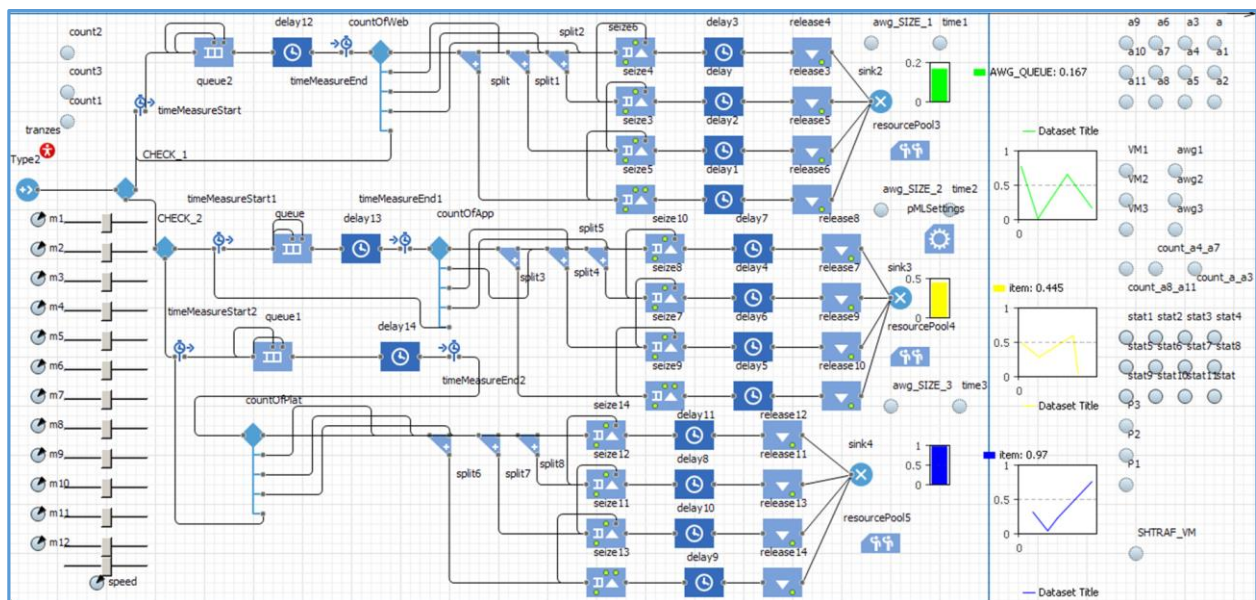


Рисунок 2. Структура модели системы обработки данных

Заявки поступают в систему с интенсивностью μ определяемой динамическим параметром *Speed*, они имеют два параметра *Priority* и *Size*, определенные в агенте *tranzes*. Распределительные блоки *SelectOutput* имитируют работу контроллера облака, распределяющего заявки на обработку по виртуальным машинам. Далее обработка

определяется сценарием. В случае приоритетного обслуживания заявка попадает в соответствующую приоритету очередь. Блок *Delay* после очереди служит для контроля неправомерного поступления заявки на обслуживание в связи с возможным наличием свободного числа виртуальных машин для обслуживания заявок с одинаковым приоритетом, а так же для предотвращения ситуаций возникновения отказов в связи с невозможностью обработки запроса из-за недостатка числа виртуальных машин (ВМ).

Последующие блоки *Split* служат для декомпозиции запроса на отдельные функциональные задачи (ФЗ), каждая из которых занимает случайную, свободную ВМ, где для них выделяются ресурсы при помощи блока *ResourcePool*. После получения необходимого числа ресурсов ФЗ поступают в блоки задержки, имитирующие обработку ФЗ запроса виртуальными машинами, где им выделяется случайное время обработки запроса, распределенное по экспоненциальному закону. По завершению обработки, выделяемые для заявки ресурсы освобождаются блоком *Release* и заявка покидает систему.

Эксперименты с моделью. На основе разработанной модели был проведен ряд экспериментов, по которым оценивался критерий эффективности (1) при изменении входных параметров системы.

Сценарий 1. Сценарий с очередями и приоритетами. Расчет основных характеристик системы приведен в таблице 1.

Таблица 1 – Расчет характеристик для модели по первому сценарию

№	λ	$n_{обр}$	R	L	F	T	№	λ	$n_{обр}$	R	L	F	T
1	10	4870	3,709	4,429	1595,82	28,09	6	5	2394	2,4	0,609	768,27	21,78
2	9	4344	3,623	2,674	1419,93	25,02	7	4	1917	1,843	0,27	608,94	20,49
3	8	3839	3,231	2,143	1250,93	24,79	8	3	1439	1,416	0,158	449,74	18,63
4	7	3423	3,093	1,882	1112,02	24,52	9	2	937	0,917	0,042	281,95	18,49
5	6	2836	2,766	1,441	915,98	23,71	10	1	449	0,399	0,009	119,22	17,24

На рисунке 3 приведен график изменения критерия эффективности в зависимости от интенсивности входного потока при использовании первой стратегии.

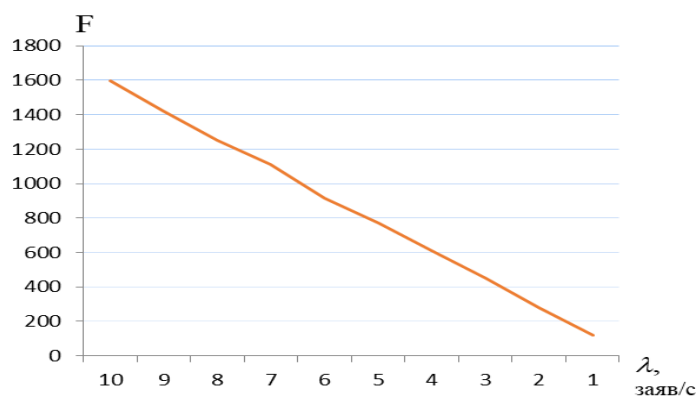


Рисунок 3. График изменения критерия эффективности от интенсивности входного потока

Из полученных значений видно, что при малой интенсивности и хорошей скорости обработки функциональных задач ВМ средняя длина очереди не превышает единицу, а скорость обработки занимает менее 20 секунд. Однако, при увеличении интенсивности входного потока функциональная задача может ждать обработки слишком долго. Штраф за ожидание может существенно превышать доход от обработки в системе, таким образом при интенсивном входном потоке высокоприоритетных задач заявка с минимальным

приоритетом может быть вообще не обработана, что в свою очередь для корпоративных информационных систем ведет к существенному понижению качества SLA.

Сценарий 2. Рассмотрим сценарий обработки заявок для структуры с неограниченными очередями и отсутствием приоритетов в обслуживании. Будем предполагать, что в отсутствие приоритетности все заявки дают одинаковый размер дохода за обслуживание. Результаты экспериментов, с учетом трех градаций изменения диапазона интенсивности обслуживания, приведены в таблице 2. В данном случае мы получаем значительно худшие результаты по критерию эффективности (1), несмотря на отсутствие потерь в обслуживании, в связи с уравниванием ценовой политики обработки функциональных задач.

Сценарий 3. В третьем сценарии мы рассматриваем ту же систему обработки данных, но уже с ограничением на число мест в буфере хранения. Результаты моделирования приведены в таблице 3.

Таблица 2 – Расчет характеристик для модели по второму сценарию

№	μ	$n_{обр}$	R	L	F	T
1	$1 < \mu \leq 5$	3429	4.345	41.22	910.36	26.88
2	$5 < \mu \leq 10$	3372	2.793	0.74	1085.29	15.66
3	$10 < \mu \leq 15$	3462	1.587	0.165	1115.54	9.06

Таблица 3 – Расчет характеристик для модели с ограничением на число мест в буфере хранения

№	λ	$n_{обр}$	R	F	№	λ	$n_{обр}$	R	F
1	10	4747	1.118	1237.62	6	5	2454	0.701	698.47
2	9	4311	1.083	1126.6	7	4	1907	0.544	536.38
3	8	3893	1.041	1023.01	8	3	1456	0.436	434.098
4	7	3466	0.961	928.26	9	2	970	0.3	296.05
5	6	2849	0.806	795.97	10	1	451	0.214	137.34

При незначительных интенсивностях поступления заявок, эффективность системы возрастает по отношению к рассмотренным ранее стратегиям, однако затраты на содержание заявки в системе до отказа не окупаются.

В рамках данной статьи рассмотрена модель и сценарии обработки данных в корпоративной информационной системе на основе дискретно-событийного подхода. Анализируя результаты экспериментов, было установлено, что при незначительных интенсивностях относительно скорости обработки функциональных задач стратегия обработки для системы с отказами работает эффективнее. При повышении интенсивности поступления задач большую эффективность дает стратегия увеличения буфера хранения. Разработанная модель СОД может быть использована как элемент системы поддержки принятия решений по организации эффективного управления распределением задач в облачных вычислительных средах.

Библиографический список:

1. Григорьева Т. Е. Дискретно-событийное моделирование в СМ «Марс» для курса «Системы массового обслуживания» // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2014. – 1 (31). – С. 152-155.
2. Шевченко В.И., Рыздывыло К.Г. Исследование модели эффективного размещения данных в распределенной файловой сети хранения // Сборник статей Всероссийской научно-технической конференции «ИНФОТЕХ-2017». – 2017. – С. 105-111.

УДК 004.9:519.6

Ю.Е.Шишкин, аспирант,

А.В.Скатков, д-р техн. наук, профессор,

Севастопольский государственный университет

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ ВИЗУАЛИЗАЦИИ И ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ РАСПРОСТРАНЕНИЯ АНОМАЛИЙ ПРИ ТОЧЕЧНЫХ ДЕГРАДАЦИОННЫХ ВОЗДЕЙСТВИЯХ

Аннотация: рассматриваются информационные технологии имитационного моделирования для повышения достоверности выявления аномалий, возникающих в данных мониторинга последствий деградационных процессов в среде. Построена математическая модель моделирующего стенда, разработанного на мультиагентной версии языка Лого, предназначенная для визуализации динамики процессов распространения множества параметрически заданных деградационных процессов в среде. Представлены результаты моделирования недетерминированного распространения аномалий в двумерном факторном пространстве, на базе модели случайных блужданий.

Ключевые слова: мониторинг, математическое моделирование, интеллектуальный анализ данных, выявление аномалий, критические процессы

Annotation: information technologies of simulation modeling are considered to increase the reliability of detecting anomalies arising in the monitoring data on the effects of degradation processes in the environment. A mathematical model of a simulation bench developed on a multi-agent version of the Logo language has been built, designed to visualize the dynamics of the propagation processes of a set of parametrically defined degradation processes in the environment. The results of modeling non-deterministic propagation of anomalies in a two-dimensional factorial space, based on the random walk model, are presented.

Key words: monitoring, mathematical modeling, data mining, anomaly detection, critical processes

На сегодняшний день информационные технологии имитационного моделирования (ИМ) активно и успешно применяются для решения многих задач оптимизации бизнес-процессов [1], инженерно-технических областей [2,3], критических объектов и процессов, когда цена ошибки велика [4,5], мониторинга окружающей среды [6], распространения аномалий при деградационных воздействиях [7,8] и других фундаментальных и прикладных областей человеческой деятельности [9]. В литературе активно освещается использование подхода ИМ в CAD/CAM системах для оптимизации внутренней структуры сложных объектов, например, автономных роботов [10], в системах ИМ общего назначения для оценки эффективности систем управления, обучения персонала работе с ответственными критическими системами [11], и др.

При анализе аномалий важную роль играет визуализация моделирования. Грамотно построенная визуализация повышает степень информированности и обоснованности при принятии решений о возникновении аномалий в данных мониторинга и оценки динамики протекания деградационных процессов во времени. Наглядной и достаточно универсальной моделью распространения деградационного процесса может послужить процесс независимого движения множества источников аномалий A в факторном пространстве M , оставляющих за собой «аномальное пятно» (АП) в среде E . На рисунке 1 представлена структурная схема моделирующего стенда, реализующего модель протекания деградационных процессов.

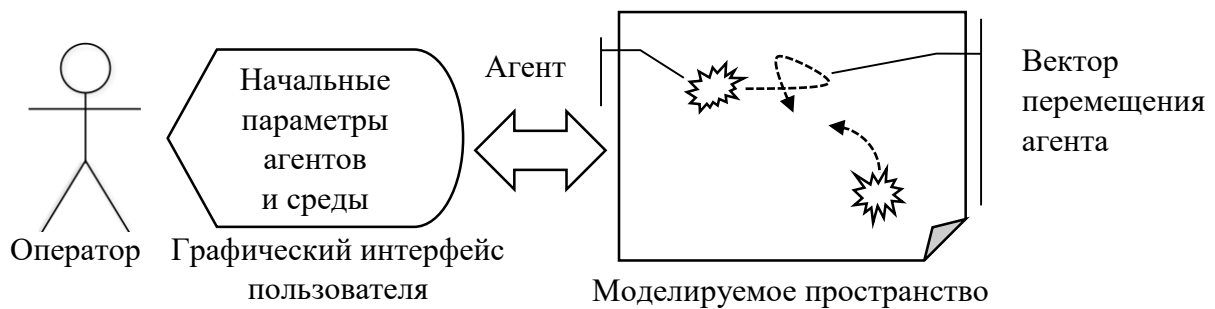


Рисунок 1 – Структурная схема моделирующего стенда

Зададим математическую модель агентов моделирования A , распространяющихся в непрерывном пространстве M , отражающем физическое евклидово пространство. Здесь и далее будем рассматривать исключительно движение агентов на плоскости. Рассмотрим целочисленную решетку $\mathbb{R}^2$ в которой задано множество агентов $a_i \in A | i \in [0; |A|]$, где $|A| = N$ – параметрически заданное число агентов. Для каждого агента a_i существует начальная точка в пространстве M , заданная парой $\langle x_{i,t}, y_{i,t} \rangle$, такая что определяет расположение i -го агента в момент $t = t_0$ и задается отображением $C: a_i, t \rightarrow c_{i,t} | \forall i \in [0; |A|], t \in [0, T]$, где T – общее модельное время. Для упрощения дальнейших выкладок $C(a_i, t)$ будем обозначать как $c_{i,t}$. Любая точка физического непрерывного пространства M с некоторыми допущениями может быть отображена в соответствующую точку дискретного моделируемого пространства $\mathbb{R}^2$, $M \rightarrow \mathbb{R}^2$ причем масштаб целочисленной решетки $\mathbb{R}^2$ определяет погрешность дискретизации расположения агента. Любой агент a_i стартуя из точки  за время Δt переходит в точку $c_{i,t+\Delta t}$, причем правило перехода определяется компонентами вектора перемещения $d_{i,t} = \langle l_{i,t}, r_{i,t} \rangle$ – длиной и направлением. Для упрощения процесса функционирования модели будем считать t целочисленным и $\Delta t = 1$, таким образом получим модель с дискретным временем что решает проблему неопределенности $\lim_{\Delta t \rightarrow 0} \frac{c_{i,t+\Delta t} - c_{i,t}}{\Delta t}$, обусловленной стохастической природой компонентов вектора перемещения $d_{i,t}$. В условиях принятых допущений и ограничений движение агента a_i будет определяться некоторым функционалом $D: c_{i,t}, d_{i,t} \rightarrow c_{i,t+1}$, причем реализация D задается исследователем в соответствии с его представлениями о предметной области и зависит от физической природы агентов и среды.

Одной из эффективных часто используемых моделей движения объектов в n -мерном факторном пространстве является модель случайных блужданий. На рисунке 2 представлена визуализация трека движения агента в пространстве.

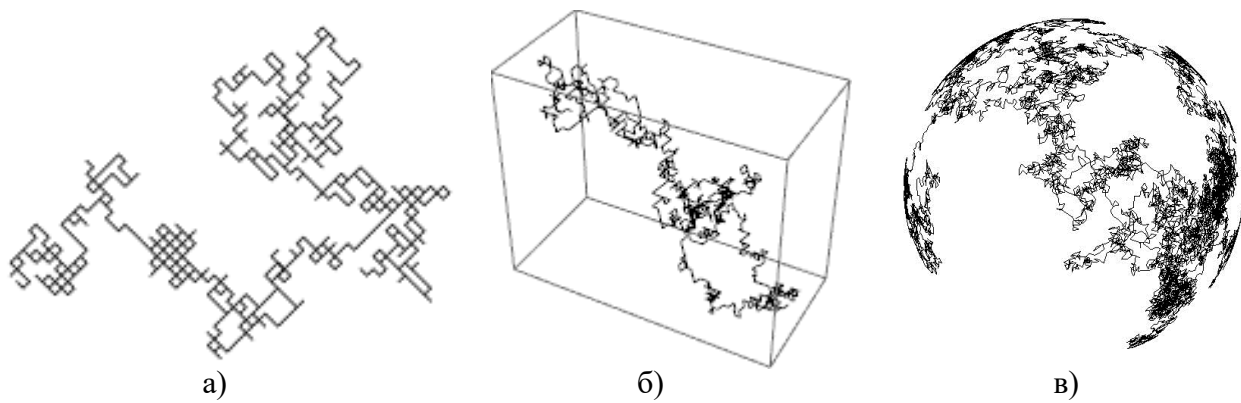


Рисунок 2 – Визуализация трека движения агента: а) на двумерной плоскости, б) в трехмерном пространстве, в) на поверхности сферы

Реализация функционала случайного движения D для агента в евклидовом пространстве в случае, когда его перемещения описываются моделью случайных блужданий выполняется в соответствии с соотношением (1). В процессе движения агентов a_i допускается самопересечение их траекторий.

$$\begin{aligned}
 D: c_{i,t} \langle x_{i,t}, y_{i,t} \rangle, d_{i,t} \langle l_{i,t}, r_{i,t} \rangle &\rightarrow c_{i,t+1} \langle x_{i,t+1}, y_{i,t+1} \rangle \\
 \begin{cases} x_{i,t+1} = x_{i,t} + l_{i,t} \cos(r_{i,t}) \\ y_{i,t+1} = y_{i,t} + l_{i,t} \sin(r_{i,t}) \end{cases} &, \\
 r_{i,t} \sim U[0; 2\pi] & \\
 l_{i,t} \sim U[0; l_{\max}] &
 \end{aligned}
 \quad (1)$$

где функция $U[0; 2\pi]$ задает случайную величину, заданную равномерным случайным распределением на отрезке $[0; 2\pi]$, $l_{\max}$ – максимальная длина шага, на который может сместиться агент a_i за время $\Delta t = 1$.

В процессе моделирования любой агент a_i в каждый момент времени $t \in [0, T]$ генерирует вокруг себя в среде E аномальное пятно (АП) радиусом R , заданным пользователем. Выбор R обуславливается природой решаемой задачи но в общем случае адекватный реальности результат моделирования можно достичь при $R \gg l_{\max}$.

Параметрическое представление лабораторного стенда, построенного по представленной выше модели, будет иметь вид:

$$L \langle N, C \{c_{1,0} \dots c_{n,0}\}, T, R, l_{\max} \rangle, \quad (2)$$

где N – число агентов, $C \{c_{1,0} \dots c_{n,0}\}$ – начальные положения агентов, T – внутримодельное время, R – радиус АП агента, $l_{\max}$ – максимальный шаг агента за время Δt .

Произведем моделирование деградационного процесса, протекающего в среде E , на лабораторном стенде, с параметрами $N = 1$, $C = \{0,0\}$, $T = 1000$, $R = 10$, $l_{\max} = 1$ и представим результат на рисунке 3. Агенты будем обозначать красными маркерами.

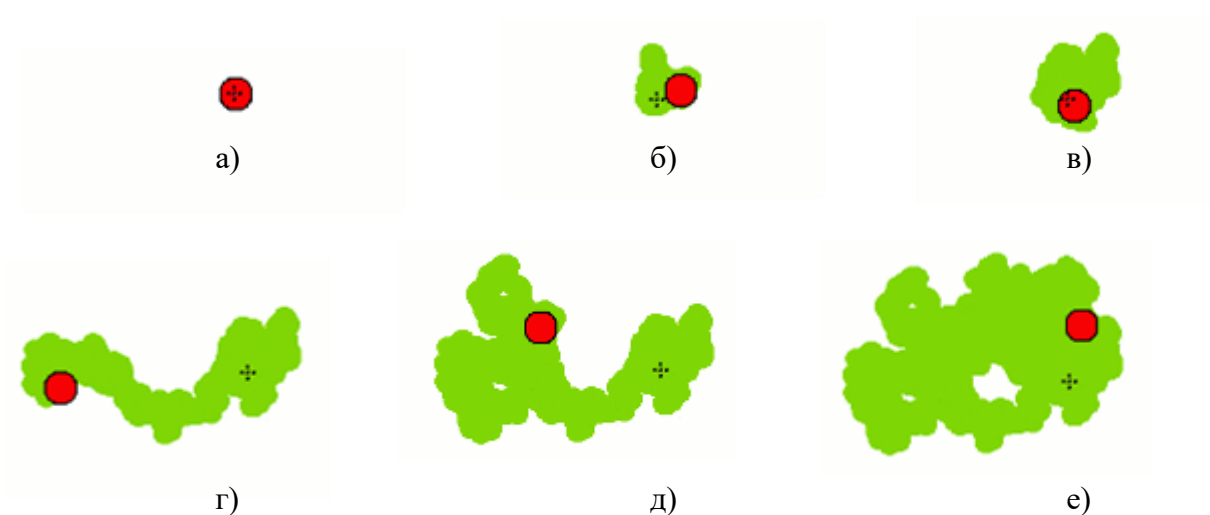


Рисунок 3 – Визуализация результата моделирования распространения «аномального пятна» единичного агента: а) $t=0$, б) $t=20$, в) $t=50$, г) $t=200$, д) $t=400$, е) $t=1000$

Представленный вариант использования модели может иметь физическую интерпретацию в виде распространения загрязнителя, генерируемого некоторым подвижным агентом в стационарной среде. Данное представление весьма условно, потому что на практике распространение происходит по нескольким направлениям одновременно. Изменим в модели параметр $N=5$, при этом остальные параметры, в частности начальная точка моделирования для каждого из агентов останутся неизменными $c_i = \{0,0\} \forall i[0; N]$, результаты моделирования такой системы представлены на рисунке 4.

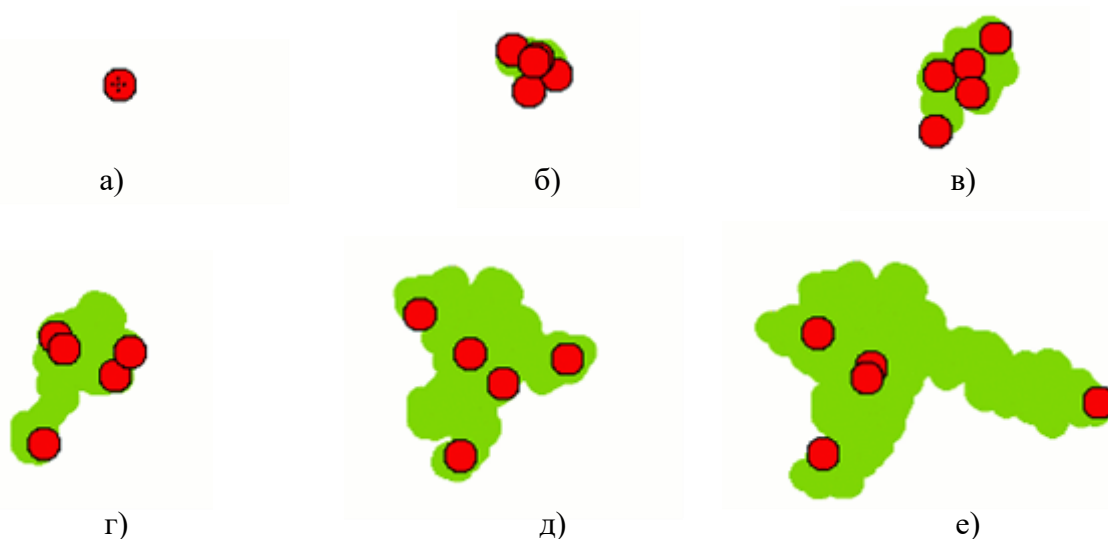


Рисунок 4 – Визуализация результата моделирования распространения «аномального пятна» нескольких агентов: а) $t=0$, б) $t=20$, в) $t=50$, г) $t=200$, д) $t=400$, е) $t=1000$

Данный вариант использования моделирующего стенда может иметь физическую интерпретацию модели распространения коррозии в объекте из единой начальной точки в результате дефекта этого объекта.

Второй вариант представления результата моделирования основан на расширенной модели, где факт воздействия агента на среду имеет вероятностную природу, в таком случае при повторном прохождении агента в некоторой области пространства E , которую обозначим $P(A, C, E)$, вероятность возникновения АП – $P(АП)$ возрастает. Если

предположить что события появления агента в области $P(A, C, E)$ независимы то вероятность возникновения АП в области C примет вид $P(АП) = 1 - (1 - P(A, C, E))^k$, где k – число появлений агента в области C . Обозначив $P(АП) \in [0; 1]$ как параметр интенсивности цвета, для параметров $P(A, C, E) = 5\%$, $N = 1$, $C = \{0, 0\}$, $T = 1000$, $R = 10$, $l_{\max} = 1$ получим визуализацию, представленную на рисунке 5.

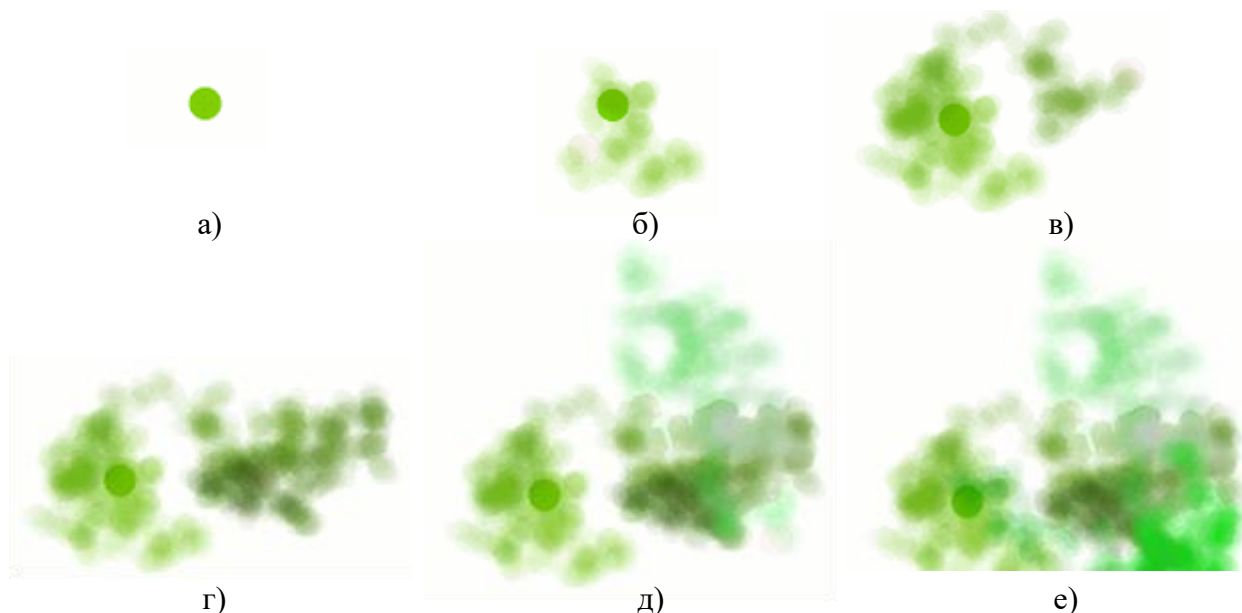


Рисунок 5 – Визуализация результата моделирования распространения вероятностного «аномального пятна» единичного агента: а) $t=0$, б) $t=20$, в) $t=50$, г) $t=200$, д) $t=400$, е) $t=1000$

С использованием предложенной информационной технологии мультиагентного имитационного моделирования распространения аномалии при деградиционных воздействиях на базе разработанного лабораторного стенда была произведена серия целенаправленных экспериментов, направленных на разработку аппарата проверки качества методов выявления аномалий в различных средах. В представленных примерах рассматривались решения задач в евклидовом пространстве, но предложенная математическая модель может быть дополнена необходимым числом измерений создавая факторное пространство требуемой размерности.

С точки зрения удобства пользователя лабораторный стенд позволяет параметрически задавать неограниченное число агентов системы моделирования, на графическом интерфейсе курсором указывать их начальное положение, поддерживает масштабирование скорости течения внутримодельного времени, и в отличие от альтернативных моделей позволяет задать радиус воздействия агента на среду и вероятность возникновения АП при повторном прохождении агента в области, на которой уже было произведено деградиционное воздействие.

Применение предложенного подхода визуализации данных моделирования дает возможность развить качественные представления о динамике деградиционных воздействий и порождаемых ими процессов. Это позволит разработать новые подходы к моделированию механизмов развития таких процессов, обладающие большей адекватностью и достоверностью.

Исследование выполнено при финансовой поддержке РФФИ и города Севастополь в рамках научных проектов № 18-47-920005\18, № 18-47-920007\18.

Библиографический список:

1. Боев В. Д. Имитационное моделирование систем: учебное пособие для прикладного бакалавриата. М.: Юрайт, 2018. 253 с.
2. Emiliano C., Daniele P. Handling obstacles in pedestrian simulations: Models and optimization // *Applied Mathematical Modelling*, № 45, 2017, С. 285–302.
3. Шишкин Ю.Е., Скатков А.В., Маловик К.Н. Модельные представления для мобильных приложений оценки и прогнозирования аномальных и паталогических состояний // *Качество и жизнь*. 2019. № 1 (21). С. 55–60.
4. Hee-Soo K., Seok-Won L., Dependability-enhanced unified modeling and simulation methodology for Critical Infrastructures // *Information and Software Technology*, № 102, 2018, С 175–192. DOI: 10.1016/j.infsof.2018.06.002
5. Шишкин Ю.Е., Скатков А.В. Метрики качества для оценки и прогнозирования критических состояний // *Качество и жизнь*. 2019. № 1 (21). С. 61–66.
6. Шишкин Ю.Е., Греков А.Н. Концепция интеллектуальной системы автоматизированного экологического мониторинга севастопольской бухты на базе малогабаритных автономных роботов // *Системы контроля окружающей среды*. 2018. № 14 (34). С. 63–69. DOI: 10.33075/2220-5861-2018-4-63-69
7. Лисов А.А., Чернова Т.А., Горбунов М.С. Системный подход к анализу деградационных процессов в электротехнических устройствах // *Труды МАИ*. 2018. № 102. С. 23.
8. Скатков А.В., Воронин Д.Ю., Скатков И.А. Имитационное моделирование деградационных отказов первичных измерителей системы мониторинга // *Системы контроля окружающей среды*. 2017. № 9 (29). С. 50–58.
9. Горбунов А.Р., Лычкина Н.Н. Парадигмы имитационного моделирования: новое в решении задач стратегического управления (объединенная логика имитационного моделирования) // *Бизнес-информатика*. 2007. № 2 (2). С. 60-66.
10. Греков А.Н., Шишкин Ю.Е., Никишин В.В. Экологический мониторинг севастопольской бухты на основе автономного надводного мини-корабля // *Комплексные исследования Мирового океана: материалы IV Всероссийской научной конференции молодых ученых*. 2019. С. 357–358.
11. Дозорцев В.М. Компьютерные тренажеры для обучения операторов технологических процессов. М.: Синтег. 2009. 365 с.

СЕКЦИЯ 3

Системы и средства искусственного интеллекта



УДК 004.048

Т.А. Васяева, к.т.н., доцент

Т.В. Мартыненко, к.т.н., доцент

Н.С. Суббота, студент

Донецкий национальный технический университет

ПРОГНОЗИРОВАНИЕ ФИНАНСОВЫХ ВРЕМЕННЫХ РЯДОВ С ПОМОЩЬЮ GRU СЕТЕЙ

Аннотация: В статье рассмотрено прогнозирование финансовых временных рядов с использованием глубоких нейронных сетей на примере прогнозирования цен акций сети аптек «3би6». Показано получение котировок акций, описан формат полученных данных. Рассмотрено применение GRU сетей для решения проблемы исчезающего градиента. Для разработки и обучения нейронной сети использован язык программирования Python, а также библиотеки pandas; numpy; keras (в качестве основы tensorflow).

Ключевые слова: машинное обучение, глубокое обучение, рекуррентные нейронные сети, GRU, временной ряд, цены на акции.

Annotation: The article describes the forecasting of financial time series using deep neural networks on the example of predicting the stock prices of pharmacy chains "36 and 6". The main methods of stock prices analysis is described in this article. Deep neural networks predicting for the stock prices of pharmacy chains 36 and 6 is discussed in this work. The financial time series forecasting using deep neural networks on the example of predicting the stock prices of pharmacy chains "36 and 6" is described in the article. The receipt of stock quotes is shown. The format of this data is described. There was developed and trained neural network for exchange trade forecasting with the use of the programming language Python, as well as libraries pandas; numpy; keras (as the basis of tensorflow).

Key words: Machine Learning, Deep Learning, Recurrent Neural Networks, GRU, Time Series, Stock Prices.

Введение

Рыночный путь развития предполагает, что главным законом функционирования субъектов хозяйственной деятельности является соотношение между спросом и предложением, а биржа как его элемент приобретает определяющее значение. Эффективные действия на бирже связаны с тщательным анализом происходящего на рынке. Прогнозирование является основой биржевой торговли. Способность предугадывать движение цен лучше, чем остальные может сделать из трейдера обеспеченного человека в короткие сроки. Для того чтобы прогнозирование было максимально достоверным, а прогнозы – обоснованными, трейдеры используют биржевой анализ [1-2]. В последние годы, у финансовых аналитиков стал вызывать большой интерес к такому направлению как машинное обучение[3] и в частности искусственные нейронные[4-5].

Анализ предметной области

В упрощённом варианте под биржей понимают место, где производится обмен. Это может быть обмен денег на товар или товар меняют на деньги, то есть происходит покупка и продажа чего-либо. При этом торг на биржах проходит не на физический товар, а на финансовые инструменты. Особенностью биржевого товара является отсутствие понятия качества. Биржевой товар одного вида является полностью взаимозаменяемым, и только цена имеет значение. Механизм работы биржи устроен так, что помогает покупателю и продавцу найти цену, которая устроит их обоих. Таким образом, биржа имеет существенные отличия от рынка: и в первую очередь отличие это в том, что на биржах торгуется однородный товар; обмен организован по принципу «гарантия лучшей цены, что даёт возможность однозначно определять цену биржевого товара в определенный момент времени. Еще одно важное и существенное отличие от обычного рынка это невозможность выбирать, с кем заключать, а с кем не заключать сделку. Заключая сделку, вообще неизвестно, у кого товар покупается и кому продается, так как напрямую вся работа ведется только с биржей. Именно биржа выступает арбитром всех сделок.

Постановка проблемы

Предметом исследования данной работы являются методы анализа фондовых бирж. В данном контексте финансовые инструменты представляются ценными бумагами, такими как акции, облигации, фьючеры, пифы и др. Прогнозирование биржевой торговли связано с анализом данных котировок. Мы будем прогнозировать цены акций, основываясь на ценах акций в предыдущие годы, а также дополнительных данных биржевой торговли за рассматриваемый период.

Получение данных

Для получения котировок данных будем использовать услуги компании «ФИНАМ» и их интернет-ресурс[6]. Сервера «ФИНАМ» предоставляют возможность — скачивать котировки. Перечень бумаг (инструментов) которые предоставляются компанией «ФИНАМ» будем брать из источника [6].

С помощью инструментов, которые предоставляет данный ресурс (рисунок 1), сформируем файл, данные из которого в дальнейшем будем использовать для анализа. Возьмем, например, данные, предоставленные ПАО Московская Биржа, котировки акций сети аптек "3би6".

Рисунок 1 – Получение данных для анализа

Данные получены в следующем формате (табл. 1): дата, время, цена открытия, максимальная цена, минимальная цена, цена закрытия, объем.

Таблица 1 – Входные данные

DATE	OPEN	HIGH	LOW	CLOSE	VOL
20140326	13.41000	13.75000	13.40000	13.59000	255490
20140327	13.48000	13.69000	13.33000	13.42000	151960
20140328	13.54000	14.61000	13.44000	14.52000	433440
20140331	14.69000	14.91000	14.41000	14.76000	178030
...	...	...	...	...	...
20140401	14.75000	15.23000	14.65000	15.10000	151950
20140402	14.88000	14.95000	14.65000	14.79000	77570

Данные представляют собой многомерный временной ряд. Период для анализа выберем 5 лет, интервал – один день. Традиционно наборы данных для работы с временными рядами формируют «скользящим окном» шириной равной глубине погружения. Определение глубины погружения является нетривиальной задачей и рассматривается в [7]. В данной работе эта задача не рассматривается.

Рекуррентные нейронные сети

В качестве базовой модели прогнозирования будем использовать нейронные сети. В настоящее время глубокие нейронные сети становятся одним из самых популярных подходов к решению очень разных задач, в том числе к прогнозированию временных рядов. В данной работе для решения поставленной задачи используются рекуррентные нейронные сети (Recurrent Neural Networks) [8-10]. Обратные связи придают рекуррентным нейронным сетям дополнительные возможности. На рис. 2 показан фрагмент простой рекуррентной нейронной сети.

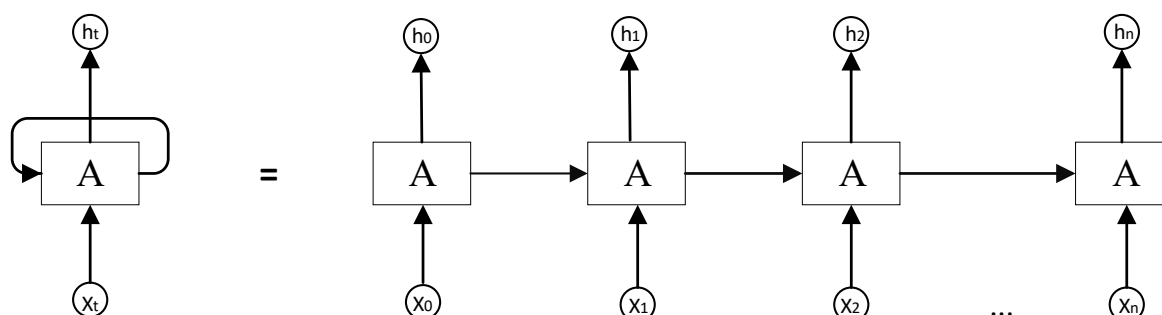


Рисунок 2 – Фрагмент рекуррентной нейронной сети и рекуррентная нейронная сеть в развертке

Сеть принимает входное значение x_i и возвращает значение h_i . Наличие обратной связи позволяет передавать информацию от одного шага сети к другому. Рекуррентную сеть можно рассматривать, как несколько копий одной и той же сети, каждая из которых передает информацию последующей копии. Для обучения нейронных сетей, в том числе и рекуррентных, используется алгоритм обратного распространения ошибки [8], основанный на методе градиентного спуска. Рекуррентные нейронные сети, так же как сети с большим количеством скрытых слоев, трудно обучать из-за проблемы исчезающего градиента. Проблему исчезающего градиента позволяют решить другие архитектуры RNN, например GRU (UpdateGate and ResetGate) [9]. В GRU (рис. 3) используются фильтры обновления и сброса, которые определяют, какую информацию следует передавать на выход. Особенность такой архитектуры заключается в том, что можно научить сеть хранить только ту информацию, причем достаточно долго, которая имеет отношения к прогнозу.

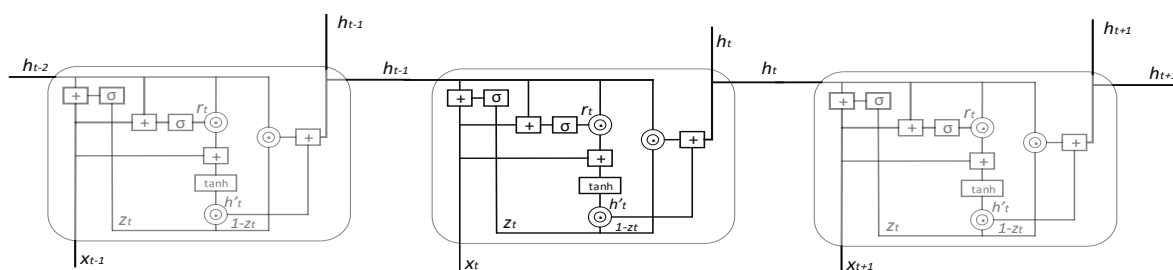


Рисунок 3 – Повторяющийся модуль в GRU

Рассмотрим представленную на рисунке 3 нейронную сеть более подробно. Начнем с вычисления обновления z_t для временного шага t по формуле:

$$z_t = \sigma(W^{(z)}x_t + U^{(z)}h_{t-1}) \quad (1)$$

Когда x_t подается на вход, он умножается на собственный вес $W(z)$. Так же как и h_{t-1} , который содержит информацию с предыдущих нейронов развернутой сети GRU, умножается тоже на собственный вес $U(z)$. Оба результата суммируются, и применяется функция активации сигмоид для получения результата между 0 и 1. Таким образом, фильтр обновления помогает модели определить, какую часть прошлой информации (из предыдущих временных шагов) необходимо передать в будущее.

Фильтр сброса используется в модели, чтобы решить, какую часть прошлой информации нужно забыть. Для его расчета мы используем:

$$r_t = \sigma(W^{(r)}x_t + U^{(r)}h_{t-1}) \quad (2)$$

Эта формула такая же, как и для фильтра обновления. Разница заключается в весе и использовании, на текущее значение сброса она влияет следующим образом:

$$h'_t = \tanh(Wx_t + r_t \odot Uh_{t-1}) \quad (3)$$

В качестве последнего шага необходимо вычислить h_t -вектор, который содержит информацию для текущего блока и передает ее в сеть. Это делается следующим образом:

$$h_t = z_t \odot h'_{t-1} + (1 - z_t) \odot h'_t \quad (4)$$

Экспериментальным способом разработана архитектура нейронной сети. Сеть состоит из входного слоя, трех GRU слоев и одного полносвязного выходного слоя, с функцией активации Relu. На вход подаются наборы данных размером 5x100 (глубина погружения = 100). На первом GRU слое 500 нейронов, на двух других по 100. На последнем слое 1 нейрон (горизонт прогноза = 1). После каждого GRU слоя используется Dropout [11], т.е. при обучении для каждого входного образца сеть перестраивается таким образом, что выпадает некоторый набор нейронов.

Экспериментальные исследования

В среде Colab Laboratory [12] выполнена программная реализация нейронной сети для прогнозирования цен на акции. Для моделирования использовался язык Python, с применением библиотек pandas; numpy; keras[13] (в качестве основы tensorflow).

Проведем эксперименты обучения Simple RNN нейронной сети (табл. 2-4) и GRU (табл. 5-7). Функция потерь (функция оценки оптимизации) – mean squared error. Метрика (функция, которая используется для оценки модели) – mean absolute error.

Таблица 2– Эксперименты обучения сети Simple RNN, optimizer = «SGD»

Epoch	Training set		Validation set		Test set	
	MSE	MAE	MSE	MAE	MSE	MAE
25	0.1305	0.3119	0.0538	0.2313	0.0148	0.1039
50	0.0931	0.2512	0.0505	0.2245	0.0938	0.3038
75	0.0115	0.0837	0.0837	0.0033	0.0012	0.0283
100	0.0050	0.0562	0.0053	0.0711	0.0039	0.0583
125	0.0056	0.0596	0.0010	0.0285	0.0005	0.0163
150	0.0041	0.0502	9.0535e-04	0.0244	0.0004	0.0138
175	0.0026	0.0400	0.0010	0.0267	0.0305	0.1746
200	0.0026	0.0367	6.5440e-04	0.0236	0.0020	0.0420
225	0.0020	0.0334	6.8846e-04	0.0228	0.0006	0.0210
250	0.0019	0.0321	0.0034	0.0537	0.0032	0.0530
275	0.0023	0.0371	9.2953e-04	0.0259	0.0004	0.0160
300	0.0023	0.0356	4.9935e-04	0.0176	0.0004	0.0148

Epoch	Training set		Validation set		Test set	
	MSE	MAE	MSE	MAE	MSE	MAE
325	0.0017	0.0301	5.0991e-04	0.0196	0.0005	0.0170
350	0.0016	0.0306	3.3104e-04	0.0147	0.0004	0.0162
375	0.0015	0.0293	2.6859e-04	0.0139	0.0005	0.0168
400	0.0022	0.0351	1.8592e-04	0.0116	0.0005	0.0170
425	0.0013	0.0275	5.6806e-04	0.0191	0.0012	0.0228
450	0.0014	0.0275	6.8861e-04	0.0215	0.0014	0.0286
475	0.0019	0.0322	1.4063e-04	0.0079	0.0003	0.0155
500	0.0014	0.0274	3.0688e-04	0.0141	0.0007	0.0223

Таблица 3– Эксперименты обучения сети Simple RNN, optimizer = «RMSprop»

Epoch	Training set		Validation set		Test set	
	MSE	MAE	MSE	MAE	MSE	MAE
25	0.1967	0.4219	0.0538	0.2313	0.0148	0.1039
50	0.1405	0.3235	0.1295	0.3595	0.0352	0.1877
75	0.1967	0.4219	0.0538	0.2313	0.0352	0.1877
100	0.0206	0.1183	0.0569	0.2380	0.1375	0.3654
125	0.0255	0.1280	0.0177	0.1322	0.0714	0.2595
150	0.0202	0.1166	0.0306	0.1741	0.0950	0.3015
175	0.0205	0.1180	0.0122	0.1093	0.0600	0.2367
200	0.1967	0.4219	0.0538	0.2313	0.0148	0.1039
225	0.0208	0.1186	0.0822	0.2862	0.1751	0.4136
250	0.0210	0.1201	0.0518	0.2271	0.1297	0.3545
275	0.0211	0.1202	0.0512	0.2184	0.0261	0.1213
300	0.1967	0.4219	0.0538	0.2313	0.0148	0.1039
325	0.0205	0.1177	0.0272	0.1642	0.1571	0.2648
350	0.0205	0.1180	0.0251	0.1575	0.0852	0.2849
375	0.0210	0.1185	0.0143	0.1187	0.0645	0.2460
400	0.0094	0.0755	2.8791e-04	0.0133	0.0215	0.1319
425	0.1430	0.3214	0.0538	0.2313	0.0148	0.1039
450	0.1967	0.4219	0.0538	0.2313	0.0148	0.1039
475	0.1967	0.4219	0.0538	0.2313	0.0148	0.1039
500	0.1967	0.4219	0.0538	0.2313	0.0352	0.1877

Таблица 4– Эксперименты обучения сети Simple RNN, optimizer = «adam»

Epoch	Training set		Validation set		Test set	
	MSE	MAE	MSE	MAE	MSE	MAE
25	0.1967	0.4219	0.0538	0.2313	0.0148	0.1039
50	0.0918	0.2481	0.0524	0.2283	0.1300	0.3551
75	0.0247	0.1248	0.0444	0.2100	0.1170	0.3362
100	0.0209	0.1192	0.0516	0.2265	0.1292	0.3538
125	0.0212	0.1195	0.0646	0.2536	0.1491	0.3809
150	0.1967	0.4219	0.0538	0.2313	0.0148	0.1039
175	0.0201	0.1170	0.0619	0.2484	0.1424	0.3723
200	0.0017	0.0306	5.3247e-05	0.0050	0.0012	0.0290
225	0.0149	0.0991	0.1101	0.3316	0.1645	0.4044
250	0.1967	0.4219	0.0538	0.2313	0.0321	0.1468
275	0.1967	0.4219	0.0538	0.2313	0.0148	0.1040
300	0.1967	0.4219	0.0538	0.2313	0.0148	0.1039
325	0.0045	0.0535	0.0038	0.0572	0.0032	0.0497
350	0.1967	0.4219	0.0538	0.2313	0.0148	0.1040
375	0.1967	0.4219	0.0538	0.2313	0.0148	0.1039
400	0.1967	0.4219	0.0538	0.2313	0.0148	0.1039
425	0.1967	0.4219	0.0538	0.2313	0.0148	0.1039
450	0.1967	0.4219	0.0538	0.2313	0.0148	0.1039

Epoch	Training set		Validation set		Test set	
	MSE	MAE	MSE	MAE	MSE	MAE
475	0.0012	0.0265	6.0529e-05	0.0062	0.0016	0.0331
500	0.1967	0.4219	0.0538	0.2313	0.0148	0.1039

Таблица 5. Эксперименты обучения сети GRU, optimizer = «SGD»

Epoch	Training set		Validation set		Test set	
	MSE	MAE	MSE	MAE	MSE	MAE
25	0.0034	0.0451	0.0032	0.0560	0.0080	0.0880
50	0.0021	0.0336	2.4674e-04	0.0140	0.0005	0.0213
75	0.0014	0.0288	8.2823e-05	0.0085	0.0003	0.0160
100	0.0018	0.0300	9.5337e-05	0.0073	0.0002	0.0110
125	9.3750e-04	0.0246	3.4187e-05	0.0050	0.0004	0.0198
150	0.0015	0.0270	6.0132e-05	0.0053	5.9330e-05	0.0052
175	8.0759e-04	0.0208	1.1295e-04	0.0104	0.0002	0.0122
200	0.0014	0.0255	6.2011e-05	0.0055	6.9462e-05	0.0065
225	5.0359e-04	0.0179	1.1111e-05	0.0026	0.0002	0.0143
250	0.0013	0.0249	6.4731e-05	0.0055	6.7806e-05	0.0062
275	3.9364e-04	0.0153	8.2145e-06	0.0021	3.5834e-05	0.0052
300	3.2893e-04	0.0135	3.0828e-05	0.0045	0.0006	0.0233
325	3.0425e-04	0.0135	1.4492e-05	0.0031	1.9409e-05	0.0040
350	0.0012	0.0238	6.0974e-05	0.0053	6.1079e-05	0.0055
375	2.4991e-04	0.0120	8.3227e-05	0.0088	0.0002	0.0147
400	0.0011	0.0232	5.9779e-05	0.0053	7.2583e-05	0.0067

Таблица 6. Эксперименты обучения сети GRU, optimizer = «RMSprop»

Epoch	Training set		Validation set		Test set	
	MSE	MAE	MSE	MAE	MSE	MAE
25	0.0033	0.0432	5.3356e-04	0.0221	0.0004	0.0183
50	0.0019	0.0346	1.0801e-04	0.0096	0.0002	0.0129
75	0.0012	0.0281	2.6066e-04	0.0157	0.0004	0.0179
100	0.0012	0.0280	2.8882e-05	0.0047	2.2457e-05	0.0042
125	0.0011	0.0258	1.1634e-05	0.0028	0.0002	0.0114
150	9.0187e-04	0.0233	1.6808e-04	0.0127	0.0005	0.0227
175	7.6243e-04	0.0217	1.4423e-04	0.0117	7.7916e-05	0.0082
200	5.2413e-04	0.0175	1.1297e-05	0.0027	7.7195e-05	0.0075
225	6.4330e-04	0.0193	8.5945e-05	0.0088	0.0004	0.0182
250	4.7044e-04	0.0169	6.3979e-04	0.0251	0.0018	0.0417
275	4.4681e-04	0.0162	1.8000e-05	0.0036	7.5121e-05	0.0080
300	3.7252e-04	0.0151	1.4825e-05	0.0032	3.1399e-05	0.0049
325	3.6205e-04	0.0144	5.2292e-05	0.0069	0.0001	0.0107
350	1.8596e-04	0.0100	6.2419e-06	0.0018	1.1842e-05	0.0019
375	3.2618e-04	0.0140	3.7247e-04	0.0191	0.0007	0.0265
400	2.7782e-04	0.0127	1.5547e-04	0.0121	0.0006	0.0231

Таблица 7. Эксперименты обучения сети GRU, optimizer = «adam»

Epoch	Training set		Validation set		Test set	
	MSE	MAE	MSE	MAE	MSE	MAE
25	0.0012	0.0257	3.8994e-05	0.0042	6.1829e-05	0.0065
50	8.2893e-04	0.0218	2.8236e-05	0.0041	2.1816e-05	0.0033
75	8.2871e-04	0.0224	8.4012e-05	0.0078	5.2994e-05	0.0064
100	5.1159e-04	0.0164	1.4284e-05	0.0027	3.8465e-05	0.0055
125	4.9456e-04	0.0164	4.5793e-05	0.0058	8.3688e-05	0.0086
150	3.2748e-04	0.0136	3.2897e-05	0.0053	2.7063e-05	0.0047
175	3.5142e-04	0.0138	6.1541e-05	0.0073	7.8827e-05	0.0086

Epoch	Training set		Validation set		Test set	
	MSE	MAE	MSE	MAE	MSE	MAE
200	2.9123e-04	0.0123	2.9800e-05	0.0047	0.0148	0.1039
225	3.2575e-04	0.0137	8.1865e-06	0.0022	1.9035e-05	0.0038
250	2.3602e-04	0.0111	2.6245e-05	0.0045	1.1189e-05	0.0027
275	2.3157e-04	0.0113	1.2040e-05	0.0028	3.6610e-05	0.0055
300	1.8701e-04	0.0099	9.9938e-06	0.0026	6.2019e-05	0.0066
325	2.0431e-04	0.0106	1.3269e-05	0.0029	3.9424e-05	0.0057
350	2.0833e-04	0.0109	2.6416e-05	0.0047	1.5220e-05	0.0032
375	1.7286e-04	0.0095	5.4383e-06	0.0016	9.6958e-06	0.0025
400	1.8427e-04	0.0099	1.2043e-05	0.0028	5.2759e-06	0.0017

На рисунках 4, 5 представлены результаты работы сети Simple RNN, обученной за 275 итераций при использовании оптимизатора «SGD». Для данной нейронной сети ошибка обучения достаточно маленькая, по сравнению с другими (табл. 2-4). Рис. 4 наглядно показывает результаты работы на обучающих данных, а рис. 5 – на проверочных. Для сравнения представлены результаты работы сети GRU на тестовых данных (рис. 6), при таких же параметрах обучения НС как в случае Simple RNN.

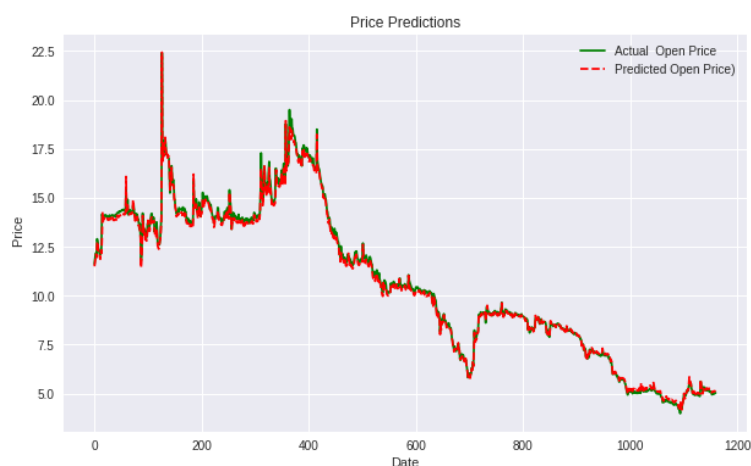


Рисунок 4 – Результаты на обучающей выборке (Шагов = 275, optimizer = «SGD», Simple RNN)

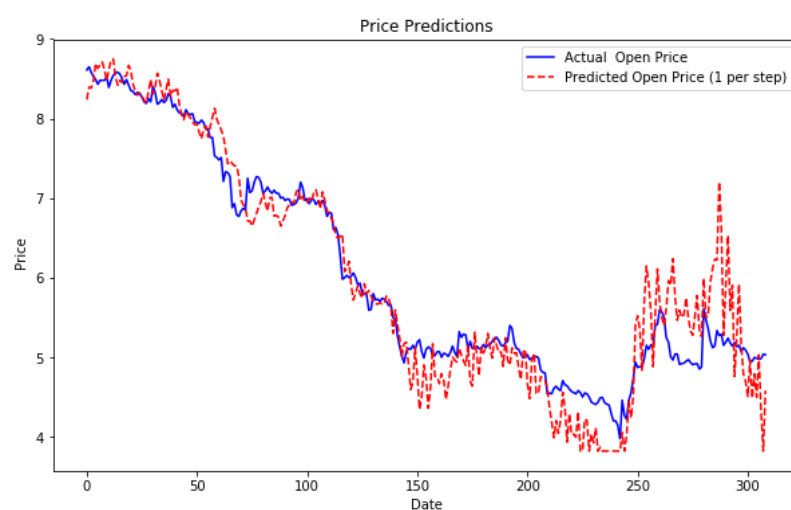


Рисунок 5 – Результаты на тестовой выборке (Шагов = 275, optimizer = «SGD», Simple RNN).

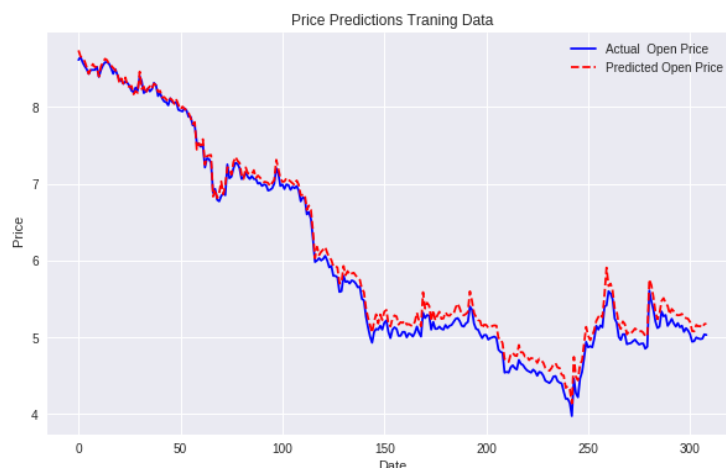


Рисунок 6 – Результаты прогноза на тестовой выборке (Шагов = 275, optimizer = «SGD», GRU).

Таким образом, хорошо видно, что GRU гораздо эффективнее справляется с решением нашей задачи, причем согласно экспериментальным исследованиям (табл. 5-7), показанный на рисунке 6 результат не является лучшим.

Заключение

Рассмотрена задача прогнозирования биржевой торговли. Показано, что машинное обучение и в частности искусственные нейронные могут быть инструментом для финансовых аналитиков. Выполнены экспериментальные исследования по обучению рекуррентных нейронных сетей. Согласно табл. 2-4 ошибка обучения для Simple RNN достаточно большая, при использовании оптимизатора «SGD», а при «RMSprop» и «adam» сеть практически вообще не обучается. Проведены аналогичные исследования для сети GRU (табл. 5-7). Показано, что для сети GRU ошибка, как для обучающих, так и для тестовых данных существенно меньше, чем для Simple RNN. Проведенные эксперименты подтверждают решение проблемы исчезающего градиента сетью GRU и показывают возможность применения такой нейросетевой модели для прогнозирования биржевой торговли.

Библиографический список:

1. Thomas S. Y. Ho, Sang Bin Lee.: The Oxford Guide to Financial Modeling: Applications for Capital Markets, Corporate Finance, Risk Management and Financial Institutions. Oxford University Press, Inc. 198 Madison Avenue, New York, New York 10016 (2004) г.
2. Iacomin R.: Stock market prediction. In: 19th International Conference on System Theory, Control and Computing (ICSTCC), Cheile Gradistei, pp. 200-205. (2015), DOI: 10.1109/ICSTCC.2015.7321293.
3. Shai Shalev-Shwartz, Shai Ben-David: Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press. 32 Avenue of the Americas, New York, NY 10013-2473, USA (2014)
4. Schmidhuber J. Deep Learning in Neural Networks: an Overview // Neural Networks. 2015. Vol. 1. P. 85–117, DOI: 10.1016/j.neunet.2014.09.003
5. Michael A. Nielsen: Neural Networks and Deep Learning, Determination Press. (2015)
6. Stock quotes, <https://www.finam.ru/>, last accessed 2019/03/29.
7. Sergii, K., Yurii, S., Tatyana, V., Natalia, A.: Feature Selection for Time-Series Prediction in Case of Undetermined Estimation. In: Samsonovich A., Klimov V., Rybina G. (eds.) Biologically Inspired Cognitive Architectures (BICA) for Young Scientists, Advances in Intelligent

Systems and Computing, vol. 449, pp. 85-97. Springer, Cham (2016). DOI: 10.1007/978-3-319-32554-5_12

8. Созыкин А.В. Обзор методов обучения глубоких нейронных сетей // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2017. Т. 6, № 3. С. 28–59. DOI: 10.14529/cmse170303

9. Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Gated feedback recurrent neural networks. arXiv preprint arXiv:1502.02367, 2015

10. Manaswi N.K. RNN and LSTM. In: Deep Learning with Applications Using Python. Apress, Berkeley, CA Recurrent Neural Networks (2018)

11. Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. // The Journal of Machine Learning Research, 2014. Vol. 15, no. 1.

12. Google Colaboratory. - <https://colab.research.google.com/notebooks/welcome.ipynb>.

13. Keras: The Python Deep Learning library. - <https://keras.io/>

УДК 681.518.54:572.983

А.Ю. Дорогов, д.т.н., профессор*Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»***БЫСТРЫЕ НЕЙРОННЫЕ СЕТИ ГЛУБОКОГО ОБУЧЕНИЯ
С КОММУТАЦИЕЙ ПЛОСКОСТЕЙ**

Аннотация: рассматривается алгоритм построения сетевой топологии и обучения двумерных быстрых нейронных сетей с дополнительными коммутируемыми плоскостями. Построенная топология идеологически близка к топологиям свёрточных нейронных сетей глубокого обучения, но является регулярной с числом слоев, устанавливаемым факторным представлением размерностей изображения и выходной плоскости классов. Алгоритм обучения имеет аналитическое представление, является абсолютно устойчивым и сходится за конечное число шагов. Дополнительные плоскости расширяют информационную емкость перестраиваемого преобразования до максимально возможной. Управление плоскостями в режиме обучения и обработки реализуется числовыми кодами координат выходной плоскости

Ключевые слова: быстрое перестраиваемое преобразование; нейронная сеть; быстрое обучение; свёрточная нейронная сеть; растровые изображения; плоскости нейронных слоев

Annotation: algorithm of network topology construction and training of two-dimensional fast neural networks with additional switched planes is considered. The constructed topology is ideologically close to the topology of convolutional neural networks of deep learning, but it has regular topology with the number of layers established by the factor representation of the dimensions of the image and the output plane of the classes. The learning algorithm has an analytical representation; it is absolutely stable and converges in a finite number of steps. Additional planes extend the information capacity of the tunable transformation to the maximum possible. Control of the planes in the training and processing mode is realized by numerical coordinate codes of the output plane.

Keywords — fast tunable transformation; neural network; learning; convolutional neural network; the bitmap; the planes of the neural layers

Введение

Технология глубокого обучения предполагает наличие процесса конфигурационного усложнения информативных признаков в последовательности нейронных слоев. Начиная с неокогнитрона К. Фукусима [1] предложено множество вариантов реализации этой идеи. Одним из удачных решений являются архитектура свёрточных нейронных сетей [2], которая показала высокую эффективность при решении различных задач. Отличительной особенностью этой архитектуры является наличие в свёрточных слоях нескольких каналов обработки данных (называемых картами или плоскостями). В каждой плоскости выполняется свертка выходного образа предшествующего слоя с фиксированным ядром небольшой размерности. Свёрточные слои чередуются со слоями пулинга, кратно снижающих размерность пространства признаков. Слои пулинга не являются обязательными и существуют варианты их полного устранения из архитектуры сети [3]. Второй отличительной особенностью архитектуры является использование полулинейные функции активации, выполняющих роль коммутационных ключей, управляемых значениями переменных скрытых слоев.

Недостатком архитектуры свёрточных сетей является отсутствие теоретически-обоснованных методов выбора структуры сети и параметров ядер свертки. Существует несколько прекрасно работающих конфигураций сетей для решения конкретных задач, но не понятно, как нужно строить сеть для новой задачи. До сих пор выбор структуры свёрточной сети является предметом искусства. Второй существенный недостаток связан с временем обучения свёрточных сетей. На типовом процессоре время может варьироваться от нескольких часов до нескольких суток, поэтому для обучения сетей часто используют высокопроизводительные графические процессоры. В работе [4] авторами была предложена идея использовать алгоритмы быстрых преобразований для построения структуры и топологии многослойных персептронных нейронных сетей. Покажем, что этот подход с некоторой модификацией можно использовать также для построения структуры и топологии свёрточных нейронных сетей.

В настоящее время широко известны быстрые алгоритмы линейных преобразований Фурье, Уолша, Хаара и других им подобных. При использовании быстрых алгоритмов выигрыш по вычислительным операциям экспоненциально растет при увеличении размерности преобразования. В конце 20 века возникло направление быстрых перестраиваемых преобразований [5], которые по существу являются нейронными сетями с ограниченными связями и линейными функциями активации. Для обучения подобных нейронных сетей были разработаны методы, абсолютно сходящиеся за конечное число шагов. Число слоев в быстрых преобразованиях и их конфигурация определяются размерностью обрабатываемых образов. Для построения быстрых алгоритмов размерность преобразования должна быть составным числом и чем больше множителей в разложении размерности, тем выше вычислительная эффективность быстрого алгоритма. Несмотря на большое разнообразие быстрых алгоритмов, конфигурации их структур удовлетворяют системному инварианту самоподобия [6]. Как известно таким же свойством обладают фракталы. Поэтому быстрые алгоритмы можно интерпретировать как квазифракталы. Свойство структурной фрактальности позволяет решить одновременно две задачи: реализовать быструю обработку данных и быстрое обучение преобразования.

В настоящей работе будет показано, что небольшая модификация системного инварианта быстрых алгоритмов приводит к структурам свёрточных нейронных сетей. При этом удастся сохранить алгоритм быстрого обучения и расширить информационную емкость сети по распознаваемым образом вплоть до максимально-возможной, определяемой числом нейронов выходного слоя сети. Предлагаемый вариант архитектуры нельзя назвать свёрточными сетями, поскольку в плоскостях нейронных слоев используется более общие преобразования, чем свертка, слои пулинга в преобразовании отсутствуют, но принцип пулинга все же используется при обучении сети. Все нейроны имеют линейные функции активации, нелинейная обработка существует, но реализуется не за счет функций активации, а за счет коммутации плоскостей нейронных сетей. В какой-то мере это подобно коммутирующим полулинейным функциям активации свёрточных нейронных сетей. Управление коммутацией плоскостей осуществляется по координатам нейронов в выходной плоскости сети. Назовем данный класс сетей нейронными сетями с управляемой коммутацией плоскостей (Control Switching of Planes Neural Networks - CSPNN).

Топология двумерных быстрых перестраиваемых преобразований

Обозначим через $F(U_y, U_x)$ матрицу изображения размерностью $N_y \times N_x$. При воздействии на изображение линейного преобразования $H(U_y, U_x; V_y, V_x)$ получается массив из $M_y \times M_x$ коэффициентов. Двумерное преобразование выполняется по правилу:

$$S(V_y, V_x) = \sum_{U_y=0}^{N_y-1} \sum_{U_x=0}^{N_x-1} F(U_y, U_x) H(U_y, U_x; V_y, V_x). \quad (1)$$

Необходимым условием существования быстрого алгоритма является возможность мультипликативной декомпозиции значений входных и выходных размерностей преобразования в равное число сомножителей:

$$\begin{aligned} N_y &= p_0^y p_1^y \dots p_{n-1}^y, & M_y &= g_0^y g_1^y \dots g_{n-1}^y, \\ N_x &= p_0^x p_1^x \dots p_{n-1}^x, & M_x &= g_0^x g_1^x \dots g_{n-1}^x \end{aligned} \quad (2)$$

Индексы x, y здесь означают принадлежность к осям координат исходного изображения, а значение n определяет число слоев в графе быстрого алгоритма. Для регулярных преобразований общего вида данное условие не является жестким ограничением, поскольку некоторые сомножители могут быть единичными. Тем не менее, чем больше число не единичных сомножителей, тем выше вычислительная эффективность алгоритма преобразования. Используя сомножители декомпозиций (2), координаты точек изображения представим в позиционной системе счисления со смешанными основаниями:

$$\begin{aligned} U_y &= \langle u_{n-1}^y u_{n-2}^y \cdots u_1^y u_0^y \rangle, \\ U_x &= \langle u_{n-1}^x u_{n-2}^x \cdots u_1^x u_0^x \rangle, \end{aligned} \quad (3)$$

где вес m -го разряда определяется выражением $p_{m-1}^* p_{m-2}^* \cdots p_1^* p_0^*$, а u_m^* - является разрядной переменной, принимающей значения $[0, p_m^* - 1]$ (звездочка здесь заменяет индексы x, y). Аналогично можно представить координаты спектральных коэффициентов в плоскости $[V_y, V_x]$:

$$\begin{aligned} V_y &= \langle v_{n-1}^y v_{n-2}^y \cdots v_1^y v_0^y \rangle, \\ V_x &= \langle v_{n-1}^x v_{n-2}^x \cdots v_1^x v_0^x \rangle, \end{aligned} \quad (4)$$

где вес m -го разряда определяется выражением $g_{m-1}^* g_{m-2}^* \cdots g_1^* g_0^*$, а v_m^* является разрядной переменной, принимающей значения $[0, g_m^* - 1]$.

Алгоритм быстрого преобразования обычно представляется в виде графа с топологией различного вида. Поразрядную форму удобно использовать для аналитического описания графа топологии быстрого алгоритма. Например, для топологии «Кули-Тьюки с прореживанием по времени» граф может быть описан в виде лингвистического предложения [6] (топологической модели):

$$\left[\langle u_{n-1}^* u_{n-2}^* \cdots u_1^* u_0^* \rangle \langle u_{n-1}^* u_{n-2}^* \cdots u_1^* v_0^* \rangle \cdots \langle u_{n-1}^* u_{n-2}^* \cdots u_{m+1}^* u_m^* v_{m-1}^* v_{m-2}^* \cdots v_0^* \rangle \cdots \langle v_{n-1}^* v_{n-2}^* \cdots v_1^* v_0^* \rangle \right],$$

где словами являются поразрядные представления координатных чисел, а буквами – имена разрядных переменных. Число слов в предложении равно $n+1$. Первое и последнее слова в предложении соответствуют координатам точек терминальных плоскостей из выражений (3) и (4). Промежуточные слова определяют координаты U_y^m, U_x^m и V_y^m, V_x^m в плоскостях внутренних слоев быстрого алгоритма. Для алгоритма с замещением значений выполняется условие:

$$U_y^{m+1} = V_y^m, \quad U_x^{m+1} = V_x^m. \quad (5)$$

Граф топологии в слое m содержит базовые операции (ядра) $W_{i_x^m, i_y^m}^m(u_m^y u_m^x; v_m^y v_m^x)$, представляющие собой четырехмерные матрицы размерности $[p_m^y, p_m^x; g_m^y, g_m^x]$. Где поразрядные выражения индексов ядер слоя m для выбранной топологии имеют вид:

$$i_x^m = \langle u_{n-1}^x u_{n-2}^x \cdots u_{m+1}^x v_{m-1}^x v_{m-2}^x \cdots v_0^x \rangle, \quad i_y^m = \langle u_{n-1}^y u_{n-2}^y \cdots u_{m+1}^y v_{m-1}^y v_{m-2}^y \cdots v_0^y \rangle. \quad (6)$$

Выражение (6) является аналитическим представлением системного инварианта быстрых алгоритмов [6]. В общем случае топологии для направлений x и y могут быть различными. Взаимосвязь между базовыми операциями определяется фактор-графом (структурной моделью быстрого преобразования), где каждой вершине соответствует нейронное ядро. Для выбранной топологии граф структурной модели описывается лингвистическим предложением:

$$\left[\langle u_{n-1}^* u_{n-2}^* \dots u_1^* \rangle \langle u_{n-1}^* u_{n-2}^* \dots u_2^* v_0^* \rangle \dots \langle u_{n-1}^* u_{n-2}^* \dots u_{m+1}^* v_{m-1}^* v_{m-2}^* \dots v_0^* \rangle \dots \langle v_{n-1}^* v_{n-2}^* \dots v_1^* v_0^* \rangle \right]$$

Каждое слово в этом предложении определяет номер базовой операции i_*^m в слое m . Число слов в предложении равно n . На рис. 1 показана структурная модель быстрого двумерного преобразования для размерности изображения 8×8 . Входное изображение поступает на нижний слой, а спектральные коэффициенты получаются в верхнем слое. Вершинам модели соответствуют базовые операции (далее называемые нейронными ядрами) размерности $[2, 2; 2, 2]$. Ядро в слое m выполняет двумерное преобразование над пространственным блоком размером $p_m^y \times p_m^x$:

$$S^m(v_y^m, v_x^m) = \sum_{u_m^y} \sum_{u_m^x} F^m(U_y^m, U_x^m) W_{i_x^m, i_y^m}^m(u_m^y u_m^x; v_m^y v_m^x). \quad (7)$$

Соответствия $U_*^m \leftrightarrow (i_*^m, u_*^m)$ взаимно однозначно определяются по лингвистическим предложениям топологической и структурной модели.

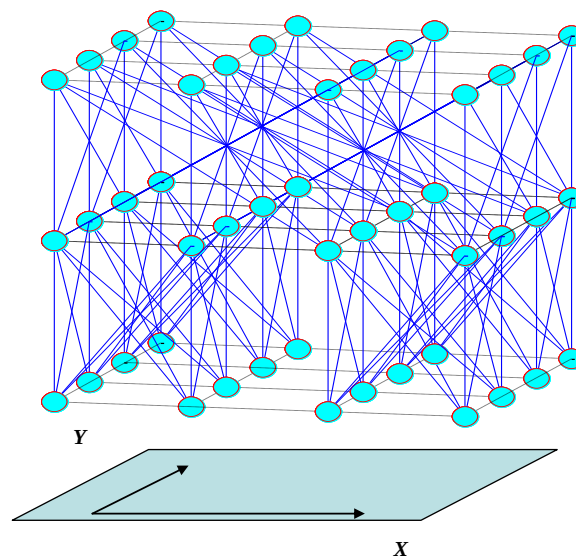


Рисунок 1 - Структурная модель двумерного быстрого преобразования

Мультипликативная декомпозиция элементов матрицы быстрого преобразования

Задание конкретных значений для всех разрядных переменных u_m^*, v_m^* (где m пробегает значения $0, 1, \dots, n-1$) определяет некоторый путь в топологическом графе между парой вершин начального и конечного слоя. Из однозначности поразрядного представления чисел следует, что такой путь единственен для каждого парного сочетания пространственных точек входной и выходной плоскости. Это обстоятельство позволяет получить удобное

аналитическое выражение, связывающее элементы матрицы быстрого преобразования с элементами ядер. Из выражения (1) следует что

$$H(U_y, U_x; V_y, V_x) = \frac{\partial S(V_y, V_x)}{\partial F(U_y, U_x)}. \quad (8)$$

Дифференцируя (8) по правилу дифференцирования сложной функции получим:

$$H(U_y, U_x; V_y, V_x) = \frac{\partial S^{n-1}}{\partial F^{n-1}} \frac{\partial F^{n-1}}{\partial S^{n-2}} \frac{\partial S^{n-2}}{\partial F^{n-2}} \cdots \frac{\partial F^1}{\partial S^0} \frac{\partial S^0}{\partial F^0}.$$

Из условия (5) следует, что для всех m имеет место $\frac{\partial F^m}{\partial S^{m-1}} = 1$, а из (7), – что

$\frac{\partial S^m}{\partial F^m} = W_{i_x^m, i_y^m}^m(u_m^y u_m^x; v_m^y v_m^x)$. Таким образом получим, что каждый элемент четырехмерной матрицы преобразования H выражается через элементы ядер в виде произведения:

$$H(U_y, U_x; V_y, V_x) = W_{i_x^{n-1}, i_y^{n-1}}^{n-1}(u_{n-1}^y u_{n-1}^x; v_{n-1}^y v_{n-1}^x) W_{i_x^{n-2}, i_y^{n-2}}^{n-2}(u_{n-2}^y u_{n-2}^x; v_{n-2}^y v_{n-2}^x) \dots W_{i_x^0, i_y^0}^0(u_0^y u_0^x; v_0^y v_0^x), \quad (9)$$

где поразрядные выражения индексов ядер слоя m для выбранной топологии определяются выражением (6).

Мультипликативная декомпозиция двумерных изображений

Алгоритм мультипликативной декомпозиции основан на идеях фрактальной фильтрации [6] (в терминологии свёрточных нейронных сетей эта операция называется пулингом). Для двумерного случая фрактальная фильтрация представляет собой кратномасштабную обработку изображений, последовательно сжимающую его размеры вплоть до единственной точки. Схему фрактальной фильтрации можно представить в виде пирамиды показанной на рис 2. Основанием пирамиды является исходное изображение, $F(U_y, U_x)$ для которого аргументы U_y и U_x представлены в позиционной системе счисления (см. выражение (3)). В этом позиционном представлении зафиксируем все разряды кроме двух младших u_0^y и u_0^x . Если варьировать эти разряды по всем возможным значениям, то получим двумерную выборку размером $p_0^y \times p_0^x$. Под фрактальным фильтром понимается произвольный функционал Φ , действующий на этой выборке. Формально это можно записать в виде выражения:

$$F_1(\langle u_{n-1}^y u_{n-2}^y \cdots u_1^y \rangle, \langle u_{n-1}^x u_{n-2}^x \cdots u_1^x \rangle) = \Phi_{(u_0^y, u_0^x)} \left[F(\langle u_{n-1}^y u_{n-2}^y \cdots u_1^y u_0^y \rangle, \langle u_{n-1}^x u_{n-2}^x \cdots u_1^x u_0^x \rangle) \right]$$

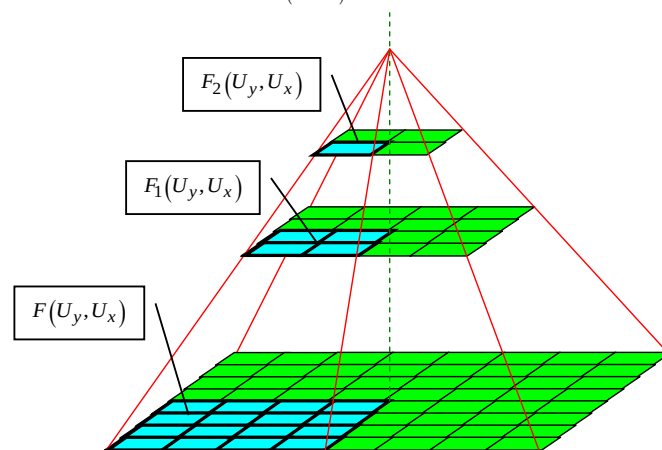


Рисунок 2 - Схема фрактальной фильтрации изображения

Очевидно, что изображение F_1 будет кратно уменьшенным по размерам по отношению к исходному изображению. Функционалом, например, может быть правило вычисления среднего значения выборки или ее медианы. Исходное изображение теперь можно формально представить в виде произведения:

$$F\left(\langle u_{n-1}^y u_{n-2}^y \cdots u_1^y u_0^y \rangle, \langle u_{n-1}^x u_{n-2}^x \cdots u_1^x u_0^x \rangle\right) = F_1\left(\langle u_{n-1}^y u_{n-2}^y \cdots u_1^y \rangle, \langle u_{n-1}^x u_{n-2}^x \cdots u_1^x \rangle\right) f_{j_y^0 j_x^0}(u_0^y, u_0^x),$$

где $f_{j_y^0 j_x^0}(u_0^y, u_0^x)$ - набор двумерных функций-множителей, зависящих от разрядных переменных u_0^y и u_0^x , а индексы j_y^0, j_x^0 выделяют двумерную функцию из этого набора. Значение этих индексов устанавливается равным значениям аргументов изображения F_1 , т.е. $j_y^0 = \langle u_{n-1}^y u_{n-2}^y \cdots u_1^y \rangle$ и $j_x^0 = \langle u_{n-1}^x u_{n-2}^x \cdots u_1^x \rangle$. Для получения функций множителей достаточно выполнить скалярное деление изображения F на изображение F_1 при варьировании всех разрядных переменных. Изображение F_1 в свою очередь также может быть представлено как произведение изображения F_2 на множители из набора $f_{j_y^1 j_x^1}(u_1^y, u_1^x)$. Повторяя многократно операцию фрактальной фильтрации и разложения, достигнем вершины пирамиды изображений и получим мультипликативную декомпозицию:

$$F\left(\langle u_{n-1}^y u_{n-2}^y \cdots u_1^y u_0^y \rangle, \langle u_{n-1}^x u_{n-2}^x \cdots u_1^x u_0^x \rangle\right) = f_{j_y^{n-1} j_x^{n-1}}(u_{n-1}^y, u_{n-1}^x) f_{j_y^{n-2} j_x^{n-2}}(u_{n-2}^y, u_{n-2}^x) \cdots \\ \cdots f_{j_y^1 j_x^1}(u_1^y, u_1^x) f_{j_y^0 j_x^0}(u_0^y, u_0^x), \quad (10)$$

где индексы сомножителей определяются выражениями:

$$j_x^m = \langle u_{n-1}^x u_{n-2}^x \cdots u_{m+1}^x \rangle, \quad j_y^m = \langle u_{n-1}^y u_{n-2}^y \cdots u_{m+1}^y \rangle. \quad (11)$$

Настройка регулярных приспособленных преобразований

Преобразование назовем приспособленным к изображению, если одна из базисных функций преобразования с координатами (V_y, V_x) совпадает с данным изображением. Очевидно, что значение скалярного произведения изображения с этой функцией будет максимальным среди прочих коэффициентов спектральной области преобразования. В этом заключается цель настройки преобразования. Значение координат (V_y, V_x) функции в спектральной плоскости назовем точкой приспособления.

Приспособление может быть осуществлено и к нескольким изображениям. Если сравнить полученную мультипликативную декомпозицию изображения с декомпозицией быстрого преобразования (9), то нетрудно заметить, что они подобны. Причем множество индексов ядер в каждом слое покрывает множество индексов функций-сомножителей. Отсюда следует конструктивный вывод, что быстрое преобразование будет приспособлено, к изображению, когда ядра преобразования будут приспособлены к функциям-сомножителям разложения (10). Настройка ядер преобразования определяется правилом:

$$W_{i_x^m, i_y^m}^m(u_m^y u_m^x; v_m^y v_m^x) = f_{j_x^m j_y^m}(u_m^y, u_m^x). \quad (12)$$

Сравнивая выражения (6) для i_x^m, i_y^m и (11) для j_x^m, j_y^m , можно сделать вывод, что количество компонентов в мультипликативном разложении изображения (10) и количество ядер преобразования совпадают для слоя $m = 0$ (т.е. имеют место равенства $i_x^0 = j_x^0$ и $i_y^0 = j_y^0$), и меньше числа ядер для всех остальных слоев. Поэтому при настройке часть степеней свободы преобразования не используется. Свободно варьируемыми являются разрядные переменные $v_0^y v_0^x$, следовательно, ядро может быть приспособлено одновременно к $D = g_0^y g_0^x$ изображениям. Остальные слои имеют большее число степеней свободы и не могут ухудшить это значение. Таким образом, можно заключить, что быстрое преобразование можно приспособить не более чем к D различным изображениям. На этом возможности данного алгоритма настройки исчерпываются. Значение D назовем уровнем приспособленности преобразования.

Регулярные преобразования с дополнительными плоскостями

Оставаясь в рамках рассмотренной топологии, не использованные при настройке степени свободы можно доопределить несколькими способами (более детально см. [6]). В этом случае уровень приспособленности не меняется, но при этом видоизменяются остальные функции преобразования.

Рассмотрим альтернативное решение, которое заключается в расширении топологии дополнительными плоскостями, так чтобы в новой топологии использовать оставшиеся степени свободы для увеличения уровня приспособленности.

Уточним вначале правило выбора настраиваемых ядер для прежней топологии. Выразим координаты точки приспособления в позиционной системе счисления, обозначив разрядные переменные через y и x :

$$V_y = \langle y_{n-1}, y_{n-2} \dots y_0 \rangle, \quad V_x = \langle x_{n-1}, x_{n-2} \dots x_0 \rangle \quad (13)$$

Фиксированные значения разрядных переменных y_m, x_m соответствуют переменным v_m^y, v_m^x , поэтому (как это следует из (6)) при выборе данной точки приспособления, согласно правилу (12) должны настраиваться только ядра с номерами:

$$\begin{aligned} i_x^m &= \langle u_{n-1}^x u_{n-2}^x \dots u_{m+1}^x x_{m-1} x_{m-2} \dots x_0 \rangle, \\ i_y^m &= \langle u_{n-1}^y u_{n-2}^y \dots u_{m+1}^y y_{m-1} y_{m-2} \dots y_0 \rangle. \end{aligned} \quad (14)$$

В частности для $m = 0$ имеем:

$$i_x^0 = \langle u_{n-1}^x u_{n-2}^x \dots u_{m+1}^x \rangle, \quad i_y^0 = \langle u_{n-1}^y u_{n-2}^y \dots u_{m+1}^y \rangle.$$

Т.е. независимо от выбора точек приспособления все ядра нулевого слоя всегда должны настраиваться. При этом уровень приспособленности преобразования ограничен значением D .

Для увеличения уровня приспособленности введем в каждом слое преобразования дополнительные плоскости, структурно-копирующие основную плоскость слоя. Порядковые номера дополнительных плоскостей в пределах слоя определим правилом:

$$\pi_m = \langle x_{n-1} x_{n-2} \dots x_{m+1}, y_{n-1} y_{n-2} \dots y_{m+1} \rangle \quad (15)$$

Очевидно, что максимальное количество дополнительных плоскостей будет в нулевом слое, а по мере увеличения номера слоя количество дополнительных плоскостей будет уменьшаться, и в последнем слое получим $\pi_{n-1} = \langle \rangle$, т.е. дополнительных плоскостей не будет совсем. Таким образом, в новой топологии плоскость последнего слоя останется прежней, а в младших слоях появятся дополнительные плоскости. Поскольку правило порождения новых плоскостей не противоречит условию (14), то для настройки ядер преобразования можно использовать прежнее правило (12), расширив его для дополнительных плоскостей аргументом выбора плоскости:

$$W_{i_x^m, j_y^m}^m \langle \pi_m \rangle (u_m^y u_m^x; v_m^y v_m^x) = f_{j_x^m, j_y^m}^k (u_m^y, u_m^x). \quad (16)$$

Индекс k в правой части нумерует точку приспособления. Для $m = 0$ имеем $i_x^0 = j_x^0$ и $i_y^0 = j_y^0$, варьируемыми переменными в левой части являются номер плоскости $\pi_0 = \langle x_{n-1} x_{n-2} \dots x_1, y_{n-1} y_{n-2} \dots y_1 \rangle$ и разрядные переменные $v_0^y v_0^x$. Вместе они покрывают весь диапазон координат выходной плоскости. Этому диапазону отвечают возможные значения индекса k в правой части (16). Остальные слои не ухудшают степень приспособленности. Таким образом, преобразование с дополнительными плоскостями может быть приспособлено к $D_\pi = M_y \times M_x$ изображениям, т.е. каждой точке выходной плоскости будет соответствовать один опорный образ.

Архитектура сети с дополнительными плоскостями показана на рис. 3. Входное изображение подается одновременно на все плоскости входного слоя, слои разделены коммутаторами, которые управляются разрядными переменными координатных чисел выходного класса.

На рисунке 4 показаны двумерные функции преобразования, приспособленного к усредненным образам рукописных цифр [2]. Для данного преобразования выбрано:

$$N_y = N_x = p_0 p_1 p_2 = 4 \cdot 3 \cdot 2 = 24, \quad M_y = M_x = g_0 g_1 g_2 = 2 \cdot 2 \cdot 2 = 8.$$

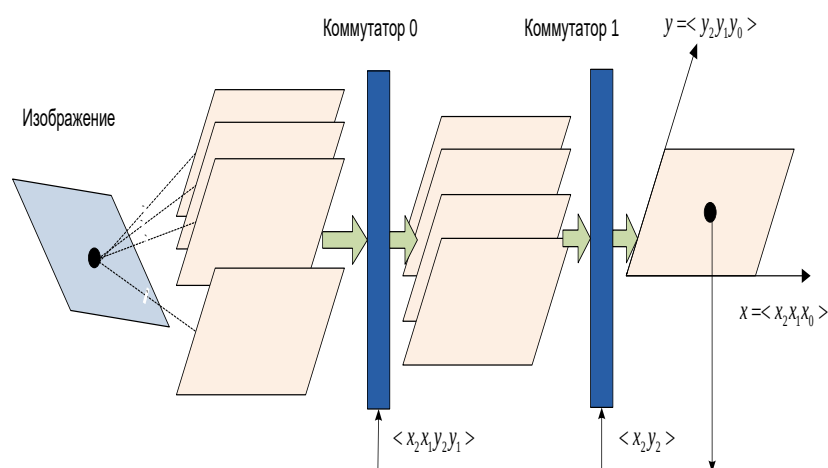


Рисунок 3 - Архитектура регулярной нейронной сети с дополнительными плоскостями

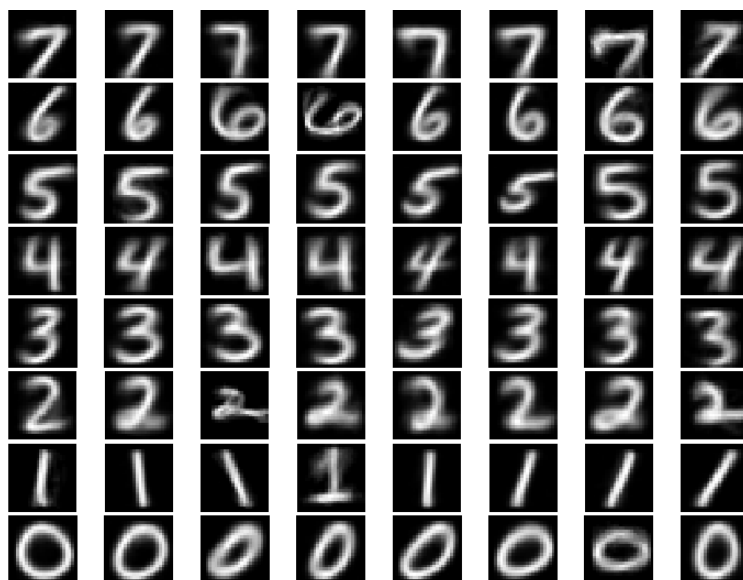


Рисунок 4 - Двумерные функции приспособленного преобразования

Настройка преобразования выполнена для подклассов рукописных цифр от 0 до 7. В общем случае все функции преобразования могут быть различны и соответствовать разным классам, т.е. в максимальном варианте данное преобразование может быть настроено на классификацию 64 различных классов.

При выполнении преобразования входное изображение поступает одновременно на все входные плоскости. Если изображение отвечает одной из точек приспособления, то значение этого коэффициента спектральной плоскости будет максимальным. Результат выполнения преобразования над образом цифры «0» показан на рисунке 5.

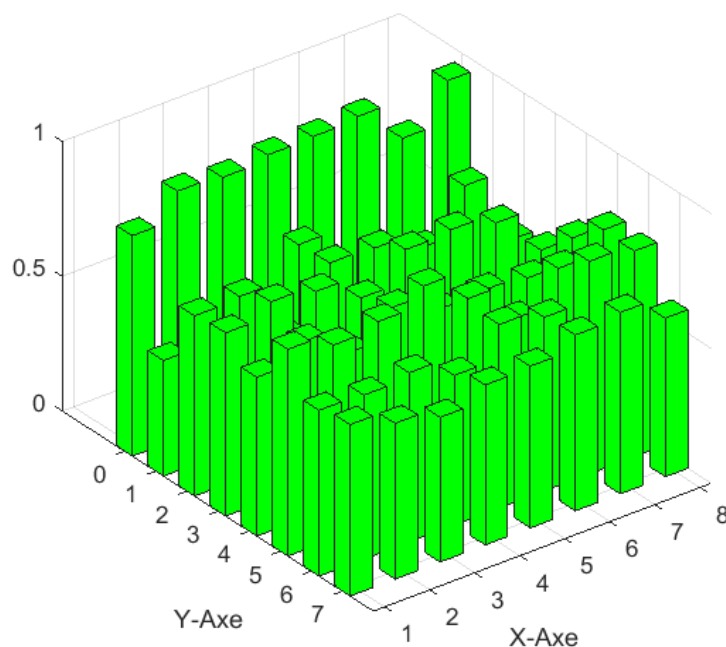


Рисунок 5 - Результат выполнения двумерного преобразования

Наглядно видно, что коэффициенты, отвечающие подклассам цифры «0», имеют максимальные значения.

Выводы

Регулярные перестраиваемые преобразования обладают уникальной возможностью аналитического представления топологии реализующей сети, что позволяет разработать алгоритмы обучения, абсолютно-сходящиеся за конечное число шагов. В работе показано, что реализующая топология легко расширяется дополнительными плоскостями, при этом число распознаваемых образов кардинально возрастает и покрывает все элементы выходной плоскости. Причем расширение топологии не нарушает принципа построения обучающего алгоритма. Построенная топология идеологически близка к топологиям свёрточных сетей глубокого обучения [2], но является регулярной. Представленное решение позволяет дать конструктивный ответ на принципиальные вопросы нейронных сетей глубокого обучения: как выбрать топологию и как сократить время обучения сети.

Библиографический список:

1. Fukushima K., Miyake S., Takayuki I. Neocognitron: A neural network model for a mechanism of visual pattern recognition. IEEE Transaction on Systems, Man and Cybernetics SMC-13(5):826–34. — 1983.
2. Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel: Backpropagation Applied to Handwritten Zip Code Recognition, Neural Computation, 1(4):541-551, Winter 1989.
3. Springenberg, Jost Tobias; Dosovitskiy, Alexey; Brox, Thomas & Riedmiller, Martin (2014-12-21), "Striving for Simplicity: The All Convolutional Net", arXiv:1412.6806 <https://arxiv.org/pdf/1412.6806>
4. Дорогов А.Ю., Алексеев А.А. // Математические модели быстрых нейронных сетей. В сб. научн. Тр. СПбГЭТУ «Системы управления и обработки информации». Вып.490, 1996, с.79-84.
5. Солодовников А.И., Спиваковский А.М. Основы теории и методы спектральной обработки информации: Учеб. пособие. Л.: Изд-во ЛГУ, 1986. 272с.
6. Дорогов А.Ю. Теория и проектирование быстрых перестраиваемых преобразований и слабосвязанных нейронных сетей. СПб.: «Политехника», 2014. 328 с.

УДК 004

В.А. Евсин, студент

В.А. Евсина, студент

С.Н. Широкова, канд. экон. наук, доцент

Южно-Российский государственный политехнический университет (НПИ)
имени М.И. Платова

ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ МОДУЛЯ ОЦЕНКИ СТОИМОСТИ АРЕНДЫ НЕДВИЖИМОСТИ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ

Аннотация: в данной статье представлены математические и проектные модели при реализации искусственной нейронной сети для оценки стоимости аренды недвижимости. В частности, рассмотрен процесс формирования входных параметров нейронной сети. Представлены математические модели определения интегральных показателей недвижимости для формирования обучающей и тестовой выборки данных. Рассмотрен процесс формирования искусственной нейронной сети для оценки стоимости аренды недвижимости с использованием нотации UML. Приведен графический интерфейс процесса обучения искусственной нейронной сети. В заключение, представлены итоги и перспективы использования искусственных нейронных сетей для оценки стоимости аренды недвижимости.

Ключевые слова: аренда недвижимости, нейронные сети, data mining, машинное обучение, информационные системы.

Annotation: this article presents mathematical and design models for the implementation of artificial neural network to assess the cost of real estate rental. In particular, the process of forming the input parameters of the neural network is considered. Mathematical models of definition of integral indicators of real estate for formation of training and test sample of data are presented. The process of forming an artificial neural network to estimate the cost of real estate lease using UML notation is considered. The graphical interface of the learning process of an artificial neural network is presented. In conclusion, the results and prospects of using artificial neural networks to assess the cost of renting real estate are presented.

Key words: real estate rental, neural networks, data mining, machine learning, information systems.

В ходе массовой информатизации последних лет и увеличения объемов хранимых данных все большее значение приобретает способность их анализировать для прогнозирования наиболее оптимальных результатов. Одним из способов для проведения оценки являются искусственные нейронные сети. Для реализации оценки стоимости аренды недвижимости с использованием искусственных нейронных сетей необходимо подать на вход набор данных, которые будут обработаны послойной нейронной сетью, к которым можно отнести следующие:

- благоустроенность помещения: интегральный показатель, характеризующий наличие различных инвентарных предметов в помещении;
- благоустроенность региона: интегральный показатель, характеризующий наличие различных типов объектов инфраструктуры в приемлемом радиусе;
- коэффициент технического оснащения помещения: показатель, характеризующий наличие предметов технического оснащения помещения;
- материал помещения;
- удаленность от центра.

Подробнее о данных коэффициентах описано в [1]. Коэффициент влияния объектов инфраструктуры региона на объект недвижимости имеет следующую форму:

$$IR = \sum_{t=0}^{nir} IR_t = \sum_{t=0}^{nir} M_t * e^{-a_t \bar{r}_t}, \quad (1)$$

где IR – интегральный показатель влияния объектов инфраструктуры региона, IR_t – показатель влияния объектов инфраструктуры t -го типа, M_t – мультипликатор влияния объектов инфраструктуры t -го типа, a_t – коэффициент насыщения объектами

инфраструктуры t -го типа, $\bar{r}_t$ – усредненное расстояние до объектов инфраструктуры t -го типа.

Интегральная оценка благоустроенности помещения складывается из оценки характеристических показателей помещения, а также показателей состояния отдельных предметов в помещении и имеет следующий вид:

$$IH = \sum_{sh=1}^{rih} W_{1,sh} * C_{1,sh} + \sum_{iih=1}^{nih} W_{2,iih} * C_{2,iih} * (1 - a_{2,iih}), \quad (2)$$

где IH – коэффициент благоустроенности помещения, $W_{1,sh}$ – весовой коэффициент sh -го характеристического показателя, $C_{1,sh}$ – степень качества sh -го характеристического показателя, $W_{2,iih}$ – весовой коэффициент от наличия iih -го предмета в помещении, $C_{2,iih}$ – степень ценности iih -го предмета в помещении, $a_{2,iih}$ – коэффициент износа iih -го предмета в помещении.

Коэффициент технического оснащения квартиры включает следующие данные: газовое обеспечение, наличие интернета, холодное водоснабжение, горячее водоснабжение, водоотведение, электричество. Для определения весовых коэффициентов от наличия каждого из компонент необходимо определить следующий функционал качества:

$$\begin{cases} TSJ(x) = \sum_{its=1}^{mts} [C^T - (k_{1jts} * x_1 + k_{2jts} * x_2 + \dots + k_{7jts} * x_7)]^2 \rightarrow \min \\ 0 < x_{its} < 1 \\ \sum_{its} x_{its} = 1 \end{cases}, \quad (3)$$

где $its = \overline{1, nts}$ – компоненты технического оснащения, x_{its} – весовой коэффициент its -й компоненты, $k_{its,jts}$ – коэффициент наличия its -й компоненты в jts -м помещении, причем:

$$k_{its,jts} = \begin{cases} 1, \text{ если } jts - \text{е помещение обеспечено } its - \text{м компонентом} \\ 0, \text{ иначе} \end{cases}. \quad (4)$$

Данный функционал качества исходит из предположения, что наличие каждой компоненты положительно сказывается на стоимости недвижимости. Для решения данного функционала может быть использован аппарат генетических алгоритмов.

Хромосома может быть представлена в следующей форме:

$$Ch = [b_1; b_2; \dots; b_{41}; b_{42}], b_{its} = \{0, 1\}. \quad (5)$$

В качестве начальной популяции принимаются наборы бинарных данных, которые в десятичном представлении формируют допустимое множество решений. Фитнес-функция модели имеет следующую форму:

$$FF = \sum_{jts=0}^{mts} \left(C_{its}^T - \left| \sum_{its} \frac{Dec(BinArray_{its})}{100} * p_{jts,its} \right| \right), \quad (6)$$

где FF – фитнес-функция модели, $Dec(BinArray_{its})$ – десятичное представление its -го параметра хромосомы, $p_{jts,its}$ – значение its -го параметра jts -й строки набора данных. В качестве модели селекции используется модель элитарного отбора по критерию минимальной фитнес-функции хромосом.

При проведении оценки стоимости недвижимости необходимо провести комплекс следующих мер:

- сформировать реестр параметров недвижимости с указанием итоговой стоимости: в ходе выполнения данной процедуры должны быть вычислены интегрированные показатели помещения;

- на основании реестров параметров недвижимости сформировать модель нейронной сети: в ходе выполнения данного этапа необходимо сформировать модель нейронной сети, обучить ее и сохранить в файл;

- провести анализ сформированной модели с учетом скорости и эффективности обучения;

– использовать сохраненную модель для оценки стоимости аренды недвижимости.

На рисунке 1 представлена диаграмма деятельности процесса формирования нейронной сети в нотации UML.

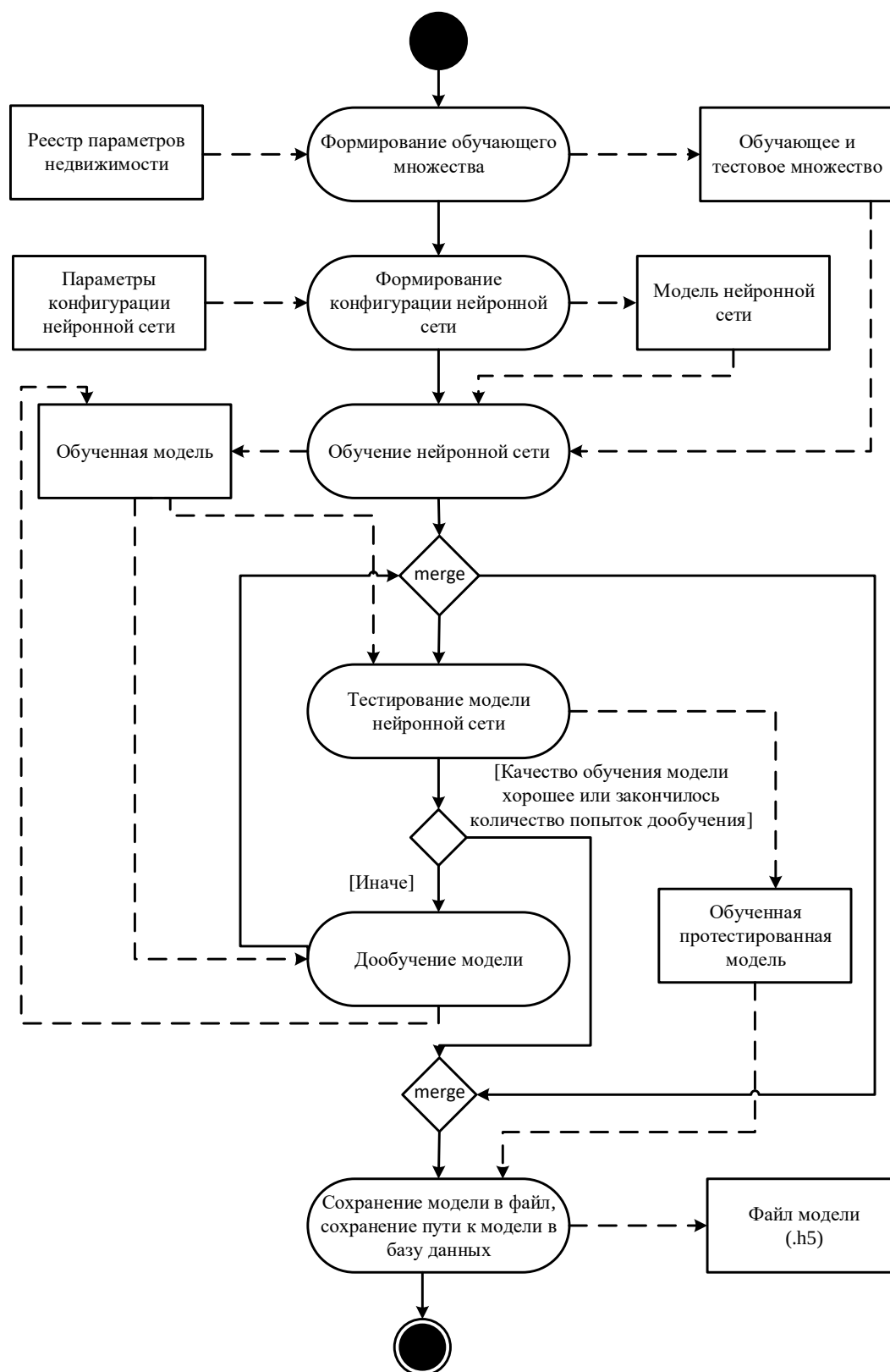


Рисунок 1 – Диаграмма деятельности процесса формирования искусственной нейронной сети для оценки стоимости аренды недвижимости

На рисунке 2 представлена диаграмма деятельности *UML* процесса определения стоимости аренды недвижимости по сохраненной модели.

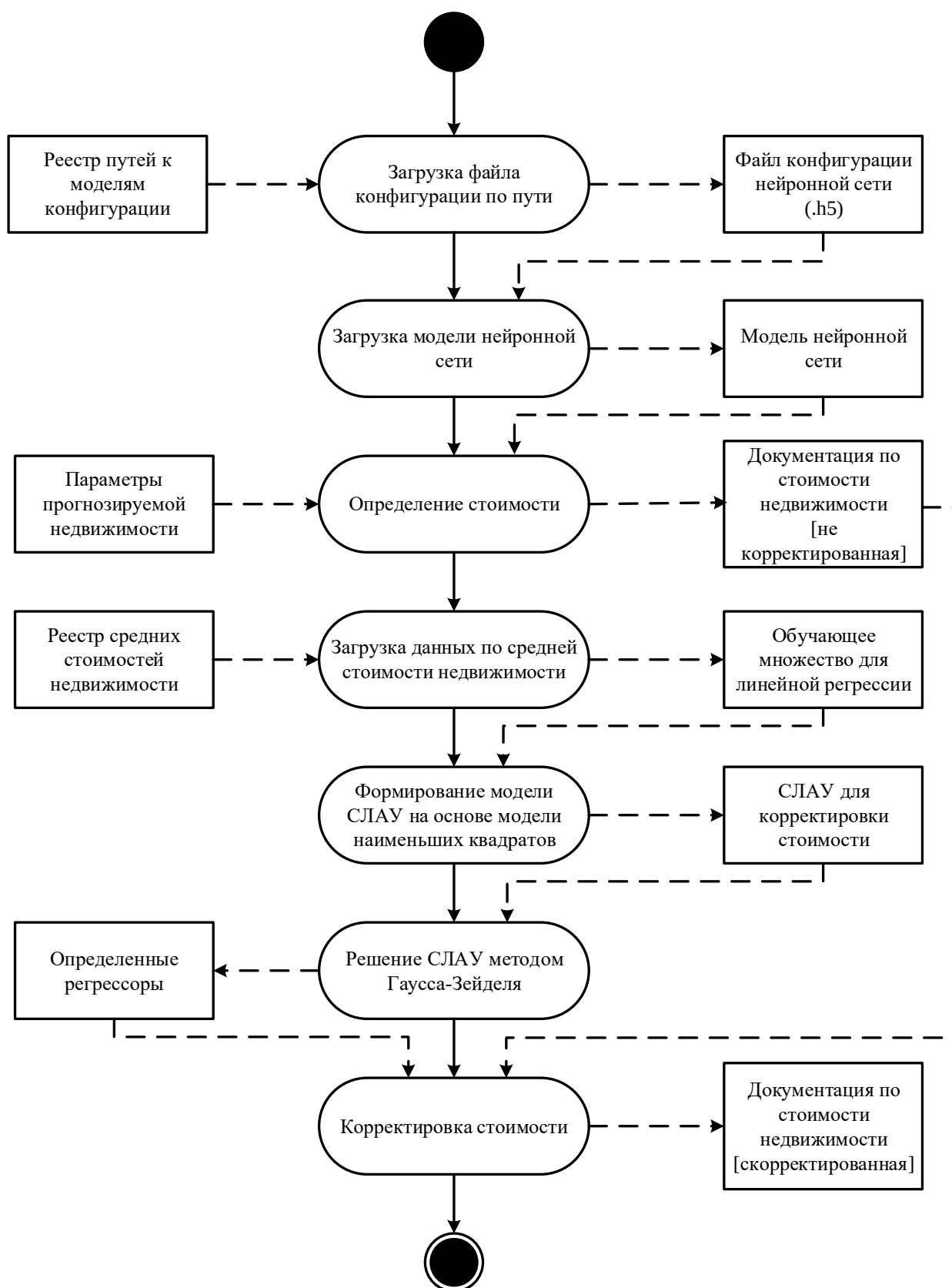


Рисунок 2 – Диаграмма деятельности процесса оценки стоимости аренды на основе сохраненной нейронной сети

Для реализации нейронной сети была использована послойная модель сети, с конфигурацией нейронов: 7/64/128/64/32/16/8/4/2/1. В качестве функции активации использована масштабная экспоненциальная линейная модель, которая имеет следующую форму:

$$SELU(x) = \lambda \begin{cases} x, & \text{если } x > 0 \\ ar * e^x - a, & \text{иначе} \end{cases}$$

Для проведения оптимизации была использована адаптивная ускоренная моментная модель Нестерова. Процесс обучения нейронной сети представлен на рисунке 3.

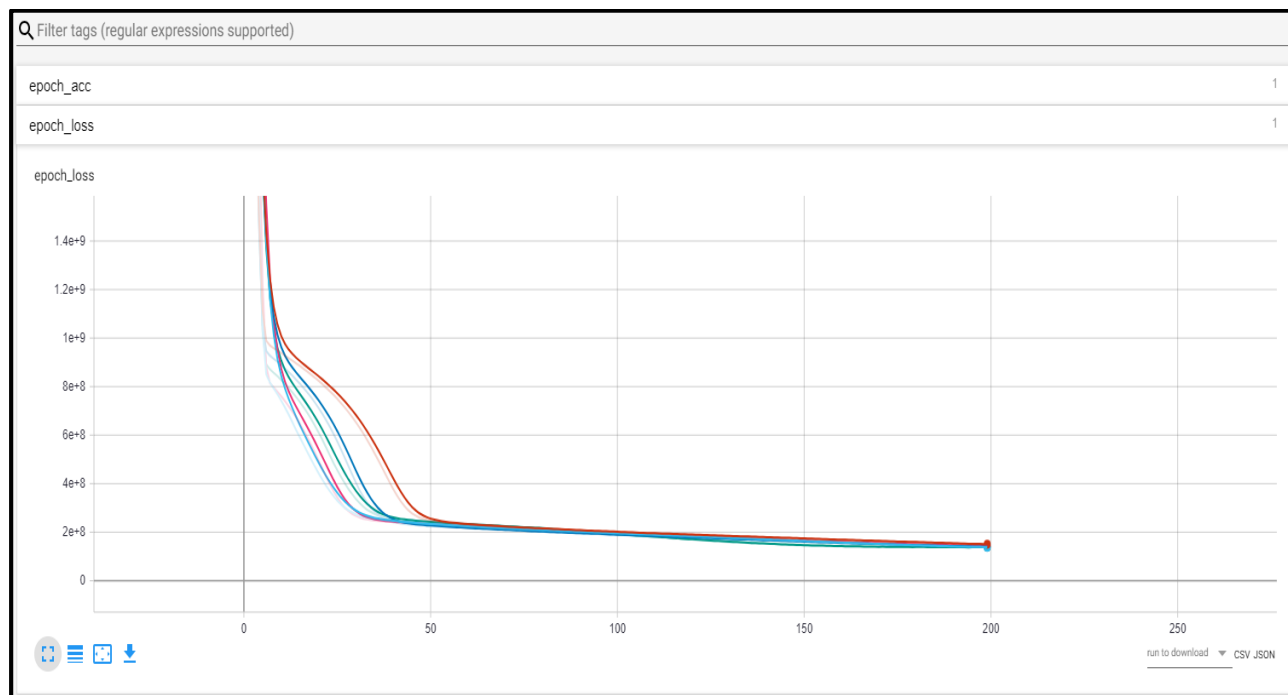


Рисунок 3 – Диаграмма динамики обучения нейронной сети для оценки стоимости аренды недвижимости

В ходе обучения нейронной сети ошибка обучения минимизируется, что говорит о положительной динамике обучения. Таким образом, можно сделать вывод о положительном результате использования искусственной нейронной сети для оценки стоимости аренды недвижимости, например, в распределенном реестре в сфере аренды недвижимости, подробнее о котором в [2,3].

Библиографический список:

1. Евсин В.А., Продан Е.А., Евсина В.А., Широбокова С.Н. Математическое моделирование процесса оценки стоимости аренды недвижимости с использованием искусственных нейронных сетей // Инженерный вестник Дона. - 2019. - №1. URL: http://www.ivdon.ru/uploads/article/pdf/IVD_6_Evsin.pdf_c5bba2b6b0.pdf
2. Евсин В.А., Широбокова С.Н., Продан Е.А. Использование технологии распределенных реестров при проектировании информационной системы «Аренда недвижимости» с применением искусственных нейронных сетей // Инженерный вестник Дона. - 2018. - №1. URL: http://www.ivdon.ru/uploads/article/pdf/IVD_39_Evsin.pdf_684c1efe21.pdf
3. Евсин В.А., Широбокова С.Н., Евсина В.А., Продан Е.А. Математическое моделирование распределенного реестра в сфере аренды недвижимости как сети массового обслуживания // Инженерный вестник Дона. - 2018. - №3. URL: http://www.ivdon.ru/uploads/article/pdf/IVD_27_Evsin_Shirobokova.pdf_14e9e70646.pdf

УДК 004

Д.М. Жевакин, студент

С.Н. Широкова, канд. экон. наук, доцент

Т.И. Перекрестова, студент

Южно-Российский государственный политехнический университет (НПИ)
имени М.И. Платова

ИСПОЛЬЗОВАНИЕ НЕЙРОСЕТЕВОГО ПОДХОДА И ТЕХНОЛОГИЙ BIG DATA ДЛЯ ПРОГНОЗИРОВАНИЯ ПОКАЗАТЕЛЕЙ РЕАЛЬНЫХ СЕКТОРОВ ЭКОНОМИКИ СУБЪЕКТОВ РФ

Аннотация: в статье предлагаются модели и алгоритмы, используемые для решения задач анализа и прогнозирования показателей реального сектора экономики субъектов РФ. Описаны данные, оказывающие влияние на принятие решений в этой сфере, алгоритмы, способные вывести необходимую рекомендацию по востребованным регионам и секторам экономики. Описан программный продукт, реализующий предложенные модели.

Ключевые слова: анализ, прогнозирование, Big Data, реальный сектор экономики.

Annotation: The article suggests models and algorithms used to solve the problems of analyzing and forecasting indicators of the real sector of the economy of the constituent entities of the Russian Federation. Describes the data that influence decision making in this area. Algorithms capable of deriving the necessary recommendation for the demanded regions and sectors of the economy. A software product that implements the proposed model is described.

Key words: analysis, forecasting, Big Data, real sector of economy.

Целью создания информационной системы (далее ИС) по анализу и прогнозированию реальных секторов экономики субъектов РФ является автоматизация формирования рекомендаций инвестору, а именно: в какой регион стоит вкладывать средства и в какой сектор экономики [1]. На рисунке 1 представлена основная бизнес-функция данной ИС.



Рисунок 1 – Главная бизнес-функция ИС

Источники данных, необходимые для работы ИС:

1. Центральный банк РФ:

– кредиты, предоставленные юридическим лицам - резидентам и индивидуальным предпринимателям, по 30 крупнейшим банкам (региональный разрез);

- кредиты, предоставленные субъектам малого и среднего предпринимательства (региональный разрез);
- кредиты, предоставленные физическим лицам - резидентам (региональный разрез).

2. Федеральная служба государственной статистики:

- объём инвестиций в основной капитал;
- структура инвестиций в основной капитал по источникам финансирования.

Показатели для анализа:

- динамика суммы выданных кредитов (по сравнению с 2016 г.);
- коэффициенты просроченности задолженностей;
- коэффициент прибыли предприятий;
- количество убыточных предприятий;
- коэффициент инвестиций в регион;
- коэффициент инвестиционной привлекательности объектов инвестирования.

Все методы, которые использовались для анализа всех этих показателей, опирались на основные методы математической статистики. Рекомендации инвестору, получаемые на выходе работы ИС:

- 1) наиболее привлекательные для инвестирования регионы;
- 2) наиболее перспективные объекты инвестирования;

Алгоритм «надежного субъекта» предназначен для определения субъектов, в которые можно инвестировать свои средства. Данный алгоритм может работать совместно с нейросетью для того, чтобы рассчитать будущие значения показателей и определить, какие регионы являются стабильными для вложения в будущие годы, а какие – нет. Данные планируется хранить в NoSQL СУБД *MongoDB*, которая позволяет хранить большое количество информации. Основным языком для взаимодействия всех модулей и компонентов системы и анализа статистики – *Python*.

Модель взаимодействия классов в системе можно представить на *UML* диаграмме классов, как показано на рисунке 2:

Алгоритм «надежного субъекта» позволяет каждому региону подобрать его коэффициент значимости, а также отсортировать все регионы в порядке убывания. Для начала работы алгоритма необходимо загрузить все показатели, отображающие денежные возможности субъекта, затем сложить сумму всех показателей каждого региона и найти вес для каждого показателя, потом необходимо посчитать сумму всех показателей с учетом их веса и отсортировать. Математически можно представить следующим образом:

$Subject = \{sub_i\}, i = \overline{1, m}$, где m – количество субъектов в стране,

$Sector = \{sector_j\}, j = \overline{1, n}$, где n – количество секторов экономики в стране.

$Money = \{money_k\}, k = \overline{1, p}$, где p – количество показателей денежной характеристики.

Множество $\{sub_i, money_k\}$ показывает информацию о денежной возможности региона по определенной характеристике. В состав множества входит $\langle CI, RI, DI, VI \rangle$, где CI – информация о загруженном показателе, RI – информация о субъекте, DI – дата загрузки, VI – значение показателя. Для того, чтобы сравнить между собой регионы и отсортировать их, необходимо найти значение индекса с учетом веса.

Пусть ω_k – вес каждого показателя, тогда

$$\omega_k = \frac{\sum_{k=1}^p money}{money_k}, \quad (1)$$

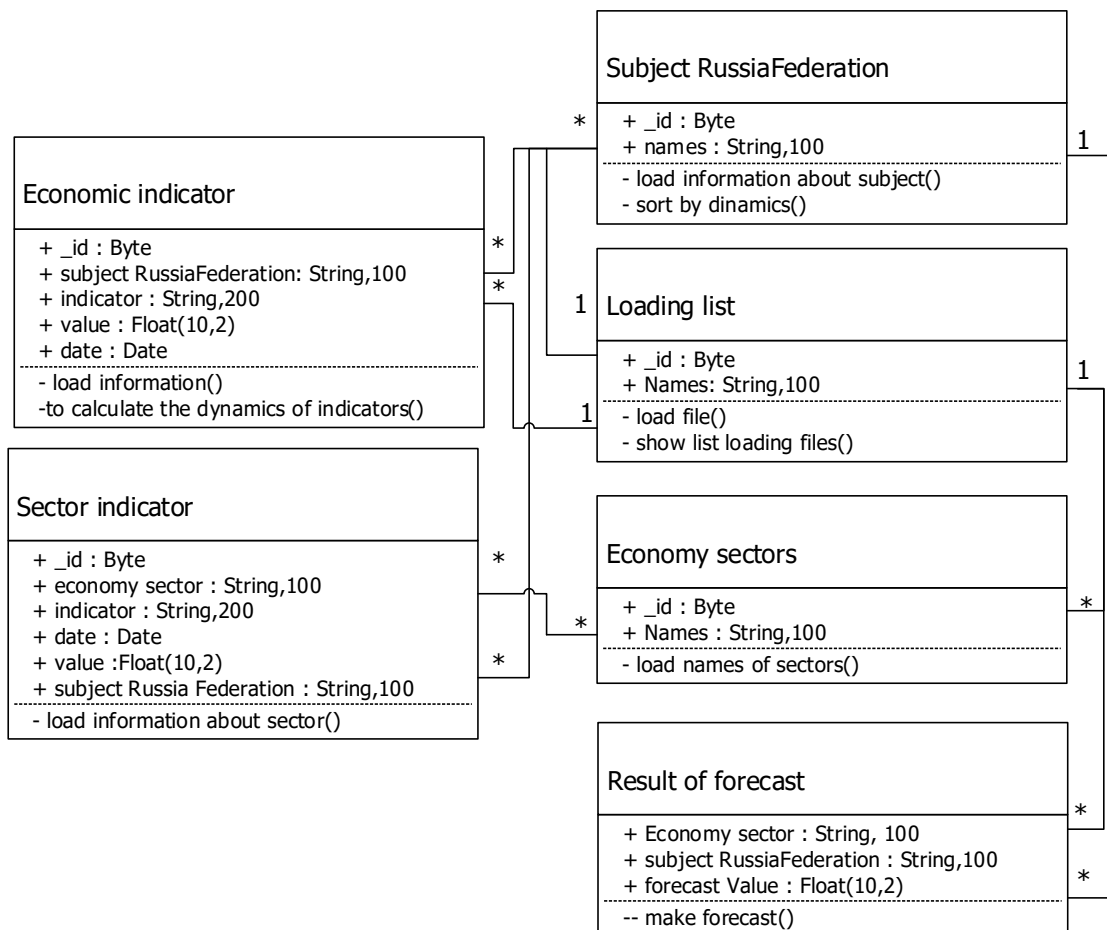


Рисунок 2 – UML классов информационной системы

$$money\omega_k = \sum_{k=1}^p \omega_k * money_k, \quad (2)$$

где $money\omega_k$ – это суммарный денежный показатель с учетом веса. По этому показателю в дальнейшем будет производиться сортировка регионов по денежной способности.

Множество $\{sub_i, sector_j\}$ показывает информацию о возможностях региона в определенном реальном секторе экономики. Для лучшего прогнозирования результатов необходимо, чтобы тестовое множество состояло из как можно большего количества периодов. В состав множества входят: $\langle SI, RI, DI, VI \rangle$, где SI – информация о реальном секторе экономики. Основная особенность данного алгоритма состоит в том, что необходимо распознавать дату в любом ее представлении для того, чтобы передать информацию для нейронной сети. Нейронная сеть в ИС необходима для прогноза будущих результатов. Для того, чтобы спрогнозировать данные необходимо предоставить тестовое множество, состоящее из элементов множества $\{sub_i, sector_j\}$. После того, как данные были переданы на первый входной слой, они переходят на внутренний слой, после которого они перейдут на выходной слой, состоящий из одного выхода.

Основную бизнес-функцию можно разделить на три этапа:

- определение субъектов РФ с надежной выплатой;
- анализ показателей реальных секторов экономики в субъектах;
- прогнозирование будущих показателей реальных секторов экономики в субъектах

РФ.

Диаграмма, описывающая данную бизнес-функцию, изображена на рисунке 3.

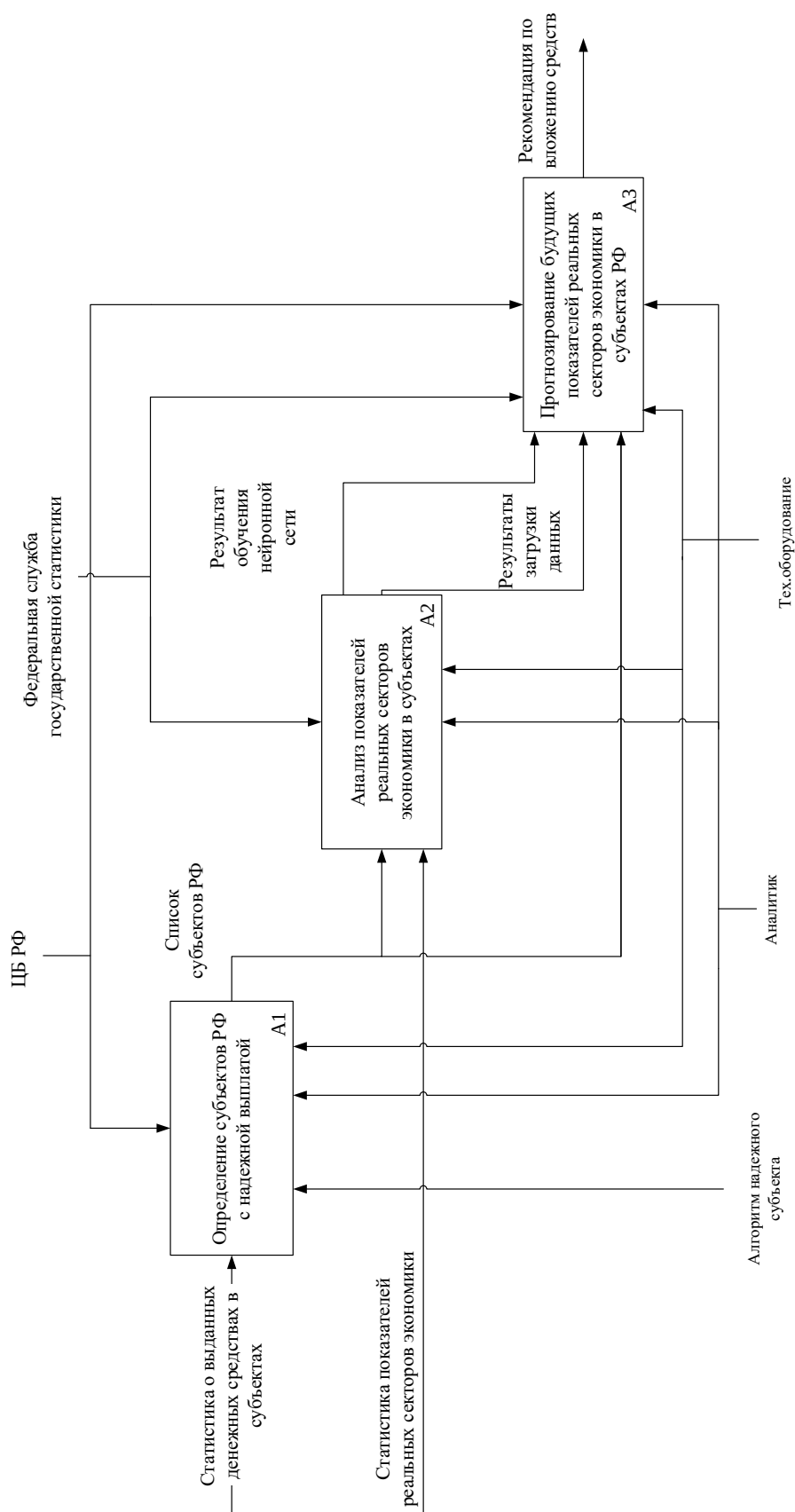


Рисунок 3 – Декомпозиция главной бизнес-функции ИС

Для данной ИС нейронная сеть должна правильно прогнозировать будущие данные, основываясь на результатах прошлых лет. Для формирования алгоритма и обучения будем использовать сигмоидальную функцию, также ее называют логистической функцией.

Реализация данной ИС представлена на рисунках 4-5. Для того, чтобы отобразить регионы по лучшему денежному показателю, с помощью алгоритма «Надежного субъекта», необходимо загрузить данные. Данные могут быть представлены в разном виде, а также иметь различные форматы даты, поэтому данные необходимо привести к общему формату. Для этого используются флажки «изменять дату» и «составная дата». Первый предполагает изменение даты из формата текстового, например, «январь 2019» в формат числовой «01.01.2019». Второй предполагает, что даты могут находиться на разных строках, например, в строке номер 3 может быть указан год, а в строке номер 4 указан месяц. Поэтому их необходимо будет объединить и использовать в системе. Также для будущего прогноза необходимо загрузить информацию о названиях регионов и название секторов экономики. Для этого есть кнопки «загрузить регионы» и «загрузить сектора экономики». После того, как названия секторов были загружены необходимо загрузить информацию о каждом секторе с помощью кнопки «загрузить данные о секторе», предварительно выбрав один из секторов экономики. После того, как информация была загружена, необходимо перейти на экран отбора регионов.

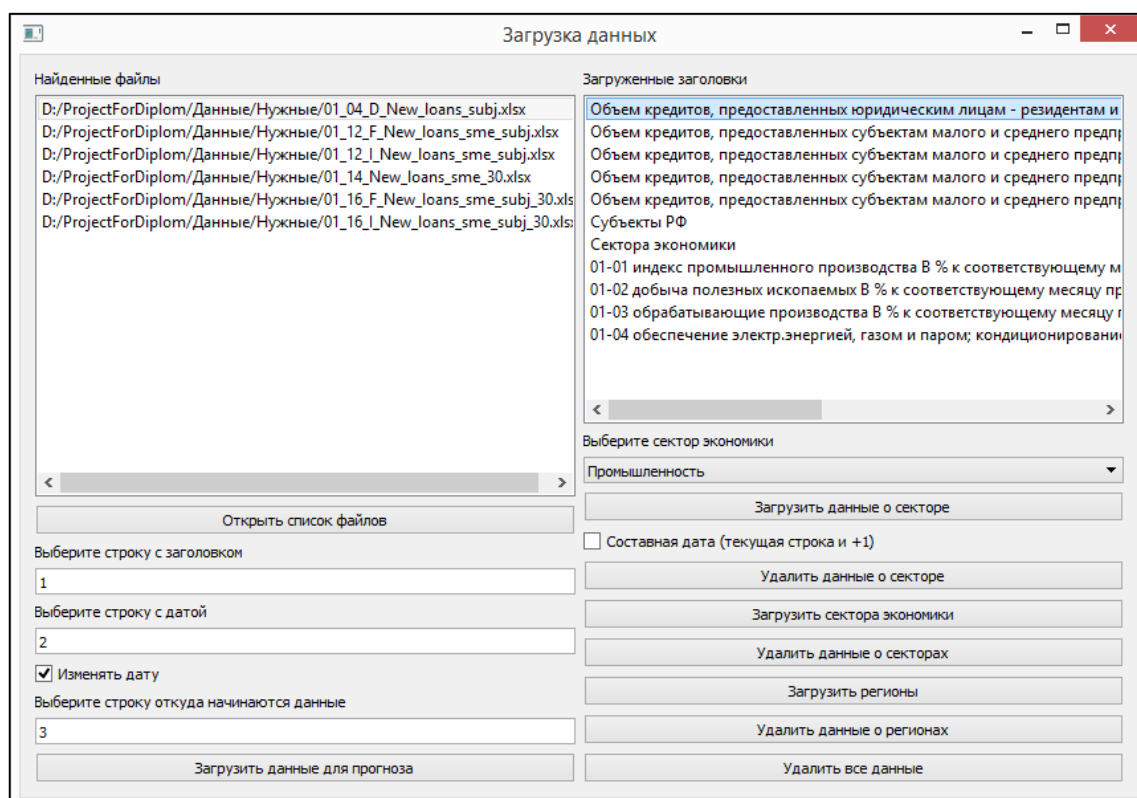


Рисунок 4 – Экранная форма загрузки данных

Далее можно отобразить необходимо количество субъектов. Все они будут отфильтрованы с помощью алгоритма «Надежного субъекта» по убыванию. После можно сделать прогноз на будущий месяц одного из субъектов. Необходимо выбрать из списка слева и нажать на кнопку «спрогнозировать для региона», затем появится информация на следующий месяц по каждому из загруженных показателей реальных секторов экономики.

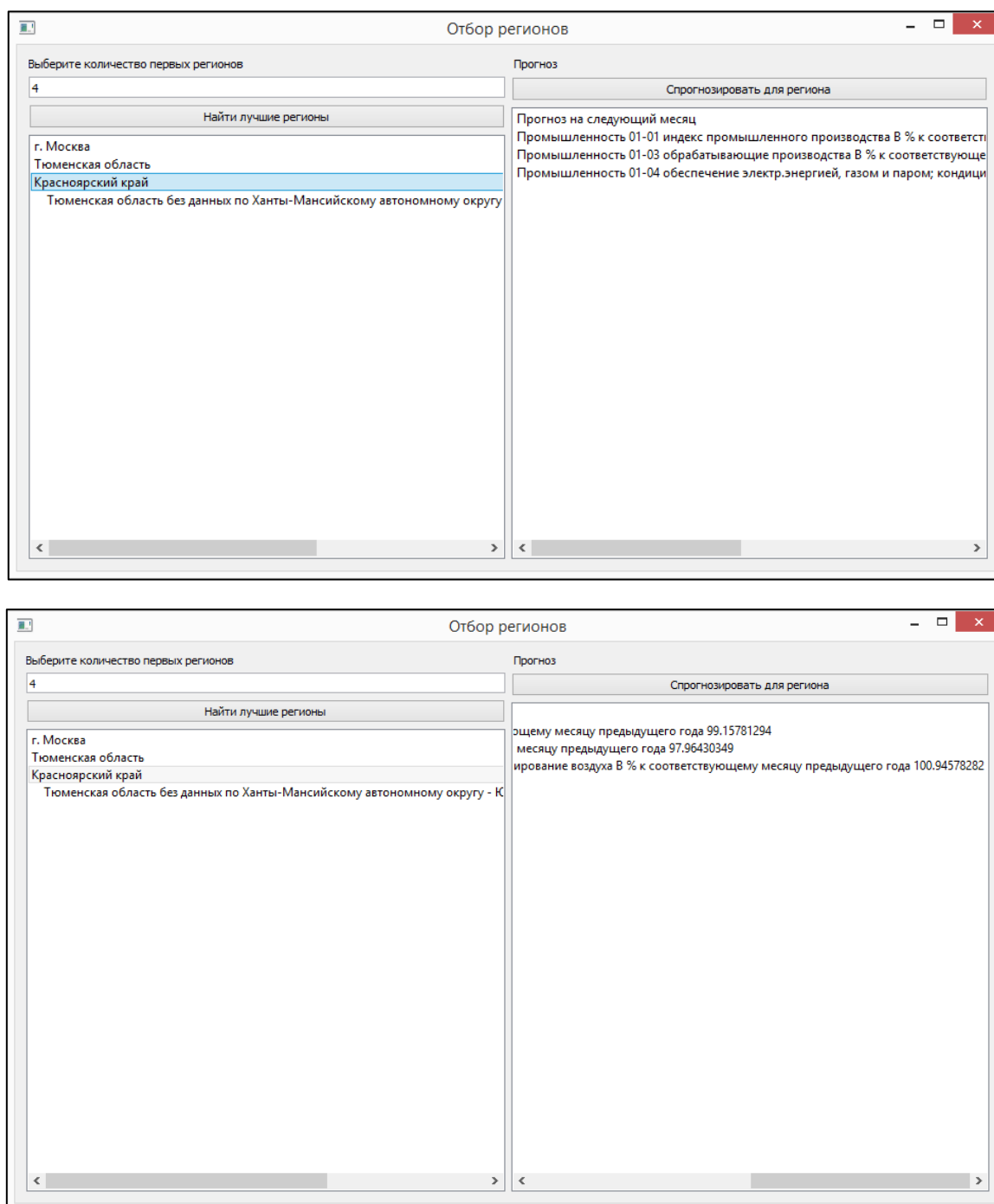


Рисунок 5 – Экранные формы отбора регионов и прогнозирования

Данная информационная система должна помочь аналитикам оценивать возможные изменения на рынке реальных секторов экономики, а также помогать правильно приводить рекомендации по вложению денежных средств в определенный субъект РФ в определенный сектор экономики.

Библиографический список:

1. Жевакин Д.М., Перекрестова Т.И., Диков М.Е. Анализ и прогнозирование показателей реальных секторов экономики субъектов РФ с использованием технологии Big Data // Статистический анализ социально-экономического развития субъектов Российской Федерации: матер. 6-ой Междунар. науч.-практ. конф. / Брянский государственный инженерно-технологический университет.– Брянск, 2019. – С 157-160.

УДК 004.048

Д.А. Ситков, магистрант

В.И. Шевченко, канд. техн. наук, доцент

Севастопольский государственный университет

ИССЛЕДОВАНИЕ МОДЕЛИ СИСТЕМЫ РАСПОЗНАВАНИЯ ЛИЦ НА ОСНОВЕ МЕТОДА ГЛАВНЫХ КОМПОНЕНТ

Аннотация: приведены общие сведения о распознавании лиц методом главных компонент, а также результаты экспериментальных исследований над моделью системы распознавания лиц. На основе анализа данных полученных в ходе экспериментов были выделены оптимальные параметры, которые могут быть использованы при разработке системы идентификации пользователя по его изображению.

Ключевые слова: компьютерное зрение, распознавание лиц, метод главных компонент, метрики расстояния, рса

Annotation: provides general information about facial recognition by the method of principal components, as well as the results of experimental studies on the model of facial recognition. Based on the analysis of the data obtained during the experiments, the optimal parameters were identified that can be used in the development of a user identification system based on his image.

Keywords: computer vision, face recognition, principal component method, distance metrics, pca

В настоящее время все более актуальной становится задача автоматизации идентификации пользователей. Несмотря на множество возможных способов решения данной задачи, наиболее перспективным является способ распознавания по лицу благодаря своей естественности и отсутствию необходимости в физических контактах для идентификации. Актуальность данной задачи также обусловлена растущими потребностями в распознавании лиц в таких областях деятельности, как организация видеоконференций, биометрия, интеллектуальные системы контроля доступа и безопасности, машинное зрение.

В рамках данной статьи рассматриваются результаты экспериментальных исследований над моделью системы идентификации пользователя по его изображению, а именно исследуется влияние числа главных компонент и выбор меры близости на точность распознавания. Целью проведения исследований является задача повышения эффективности процессов идентификации пользователя по его изображению для осуществления контроля доступа при работе студентов с дистанционными курсами. Предлагаемая работа, является продолжением исследований, изложенных в [1], в основу исследований положен метод главных компонент.

Метод главных компонент. Данный метод является статистическим и оперирует не изображениями, а векторами в линейном пространстве [2]. Идея метода состоит в представлении изображений лиц в виде набора (вектора) главных компонент изображений, называемых «собственные лица», для дальнейшего сравнения с эталонными векторами из базы изображений лиц [3]. При этом предполагается, что изображения лиц, соответствующих одному человеку, сгруппированы в кластеры в собственном пространстве. Процесс вычисления главных компонент сводится к вычислению собственных векторов и собственных значений ковариационной матрицы исходных данных или к сингулярному разложению матрицы данных. Из базы данных выбираются изображения-кандидаты, имеющие наименьшее расстояние от входного (неизвестного) изображения. Данный метод является одним из наиболее распространенных методов для уменьшения размерности данных, с потерей наименьшего количества информации [4]. Метод применяют в различных областях, таких как, компьютерное зрение, распознавание образов, сжатие данных и т. п. Также метод главных компонент применяется для обнаружения лица на изображении [5].

Для лиц значения компонент в собственном пространстве имеют большие значения, а в дополнении собственного пространства - близки к нулю. По этому факту можно обнаружить, является ли входное изображение лицом. Для этого проверяется величина

Клиентское приложение для захвата фотографии с веб-камеры предполагает наличие подсистемы выдачи инструкций пользователю по созданию «удачного» фото, таких как отметка специальной области размещения лица в кадре для съемки и рекомендаций принять нейтральное выражение лица. Тем не менее, пользователь не всегда может соблюдать рекомендации в полной мере, в связи с этим при отборе фотографий для распознавания допускаются небольшие изменения в выражении и положении лица, взгляда и прочего.

Проведение экспериментов. В качестве метрик расстояния были взяты следующие метрики: Эвклидово расстояние, квадрат Эвклидова расстояния, косинусное расстояние, метрика Манхэттена, расстояние Чебышева, расстояние Минковского. Используемые метрики расстояния приведены в таблице 1.

Таблица 1 – Метрики расстояния

Метрика	Формула	Метрика	Формула
Эвклидово расстояние	$d(x, y) = \sqrt{\sum_{i=1}^D (x_i - y_i)^2}$	Расстояние Манхэттена	$d(x, y) = \sum_{i=1}^D x_i - y_i $
Квадрат Эвклидова расстояния	$d(x, y) = \sum_{i=1}^D (x_i - y_i)^2$	Расстояние Чебышева	$d(x, y) = \max_{i=1,2,\dots,D} x_i - y_i $
Косинусное расстояние	$d(x, y) = 1 - \frac{\sum_{i=1}^D x_i y_i}{\sqrt{\sum_{i=1}^D (x_i)^2} \sqrt{\sum_{i=1}^D (y_i)^2}}$	Расстояние Минковского	$d(x, y) = \sqrt[p]{\sum_{i=1}^D (x_i - y_i)^p}$

По результатам исследований были построены графики. На рисунке 2 приведен график влияния числа главных компонент на точность распознавания при заданной метрике.

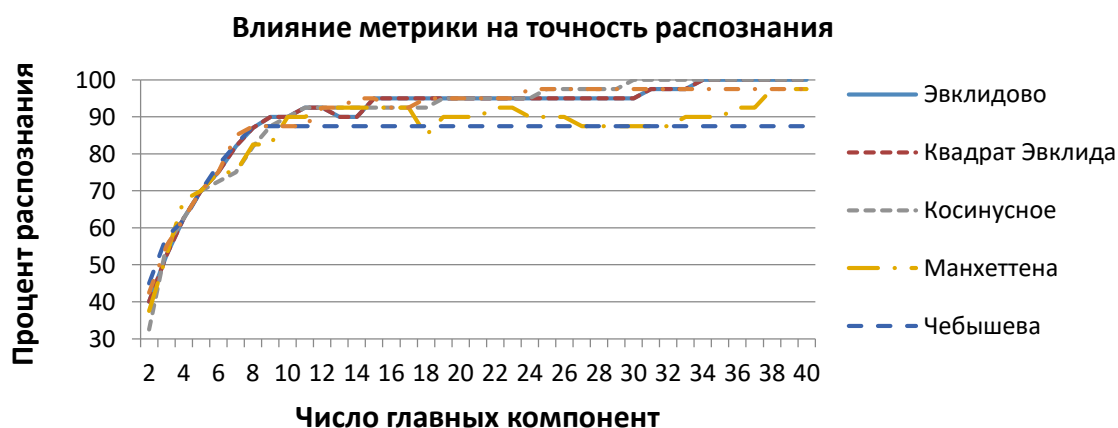


Рисунок 2. График влияния числа главных метрики на точность распознавания при заданной метрике

Как видно из графика, первые 3 метрики показали наилучшие результаты, для них уже при 30 главных компонентах достигнута точность распознавания более 95%. Наихудшие результаты у метрики Чебышева, при этом точность распознавания не привысила 87,5%.

Следующие эксперименты проведены с целью определения оптимального порогового значения расстояния для решения о принадлежности лица данному классу, а также выявления зависимости расстояний от числа главных компонент. Стоит отметить, что пороговое значение определяются опытным путем для каждой конкретной системы, но

некоторые зависимости все же можно выделить для опоры в дальнейших суждениях. На рисунке 3 приведены графики для различных метрик расстояния.

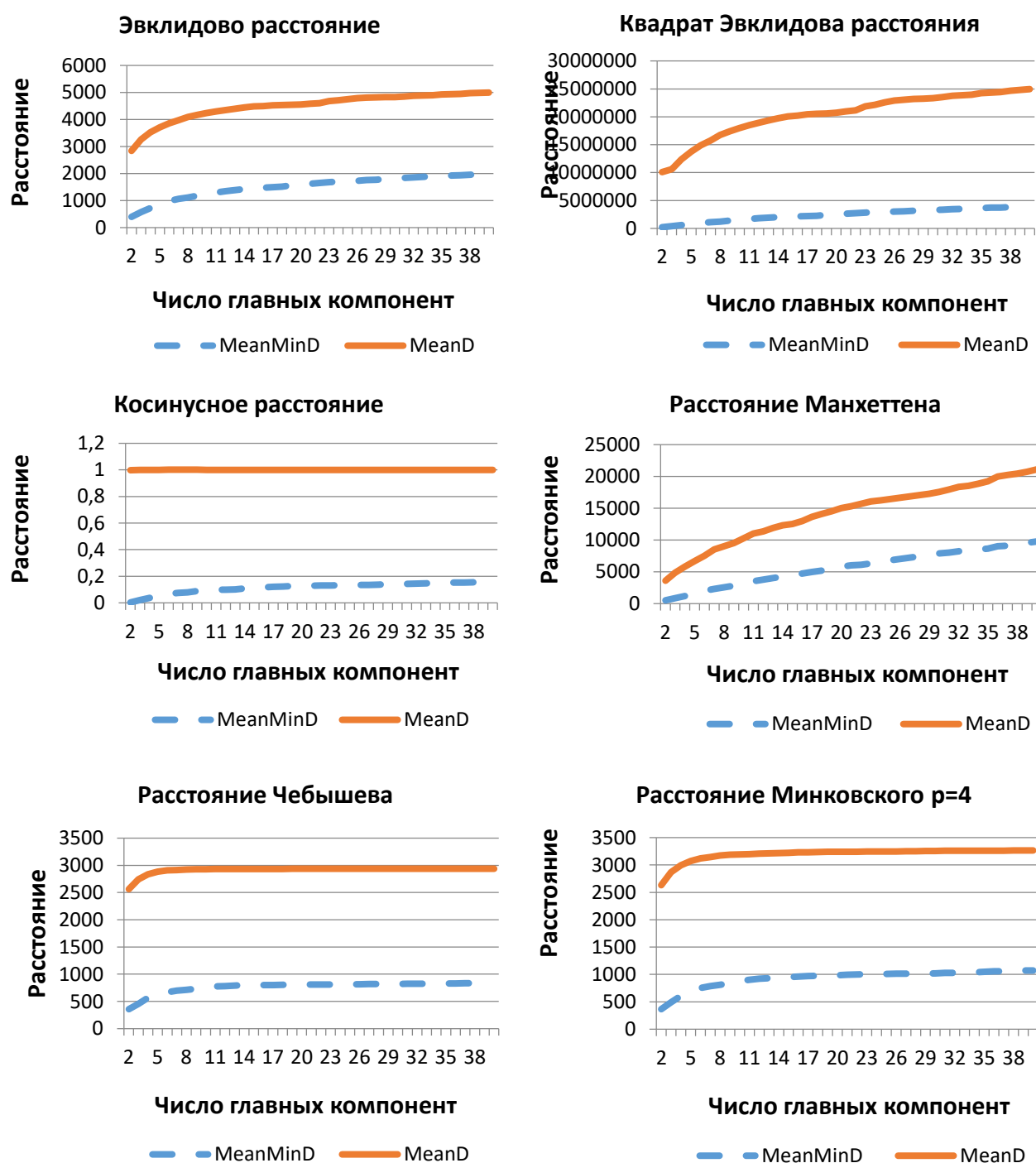


Рисунок 2. Зависимость расстояний от числа главных компонент для различных метрик

Как видно по графику среднее минимальное расстояние для найденных в базе данных устанавливается на уровне 2000 для 36 метрик, среднее же значение расстояния для других классов установилось на уровне 5000. Можно сказать что пороговое значение для верной классификации должно быть в пределах от 2000 до 5000, рекомендуемый уровень не выше 3000.

Благодаря возведению в квадрат при расчёте лучше учитываются большие разности.

Для квадрата Эвклидова расстояния среднее минимальное расстояние для найденных в базе данных устанавливается на уровне 380000 для 36 метрик, среднее же значение расстояния для других классов установилось на уровне 2225000. Можно сказать что пороговое значение для верной классификации должно быть в этих пределах, рекомендуемый уровень не выше 1000000.

Косинусное расстояние — это мера расстояния между двумя векторами предгильбертового пространства, которая используется для измерения косинуса угла между ними. Одна из причин популярности косинусного расстояния состоит в том, что оно эффективно в качестве оценочной меры, особенно для разреженных векторов, так как необходимо учитывать только ненулевые измерения. Можно заметить, что среднее значение расстояния для других классов практически не зависит от числа главных компонент и равно 1.

Среднее минимальное расстояние установилось на уровне 0,18 для 36 компонент необходимых для высокой точности распознавания. Оптимальным значением порога можно принять 0,4-0,6.

Манхэттенское расстояние зависит от вращения системы координат, но не зависит от отражения относительно оси координат или переноса.

Как видно по графику, среднее минимальное расстояние для найденных в базе данных равно 9300 для 37 метрик, обеспечивая 97,5% точность распознавания, среднее же значение расстояния для других классов для данной точности равно 20400. Можно сказать что пороговое значение для верной классификации должно быть в пределах от 10000 до 15000, рекомендуемый уровень не выше 13000. Стоит отметить, что при повышении числа компонент, что возможно при большей базе данных, средние значения продолжают расти с увеличением числа главных компонент.

Расстоянием Чебышёва между n -мерными числовыми векторами называется максимум модуля разности компонент этих векторов.

Данная метрика дает 87,5% точность при 7 компонентах, но превышая значение 10 точность распознавания прекращает расти, а среднее значение устанавливается на 2937, среднее минимальное практически прекращает расти, достигнув отметки 800 для 15 компонент. Рекомендуемый уровень порогового значения в районе 1500.

Расстояние Минковского (метрика Минковского) — параметрическая метрика на евклидовом пространстве, которую можно рассматривать как обобщение евклидова расстояния и расстояния городских кварталов.

Средние значения для данной метрики прекращают активный рост при 15 и более компонентах установившись на значениях 1000 и 3200 соответственно.

Можно сказать что пороговое значение для верной классификации должно быть в пределах от 1600 до 2000.

Итоговое сравнение результатов для разных метрик приведено в таблице 2.

Таблица 2 – Итоговое сравнение результатов для разных метрик

Метрика	Число главных компонент	Среднее минимальное расстояние	Среднее расстояние	Пороговое значение	Достигнутая точность
Эвклидово расстояние	34	1894	4885	3000	100%
Квадрат Эвклидова расстояния	34	3587236	23961025	1000000	100%
Косинусное расстояние	30	0,141	1	0,5	100%

Метрика	Число главных компонент	Среднее минимальное расстояние	Среднее расстояние	Пороговое значение	Достигнутая точность
Расстояние Манхэттена	38	9363	20420	12000	97,5%
Расстояние Чебышева	8	712	2924	1500	87,5%
Расстояние Минковского	24	1005	3247	1850	97,5%

В рамках данной статьи приведен краткий обзор метода главных компонент, проведены экспериментальные исследования, а именно проведен сравнительный анализ точности распознавания при разном числе главных компонент и различных метриках расстояния. Выяснено, что применение Евклидова, квадрата Евклидова и косинусного расстояния показывают наилучшие результаты (100% точность при 34, 34 и 30 главных компонент соответственно на тестовом наборе данных).

Также были определены количественные характеристики значений расстояния при применении различных метрик, с целью выявления оптимального порога верной классификации для каждой метрики с достижением необходимой точности распознавания.

В заключение можно сделать вывод, что метрика косинусного расстояния является наиболее оптимальным выбором, так как показывает максимальную точность при меньшем числе главных компонент на фоне остальных. Также она показывает максимальную разницу в соотношении среднего минимального расстояния к общему минимальному расстоянию, что подтверждает ее эффективность при поиске изображения в базе данных с минимальным расстоянием от входного (неизвестного) изображения.

Библиографический список:

1. Ситков Д.А., Шевченко В.И. Обзор методов распознавания лиц на изображении // Сборник статей всероссийской студенческой НТК "Мир компьютерных технологий". – Севастополь: Федеральное государственное автономное образовательное учреждение высшего образования "Севастопольский государственный университет". 2019. С. 118–121.
2. Р. Гонсалес, Р. Вудс. Цифровая обработка изображений. – Москва: Техносфера, 2005. – 1072с.
3. Мищенко Е.С. Сравнительный анализ алгоритмов распознавания лиц // Вестник Волгоградского государственного университета. Серия 9: Исследования молодых ученых. 2013. №11. С. 74–77.
4. Шерстобитов А.И., Федосов В.П., Приходченко В.А., Тимофеев М.В. распознавание лиц на групповых фотографиях с использованием алгоритмов сегментации // Известия ЮФУ. Технические науки. 2013.
5. Мокеев, В.В. О повышении эффективности вычислений главных компонент в задачах анализа изображений / В.В. Мокеев // Цифровая обработка сигналов. –2011. – № 4. – С. 29–36.
6. Мокеев, В.В. О задаче нахождения собственных значений и векторов больших матричных систем / В.В. Мокеев // Журнал Вычислительной Математики и Математической Физики. – 1992. – 32 (10). – С. 1652–1657.

УДК 004.896

В.А. Сколота, аспирант

И.А. Белова, аспирант

М.В. Мартинович, канд. техн. наук, доцент

Г.С. Зиновьев, докт. техн. наук, профессор

Новосибирский государственный технический университет

ОБЗОР ТЕХНИЧЕСКИХ СРЕДСТВ РЕАЛИЗАЦИИ НЕЙРО-ФАЗЗИ-АЛГОРИТМОВ ПОЛУЧЕНИЯ ПОКАЗАТЕЛЕЙ КАЧЕСТВА ЭЛЕКТРИЧЕСКОЙ ЭНЕРГИИ

Аннотация: Измерение показателей качества электрической энергии это очень емкий в плане вычислений процесс, осложняемый ситуацией наложения помех друг на друга и необходимость разделять помехи перед измерением конкретных показателей качества электрической энергии. Здесь в качестве одного из способов обработки информации можно использовать нейронные сети и фаззи-логику. Это может позволить избавиться от сложных математических вычислений на этапе обработки сигнала, проведя их при обучении сети. В таком случае встает вопрос об аппаратной реализации нейронной сети и фаззи-логики.

Ключевые слова: нейронные сети, микроконтроллеры, ПЛИС, аппаратная реализация, нечеткая логика

Abstract: Measurement of indicators of the quality of electric energy is a very computationally intensive process, complicated by the situation of interference with each other and the need to separate the interference before measuring specific indicators of the quality of electric energy. Here, as one of the methods of processing information, you can use neural networks and fuzzy logic. This can make it possible to get rid of complex mathematical calculations at the signal processing stage, having performed them during the training of the network. In this case, the question arises of the hardware implementation of the neural network and fuzzy logic.

Key words: neural networks, microcontrollers, FPGAs, hardware implementation, fuzzy logic

ВВЕДЕНИЕ

Искусственные нейроны реализуются как программно, так и аппаратно. Большей частью искусственные нейронные сети реализуются программным способом. Такая реализация позволяет сосредоточиться на применении нейронной сети, но по скорости вычислений она проигрывает аппаратной реализации искусственных нейронных сетей. Аппаратная реализация бывает цифровой и аналоговой. При аналоговой реализации можно воспользоваться нелинейными свойствами КМОП-устройств, но характеристики сети будут довольно сильно изменяться с изменением температуры, а результаты будут не точными. С увеличением доступности субмикронных процессов многие исследователи обращаются к аналоговой конструкции нейронных сетей, как способу реализации функции обработки сигналов меньшим числом активных приборов.

КРАТКИЙ ОБЗОР ИСТОЧНИКОВ

Для начала требуется определиться с основными требованиями к аппаратной реализации нейронной сети. При аппаратной реализации самыми важными параметрами являются время и точность вычислений. Также важными являются размеры аппаратной реализации. Точность вычислений и размеры аппаратной реализации могут оказаться конфликтующими параметрами. Так, в случае аппаратной реализации вычисление идет параллельно во всей схеме. При программной реализации нейроны последовательно рассчитываются определенной реализацией количеством вычислителей. Одно из возможных применений нейронной сети для диагностики и улучшения качества напряжения приведено в [16]. Там предлагается применять нейронную сеть в качестве ПИ-звена. Предполагается, что нейронная сеть будет быстрее реагировать и, соответственно, лучше компенсировать искажения сети. Использование искусственного интеллекта для решения проблемы качества

электроэнергии с использованием параллельного фильтра активной мощности описано в статье [20]. Подробный обзор интеллектуальной обработки сигналов и классификации событий представлен в статье [19].

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

Программная реализация нейронной сети - это реализация на микроконтроллерах общего назначения или на компьютере (суперкомпьютере). В данном случае нейронная сеть пишется в виде программного кода или создается с помощью программных средств, способных генерировать код. Преимущество данного способа: удобство использования, широкая доступность аппаратных средств. Предельная сложность реализуемой сети зависит от микроконтроллера, на котором происходит данная реализация, и упирается в большей степени в память, необходимую для хранения весовых коэффициентов и скорость работы нейронной сети. Скорость и является недостатком, в этом плане программная реализация уступает аппаратной.

Сейчас крупнейшими производителями микроконтроллеров являются фирмы STMicroelectronics и Atmel. Обе компании имеют очень широкий диапазон решений для разных задач. Среди высокопроизводительных микроконтроллеров имеются решения выполненные на ядре ARM Cortex M7. Это микроконтроллеры с ядром, имеющим 200-300 МГц рабочей частоты и 1-2 МБ флеш-памяти. Также контроллеры имеют до 24 АЦП, что может быть удобно при встраивании их в аналоговую нейронную сеть. Имеются 2 ЦАП для выходного сигнала рассчитываемой сети. При этом все микроконтроллеры имеют доступную цену, до нескольких десятков долларов.

ЦИФРОВАЯ РЕАЛИЗАЦИЯ

ИНС могут быть реализованы с использованием цифровых схем. Широко распространен способ реализации нейронной сети на ПЛИС, известна также и реализация нейронной сети на ПЛИС совместно с сигнальным процессором [7]. Также известна реализация самоорганизующейся нейронной сети с цифровыми ФАПЧ на ПЛИС [8]. Цифровые реализации специализированных схем являются достаточно редкими, причина этого в том, что благодаря постоянной миниатюризации компонентов развитие ПЛИС и микроконтроллеров позволяет реализовывать достаточно крупные нейронные сети [11]. Цифровые схемы опираются на существующие логические элементы и могут в полной мере воспользоваться достижениями в области цифровых схем [9]. Еще одним преимуществом такой реализации является возможность интеграции наряду со стандартными схемами. Такие возможности были продемонстрированы, например, чипом Synapse от IBM [10], или процессором Zeroth от Qualcomm [12].

Свою аппаратную реализацию под названием Nirvana разрабатывает и Intel [5], в компании IBM была разработана архитектура сверточных нейросетей, способная работать на нейроморфных чипах [6].

АНАЛОГОВАЯ РЕАЛИЗАЦИЯ

Существует множество аналоговых решений. Главной трудностью разработки аналоговых решений является то, что прототипирование возможно только на дискретных компонентах, а массовое производство на дискретных компонентах или созданием интегральных схем специального назначения. Программируемые аналоговые интегральные схемы (ПАИС), в отличие от своих цифровых аналогов ПЛИС, не получили широкого

распространения, и на текущий момент существует лишь очень ограниченные версии ПАИС. Аналоговые решения можно разделить на импульсные и непрерывные решения. Схема [1] относится к концу 60-х и является схемой нейрона, включающей в себя операционный усилитель. Схема состоит из операционных усилителей, резисторов, диодов и полевых транзисторов.

Аналоговые реализации представлены в статье [18] и представляют из себя разновидности транзисторных реализаций на МОП-технологии, но с различной спецификой: на транзисторах с плавающим затвором, на инжекции горячих электронов, с применением операционных усилителей, на токовых зеркалах и преобразователях напряжения.

Схема [2] представляет из себя схему нейрона, в которой разделены тело нейрона и синапс. Схема довольно компактная, так синапс состоит всего из 3 транзисторов, а тело нейрона из 5. Синапс имеет управление весом входа, а также управлением знаком синапса - тормозящий или ускоряющий синапс. Сигналы от синапсов суммируются на 2 линиях, соединяющих нейроны между собой и с телом нейрона. Вес w сигнала определяется уровнем напряжения на подложке транзистора, что не дает оптимальный результат.

Развитием данной схемы стали аналоговые схемы [14] и [15]. В первом случае весовой коэффициент задается резистивным делителем, во втором – напряжением на затворе транзистора.

Также разработки ведутся и в настоящее время. Так Японские ученые разрабатывают магнитные устройства, способные выполнять роль синапсов нейронов [3], а команда исследователей из Университета Стоуни Брук и Университета Калифорнии занимается разработкой нейрочипа, в котором роль синапсов выполняют пары мемристоров [4].

НЕЧЕТКАЯ ЛОГИКА

Распространенным подходом при создании нечеткой логики является использование микроконтроллеров. Для реализации нечеткого алгоритма существует два пути: аппаратная и программно-аппаратная реализация нечеткого контроллера

Аппаратная реализация возможна как:

1. Аппаратная реализация на спроектированном под конкретный алгоритм промышленном образце.
2. Микроконтроллеры специального назначения (Fujitsu F2RU-8, TogaiInfraLogic VY86C570, Siemens SAE 81C99).
3. Микроконтроллеры общего назначения с аппаратной поддержкой нечеткой логики (Motorolla 68HC12, STMicroelectronics ST52 Dualogic).

Аппаратная реализация обладает такими преимуществами, как скорость и удобство работы, однако не всегда может быть доступна в силу малой распространенности и высокой цены.

Программная реализация - это реализация на микроконтроллерах общего назначения, с использованием специальных "нечетких" библиотек, конструкторов нечетких систем, или программных средств, способных генерировать код с нечеткой системой под любой микроконтроллер.

Программная реализация проста в использовании, легкодоступна, гибка. Однако она имеет меньшую скорость работы и большую погрешность вычислений [13].

Некоторые способы применения нечеткой логики описаны в [17]: автоматическое обнаружение и классификация нарушения качества электроэнергии, конструкция контроллера для компенсации, определение характеристик нарушения качества электроэнергии, определение местоположения источника нарушения качества электроэнергии, планирование и контроль реактивной мощности, анализ устойчивости энергосистем, оценка нарушения качества электроэнергии, выявление неисправного оборудования, оценка стоимости простоев, связанных с нарушениями качества электроэнергии.

ВЫВОДЫ

Самым доступным средством реализации нейро-фаззи-алгоритмов является программная реализация. Аппаратно она может представлять из себя микроконтроллер, выполняющий программу с нейронной сетью. Данный способ прекрасно подходит как для прототипирования, так и для производства. Обеспечивается это доступностью аппаратных средств и их ценой. Как недостаток можно отметить скорость работы сети, реализованной таким образом. Еще одним хорошим инструментом прототипирования является цифровая реализация на ПЛИС. Преимущество здесь простота построения сети и параллельность ее работы. Недостаток это дороговизна производства и ограниченность размеров реализуемой сети. Для массового производства хорошим решением может стать аналоговая реализация. Ее преимущество это компактность и скорость работы, благодаря чему с ее помощью можно реализовывать схемы высокой сложности. Однако возможности прототипирования ограничены дискретными компонентами, а массовое производство - интегральными схемами специального назначения.

Существуют разработки западных компаний, реализующие в себе нейрочип, однако они не доступны на рынке, и производиться только по заказу, соответственно цена таких изделий делает их недоступными.

Библиографический список:

1. L. Wennik, "Electrical neuron circuit that includes an operational amplifier", Aug. 23 1966, US Patent 3,476,954.
2. K. Bernstein, N. Rohrer, "Neuron Circuit", Nov. 14 2002, US Patent 0167332.
3. William A. Borders, Hisanao Akima, Shunsuke Fukami, Analogue spin-orbit torque device for artificial-neural-network-based associative memory operation, Applied Physics Express 10 2017.
4. M. Prezioso, F. Merrih-Bayat, B. D. Hoskins, Training and operation of an integrated neuromorphic network based on metal-oxide memristors, Nature 521, 61–64 07 May 2015
5. <https://newsroom.intel.com/news-releases/intel-ai-day-news-release/>
6. Steven K. Essera, Paul A. Merollaa, John V. Arthura, Convolutional networks for fast, energy-efficient neuromorphic computing, PNAS, vol. 113 no. 41
7. S. Jung and S. S. Kim, "Hardware implementation of a real-time neural network controller with a dsp and an fpga for nonlinear systems," Industrial Electronics, IEEE Transactions on, vol. 54, no. 1, pp. 265–271, 2007.
8. H. Hikawa, "FPGA implementation of self organizing map with digital phase locked loops," Neural Networks, vol. 18, no. 56, pp. 514 – 522, 2005, IJCNNg 2005. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0893608005001103>
9. P. Moerland and E. Fiesler, "Neural network adaptations to hardware implementations," Handbook of Neural Computation, vol. 1, p. 2, 1997.
10. P. A. Merolla, "A million spiking-neuron integrated circuit with a scalable communication network and interface," Science, vol. 345, no. 6197, pp. 668–673, 014. [Online]. Available: <http://www.sciencemag.org/content/345/6197/668.abstract>
11. M. T. Tommiska, "Efficient digital implementation of the sigmoid function for reprogrammable logic," IEEE Proceedings, Computers and Digital Techniques, vol. 150, no. 6, pp. 403–411, 2003.
12. V. Aparin and J. A. Levin, "Methods and systems for cmos implementation of neuron synapse," Apr. 8 2014, US Patent 8,694,452.
13. D.E. Bekasov, "Software and hardware implementation of fuzzy microcontroller", Youth Science and Technology messenger, may 2013 (in Russian).

14. В.А. Сколота, Аппаратная реализация искусственного нейрона – Патент на полезную модель №170159.
15. В.А. Сколота, И.А. Белова, М.В. Мартинович, Схема нейрона - патент на изобретение №2678153.
16. Ravi Waswani, Ketan Fegade, Abhijeet Singh, Rajankumar Patel, Power Quality Improvement by Artificial Neural Network Using Dstatcom in Electric Ship, 2017 International Conference on Innovation in Information, Embedded and Communication Systems.
17. Padmanabh Thakur, Asheesh K Singh, Signal Processing and AI Based Diagnosis of Power Quality Disturbances: A Review
18. Vladimir A. Skolota, Irina A. Belova, Miroslav V. Martinovich, Overview of Technical Means of Implementation of Neuro-Fuzzy-Algorithms for Obtaining Quality Factor of Electric Power, 18th INTERNATIONAL CONFERENCE ON MICRO/NANOTECHNOLOGIES AND ELECTRON DEVICES EDM 2017, pp 469-472.
19. Om Prakash Mahela, Abdul Gafoor Shaik, Neeraj Gupta, A critical review of detection and classification of power quality events, Renewable and Sustainable Energy Reviews 41 (2015) , pp 495–505.
20. Aliyu Sabo, Noor Izzri Abdul Wahab, Mohd Amran Mohd Radzi, Nashiren Farzilah Mailah, A Modified Artificial Neural Network (ANN) Algorithm to Control Shunt Active Power Filter (SAPF) for Current Harmonics Reduction, 2013 IEEE Conference on Clean Energy and Technology (CEAT), pp 348-352

УДК 004.9:517.97

Ю.Е. Шишкин, аспирант,

А.В. Скатков, докт. техн. наук, профессор

Севастопольский государственный университет

ИНТЕЛЛЕКТУАЛЬНАЯ ТЕХНОЛОГИЯ ОБНАРУЖЕНИЯ АНОМАЛИЙ ПАРАМЕТРОВ СЛОЖНЫХ ОБЪЕКТОВ КАК ЗАДАЧА КЛАССИФИКАЦИИ НА ОСНОВЕ ЭКСПЕРИМЕНТАЛЬНОЙ И ЭКСПЕРТНОЙ ИНФОРМАЦИИ

Аннотация: рассматриваются интеллектуальные информационные технологии выявления аномалий, возникающих в данных мониторинга сложных объектов и процессов, как задачи классификации в условиях неточной, неполной и противоречивой априорной информации о параметрах этих объектов и процессов. Предложена математическая модель системы классификации, использующая механизмы адаптации параметров решающих правил и функции расстояния кластеризуемых объектов в факторном пространстве, направленная на повышение качества кластеризации в случае, когда априорная информация о статистических законах распределения этих параметров отсутствует.

Ключевые слова: классификация, мониторинг, математическое моделирование, интеллектуальный анализ данных, выявление аномалий, критические процессы

Annotation: intellectual information technologies for identifying anomalies arising in the monitoring data of complex objects and processes are considered as classification tasks in the context of inaccurate, incomplete and contradictory a priori information about the parameters of these objects and processes. A mathematical model of the classification system is proposed, using the mechanisms for adapting the parameters of the decision rules and the distance function of clustered objects in the factor space, aimed at improving the quality of clustering when there is no a priori information about the statistical laws of the distribution of these parameters.

Key words: classification, monitoring, mathematical modeling, data mining, anomaly detection, critical processes

На сегодняшний день формальные методы, используемые для решения задач классификации, широко и успешно применяются для решения задач поддержки принятия решений по выявлению аномальных состояний и управлению объектами критических инфраструктур, в условиях, когда система достаточно хорошо определена [1–4]. В таком случае задача классификации может быть сформулирована как поиск наилучшего отображения множества значений параметров объекта на какое-либо подмножество классов, где под классами подразумеваются возможные варианты управления этими объектами [5,6]. При этом в меньшей степени изучен вопрос, когда осуществляется решение задачи классификации в условиях неопределенности, где неопределенность обусловлена использованием в ходе обучения системы распознавания нечеткой, неточной, неполной и противоречивой априорной информации о параметрах, характеризующих состояние наблюдаемых объектов и сред, и в частности предполагается отсутствие априорной информации о статистических законах распределения этих параметров [7–9]. Удовлетворение требований возможности работы в условиях нечетких, неполных и противоречивых сведений включает в себя необходимость обнаруживать противоречия между вновь поступившими данными непрерывного мониторинга с уже имеющимся сведениями, определять несоответствия и принимать решения о необходимости переобучения системы [10].

Задача интеллектуального выявления аномалий рассматривается как частный случай задачи классификации и задается в виде поиска такой обучающейся модели $v^*(I)$ (1) из семейства параметрических функций V , которая ставит в соответствие произвольному $\alpha \in A$ один из N кластеров так, чтобы объекты внутри одного кластера были схожи, в соответствии с используемой метрикой расстояний, а метрики объектов из различных кластеров различались, известен широкий класс прикладных задач, допускающих такую постановку [11–13]:

$$\begin{cases} V = \{v(\alpha, I): \alpha \times I \rightarrow N\} \\ v(\alpha, I): R(v(\alpha, D), D) \rightarrow \min, \\ \alpha \in A; v \in V \end{cases} \quad (1)$$

где D – априорно заданные векторы свойств обучающего множества объектов, I – параметры классифицируемого объекта, $v(\alpha, I)$ – решающее правило ставящее в соответствие каждому объекту I соответствующий ему класс N , α – параметры функции отображения, определяющие структуру решающих правил, R – функционал потерь, численно определяющий степень невязки модели V с априорно заданным обучающим множеством D .

В детерминированном случае каждый объект описывается в виде K -мерного вектора параметров, достоверно задающего его состояние, и класса N , которому он соответствует в модели обучения:

$$I = (X, D) = \{I_k, k \in K\}, \quad (2)$$

где $x_i \in X$ – вектор параметров i -го объекта, $d_i \in D$ – класс, соответствующий i -му объекту. Данная постановка отличается от стандартной тем, что для некоторых пар x_i, d_i – значения могут быть пропущены или заданы неточно в виде наборов $x_i, d \in \{R_1 \dots R_m\}, r \in D$. Для задачи самообучения допускается вырожденный случай постановки модели в виде (2) где D частично или полностью не задано.

В интеллектуальной системе мониторинга при обучении будем считать, что в качестве априорной информации используются результаты инструментальных наблюдений численных параметров интересующих исследователя объектов, в качестве дополнительного источника используются формально заданные суждения эксперта относительно классов некоторых объектов, т.е. система компенсирует недостаток достоверной априорной информации за счет диалогового взаимодействия с экспертом. При этом возникает дополнительная задача оценки согласованности каждого нового поступающего элемента априорной информации с имеющимся набором данных базы знаний, на основе оценки которых лицо, принимающее решение может целенаправленно уточнять отдельные элементы априорного описания объектов базы знаний, что в свою очередь улучшает качество работы системы распознавания в целом. В совокупности это обеспечивает эффективность работы такого рода системы за счет использования методов, объединяющих как стандартную процедуру обучения на основе полученных экспертных оценок так самообучения и адаптации. Обеспечение интеллектуального подхода выявления аномалий данных мониторинга сложных систем производится на основе актуализации базы знаний специализированной математической модели в соответствии с вновь поступающими данными и применения механизмов адаптации системы путём итеративного изменения объёма априорной информации о наборах значений параметров наблюдаемых объектов и соответствующих им состояний для заданной выборки, что позволяет согласовывать противоречивые априорные данные при заданном допустимом уровне изменений. В рамках подхода адаптации используются наборы решающих правил, обеспечивающих построение оценок согласованности каждого поступающего элемента априорной информации в соответствии с имеющимися данными. Процесс обучения включает в себя процедуру взаимодействия эксперта с интеллектуальной системой, что обеспечивает интерактивную актуализацию поведения системы, осуществляющей классификацию, и позволяет адаптировать решающие правила изменяющимся внешним условиям.

Применение предложенной интеллектуальной технологии выявления аномалий в данных мониторинга сложных объектов и сред критического назначения, включающего этапы обучения информационной системы на основе экспериментальной и экспертной информации, дает возможность развить качественные представления о динамике аварийных

процессов и позволяет разработать новые подходы выявления аномальных состояний, обладающих большей достоверностью в условиях неопределенности.

Исследование выполнено при финансовой поддержке РФФИ и города Севастополь в рамках научных проектов № 18-47-920005\18, № 18-47-920007\18.

Библиографический список:

1. Остапенко А.Г., Ермилов Е.В., Шершень А.Н. Предупреждение и минимизация последствий компьютерных атак на элементы критической информационной инфраструктуры и автоматизированные информационные системы критически важных объектов: риск-анализ и оценка эффективности защиты // Информация и безопасность. 2013. Т. 16. № 2. С. 167–178.
2. Шайтура С.В. Интеллектуальный анализ геоданных // Перспективы науки и образования. 2015. № 6 (18). С. 24–30.
3. Скатков А.В., Брюховецкий А.А., Шишкин Ю.Е. Разработка интеллектуальной технологии обнаружения аномалий экосистем акватории г. Севастополя // Системы контроля окружающей среды. 2019. № 1 (35). С. 27–34. DOI: 10.33075/2220-5861-2019-1-27-34
4. Брюховецкий А.А., Скатков А.В., Шишкин Ю.Е. Моделирование процессов обнаружения аномалий в сложноструктурированных данных мониторинга // Системы контроля окружающей среды. 2017. № 9 (29). С. 45–49.
5. Соколов А.В. Метод критических технологий // Форсайт. 2007. Т. 1. № 4. С. 64–74.
6. Кувшинов Б.М. Адаптивная модель распознавания образов для решения задач классификации в условиях неопределенности: автореф. дис. канд. техн. наук. Южно-уральский. гос. университет. Челябинск. 2004. 16 с.
7. Логинов Е.Л. Модели и алгоритмы решения задачи структурно-параметрического синтеза информационных процессов в системах управления объектов критической инфраструктуры. Монография. Краснодар. 2010. 191 с.
8. Махутов Н.А., Резников Д.О., Петров В.П. Особенности обеспечения безопасности критических инфраструктур // Безопасность в техносфере. 2014. Т. 3. № 1. С. 3–14.
9. Shishkin Y.E. Big Data visualization in decision making // Science in Progress: тез. Всерос. науч.-практ. конф. магистрантов и аспирантов. Новосибирск 20 октября 2016 г. Новосибирск: НГТУ, 2016. С. 203-205. ISBN 978-5-7782-3094-1.
10. Полковникова Н.А., Курейчик В.М. Нейросетевые технологии, нечёткая кластеризация и генетические алгоритмы в экспертной системе // Известия ЮФУ. Технические науки. 2014. № 7 (156). С. 7–15.
11. Абрамов Н.С., Талалаев А.А., Фраленко В.П. Интеллектуальный анализ телеметрической информации для диагностики оборудования космического аппарата // Информационные технологии и вычислительные системы. 2016. № 1. С. 64–75.
12. Скатков А.В., Шишкин Ю.Е. Кластеризация данных в задачах обнаружения аномалий на основе ортогональных фильтров // Системы контроля окружающей среды. 2018. № 11 (31). С. 36–43.
13. Колесенков А.Н. Технология поддержки принятия управленческих решений на основе оперативного мониторинга пожарной обстановки // Известия Тульского государственного университета. Технические науки. 2015. № 9. С. 157–163.

СЕКЦИЯ 4

Методы и системы
защиты информации,
информационная и
сетевая безопасность



УДК 681.3

А.С. Бибик, студент 4-го курса

Е.О. Савкова, канд. техн. наук, доцент

Донецкий национальный технический университет

WEB-ОРИЕНТИРОВАННАЯ СИСТЕМА ПРОЦЕССИНГА КРИПТОВАЛЮТНЫХ ОПЕРАЦИЙ

Аннотация: В статье рассмотрен процессинг финансовых операций и предложена микросервисная архитектура системы автоматизации приема платежей в криптовалюте. Описаны основные компоненты системы, особенностью которых является работа с платформой базы блокчейна, распределённая архитектура, изолированная кодовая база, независимость развёртывания, репликация.

Ключевые слова: процессинг, криптовалюта, rpc, rest api, http, микросервисы

Annotation: The article describes the processing of financial transactions and proposed microservice architecture of the automation system for accepting payments in cryptocurrency. It describes the main components of the system, which feature is the work with the blockchain base platform, distributed architecture, isolated code base, deployment independence, replication.

Key words: processing, cryptocurrency, rpc, rest api, http, microservices

Безналичные платежи всюду окружают современного человека, совершает он покупку в магазине или оплачивает счёт за телефонную связь. В момент ожидания чека от POS-терминала, подтверждения оплаты, мы сталкиваемся с системами взаиморасчёта. В основе систем взаиморасчёта функционирует процессинг финансовых операций.

Процессинг — это деятельность по обработке информации, используемой при совершении платёжных операций. Такую деятельность осуществляют процессинговые центры (провайдеры платёжных сервисов)[1]. Наиболее широкое распространение процессинг-сервисы получили в следующих отраслях:

- процессинг электронных платежей;
- процессинг пластиковых карт.

В простейшем виде процессинг через банковский шлюз состоит из нескольких этапов:

1. магазин выставляет счёт покупателю;
2. покупатель оплачивает счёт;
3. банк подтверждает зачисление средств.

В последние годы с бумом блокчейн-технологий пришла и необходимость в разного рода процессинге криптовалют, будь то переводы или оплата товаров и услуг в сети.

Как правило, магазины для работы с безналичным расчётом не создают свои системы эквайринга и банковских операций, так как этот процесс очень сложен и требует дополнительных затрат. Вместо этого они пользуются услугами банка, который предоставляет всё, что необходимо и выгодно[2].

Также и владельцы интернет-магазинов, которые хотят принимать криптовалюту для расчётов, прибегают к использованию сервисов, представляющих данную услугу.

В свою очередь сервисы, производят консультацию и внедрение их систем в проекты партнёров. Простейшая схема их взаимодействия (рисунок 1) состоит из следующих этапов:

1. клиент магазина-партнёра оформляет заказ. Магазин-партнёр отправляет на сервис запрос с заявкой на создание платёжного обязательства, в котором указывает сумму и остальную необходимую информацию;
2. процессинговая система привязывает к платёжному обязательству кошелек из собственного пула свободных кошельков или создаёт в блокчейне новый, если пул пуст;
3. система отправляет адрес кошелька для оплаты магазину-партнёру, который предоставляет его своему клиенту;

4. система процессинга, проверяет кошелек на наличие средств, как только на счёт поступает установленная сумма, система оповещает магазин-партнёр о том, что средства поступили;
5. магазин подтверждает оплату заказа клиента;
6. партнёр может перевести средства со своего счёта в системе на свой персональный кошелек.

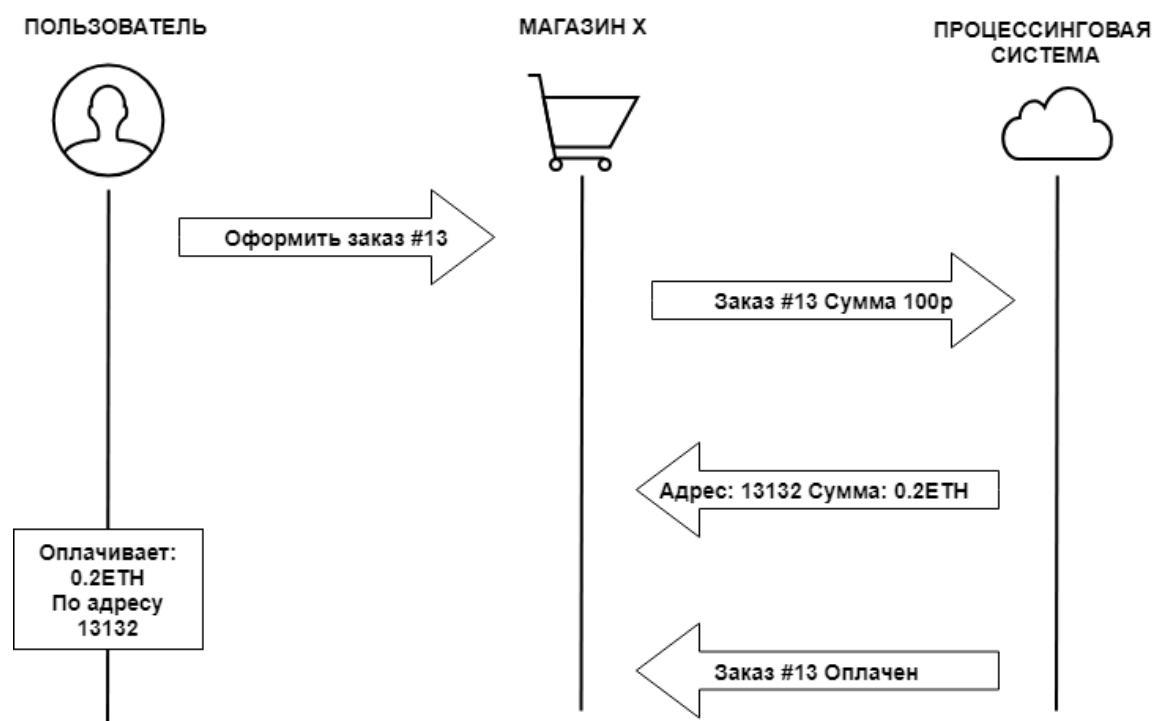


Рисунок 1 – Схема процессинга

В связи с необходимостью автоматизации приёма платежей в криптовалюте необходимо разработать систему процессинга криптовалютных операций. Лучшим выбором для данной системы будет микросервисная архитектура, по следующим характеристикам:

- репликация (дублирование) сервисов для отказоустойчивости;
- распределённая архитектура позволяет балансировать нагрузки между репликами и создавать новые, в случае необходимости (Например, нагрузка в праздничные дни);
- изоляция кодовой базы сервисов друг от друга, позволяет вносить изменения в один сервис в случае необходимости, не затрагивая остальные, что соответствует принципу единственной ответственности и упрощает командную разработку таким образом, что разработчики могут независимо работать над разными модулями системы[3];
- позволяет перезапускать сервисы независимо друг от друга, так же в случае отказа менее важных сервисов остальные не прекратят свою работу. Например, в случае отказа сервиса получения курса валют, из системы пропадёт лишь функционал выставления счёта с калькуляцией запрашиваемой суммы, таким образом будут приниматься запросы только в криптовалюте, в аварийный период, что не затронет функционал и работу остальных компонентов системы.

Архитектура системы состоит из следующих компонентов (рисунок 2):

- координатор (REST API) - источник пользовательского трафика, сервис, к которому непосредственно обращаются клиенты через интернет по протоколу HTTP/HTTPS, который координирует работу остальных сервисов. Отвечает за

учет пользователей, создание платёжных обязательств, создание заявок на переводы. Координатор общается с остальными сервисами по протоколу RPC (Remote Procedure Call);

- сервис проверки оплаты. Принимает адрес кошелька и ожидаемую сумму и обращается к блокчейну для проверки баланса кошелька;
- сервис получения курса валют. Получает тикеры валют (например, RUB/USD) и постоянно синхронизирует курс валют системы с биржевым курсом;
- сервис переводов. Получает заявки на переводы и занимается их обработкой;
- сервис создания кошельков занимается созданием кошельков в блокчейне.

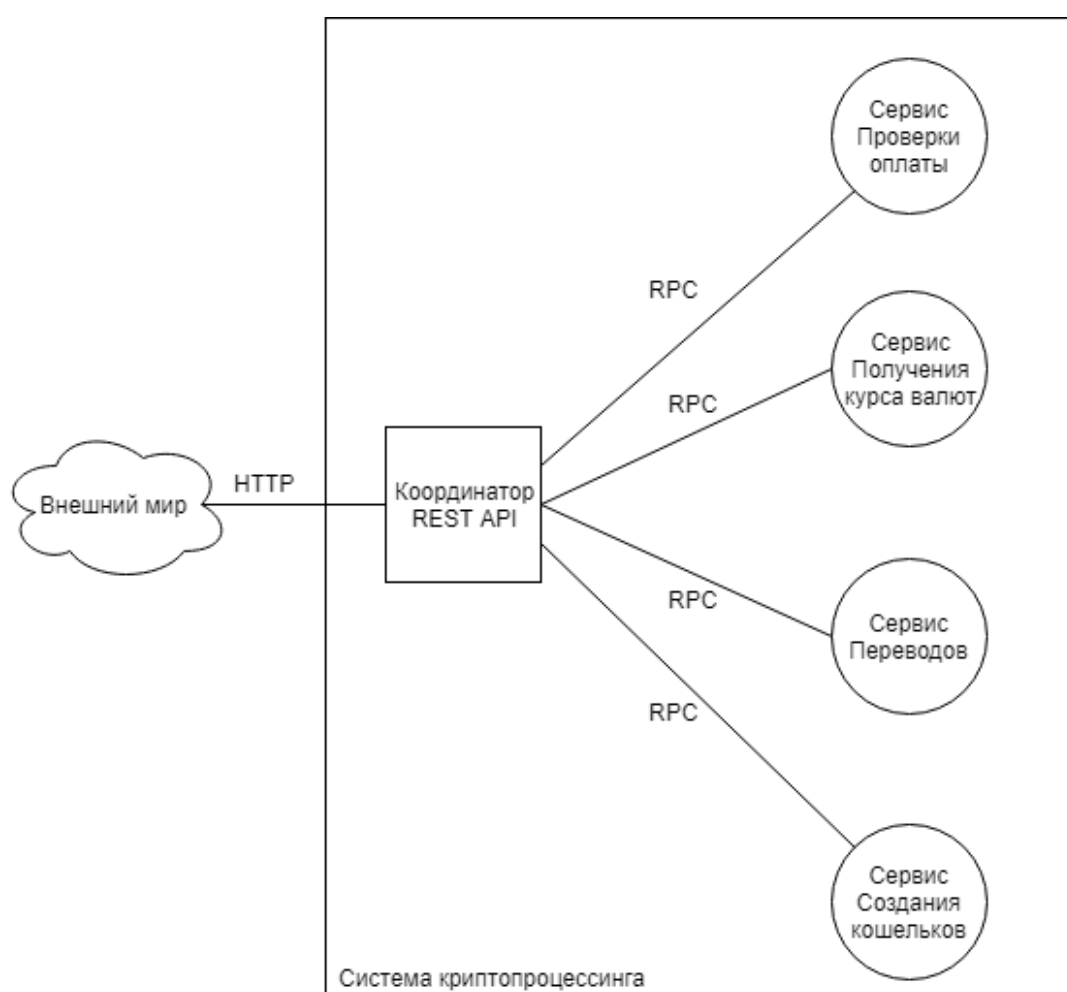


Рисунок 2 – Архитектура системы

Вывод

Выполнен анализ ситуации современных методов взиморасчёта, а также необходимости разработки систем процессинга криптовалютных операций. Была разработана предварительная архитектура системы с учётом современных тенденций, а также удобства микросервисной архитектуры.

Библиографический список:

1. Мисько О. Н. Процессинг // Большая российская энциклопедия: [в 35 т.] / Большая российская энциклопедия, 2017.
2. Пол Винья Майкл Кейси. Эпоха криптовалют // издательство МАНН, ИВАНОВ И ФЕРБЕР Максимально полезные книги. 2019
3. Сэм Ньюмен. Создание микросервисов // O'Really (Питер) 2016.

УДК 004.056

Н.Л. Корепанова, канд. техн. наук, доцент**М.А. Лебедева, старший преподаватель****Д.А. Ситников, магистрант***Севастопольский государственный университет***ПРИМЕНЕНИЕ ХЭШ-ФУНКЦИЙ ДЛЯ ЗАЩИТЫ КОМПЬЮТЕРНЫХ СЕТЕЙ**

Аннотация: в данной статье рассматривается актуальная проблема применения функций хэширования для обеспечения безопасности в компьютерных сетях. Авторы дают обобщенную характеристику существующих алгоритмов хэширования, формируют требования к ним. Описаны области защиты компьютерных сетей, в которых находят свое применение хэш-функции.

Ключевые слова: алгоритмы, хэш-функция, криптография, сеть, аутентификация.

Annotation: The relevant problem of using hashing functions which provide security in computer networks is discussed. The author gives a general description of existing hashing algorithms and forms the requirements for them. The research describes the area of computer networks protection, where hashing functions are used.

Key words: algorithms, hashing function, creptography, network, authentication.

Хэш-функция может выступать примером шифрования в целях аутентификации (установлении подлинности автора и документа), так же ее называют односторонней функции (one-way) или дайджест-функцией (digest function).

Функция хэширования (хэш-функция) представляет собой преобразование, на вход которого подается сообщение переменной длины M , а выходом является строка фиксированной длины $h(M)$. Иначе говоря, хэш-функция $h(M)$ принимает в качестве аргумента сообщение (документ) M произвольной длины и возвращает хэш-значение (хэш) $H=h(M)$ фиксированной длины (рисунок 1).

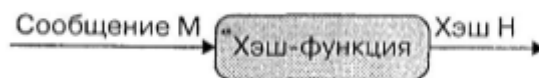


Рисунок 1 - Схема формирования хэша $H=h(M)$.

Хэш-значение $h(M)$ – это дайджест сообщения M , то есть сжатое двоичное представление основного сообщения M произвольной длины. Хэш-значение $h(M)$ формируется функцией хэширования.

Функция хэширования позволяет сжать подписываемый документ M до 128 бит и более (в частности, 128 или 256 бит), тогда как M может быть размером в мегабайт или более. Следует отметить, что значение хэш-функции $h(M)$ зависит сложным образом от документа M и не позволяет восстановить сам документ M .

Хэш-функция может применяться для проверки целостности данных. Хэш-функция, примененная к шифруемым данным, дает в результате значение (дайджест), состоящее из фиксированного небольшого числа байт. Дайджест передается вместе с исходным сообщением. Получатель сообщения, зная, какая хэш-функция была применена для получения дайджеста, заново вычисляет дайджест, используя незашифрованную часть сообщения. Если значение полученного и вычисленного дайджестов совпадают, значит, содержимое сообщения не было подвергнуто никаким изменениям. Значение дайджеста не дает возможности восстановить исходное сообщение, но зато позволяет проверить целостность данных.

Функция хэширования должна обладать следующими свойствами:

1. Хэш-функция может быть применена к аргументу любого размера.
2. Выходное значение хэш-функции имеет фиксированный размер.

3. Хэш-функцию $h(x)$ достаточно просто вычислить для любого x . Скорость вычисления хэш-функции должна быть такой, чтобы скорость выработки и проверки электронной цифровой подписи (ЭЦП) при использовании хэш-функции была значительно больше, чем при использовании самого сообщения.

4. Хэш-функция должна быть чувствительна к всевозможным изменениям в тексте M , таким как вставки, выбросы, перестановки и т.п.

5. Хэш-функция должна быть однонаправленной, то есть обладать свойством необратимости. Иными словами, задача подбора документа M' , который обладал бы требуемым значением хэш-функции, должна быть вычислительно неразрешима.

6. Вероятность того, что значения хэш-функций двух различных документов (вне зависимости от их длин) совпадут, должна быть ничтожно мала; то есть для любого фиксированного x с вычислительной точки зрения невозможно найти $x' \neq x$, такое что $h(x') = h(x)$.

Теоретически возможно, что два различных сообщения могут быть сжаты в одну и ту же свертку (так называемая коллизия, или столкновение). Поэтому для обеспечения стойкости функции хэширования необходимо предусмотреть способ избегать столкновений. Полностью столкновений избежать нельзя, поскольку в общем случае количество возможных сообщений превышает количество возможных выходных значений функции хэширования. Однако вероятность столкновения должна быть низкой.

Например, хэш-функция CRC-32, представляющая собой контрольную сумму, является линейным отображением и поэтому не удовлетворяет ни одному из этих трех свойств. Использование в качестве бесключевой хэш-функции, построенной на основе алгоритма блочного шифрования в режиме выработки имитовставки, также нецелесообразно, так как обратимость блочного шифрования позволяет подбирать входное сообщение для любого значения свертки при фиксированном и общеизвестном ключе. Для построения примера хэш-функции, удовлетворяющей свойству 1), рассмотрим функцию, заданную формулой $g_k(x) = E_k(x) \oplus x$, где E_k — алгоритм блочного шифрования. Такая функция является однонаправленной по обоим аргументам. Поэтому на ее основе можно построить хэш-функцию по правилу (1), определив одношаговую сжимающую функцию одной из следующих формул:

$$f(x, H) = E_H(x) \oplus x \text{ или } f(x, H) = E_x(H) \oplus H.$$

Первая из этих функций лежит в основе российского стандарта хэш-функции, а вторая в основе американского стандарта SHA.

Приведем без доказательства следующие утверждения:

- Если функция хэширования h построена на основе одношаговой – сжимающей функции f по правшущу (1), то из устойчивости к коллизиям функции f следует устойчивость к коллизиям функции h .

- Если хэш-функция устойчива к коллизиям, то она устойчива к нахождению второго прообраза.

- Устойчивая к коллизиям хэш-функция не обязана быть – однонаправленной.

- Пусть $h: X \rightarrow Y$ — хэш-функция и $|X| \rightarrow 2|Y|$. Тогда если существует эффективный алгоритм обращения функции h , то существует вероятностный алгоритм нахождения коллизии функции h с вероятностью успеха, большей $1/2$.

Заметим, что трудоемкость подбора прообраза для однонаправленной функции или трудоемкость поиска второго прообраза оцениваются величиной $O(2^n)$. В то же время трудоемкость поиска коллизии оценивается величиной $O(2^{n/2})$, так как в данной ситуации применима атака, основанная на парадоксе "дней рождений".

Рассмотрим конкретные примеры хэш-функций, построенных на основе некоторых алгоритмов преобразования блоков. Пусть E_k — алгоритм блочного шифрования, n —

размер блока, l — размер ключа и G — некоторое отображение, ставящее в соответствие вектору длины n вектор длины l . Рассмотрим следующие одношаговые сжимающие функции, построенные на основе алгоритма E_k :

- а) $f(x, H) = E_x(H) \oplus H$ (Дэвис — Мейер);
- б) $f(x, H) = E_G(H)(x) \oplus x$ (Матиас — Мейер — Осеас);
- в) $f(x, H) = E_G(H)(x) \oplus x \oplus H$ (Миагучи — Принель).

Значением любой из хэш-функций, построенных по правилу (1) из приведенных одношаговых сжимающих функций, является вектор длины n , равной размеру блока. В случае если эта величина оказывается недостаточной, ее можно увеличить, заменив одношаговую функцию f на функцию f с удвоенной размерностью значений. Это можно сделать, например, путем двукратного применения функции f с последующим перемешиванием полублоков согласно формуле:

$$f^*(x, H_1, H_2) = \pi(f(x, H_1), f(x, H_2)),$$

в которой π переставляет произвольные полублоки a, b, c, d по правилу $\pi((a, b), (c, d)) = (a, d, c, b)$. Такой подход реализован в конструкции одношаговой функции MDC-2. Другие примеры бесключевых хэш-функций дают известные алгоритмы MD-4, MD-5 и SHA. Они оперируют с блоками длины n , совпадающей с длиной результирующего значения свертки, причем $n = 128$ для алгоритма MD-4 и $n = 160$ для MD-5 и SHA. Указанные алгоритмы спроектированы специально с учетом эффективной реализации на 32-разрядных ЭВМ. При их использовании исходное сообщение M разбивается на блоки длиной $m = 512$ бит. Последний блок формируется путем дописывания к концу сообщения комбинации 10...0 до получения блока размера 448 бит, к которому затем добавляется комбинация из 64 бит, представляющая битовую длину сообщения. Затем вычисляется значение свертки с использованием одношаговой сжимающей функции, заданной формулой $f(x, H) = E_x(H) \oplus H$, где x — блок сообщения длины $m = 512$ бит, H — блок из n бит, а E_x — некоторое преобразование множества блоков. Значение начального вектора определяется в описании преобразования E_x .

В стандарте хэш-функции ГОСТ Р 34.11-94 приняты значения $n = m = 512$. Одношаговая сжимающая функция $f(x, H)$, используемая для вычисления последовательности значений $H_i = f(x_i, H_{i-1})$, построена на базе четырех параллельно работающих схем блочного шифрования (ГОСТ 28147-89), каждая из которых имеет 256-битовый ключ и оперирует с блоками размера 64 бита. Каждый из ключей вычисляется в соответствии с некоторой линейной функцией от блока исходного сообщения x_i и значения H_{i-1} . Значение H_i является линейной функцией от результата шифрования, блока исходного сообщения x_i и значения H_{i-1} . После вычисления значения H_N для последовательности блоков $M_1, M_2, \dots, M_N$ применяют еще два шага вычисления согласно формуле $H = h(M) = f(Z \oplus M_N, f(L, H_N))$, где Z — сумма по модулю два всех блоков сообщения, а L — длина сообщения.

Сделаем некоторые выводы, функция хэширования может быть использована для нахождения изменений в сообщениях, то есть с ее помощью можно формировать криптографическую контрольную сумму (также называемую кодом обнаружения изменений или кодом аутентификации сообщения). В данном случае хэш-функция применяется для контроля целостности сообщения, при формировании и проверке электронной цифровой подписи.

Хэш-функции используются также в целях аутентификации пользователей. В некоторых технологиях информационной безопасности применяется нетривиальный прием шифрования — шифрование с помощью односторонней хэш-функции. Особенность этого способа сводится к тому, что оно является односторонним, то есть не расшифровывается на

стороне приемника. Обе стороны (отправитель и получатель) используют одну и ту же процедуру одностороннего шифрования на основе хэш-функции.

Применение хэш-функций в криптографии:

Проверка целостности данных при их передаче или хранении

- с помощью ключевой хэш-функции

- с помощью бесключевой хэш-функции (аутентификация источника– данных)

2) Проверка парольной фразы. В большинстве случаев парольные фразы не хранятся на целевых объектах, хранятся лишь их хэш-значения.

3) Ускорение поиска данных. Для осуществления быстрого поиска нужного сообщения в большом списке сообщений различной длины удобнее сравнивать друг с другом не сами сообщения, а короткие значения их сверток, играющих одновременно роль контрольных сумм. Основным требованием к таким хэш-функциям является равномерность распределения их значений при случайном выборе значений аргументов.

Хэш-функции также используются в некоторых структурах данных — хэш-таблица и декартовых деревьях.

Требования к хэш-функции в этом случае другие:

- хорошая перемешиваемость данных;

- быстрый алгоритм вычисления.

Хэш-функции имеют также разнообразные применения при проведении статистических экспериментов, при тестировании логических устройств, при построении алгоритмов быстрого поиска и проверки целостности записей в базах данных.

Библиографический список:

1. Г.В.Ваныкина, Т.О. Сундукова. «Алгоритмы компьютерной обработки данных» (Изд-во Тул. гос. пед. ун-та им. Л.Н. Толстого, 2011. – 218 с.
2. А.П.Алферов, А.Ю.Зубов, А.С.Кузьмин. Основы криптографии.-М.: Гелиос АРВ 2002г. – 480с.
3. С.Я. Килин. Квантовая криптография: идеи и практика. Беларуская навука, 2007. – 391 с.

УДК 004.056

Е.А.Шумаева, канд. гос. упр., доцент

Г.С.Джура, студент

*Донецкий национальный технический университет***ИНФОРМАЦИОННОЕ ПРАВО КАК ИНСТРУМЕНТ ОБЕСПЕЧЕНИЯ КИБЕРБЕЗОПАСНОСТИ ГОСУДАРСТВА**

Аннотация: В статье исследованы проблемы обеспечения «цифровых прав» граждан с точки зрения государственного управления. Проанализированы основные тенденции развития цифровой отрасли в целом и киберпреступлений в частности.

Ключевые слова: кибербезопасность, информационное право, информационная безопасность.

Annotation: The article investigates the problems of ensuring the «digital rights» of citizens from the point of view of public administration. The main trends in the development of the digital industry in General and cybercrime in particular are analyzed.

Key words: cybersecurity, information law, information security.

Исследования последних лет показали, что основой современного мирового развития служат инновации, которые являются движущей силой общего роста современного государства. Процесс «цифровизации» сегодня продолжает бурно развиваться, стимулируя новые изменения и технологические новации, которые, в свою очередь, ставят непростые проблемы правового характера в цифровой экосистеме. На первый план выходит задача создания эффективной защиты прав граждан в киберпространстве и обеспечение сохранности государственной, служебной и иных видов тайн.

Важная роль в обеспечении информационной безопасности России отводится определению приоритетных направлений и механизмов реализации государственной политики Российской Федерации в сфере международной информационной безопасности в целях противодействия основным угрозам в этой области.

Государственные и финансовые организации развитых стран, учитывая потенциальную общественную опасность вовлечения различных структур в процессы использования кибероружия, ещё в конце прошлого века стали обмениваться необходимой информацией и вырабатывать организационно-правовые меры по предотвращению отрицательного влияния киберпреступлений, как на национальные экономики, так и на мировую хозяйственную систему в целом.

Стоит заметить, что в связи с быстро меняющейся мировой обстановкой, роль государства в безопасности страны становится размытой, так как на эти процессы начинают влиять большое количество факторов, которые государство не способно быстро отследить и изменить. Однако, внимание к рассматриваемому вопросу в развитых странах становится все более пристальным. Так, по недавним оценкам органов государственного контроля Великобритании затраты на борьбу с киберпреступностью обходятся стране ежегодно в сумме от 18 до 27 млрд. фунтов стерлингов. Такая ситуация в информационно-коммуникационной среде вынуждает правительства многих стран принимать активные контрмеры по защите государственных и частных интересов в киберпространстве, включая разработку нового законодательства в этой сфере [1].

Информационное право – это наука, являющаяся фундаментом для обеспечения информационной безопасности государства, т.к. именно информационное законодательство сегодня является основой для функционирования отрасли. Именно разработка концепций, методологий, норм и в общем системный подход, формируемый данной наукой является важнейшим краеугольным элементом формирования государственной политики сегодняшнего дня.

Отправной точкой и методологическим ориентиром на пути институционального развития данной науки должны стать принципы и нормы, отраженные в Конституции. Какой

бы ни была цифровая реальность по степени ее развитости, она в конечном счете должна подпадать под действие Конституции как нормативного акта, имеющего высшую юридическую силу.

Государственное регулирование с учетом масштаба и скорости развития технологий, имеет шанс быть эффективным исключительно с учетом развития информационного права, как науки, обеспечивающей концептуальные методологические подходы к формированию безопасности информационного общества [1].

В современных условиях для повышения капитализации и получения конкурентных преимуществ как в коммерческой среде, так и на государственном уровне применяются новые цифровые технологии, которые формируют новую, четвертую мировую технологическую революцию, которая выражается в создании и бурном развитии современных информационных технологий, их широкое использование в различных сферах деятельности, формировании «цифровой» экономики и «цифровизации» правовой системы.

Совокупность совместно применяемых современных цифровых, IT-технологий, включающих, в частности, такие сферы, как: Интернет вещей, искусственный интеллект, робототехнику, большие данные, облачные вычисления, цифровое моделирование и дополненная реальность, аддитивное производство, создают технологический фундамент «цифровой экономики», новых социальных и общественных отношений в инфокоммуникационном пространстве.

В таких условиях «цифровизации» общественных отношений, предпринимательской деятельности и государственного управления сфера права под влиянием современных цифровых технологий серьезно преобразуется, что находит отражение, как отмечают исследователи феномена «цифровизации» права, во множестве новых правовых явлений, связанных с субъектами и объектами правового регулирования, спецификой правоотношений в цифровой реальности, осмысления понятия и содержания цифровых прав и т.д.

Интернет создает виртуальную информационную среду, в которой возникают, изменяются и прекращаются различные общественные отношения (гражданско-правовые, моральные, этические и др.). Глобальная сеть представляет собой яркий пример того, насколько удачно и эффективно может развиваться столь сложная техническая система практически в отсутствие государственного регулирования, основываясь только на выработанных внутри Сети правилах. Это порождает важный вопрос о том, как скоро уровень развития социальных отношений, связанный с существованием подобной системы, потребует разработки и применения соответствующего правового регулирования. Хотя, безусловно, на сегодняшний день говорить об опосредованности роли государств в регулировании цифровой среды едва ли возможно.

Путь цифровой трансформации сегодня требует фундаментальной перестройки подходов частного бизнеса и государства к взаимодействию, принятию решений, стимулированию инноваций и формированию законодательной среды, где у каждого участника системы – своя значимая роль.

Развитие инфраструктуры, снижение стоимости обработки, хранения и передачи данных подводят человечество к порогу нового, наиболее масштабного этапа цифровой революции, который в отличие от предыдущего, характеризовавшегося быстрым проникновением Интернета в жизнь потребителей, отличается быстрым и расширяющимся использованием в самых разнообразных областях деятельности более широкого спектра цифровых сервисов, продуктов и систем. Сегодня вполне обосновано говорится о слиянии онлайн- и офлайн-сфер, о появлении киберфизического мира, который характеризуется созданием и интеграцией принципиально новых революционных цифровых технологий [2].

Формирование киберпространства и использование в нем новых технологий на базовом принципе распределенного (децентрализованного) реестра (blockchain tech) привело к созданию принципиально нового правового инструментария: умные контракты,

электронные цифровые подписи, базовые технологические стандарты, правила и др. Новые цифровые технологии используются при ведении официальных государственных реестров, объектов недвижимости или реестров нормативных правовых актов, а также в таких сферах правоприменения как сбор налогов, учет и выдача документов, оказания государственных услуг. Цифровые технологии получили широкое применение в промышленном интернете, в управлении сложными технологическими процессами и структурами. С учетом всего вышесказанного, вопрос обеспечения информационной безопасности на государственном уровне выходит на беспрецедентно новый уровень.

Как отмечается в докладе Европейского агентства сетевой и информационной безопасности (European Union Agency For Network And Information Security) за 2018 год угрозы и риски, связанные с устройствами, системами и услугами Интернета вещей, многообразны и стремительно развиваются. Интернет вещей оказывает все большее влияние на безопасность и частную жизнь граждан, а сами виды угроз в отношении Интернета вещей чрезвычайно многообразны. Появление и использования новых цифровых активов (криптовалют, токенов и др.), повышение доступности цифровых финансовых услуг создают новые риски для «отмывания» доходов преступной и террористической деятельности, а также финансирование такой незаконной деятельности. Поэтому Межправительственная организация «Группа разработки финансовых мер борьбы с отмыванием денег (ФАТФ)» призывает к скоординированным усилиям всех стран, направленным на предотвращение использования виртуальных активов в преступных и террористических целях [3].

Наглядным примером рисков, связанных с повсеместной «цифровизацией» служат конфликты, связанные с невыполнением различными компаниями требований регуляторов о предоставлении средств дешифровки сообщений с которыми сталкиваются правоохранительные органы многих стран. Данный факт свидетельствует о необходимости поиска в законодательном регулировании оптимального правового компромисса между возможностью доступа правоохранительных служб к компьютерной информации и правом граждан на ее конфиденциальность.

Стоит также отметить, что с момента появления синдрома «Викиликс» стало очевидным, что крайне остро стоит вопрос о необходимости единообразного подхода к регулированию информационной сферы в общемировом масштабе. Именно это послужило началом переоценки мировым сообществом традиционных отношений к информации, которая способна не только нанести вред, распространяющий свое влияние практически на все сферы государственного управления, но также может оказать долгосрочное влияние на дальнейшее развитие того или иного правового субъекта, чьи интересы напрямую или косвенно расходятся с ее содержанием. Таким образом, ни у кого уже не вызывает сомнений тот факт, что сегодня информация способна если не изменить, то оказать реальное влияние на ход истории развития мировой государственности, а «Викиликс» – это еще один вид оружия в жестокой информационной войне, целью применения которого является стремление неких сил ограничить мировое влияние ведущих стран бескровным, но очень технологичным способом [4].

Возрастание роли информационной безопасности, связано также и с появлением новых видов правонарушений в информационной сфере. В частности, в интернет-банкинге – кража денежных средств с банковских счетов с использованием программных средств-эксплойтов (компьютерные программы, использующие уязвимости программного обеспечения для проведения атак на вычислительные системы и электронные устройства, например, смартфоны и др.). Киберсквоттинг (захват доменных имен), т.е. регистрация доменного имени, частично сходного с уже зарегистрированным, или тождественного по написанию с иным средством индивидуализации (товарным знаком). Брэндсквоттинг – регистрация на определенной территории товарного знака, ранее не зарегистрированного, с целью продажи его заинтересованным лицам и др.

К самостоятельным составам преступлений в сфере информационной безопасности можно отнести мошенничество с платежными и кредитными картами (статья 159.3 УК РФ) и мошенничество в сфере компьютерной информации (статья 159.6 УК РФ). Уголовный кодекс РФ (ст.272-274 УК РФ) предусматривает уголовную ответственность за неправомерный доступ к охраняемой законом компьютерной информации (не только ЭВМ или их сети), за создание, распространение или использование вредоносных компьютерных программ, а также за нарушение правил эксплуатации средств хранения, обработки или передачи охраняемой компьютерной информации (информационно-телекоммуникационных сетей).

Мошенничеством в сфере цифровой информации признается и совершение посредством неправомерного доступа к компьютерной информации или посредством создания, использования и распространения вредоносных компьютерных программ, а также целенаправленное воздействие программных и (или) программно-аппаратных средств на серверы, средства вычислительной техники, в том числе переносные (портативные) – ноутбуки, планшеты, смартфоны, снабженные соответствующим программным обеспечением, или на информационно-телекоммуникационные сети, которое нарушает установленный процесс обработки, хранения и передачи цифровой информации, что позволяет виновному или иному лицу незаконно завладеть чужим имуществом или приобрести право на него.

Сферы, затрагиваемые при борьбе с киберпреступностью в мировой практике, также имеют широчайший диапазон. Стоит упомянуть лишь об основных:

1. Раскрытие и расследование уголовных преступлений, в которых фигурируют компьютерная информация как объект посягательства, компьютер как орудие совершения преступления, а также какие-либо цифровые доказательства.

2. Сбор и исследование доказательств для гражданских дел, когда такие доказательства имеют вид компьютерной информации. Особенно это актуально по делам о нарушении прав интеллектуальной собственности, когда объект этих прав представлен в цифровом виде — программа для ЭВМ, иное произведение в цифровой форме, товарный знак в сети Интернет и т. п.

3. Страховые расследования, проводимые страховыми компаниями касательно возможных нарушений условий договора, страхового мошенничества, особенно когда объект страхования представлен в виде компьютерной информации или таким объектом является информационная система.

4. Внутрикorporативные расследования инцидентов безопасности, касающихся информационных систем. Работы по предотвращению утечки информации, содержащей коммерческую тайну и иной конфиденциальной информации.

5. Военные и разведывательные задачи по поиску, уничтожению и восстановлению компьютерной информации в ходе оказания воздействия на информационные системы противника и защиты своих систем.

6. Задачи по защите гражданами своей личной информации в электронном виде, самозащиты своих нрав, когда это связано с документами в цифровом виде и информационными системами.

Типичными преступными целями совершения компьютерных преступлений согласно УК РФ являются: шпионаж (экономический, политический и др.), саботаж (несанкционированная эксплуатация АИС), подделка документов (получение фальшивых дипломов, фиктивное продвижение по службе и др.); хищение компьютерной информации, денег (подделка счетов и платежных ведомостей; фальсификация платежных документов, вторичное получение уже произведенных выплат, перечисление денег на подставные счета и др.), вещей (совершение покупок с фиктивной оплатой, добывание запасных частей и редких материалов); внесение изменений в компьютерную информацию, приписка сверхурочных часов работы, кража машинного времени и др.

Что касается технических достижений в области киберпреступности, то ее можно разбить на 3 основных направления:

1) Возможность совершать преступления новыми способами и при помощи новых орудий. Например, то же старое мошенничество в наш век совершается при помощи сети Интернет. Но суть и предмет посягательства у мошенничества прежние. Новыми являются лишь орудия совершения — веб-сайт, электронная почта, платёжная система.

2) Достижения информационных технологий порождают принципиально новые общественные отношения, которые и становятся предметом преступных посягательств. При этом способ посягательства и орудия могут быть как старыми, так и новыми, с учётом достижений ИТ. Самый яркий пример — доменные имена. Такого общественного отношения, как право распоряжаться доменным именем до недавних пор просто не существовало. Не было и посягательств. Ныне доменные имена охраняются законом (они причислены к объектам интеллектуальной собственности).

3) Развитие ИТ может привести к появлению не просто новых общественных отношений, но и нового субъекта таких отношений. Появление полноценного искусственного интеллекта уже явно просматривается на научном горизонте. А пока можно говорить о первых шагах в этом направлении. Программа для ЭВМ ещё не рассматривается в качестве субъекта права, но в качестве стихийной силы уже во многих зачастую рассматривается. Программам уже дано принимать решения, которые могут существенно влиять на благосостояние и даже жизнь людей. Чего только стоят ошибки систем регулирования полетов, приведшие к непоправимым человеческим жертвам.

Программы уже могут порождать новые объекты авторского права. Словом, появление принципиально нового субъекта, нового члена общества со своими правами — искусственного интеллекта — дело времени. А его появление вызовет новые отношения и, соответственно, новые правонарушения, к которым мы с учетом нашего развивающегося законодательства совершенно не готовы [5].

Последствия неправомерного использования информации могут быть самыми разнообразными: разглашение сведений, содержащих государственную или коммерческую тайну, разглашение сведений о частной жизни граждан, нарушение неприкосновенности интеллектуальной собственности, имущественный ущерб в виде прямых убытков и неполученных доходов, потеря репутации фирмы, различные виды нарушений нормальной деятельности предприятия, отрасли и др. Поэтому совершенно оправдано то, что преступления данного вида помещены в раздел «Преступления против общественной безопасности и общественного порядка».

Диапазон способов извлечения и компрометации данных имеет широчайший спектр и по мнению экспертов будет только увеличиваться. Учитывая этот фактор наряду с всеобъемлющей по своим масштабам переводом всех видов рабочих процессов на инфокоммуникационную платформу, всесторонняя вовлеченность государства является не просто необходимым, а безальтернативно важнейшим элементом.

При рассмотрении дел, связанных с компьютерными преступлениями, судьи сегодня чувствуют себя неуверенно так как четких законов, защищающих государственную, экономическую, политическую и др. ценность информационных средств и ресурсов пока еще нет ни в одной стране. Одним из главных факторов, мешающих разработке правоохранительных законов в сфере информационно-компьютерной преступности, является то, что преступное использование компьютера не оставляет никаких видимых следов. Закон же традиционно строится на наличии внешних материальных свидетельств преступления. Отсюда, возможно, и сложность в обеспечении законодательной охраны цифровой информации.

Особенность компьютерных преступлений состоит и в том, что трудно найти другой вид преступления, после совершения которого его жертва не выражает особой заинтересованности в поимке преступника, а сам преступник, будучи пойман, всячески

рекламирует свою деятельность на поприще «компьютерного взлома», мало что утаивая от представителей правоохранительных органов. Психологически этот парадокс вполне объясним. Во-первых, жертва компьютерного преступления совершенно убеждена, что затраты на его раскрытие (включая потери, понесенные в результате утраты своей репутации) существенно превосходят уже причиненный ущерб. Во-вторых, преступник приобретает широкую известность в деловых и криминальных кругах, что в дальнейшем позволяет ему с выгодой использовать приобретенный опыт [6].

Стремительное увеличение оборота разнообразной информации, глобализация доступа к ней и появление новых средств ее формирования, распространения и использования с каждым днем все сильнее актуализируют вопросы сохранности и легального использования массивов информации. Информационная безопасность выходит за рамки потребностей отдельных обладателей и выступает уже в качестве одного из направлений национальных стратегий развития.

Контроль, надзор, выработка концепций, методик для инфокоммуникационной среды, по мнению авторов, должны сегодня воплощаться государством не только через тактический, но главным образом через стратегический подход к вопросам, связанным с обеспечением информационной безопасности. Путь к регулированию одной из наиболее быстроразвивающихся и всепроникающих сфер напрямую лежит через развитие информационного права, как инструмента по регламентации и закреплению правил, норм, методов и стандартов взаимодействия, технологий, прав гражданина в цифровой среде и иных процессов, которые все глубже и быстрее пронизывают современные государства и общества.

Обобщая вышесказанное, диапазон способов извлечения и компрометации данных имеет широчайший спектр и будет только увеличиваться. Учитывая этот фактор наряду с всеобъемлющей по своим масштабам переводом всех видов рабочих процессов на инфокоммуникационную платформу, всесторонняя вовлеченность государства является безальтернативно важным элементом.

Государственное регулирование с учетом масштаба и скорости развития технологий, имеет шанс быть эффективным исключительно с учетом развития информационного права, как науки, обеспечивающей концептуальные многоуровневые подходы к формированию безопасности информационного общества.

Библиографический список:

1. Пьянков К.А., Джура Г.С., Декунова Е.С., «Информационное право как инструмент обеспечения информационной безопасности государства» Современное государственное и муниципальное управление: проблемы, технологии, перспективы // «Вестник СГТУ». V Международная научно-практическая конференция (25.04.2019).

2. Новая волна цифровой революции URL: <http://russiaonline.info/story/new-wave-of-digital-revolution> (дата обращения: 26.06.2019).

3. Обеспечение информационной безопасности в условиях виртуализации общества: опыт европейского союза URL: <http://spkurdyumov.ru/uploads/2013/08/smirnov.pdf> (дата обращения: 30.06.2019).

4. К вопросу о состоянии информационной безопасности государства в условиях современных вызовов и угроз URL: <https://center-bereg.ru/h39.html> (дата обращения: 29.06.2019).

5. Форензика (компьютерная криминалистика). URL: https://bookucheba.com/informatsionnaya-bezopasnost_1280/predmet-45386.html (дата обращения: 30.06.2019).

6. Компьютерные преступления. URL: <http://mirznanii.com/a/124435-3/kompyuternye-prestupleniya-3> (дата обращения: 30.06.2019).

СЕКЦИЯ 5

**Компьютерная инженерия
систем, сетей и
компонентов**



УДК 004.75

В.В. Евсеева, ассистент**Е.Н. Машенко, канд. техн. наук, доцент***Севастопольский государственный университет*

ИСПОЛЬЗОВАНИЕ БЕСПРОВОДНЫХ СЕНСОРНЫХ СЕТЕЙ В СИСТЕМАХ БОЛЬШИХ ДАННЫХ

Аннотация: Одним из вопросов проектирования систем больших данных является создание инфраструктуры сбора данных. Беспроводные сенсорные сети (БСС) являются одним из возможных решений для этой задачи: большой объем данных генерируется разнородными сенсорными узлами, объединенными беспроводной сетью. Однако, в отличие от обычных беспроводных сетей, БСС имеют серьезные недостатки в вопросах надежности из-за ограниченных возможностей узлов. Кроме того, при плотном развертывании сенсорных узлов генерируется значительный объем избыточных данных, которые могут не представлять интереса. В данной статье исследуется вопрос возможности внедрения БСС в системы больших данных. Рассмотрены возможные технические проблемы сетей и инфраструктуры, а также предложены потенциальные приложения.

Ключевые слова: беспроводные сенсорные сети, большие данные, инфраструктура сбора данных, обработка данных

Annotation: Before discovering meaningful knowledge from big data systems, it is first necessary to build a data-gathering infrastructure. Wireless sensor networks (WSNs) are rich big data sources: a large amount of data is generated by various sensor nodes in large-scale networks. However, unlike typical wireless networks, WSNs have serious deficiencies in terms of data reliability and communication owing to the limited capabilities of the nodes. Moreover, a considerable amount of sensed data are of no interest and redundant when a large number of sensor nodes is densely deployed. Potential applications and technical challenges of networks and infrastructure are presented and explained in accordance with the research areas and objectives.

Key words: wireless sensor networks, big data, data collection infrastructure, data processing

За последнее десятилетие множество технологий перешло на качественно новый этап развития. Новые стандарты беспроводной связи, новые подходы к сбору и обработке данных стали основной для облачных вычислений, которая в свою очередь вместе с парадигмой «Интернет всего» (Internet of Everything) сформировали новую концепцию - большие данные (Big Data).

Большие данные – это инновационная технология, используемая для описания структурированных и неструктурированных данных для улучшения процессов принятия решений. Термин Big Data используется для больших объемов данных, имеющих наибольшую сложность. Как правило, такие данные не могут быть сохранены, управляться и анализироваться с помощью стандартных методов обработки данных. Большие данные используются при наблюдении за стихийными бедствиями, в сенсорных сетях, в телекоммуникации и т.д [1].

Существует ряд характеристик, отличающих большие данные, так называемые 6V (рисунк 1):

Volume (объем) – генерируются большие объемы данных, которые требуют огромного хранилища или состоят из большого количества записей;

Velocity (скорость) – поток данных генерируется с высокой скоростью, и/или скорость их обработки значительна;

Variety (разнообразие) – данные собираются из огромного количества разнородных источников и могут иметь разные форматы и степень структурированности;

Veracity (достоверность) – данные должны быть очищены от шума и аномальных значений;

Viability (жизнеспособность) – полученные данные должны позволять проводить различные прогнозы в будущем;

Value (ценность) – скрытые взаимосвязи и новые факты, которые будут обнаружены, должны обеспечить экономическую целесообразность сбора большого количества данных.

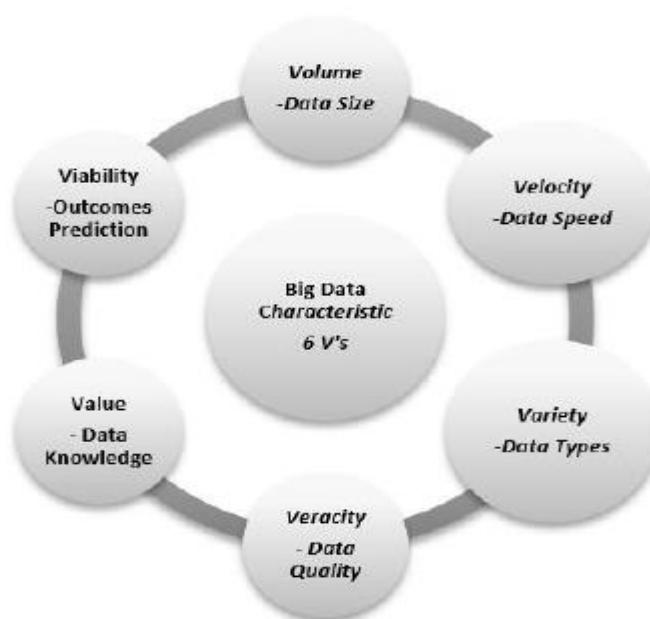


Рисунок 1 – Основные характеристики больших данных

Задача построения систем больших данных состоит из нескольких этапов: организация процессов сбора данных, хранения, обработки, анализа и визуализации данных.

Сбор данных является первым этапом построения систем больших данных. Среди многих современных технологий, порождающих генерацию больших данных, беспроводные сенсорные сети (БСС) получили значительное внимание исследователей [2]. БСС состоит из большого количества сенсорных узлов, которые отслеживают и записывают данные о физическом состоянии окружающей среды, а данные собираются в так называемом узле-приемнике. БСС используются для измерения таких параметров как: температура, уровни влажности, шума и т.д [3]. Несмотря на то, что беспроводные сенсорные сети прошли долгий путь в своем развитии, ряд вопросов все еще остается открытым и заслуживает дальнейшего изучения.

Большое количество задействованных в сети сенсорных узлов производит большое количество данных, которые должны быть проанализированы и в дальнейшем преобразованы в значимую информацию. Для достижения этой цели необходимо ответить на ряд вопросов, таких как: какие данные необходимо собирать, куда они должны передаваться, каким образом данные отправятся в пункт назначения и т.д. Кроме того, чтобы ответить на эти вопросы, необходимо принять во внимание несколько конструкторских ограничений, накладываемых на БСС, таких как ограниченная мощность сенсорного узла, его низкая пропускная способность, а также динамическая природа сенсорных узлов (узлы могут перемещаться или «умереть» из-за истощения батарей).

Основными решениями данных проблем могут стать динамическая маршрутизация, кластеризация, агрегация данных, предварительная обработка данных в сети и т.д. Однако ни одно из решений не является достаточно общим и не может быть применено для любой системы.

Поэтому существует потребность в анализе возможного эффективного применения БСС, несмотря на недостатки данной технологии.

На рисунке 2 представлена структура системы больших данных, основная на использовании беспроводной сенсорной сети. Узел-приемник собирает данные с сенсорных узлов, а затем доставляет данные во временное хранилище для последующей агрегации данных. Далее агрегированные данные обрабатываются средой больших данных с

использованием основного хранилища. Преобразованные данные обрабатываются платформами больших данных.

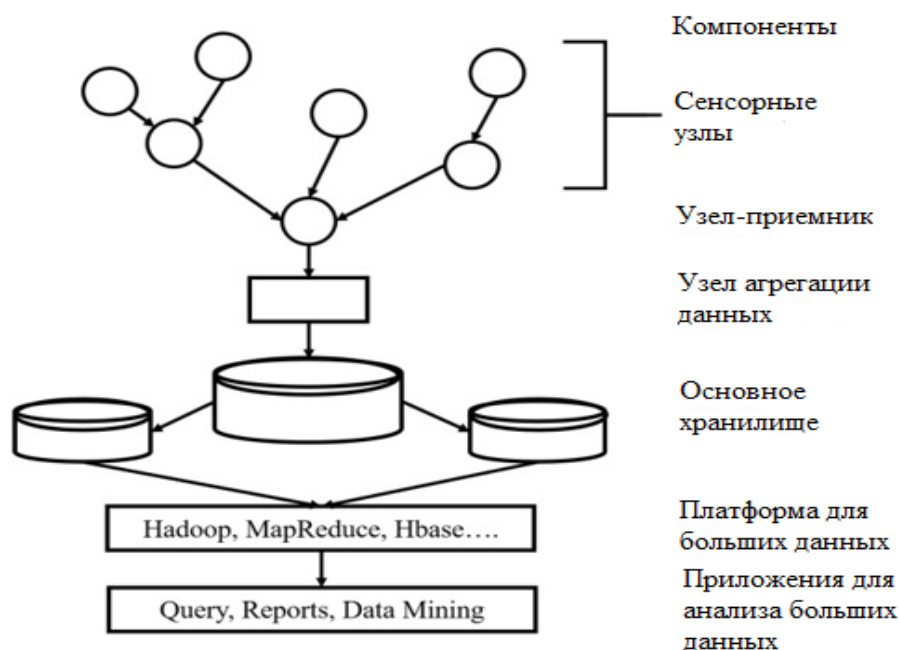


Рисунок 2 – Структура системы больших данных на основе БСС

Следует выделить следующие особенности применения БСС в системах больших данных:

Безопасность. Во время сбора данных WSN выполняет как сбор, так и передачу данных, и важно обеспечить безопасность выполнения этих задач, т.к. в БСС могут внедряться поддельные данные. Следовательно, необходима организация конфиденциальной передачи данных через безопасную сеть связи. Принимая во внимание технические ограничения сенсорных узлов, имеет смысл применить симметричную криптографию в беспроводной сенсорной сети. Это позволит поддерживать хороший баланс между эффективностью и безопасностью, который достигается с помощью шифрования с открытым ключом.

Кроме того, для эффективного и правильного обнаружения вторжения для защиты от внутренних атак имеет смысл применение доверительного подхода в БСС. Однако из-за обилия избыточных данных эффективность сети может значительно снизиться.

Требования к приложению. Как правило, БСС разворачивается и внедряется для выполнения специфических задач, а также для удовлетворения конкретных требований. Например, некоторые приложения требуют скалярных значений, в то время как другие требуют векторных. В соответствии с требуемым типом значений в приложении, данные, генерируемые узлом датчика, обладают естественно различными свойствами и характеристиками. Это подразумевает, что сеть и система данных для больших данных должны быть спроектированы в соответствии с требованиями приложения. Более того, поскольку мультимедийные данные требуют большего внимания к потреблению ресурсов, чем обычные данные, следует разработать подходящую структуру и архитектуру системы, чтобы использовать конкретные алгоритмы независимо от объема, скорости и разнообразия генерируемых данных.

Взаимодействие в режиме реального времени на основе туманных вычислений. Из-за чувствительности к задержке обнаружения инцидентов (своевременного сбора данных) в БСС обмен и обработка данных в режиме реального времени являются наиболее важными свойствами для систем больших данных. Однако жесткие ограничения в устройстве

сенсорных узлов БСС затрудняют поддержку связи в реальном времени. Более того, система больших данных в архитектуре облачных вычислений не подходит для выполнения задач в реальном времени. Для решения этой проблемы имеет смысл применение туманных вычислений, чтобы расширить облачные сервисы до границы сети путем вычислений, связи и хранения данных ближе к периферийным устройствам и конечным пользователям, что направлено на повышение мобильности и пропускной способности сети, а также безопасности и конфиденциальности.

Гибкая структура. Несмотря на то, что беспроводные сенсорные сети считаются одним из основных источников больших данных, существует множество других альтернатив. Они используются в приложениях, построенных на радиочастотной идентификации (RFID), межмашинном взаимодействии (M2M), одноранговых сетях (P2P), интернете вещей и смартфонах. Если данные из БСС удастся объединить с данными из других приложений, то можно создать более полезную систему больших данных. Это означает, что данные с этих устройств должны быть интегрированы и обрабатываться однородно. Другими словами, существует потребность в создании интегрированной, обширной и гибкой структуры для размещения различных типов данных.

Моделирование системы. Оценка эффективности системы является важной задачей при проектировании систем больших данных и беспроводных сенсорных сетей. Однако, на данный момент на рынке систем моделирования не существует интегрированного и комбинированного симулятора или испытательного стенда для моделирования такого рода систем.

Заключение. Беспроводные сенсорные сети являются ключевым компонентом систем больших данных. Благодаря все новым техническим решениям в проектировании сенсорных узлов объем данных, генерируемых БСС растет в геометрической прогрессии. Однако, технические несовершенства БСС накладывают определенные ограничения в работу систем сбора данных. В данной статье были рассмотрены основные проблемы при проектировании систем больших данных, в основе которых лежит беспроводная сенсорная сеть как подсистема сбора данных. Отмечена необходимость дальнейшего исследования применения беспроводных сенсорных сетей в системах больших данных.

Библиографический список:

1. Chen, CL Philip, and Chun-Yang Zhang. Data-intensive applications, challenges, techniques and technologies: A survey on Big Data // Information Sciences, vol. 275, pp. 314-347, 2014.
2. Chen, Min, Shiwen Mao, Yin Zhang, and Victor CM Leung. Big Data Generation and Acquisition. // In Big Data, Springer International Publishing, pp. 19-32, 2014.
3. Евсеева В.В., Мащенко Е.Н. Исследование критериев эффективности функционирования беспроводных сенсорных сетей //Мир компьютерных технологий: Сборник статей всероссийской студенческой научно-технической конференции, г.Севастополь, 2 – 6 апреля 2018 г/ М-во образования и науки РФ, Севастопольский государственный университет; науч. ред. Е.Н. Мащенко – г. Севастополь: СевГУ, 2018. – с.109
4. Жданов, В.С. Проблемы и задачи проектирования беспроводных сенсорных сетей /В.С. Жданов// Информационные, сетевые и телекоммуникационные технологии: сборник научных трудов, под ред. проф. д.т.н. Жданова В.С. – 2009. – С.8-10.
5. Иванов, В. Э. Проблемно-ориентированные сенсорные сети: практический подход [Текст] / В. Э. Иванов, А. В. Левенец, Чье Ен Ун ; под ред. Чье Ен Уна ; Тихоок. гос. ун-т. - Хабаровск : ТОГУ, 2015. - 219 с.

УДК 004

Ю.С. Мамошина, студентка**А.В. Росляков, докт. техн. наук, профессор***Поволжский государственный университет телекоммуникаций и информатики***ИССЛЕДОВАНИЕ ЗАДЕРЖЕК В СЕТИ LTE МЕТОДАМИ ТЕОРИИ NETWORK CALCULUS**

Аннотация: рассмотрены основные понятия теории сетевых исчислений Network Calculus. Проанализированы сквозные задержки в сети LTE, используя методы сетевого исчисления.

Ключевые слова: сетевое исчисление, стохастическое сетевое исчисление, сети LTE, сетевая архитектура LTE, сквозные задержки

Annotation: considered the basic concepts of Network Calculus. LTE network delays are analyzed using Network Calculus methods

Key words: network calculus, stochastic network calculus, LTE networks, LTE network architecture, end-to-end delay

Для исследования сложных пакетных сетей еще в начале 1990-х гг. был предложен качественно новый подход – теория Network Calculus (NC) [1], позволяющая определять граничные оценки основных сетевых характеристик – загрузки сетевых узлов и задержки в передаче пакетов. Ее основной идеей является использование альтернативных методов анализа систем массового обслуживания, в частности, мин-плюс и макс-плюс алгебр, преобразование сложных нелинейных сетевых систем к линейным, которые легко исследовать аналитически.

Следует отметить, что термин Network Calculus не имеет устоявшегося русскоязычного аналога. Очевидно, что дословный перевод с английского «network calculus» – «сетевое исчисление», не отражает сути моделей, положенных в основу этой теории, поэтому приблизительно названный термин можно охарактеризовать как «совокупность математических результатов для моделирования и исследования характеристик функционирования различных систем массового обслуживания, например, телекоммуникационных сетей». Однако авторы все же остановились на термине «сетевое исчисление», который уже достаточно давно используется в отечественных научных публикациях, посвященных исследованию граничных характеристик различных сетевых моделей с использованием теории NC.

Любая сеть связи описывается двумя основными понятиями: сетевыми узлами и потоками поступающих на них заявок (пакетов, вызовов, требований и т.п.) [2]. Соответственно, любая теория сетевого анализа обычно строится на двух фундаментальных положениях: модель поступающего на узел трафика A и модель обслуживания этого трафика в узле S . При этом модель входного трафика A характеризует поведение входного потока заявок, а модель обслуживания трафика S характеризует поведение сетевого узла.

Сети LTE — это двухуровневые IP-системы, работающие на основе коммутации пакетов, а не каналов, как ранние сотовые сети [3]. «Плоская» архитектура обеспечивает простоту и дешевизну развертывания: на пользовательском уровне сеть состоит из базовой станции LTE и шлюза SAE, подключающегося непосредственно к опорной сети. Все соединения в сети — IP-интерфейсы, что обуславливает легкость ее интеграции с любыми другими сетями. Технология LTE полностью обратно совместима с технологиями GPRS, EDGE, UMTS, HSPA и HSPA+, что позволяет терминалам переключаться на их использование в тех местах, где нет 4G-покрытия.

Двумя ключевыми технологиями в сети LTE являются мультиплексирование с ортогональным частотным разделением (OFDM) и мультиплексирование с несколькими входами и несколькими выходами (MIMO). Первая технология используется для нисходящей

линии связи LTE, отвечает требованиям LTE в отношении гибкости спектра и обеспечивает экономически эффективные решения для широких несущих с высокими пиковыми скоростями. Это также хорошо зарекомендовавшая себя технология, например, в стандартах IEEE 802.11 a/b/g, 802.16, HiperLAN-2, Digital Video Broadcast (DVB) и Digital Audio Broadcast (DAB). Вторая технология MIMO – использование нескольких антенн, как на передатчике, так и на приемнике для улучшения производительности связи. Она привлекла широкое внимание в беспроводной связи, поскольку обеспечивает значительное увеличение пропускной способности данных и диапазона каналов без дополнительной полосы пропускания или мощности передачи. И MIMO становится важной частью современных стандартов беспроводной связи, таких как IEEE 802.11n, 4G, LTE, WiMAX.

Сети радиодоступа LTE, использующие технологию MIMO, могут работать в следующих режимах передачи:

- 1) пространственное мультиплексирование увеличивает пропускную способность за счет передачи независимых потоков данных через разные антенны;
- 2) пространственное разнесение повышает надежность путем параллельной передачи избыточных потоков информации;
- 3) формирование луча может использоваться для направленной передачи или приема сигнала для улучшения покрытия соты;
- 4) множественный доступ с пространственным разделением (SDMA) может увеличить пропускную способность канала, повторно использовать частоту и полностью использовать частотные ресурсы.

Поскольку матрица канала MIMO оказывает глубокое влияние на пропускную способность канала и, следовательно, на задержку, поэтому используется анализ QoS системы MIMO в режиме пространственного мультиплексирования.

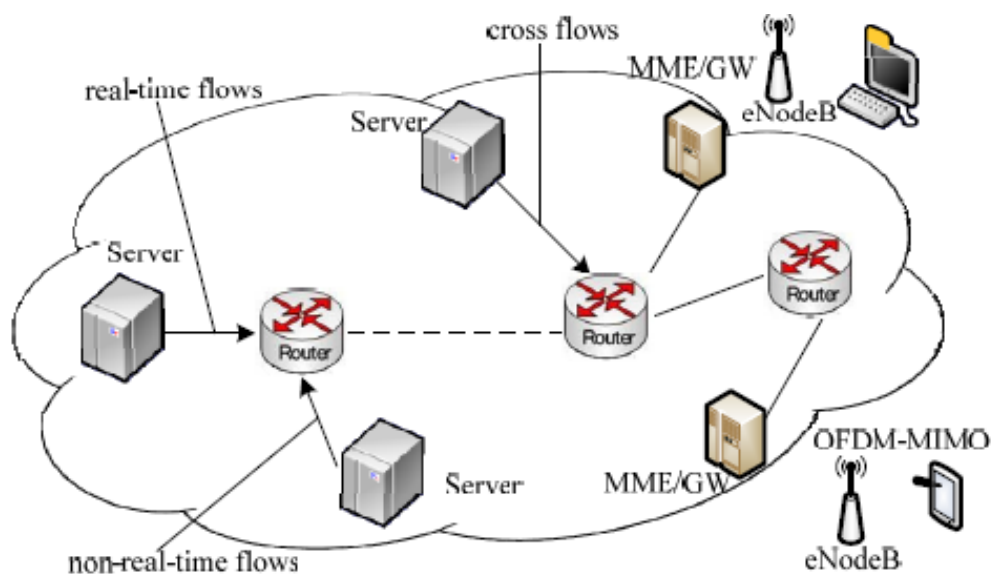


Рисунок 1 – Трафик в реальном времени (RT) и трафик не в реальном времени (NRT)

На рисунке 1 представлены два типа трафика: трафик в реальном времени (RT) и трафик не в реальном времени (NRT). Голос по IP (VoIP) и потоковое видео – это трафик RT, а просмотр веб-страниц и FTP – это трафик NRT. Трафик RT требует низкой задержки, но может допускать некоторые потери, в то время как трафик NRT намеревается получить высокую пропускную способность. Разные серверы могут передавать соответствующие данные по нескольким переходам в базовой сети CN (Core Network) при ответе на запросы на обслуживание. Будем называть такие потоки сквозными, а трафик, который проходит через

CN, образует перекрестные потоки. На рисунке 1 представлен пример перекрестных потоков. Следует обратить внимание, что перекрестные потоки могут быть потоками RT или потоками NRT. Будем рассматривать стохастические границы задержки E2E для обоих сквозных потоков, используя соответствующие модели стохастического трафика и стохастического обслуживания в узлах сети.

Чтобы упростить анализ задержек, рассмотрим всю опорную сеть CN как черный ящик. В CN на рисунке 1 пусть входы S_{1t} и S_{2t} сходятся с S_{1c}^d и S_{2c}^d на входе маршрутизатора d . Обозначим через $D_{1t,cn}$ и $D_{2t,cn}$ стохастические границы задержки для потоков трафика S_{1t} и S_{2t} соответственно в CN. С учетом модели трафика и модели обслуживания, приведенных в [4], получаем:

$$P\left\{D_{1t,cn} > \frac{x + \sigma_{1t} + \sigma_{1c}}{r_{1t,\theta}}\right\} \leq f_{1t} \otimes g_{1t}(x) \quad (1)$$

$$P\left\{D_{2t,cn} > \frac{x + \sigma_{2t} + \sigma_{2c}}{r_{2t,\theta}}\right\} \leq f_{2t} \otimes g_{2t}(x) \quad (2)$$

Рассмотрим сеть LTE-SAE. Она представляет собой плоскую архитектуру, включающую UE (пользовательское оборудование), eNodeB (усовершенствованный NodeB) и EPC (усовершенствованное пакетное ядро).

На рисунке 2, eNodeB и UE соединяются интерфейсом LTE-Uu. ENodeBs соединены посредством интерфейса X2 и к объекту/шлюзу управления мобильностью (MME/GW) посредством интерфейса S1.

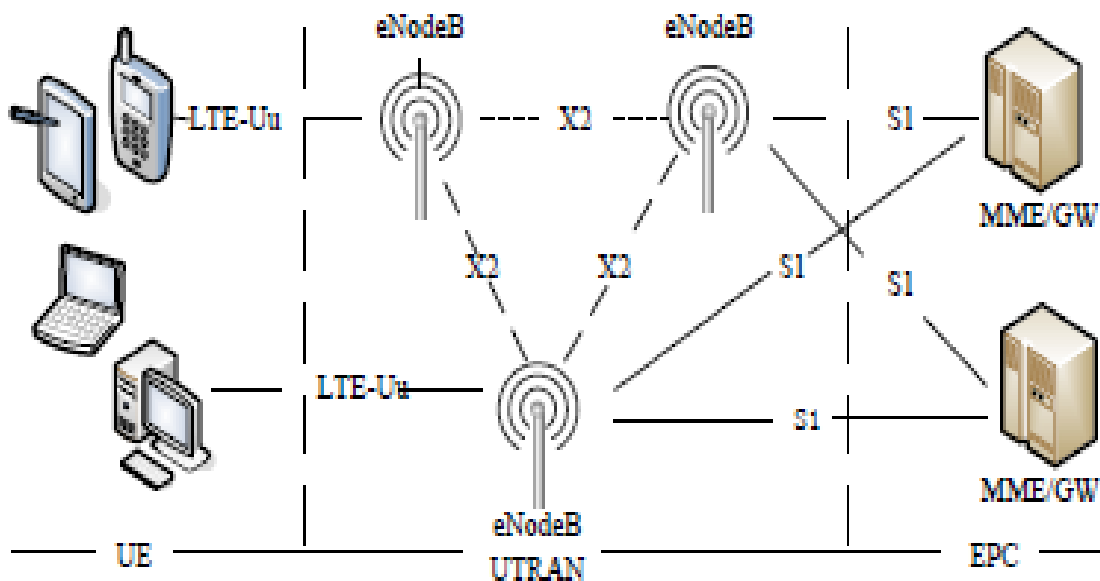


Рисунок 2 – Архитектура сети LTE-SAE

В оценке задержки будем использовать стохастическое сетевое исчисление (SNC). SNC – это развивающаяся методология для компьютерных сетей, обеспечивающая стохастические гарантии обслуживания.

Основная идея SNC заключается в использовании альтернативных алгебр, в частности алгебры min-plus и max-plus, для преобразования сложных нелинейных сетевых систем в аналитически управляемые линейные системы. Поскольку радиосети могут обеспечивать только статистические гарантии обслуживания, SNC лучше использует свойство мультиплексирования трафика в сетях по сравнению с детерминистическим сетевым

исчислением. Стохастическая кривая поступления (SAC) и стохастическая кривая обслуживания (SSC) являются двумя фундаментальными концепциями в SNC. Они используются для характеристики стохастической природы сетевого трафика и услуг, предоставляемых сетевыми элементами, соответственно.

Предположим, что все ограничивающие функции процессов поступления трафика и процессов обслуживания в узлах сети равны e^{-x} . Параметры $\theta = \theta' = \theta'' = 1$.

В CN предполагаем, что установлены следующие параметры: есть два маршрутизатора, каждый с пропускной способностью 100 Мбит/с, и все сквозные потоки пересекают оба маршрутизатора. Скорость поступления сквозных потоков RT (NRT) составляет 200 (100) пак/с, а размер каждого пакета составляет 128 (256) байт. Скорость поступления перекрестных потоков составляет 200 пак/с, а размер каждого пакета составляет 128 байт. Максимальный размер пакета каждого потока составляет 100 Кбайт. Количество сквозных (перекрестных) потоков RT и сквозных (перекрестных) потоков NRT составляет 100 (50).

В сети доступа AN предполагаем, что установлены соответствующие параметры: система MIMO с 2×2 антеннами, использующими каналы шириной 20 МГц. Соотношение сигнал/шум составляет 25 дБ. Скорость нисходящей линии связи может достигать 100 Мбит/с; вероятности переходов канала из «хорошего» состояния в «плохое» и наоборот равны соответственно $p_{gb} = 0,01$ и $p_{bg} = 0,1$. Скорость изменения каждого состояния определяется в соответствии с канальной матрицей. Учитывая указанные выше параметры AN, можно рассчитать пропускную способность канала, которая будет равна 90,909 Мбит/с.

Для приведенного выше набора сетевых параметров можно получить границы стохастической задержки для потоков S_{1t} и S_{2t} . На рисунке 3 изображены вероятности нарушения верхних границ стохастической задержки E2E для S_{1t} и S_{2t} . Из рисунка 3 видно, что вероятность нарушения уменьшается при увеличении стохастической задержки. Это следует из того, что SNC оперирует с ограничивающими функциями. Из рисунка 3 также видно, что стохастическая задержка E2E трафика RT меньше, чем задержка трафика NRT при той же вероятности нарушения границ.

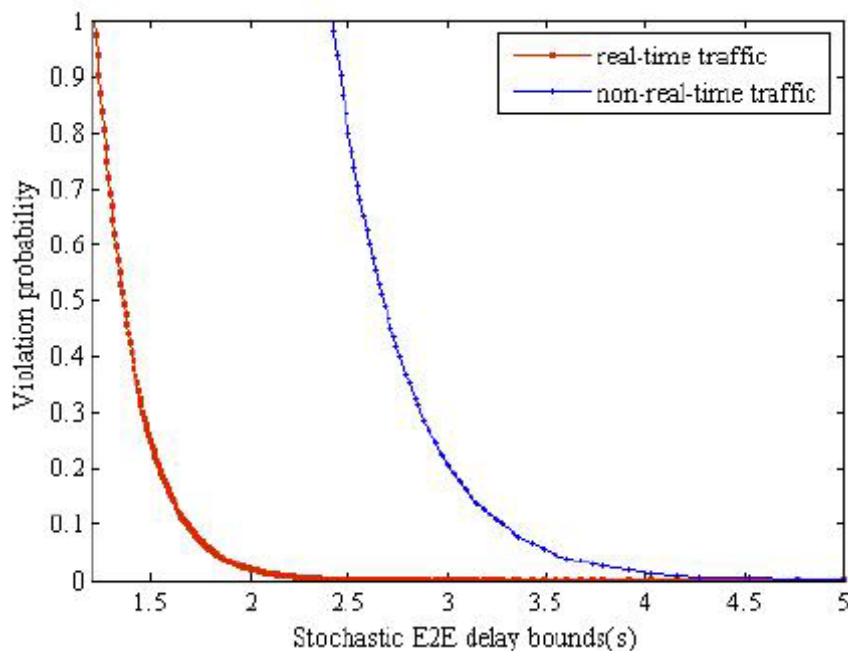


Рисунок 3 – График зависимости вероятности нарушения от стохастической задержки

Полученный результат демонстрирует, что предлагаемая структура сети LTE-SAE может дифференцировать трафик RT и трафик NRT и обеспечивает определенные стохастические гарантии для различного сквозного трафика. Следовательно, можно сделать вывод, что предложенная структура эффективна для анализа стохастических границ задержки E2E для трафика RT и трафика NRT.

Библиографический список:

1. Росляков А.В., Лыников А.А., Витевский В.Д. Сетевое исчисление. Часть 1. Теоретические основы // Инфокоммуникационные технологии. – 2018. – Т.16, №1. – С. 19-33.
2. Росляков А.В., Лыников А.А. Сетевое исчисление (Network Calculus) и его применение для оценки сетевых характеристик. – Самара, ПГУТИ, 2019. – 222 с.
3. Степутин А.Н., Николаев А.Д. Мобильная связь на пути 6G / в 2 томах. – М.: Инфра-Инженерия, 2107.
4. Zhang L., Chen X., Xiang X, Wan J. A Stochastic Network Calculus Approach For the End-to-End Delay Analysis of LTE Networks // International Conference on Selected Topics in Mobile and Wireless Networking (iCOST), 2011, pp. 30-35.

УДК 004.75

А.В. Росляков, докт. техн. наук, профессор**И.В. Поддубнов, студент***Поволжский государственный университет телекоммуникаций и информатики***ВИРТУАЛИЗАЦИЯ В СЕТЯХ 5G***Аннотация: в статье рассматривается виртуализация сетей в 5G, примеры реализации на уже существующем предприятии, технологии реализации слайсов.**Ключевые слова: стандарт 5G, Ethernet, виртуализация, слайсы, сетевые функции**Annotation: the article discusses the virtualization of networks in 5G, examples of implementation in an existing enterprise, technology implementation slices.**Keywords: 5G standard, Ethernet, virtualization, slices, network functions*

Под виртуализацией в сетях 5G понимается разделение одной сети на несколько изолированных срезов (слайсов). Данные технологии находят отличное применение в промышленности. Вычислительные и запоминающие ресурсы подключены к облаку через сеть большой емкости, а машины и приборы связаны между собой промышленными коммуникационными технологиями, такими как промышленный Ethernet.

Стандарт 5G позволяет осуществить переход к облачной архитектуре, где сеть не только предоставляет средства для передачи данных между устройствами, но также предлагает большой эластичный пул настраиваемых ресурсов, включая связь, вычисления, хранение, приложения и услуги. Это обеспечивает высокую степень гибкости производственных линий, поскольку позволяет изменять и масштабировать их без перестройки систем управления и изменения физической инфраструктуры. Наряду с внедрением облачных вычислений предполагается, что количество взаимосвязанных физических устройств резко возрастет и что устройства будут непрерывно взаимодействовать с облаком, чтобы действовать разумно и гибко [1]. Это приводит к тому, что сети связи обслуживают очень разнообразный набор пользователей с наилучшими рабочими характеристиками и приложениями, которым требуется задержка порядка миллисекунд. Кроме того, поскольку технологии беспроводной связи становятся более надежными, ожидается, что они, частично или полностью, заменят проводные технологии, которые в настоящее время доминируют в промышленных сетях [2].

На рисунке 1 представлен концепт сетевых слайсов. Как видно из рисунка 1, одни сетевые сегменты, например, могут обеспечивать доступ к информации с очень низкой задержкой путем выделения ресурсов хранения и вычислений. Точно так же другой сегмент в той же физической сети может предложить лучшую связь между датчиками и облачным сервисом. Следовательно, сетевой сегмент состоит из коммуникационных ресурсов, набора сетевых функций NF (Network Function), то есть функций обработки, которые выполняются в сети, и ресурсов для выполнения этих функций. NF включают ограничители трафика, аварийные сигналы и балансировщики нагрузки, но могут также распространяться на системы хранения и серверы или системы управления в контексте промышленности. Две технологии, облегчающие создание управления и сетевых срезов, это программно-конфигурируемые сети SDN (Software Defined Network) и виртуализации сетевых функций NFV (Network Function Virtualization). SDN позволяет разделить пересылку сообщений и указать, как пересылать сообщения. В контексте сетевого разделения это облегчает маршрутизацию между NF, а также управление очередью. NFV относится к технологии, которая позволяет виртуализировать NFs, и показана на рисунке 1.2. Физическая плоскость (также называемая субстратной сетью) содержит физические компоненты в сети, такие как серверы, системы хранения и технологии связи. Плоскость виртуализации состоит из

виртуальных ресурсов, которые являются логическими разделами или группировками базовых физических компонентов.

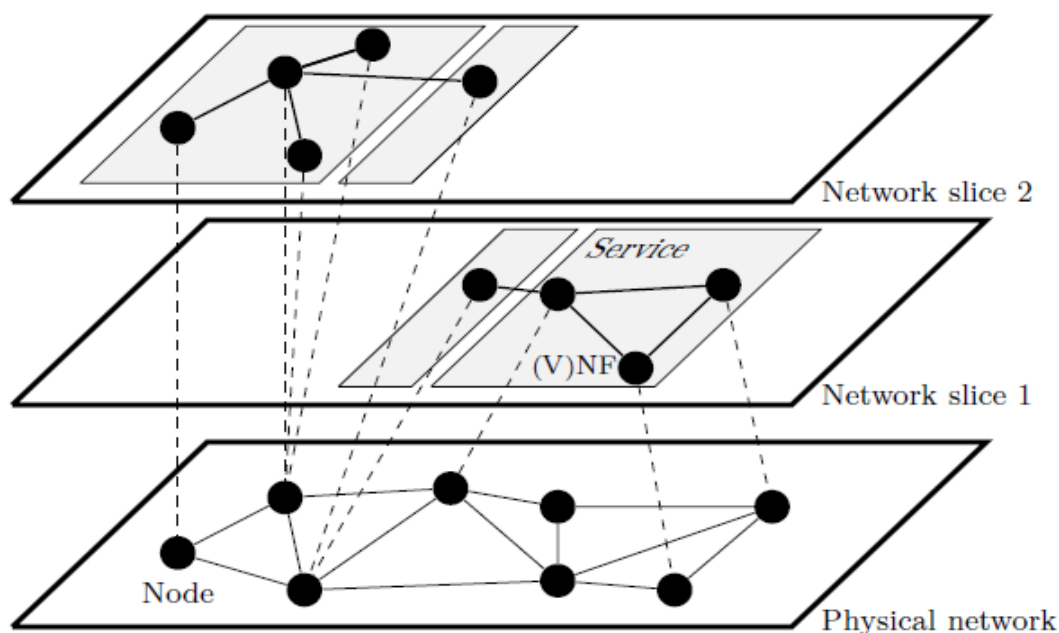


Рисунок 1 - Концепт сетевых слайсов

Виртуальный ресурс может быть виртуальной машиной, состоящей из определенного количества вычислительных ресурсов и памяти, части физического носителя данных или виртуальной сети с определенной емкостью. Наконец, сервисная плоскость содержит ряд виртуальных сетевых функций VNF (Virtual Network Function), которые созданы из виртуальных ресурсов. Три плоскости управляются механизмом управления и оркестровки NFV (NFV-MANO), который отвечает за распределение ресурсов на основе требований приложений и за настройку физических и виртуальных ресурсов, которые обеспечивают сетевые функции (Рисунок 2).

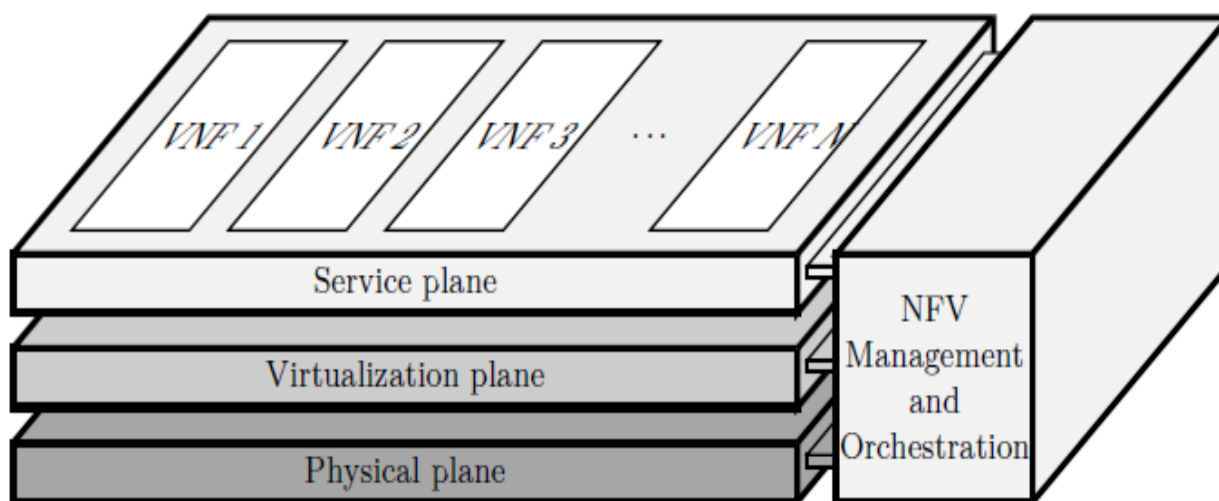


Рисунок 2 - Три уровня абстракции в виртуализации сетевых функций

В текущем состоянии процесса, стандартизации 3GPP 5G слайсов сети является центральной частью архитектуры, где она обеспечивает темы для сквозного QoS, а также изоляцию между различными службами. Например, может быть создан срез для обслуживания всех потоков, которые ищут информацию в конкретной базе данных в облаке, тогда как другой срез может быть создан для предоставления некритической высокопроизводительной услуги, такой как потоковое видео. Чтобы сделать это, различные сетевые срезы могут быть основаны на разных технологиях. Кроме того, сетевые фрагменты могут быть составлены с минимальным количеством требуемых функций предоставлять определенную услугу, чтобы пользовательское оборудование не поддерживало все функции для использования сети. Это увеличивает диапазон устройств, которые могут использовать сети 5G. Наконец, сетевое разбиение может использоваться для оценки новых технологий в полностью развернутой сети без влияния на уже предоставляемые услуги. Разрезание сетей также вводит новые бизнес-модели и партнерские отношения между субъектами бизнес-экосистемы. Например, «нарезка» позволяет использовать инфраструктуру как услугу, когда операторы или провайдеры связи арендуют один или несколько фрагментов физической сети, на которой они могут предоставлять услуги. Отдельные сегменты сети могут управляться провайдером инфраструктуры, провайдером связи, провайдером контента и т.д. В таком случае провайдеры могут оптимизировать свой собственный сетевой сегмент для обеспечения хорошего взаимодействия с пользователем.

Хотя срез сети является центральным элементом в архитектуре 5G, существует несколько проблем, которые вряд ли будут учтены при стандартизации. Это включает в себя все, что не связано с архитектурой, протоколами связи или интерфейсами, например, какие сегменты сети построить, и как распределить ресурсы, чтобы обеспечить определенные сквозные характеристики. Тем не менее, существует обширная литература по этому вопросу, и даже справочные средства для реализации среза сети.

На рисунке 3 представлен концепт сетевых слайсов, разворачиваемых в промышленной сети.

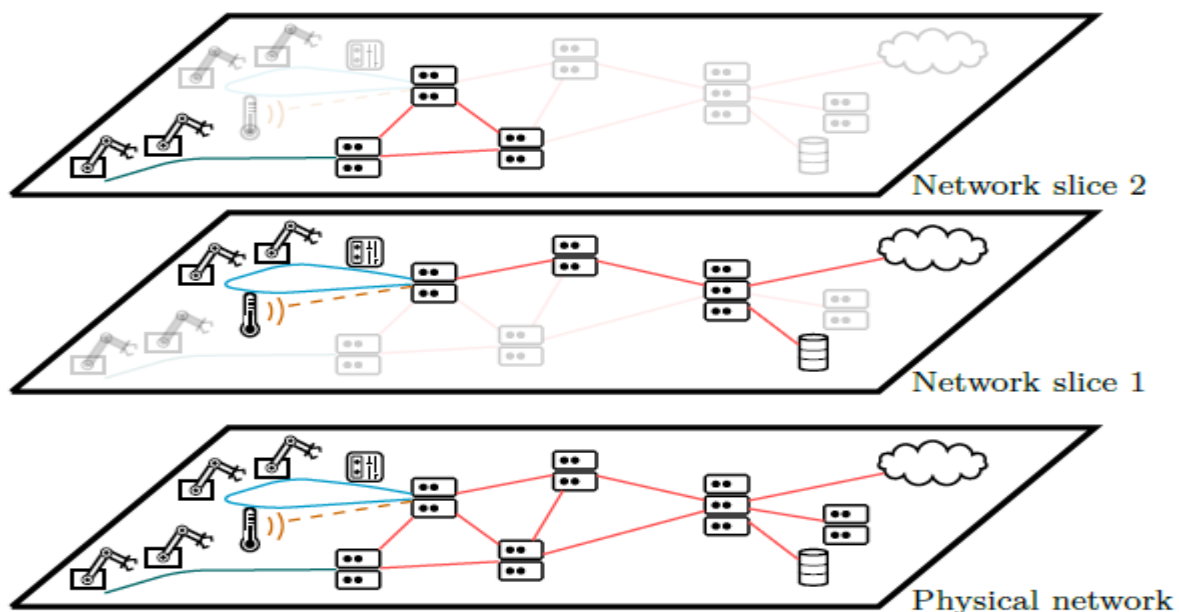


Рисунок 3 - Концепция разбиения сети в промышленных условиях, когда физическая сеть разделена на два сегмента сети

В оценке задержки будем использовать стохастическое сетевое исчисление (SNC). SNC – это развивающаяся методология для компьютерных сетей, обеспечивающая стохастические гарантии обслуживания.

Основная идея SNC заключается в использовании альтернативных алгебр, в частности алгебры min-plus и max-plus, для преобразования сложных нелинейных сетевых систем в аналитически управляемые линейные системы. Поскольку радиосети могут обеспечивать только статистические гарантии обслуживания, SNC лучше использует свойство мультиплексирования трафика в сетях по сравнению с детерминистическим сетевым исчислением. Стохастическая кривая поступления (SAC) и стохастическая кривая обслуживания (SSC) являются двумя фундаментальными концепциями в SNC. Они используются для характеристики стохастической природы сетевого трафика и услуг, предоставляемых сетевыми элементами, соответственно.

Предположим, что все ограничивающие функции процессов поступления трафика и процессов обслуживания в узлах сети равны e^{-x} . Параметры $\theta = \theta' = \theta'' = 1$.

В CN предполагаем, что установлены следующие параметры: есть два маршрутизатора, каждый с пропускной способностью 100 Мбит/с, и все сквозные потоки пересекают оба маршрутизатора. Скорость поступления сквозных потоков RT (NRT) составляет 200 (100) пак/с, а размер каждого пакета составляет 128 (256) байт. Скорость поступления перекрестных потоков составляет 200 пак/с, а размер каждого пакета составляет 128 байт. Максимальный размер пакета каждого потока составляет 100 Кбайт. Количество сквозных (перекрестных) потоков RT и сквозных (перекрестных) потоков NRT составляет 100 (50).

В сети доступа AN предполагаем, что установлены соответствующие параметры: система MIMO с 2×2 антеннами, использующими каналы шириной 20 МГц. Соотношение сигнал/шум составляет 25 дБ. Скорость нисходящей линии связи может достигать 100 Мбит/с; вероятности переходов канала из «хорошего» состояния в «плохое» и наоборот равны соответственно $p_{gb} = 0,01$ и $p_{bg} = 0,1$. Скорость изменения каждого состояния определяется в соответствии с канальной матрицей. Учитывая указанные выше параметры AN, можно рассчитать пропускную способность канала, которая будет равна 90,909 Мбит/с.

Для приведенного выше набора сетевых параметров можно получить границы стохастической задержки для потоков S_{1t} и S_{2t} . На рисунке 3 изображены вероятности нарушения верхних границ стохастической задержки E2E для S_{1t} и S_{2t} . Из рисунка 3 видно, что вероятность нарушения уменьшается при увеличении стохастической задержки. Это следует из того, что SNC оперирует с ограничивающими функциями. Из рисунка 3 также видно, что стохастическая задержка E2E трафика RT меньше, чем задержка трафика NRT при той же вероятности нарушения границ.

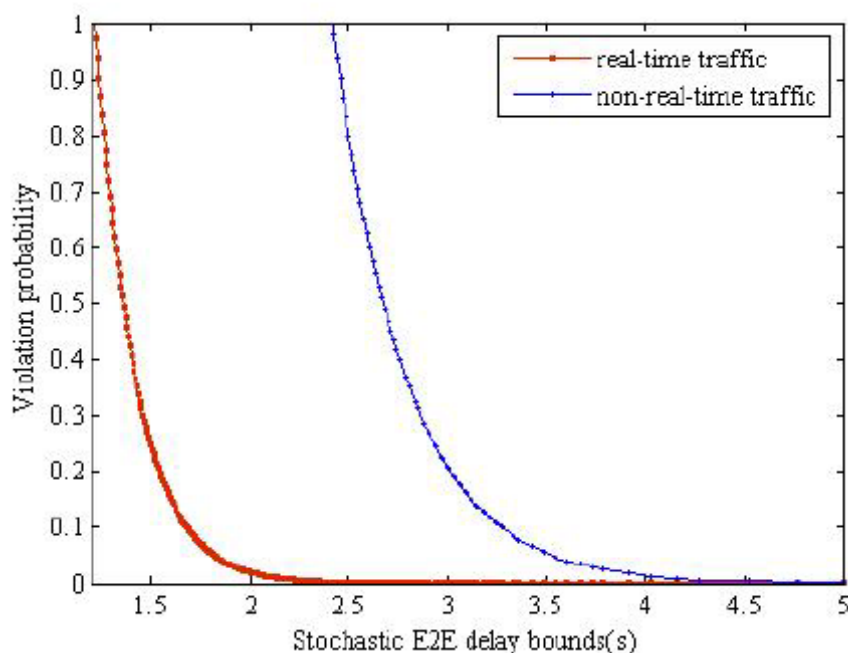


Рисунок 3 – График зависимости вероятности нарушения от стохастической задержки

Полученный результат демонстрирует, что предлагаемая структура сети LTE-SAE может дифференцировать трафик RT и трафик NRT и обеспечивает определенные стохастические гарантии для различного сквозного трафика. Следовательно, можно сделать вывод, что предложенная структура эффективна для анализа стохастических границ задержки E2E для трафика RT и трафика NRT.

Библиографический список:

1. Росляков А.В., Лыников А.А., Витевский В.Д. Сетевое исчисление. Часть 1. Теоретические основы // Инфокоммуникационные технологии. – 2018. – Т.16, №1. – С. 19-33.
2. Росляков А.В., Лыников А.А. Сетевое исчисление (Network Calculus) и его применение для оценки сетевых характеристик. – Самара, ПГУТИ, 2019. – 222 с.
3. Степутин А.Н., Николаев А.Д. Мобильная связь на пути 6G / в 2 томах. – М.: Инфра-Инженерия, 2107.
4. Zhang L., Chen X., Xiang X, Wan J. A Stochastic Network Calculus Approach For the End-to-End Delay Analysis of LTE Networks // International Conference on Selected Topics in Mobile and Wireless Networking (iCOST), 2011, pp. 30-35.

СЕКЦИЯ 7

Информационные технологии в образовании



УДК 004.418

В.В. Евсеева, преподаватель*ГБОУПО «Севастопольский торгово-экономический техникум»*

ОСОБЕННОСТИ ОРГАНИЗАЦИИ ДИСТАНЦИОННОГО ОБУЧЕНИЯ В СИСТЕМЕ MOODLE ДЛЯ ПРОФЕССИОНАЛЬНОГО ОБУЧЕНИЯ

Аннотация: Одной из основных особенностей организации дистанционного обучения является использование системы Moodle, которая способствует развитию критического мышления и творчества у учащихся, позволяя связывать теоретическое обучение с практикой, рефлексией и исследовательской деятельностью. В данной статье рассматриваются особенности использования дистанционных технологий, которые все шире входят в практику современного профессионального образования. Описаны возможности и особенности организации обучения студентов с применением системы дистанционного обучения Moodle. Показано, что высокая эффективность организации образовательного процесса достигается при сочетании различных форм дистанционного обучения с традиционными формами обучения.

Ключевые слова: дистанционное обучение, система дистанционного обучения, Moodle, профессиональное образование, современные стратегии обучения

Annotation: The main feature of distance learning programs is the use of an asynchronous platform system (Moodle), which promotes learner's critical thinking and creativity by linking learning to practice, reflection, case study and exploration. In addition, the use of an online platform (Moodle) promotes learner's individualized learning pathway, as well as the ability of participants to interact, principles that are compatible with the critical thinking that governs the principles of adult education. The article examines the features of distance technologies, which are becoming an integral part of education; offers technologies for the transition of a higher school to distance learning; forecasts possible outcomes. The description is given to possible organization of distance learning system Moodle. The article shows that the high efficiency of the educational process is achieved through the combination of various forms of distance learning with traditional forms of learning.

Key words: distance learning, distance technologies, Moodle, professional education, modern teaching strategies

Бурное развитие информационных технологий в последние годы затронуло многие аспекты современной жизни, включая преподавательскую деятельность. Это позволило компенсировать недостатки традиционного метода обучения за счет использования мультимедийных технологий.

Современные IT-технологии помогают не только увеличить вовлечение студентов в учебный процесс, но и позволяют усовершенствовать сам процесс приобретения знаний учениками.

Дистанционное обучение означает, что студент и преподаватель могут взаимодействовать, используя сеть Интернет. Благодаря такому подходу, преподаватель может изменить ход обучения в зависимости от прогресса учащегося. Таким образом, учебный процесс будет развиваться, как если бы это было взаимодействие с преподавателем один на один.

Дистанционное обучение имеет значительные преимущества над стандартными методиками, позволяя проводить обучение большего количества человек, при этом повышая качество и результативность за счет применения современных технологий. Все это позволяет создать единую образовательную среду, автоматизируя рабочий процесс.

Многие организации не располагают ресурсами для найма сотрудников, работающих на полную ставку или для приобретения необходимого оборудования, соответствующего требованиям образовательных стандартов. Внедрение платформы дистанционного обучения в учебный процесс позволяет решить эту проблему, позволяя проводить интерактивные эксперименты в симуляторах, экономя затрачиваемые ресурсы на проведение необходимых опытов.

Сделать виртуальную среду обучения по-настоящему интерактивной и привлекательной - серьезная практическая задача, с которой сталкиваются учебные заведения. Это важно с точки зрения поиска путей и средств улучшения качества обучения. Решение данной задачи требует от преподавателя не только творческого мышления и нового подхода к трансформации методов обучения, но и наличия необходимых навыков и

рекомендаций по созданию интерактивной виртуальной учебной среды. Администрация учебного заведения должна создать всеобъемлющую основу для оптимального использования виртуальной среды.

Однако, несмотря на преимущества, такой вид обучения имеет существенные недостатки. Из-за отсутствия живого контакта с преподавателем, студент должен обладать жесткой самодисциплиной. Также, несмотря на повсеместную компьютеризацию, не все студенты могут иметь необходимое техническое обеспечение.

Появление открытых систем обучения, таких как Moodle, позволяет небольшим учебным организациям адаптировать учебный процесс в условиях ограниченности учебных аудиторий. Это особенно критично в организациях, где всего несколько аудиторий, обеспеченных необходимым оборудованием.

LMS Moodle – это аббревиатура Modular Object Oriented Dynamic Learning Environment. Это бесплатное программное обеспечение с открытым исходным кодом – система управления обучением. Moodle был разработан, чтобы помочь преподавателям и учащимся разрабатывать элементы курсов дистанционного обучения. Открытый исходный код данной системы позволяет каждому желающему скачать Moodle, чтобы создать на ее базе свою собственную образовательную платформу [1].

Moodle является современной, развивающейся платформой, состоящей из множества модулей: Чат, Опрос, Форум, Глоссарий, Задание и т.д. [2].

Для обучения с помощью системы Moodle ученику необходимо иметь:

- персональный компьютер с доступом в сеть Internet (в некоторых случаях мобильный телефон);
- базовые навыки работы с электронной почтой, сетью Internet и офисными программами (для просмотра и подготовки текста и т.д.).

Система Moodle привлекательна для использования в учебном процессе в динамичных и быстро развивающихся дисциплинах, где важно дать студентам не только базовую информацию об объекте изучения, но и познакомиться с последними публикациями и свежими мнениями экспертов по изучаемому предмету. Использование Moodle позволяет широко использовать тестирование, осуществлять предварительную сдачу контрольных и лабораторных работ. Кроме того, посредством форумов или личных сообщений студенты могут получить консультацию преподавателя по интересующим их вопросам. Такой подход очень удобен и для студентов, т.к. позволяет им значительно экономить время и силы и получать более индивидуальный подход при обучении.

Несмотря на основные преимущества, Moodle имеет существенные недостатки. Данная LMS требует поддержки и сопровождения, что требует специфических знаний, которыми не всегда обладают преподаватели. Также, для ведения полноценного курса необходимы большие затраты времени на подготовку материалов и проверку работ (кроме тестов).

Начиная с 2017 года, в рамках дисциплин «Элементы математической логики» и «Теория вероятности и математическая статистика» в ГБОУПО «Севастопольский торгово-экономический техникум» активно используется LMS Moodle (рисунок 1). Для этого был куплен хостинг и домен (ваш-учитель.севастополь.рф) и на нем развернут сервер. В процессе использования данной системы отмечено, что проведение тестирования студентов на персональных мобильных телефонах значительно упрощает и сокращает время проверки работ. Сохранение материалов на сайте позволяет студентам, пропустившим занятия, а также находящимся на индивидуальном обучении не пропускать занятия и осваивать материал наравне с остальными. Также, учитывая ограниченность и большую загруженность компьютерных классов в учебном учреждении, использование Moodle позволяет за счет обширного функционала полноценно проводить учебный процесс дистанционно.

Комбинаторика

-  Лекция 1. Введение. Перестановки
-  Практическая работа №1. Решение задач на расчёт количества выборов
-  Тест 1
-  Лекция 2. Упорядоченные выборы. Размещения
-  Лекция 3. Размещения и сочетания с повторениями и без них
-  Практическая работа №2
-  Практическая работа №3
-  Схема решения задач



Рисунок 1 – Раздел комбинаторики в дистанционном курсе по теории вероятности и математической статистике

Для оценки эффективности применения LMS Moodle был проведен анкетный опрос. Анкета, предложенная студентам, была создана с помощью приложения Google Формы. Анкетирование прошли 64 студента специальности «Информационные системы (по отраслям)» ГБОУПО «Севастопольский торгово-экономический техникум». Основной целью опроса было изучение восприятия взаимодействия студентов с LMS Moodle.

Большинство студентов, заполнивших анкету, являются студентами второго курса (39%). Студенты третьего курса составили 33%, а четвертого – 28% (рисунок 2).



Рисунок 2 – Распределение участников опроса по годам обучения

Отвечая на вопрос о том, в какой степени использование платформы повышает эффективность процесса обучения, большинство студентов дали положительный ответ (85%), тогда как 3% ответили отрицательно, а 2% не смогли ответить на поставленный вопрос (рисунок 3).

В ходе опроса студенты выделили некоторые из преимуществ использования LMS Moodle:

- доступность материалов курса в любое время и с любого электронного устройства, обладающего доступом к сети Интернет;
- отсутствие проблем приобретения учебных материалов и пособий;
- возможность сотрудничества с другими студентами при выполнении домашних заданий;
- возможность связи с преподавателем напрямую;
- онлайн-оценивание более объективно, чем традиционное.

Особо стоит отметить удобство мониторинга учебной деятельности студентов. LMS Moodle позволяет накапливать, обобщать и систематизировать информацию о работах студентов, таким образом, создавая портфолио каждого обучающегося. Это позволяет преподавателю создавать индивидуальные учебные траектории для каждого студента, корректируя проблемы в обучении и развивая навыки самостоятельной работы учащегося.

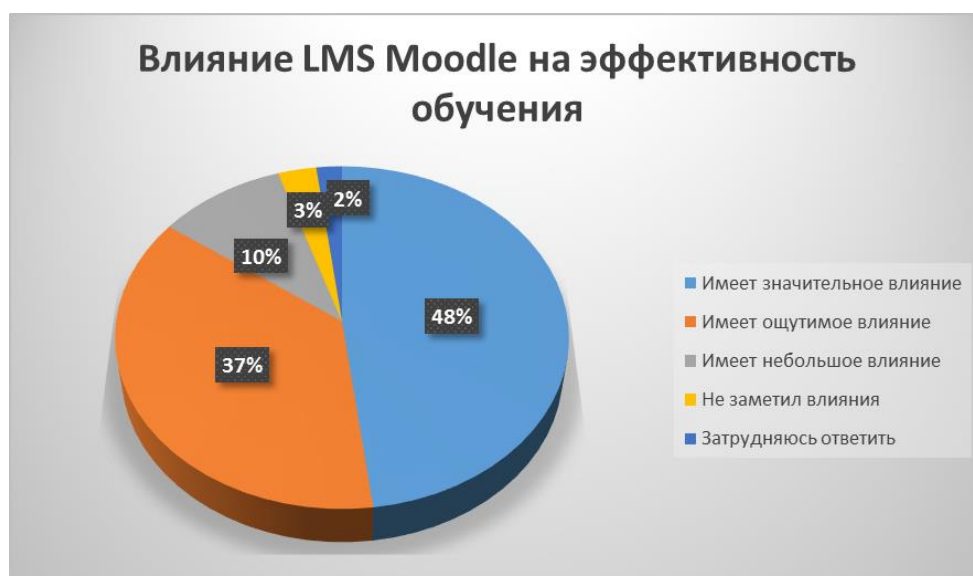


Рисунок 3 – Оказываемое влияние использования платформы Moodle на эффективность процесса обучения

Использование LMS Moodle в совокупности с другими сервисами позволяет сделать учебный процесс более интерактивным, что активно поддерживает последний федеральный государственный образовательный стандарт. Также активное внедрение дистанционного обучения позволяет решить проблему реализации государственной политики, требующей равенства и доступности образования при различных стартовых возможностях.

Разработанные автором курсы успешно используются в учебном процессе как отдельные дистанционные модули. Организация образовательного процесса и самостоятельной работы на основе платформы Moodle вызывает интерес у студентов и является важным дополнением к традиционным формам обучения, обеспечивая оптимальность формирования профессиональных компетенций студентов.

Библиографический список:

1. Андреев А.В., Андреева С.В., Доценко И.Б. Практика электронного обучения с использованием Moodle // Таганрог: ТТИ ЮФУ, 2008. – 146 с.
2. Кравченко Г.В., Волженина Н.В. Работа в системе Moodle: руководство пользователя: учебное пособие // Барнаул, 2012. – 114 с.
3. Реутовы Е.А. Применение активных и интерактивных методов обучения в образовательном процессе // Новосибирск: Изд-во НГАУ, 2012. – 58 с.
4. Смирнова Н.В. Использование системы дистанционного обучения Moodle в учебном процессе техникума [Электронный ресурс] / Костромской энергетический техникум имени Ф.В. Чижова. – Режим доступа: <http://spo-ket.ru/node/284> (дата обращения: 21.07.2019).

УДК 004.912; 519.254

П.А. Козлов, бакалавр;

А.С. Мохов, канд. техн. наук, старший преподаватель

В.О. Толчеев, докт. техн. наук, профессор

Национальный исследовательский университет «Московский энергетический институт»

ВЫЯВЛЕНИЕ ТЕМАТИК КАФЕДРАЛЬНЫХ ПУБЛИКАЦИЙ СРЕДСТВАМИ TEXT MINING

Аннотация: В статье рассматриваются способы выявления направлений исследований научного коллектива на основе анализа публикаций средствами Text Mining. Решение поставленной задачи демонстрируется на примере кафедры управления и информатики НИУ «МЭИ». На базе выборки, полученной из цифровой библиотеки eLibrary.ru, строятся терминологические профили сотрудников. С помощью применения косинусного сходства формируются группы специалистов, проводящие исследования по общим темам. Полученные результаты сравниваются с экспертными оценками тематик кафедральных НИР, анализируются причины имеющихся расхождений. Обосновывается выбор программных средств, использованных для проведения исследований.

Ключевые слова: обработка и анализ текстовых данных, лемматизация, информативные признаки, терминологический профиль, кластеризация, косинусное сходство, облака слов.

Annotation: In the article we discuss ways to identify directions of researches of the scientific group on the base of the analysis of publications using Text Mining techniques. The solution of the problem is demonstrated by the example of the department "Control and Informatics" of Moscow Power Engineering Institute. We form a sample of lecturer's publications from eLibrary.ru and build terminological profiles of them. After that, we use cosine similarity to form the groups of specialists conducting research in one direction. The results are compared with expert assessments of the topics of department's research. Then we analyze the reasons of the differences.

Key words: text mining, lemmatization, informative features, terminological profile, clustering, cosine similarity, word clouds.

Одной из основных задач, которые эффективно решаются средствами Text Mining, является задача кластеризации данных, т.е. разделение исходного текстового массива на группы связанных (тематически близких) документов, имеющих достаточно много общих терминов и близкое смысловое значение. Интерпретируемость и полезность получаемых результатов, во многом определяется выбором информативных слов, описывающих текст, и меры близости между документами.

Сделаем формальную постановку задачи [1,2]: имеется множество (русскоязычных и англоязычных) публикаций преподавателей кафедры – $\{X\}$ и набор тематик (научных направлений, рубрик), которым эти публикации соответствуют $\{Q_1, \dots, Q_g, \dots, Q_G\}$. Причем количество тематик (и их названия) заранее неизвестны.

Из множества $\{X\}$ формируется выборка $\{X_1, X_2, \dots, X_N\}$, которая состоит только из русскоязычных публикаций, содержащихся в электронной библиотеке eLibrary.ru. Все документы представлены в библиографическом виде – задаются кортежем $\langle T, A, K \rangle$ (T – название, A – аннотация, K – ключевые слова). В качестве исходной системы признаков для проводимых исследований используются ключевые слова.

Необходимо выявить в выборке $\{X_1, X_2, \dots, X_N\}$ тематики $\{Q_1, \dots, Q_g, \dots, Q_G\}$.

Выявление тематик можно осуществить двумя способами:

- кластеризация исходной выборки и определение групп терминологически близких публикаций (т.е. получаем кластеры, содержащие публикации);
- построение терминологических профилей преподавателей и объединение их в группы на основе близости между профилями (т.е. получаем кластеры, содержащие авторов – преподавателей кафедры).

Обычно в задачах Text Mining обнаружение кластеров относится к разведочному анализу данных, т.е. необходимо получить отсутствующую информацию о выборке, нарастить априорные знания. Однако в нашем случае группировка проводится для

документального массива, для которого уже имеются достаточно полные сведения. В частности, известно какие курсы читают преподаватели, кто из них находится в соотношении «научный руководитель – аспирант», какие гранты и НИР выполняются на кафедре.

В данной работе проводится выявление тематик кафедральных публикаций с помощью построения терминологических профилей преподавателей и определения степени близости между ними. Под терминологическим профилем преподавателя $\{Y_j\}$ понимается вектор частот терминов, встречающихся в его публикациях.

$$Y_j = \begin{bmatrix} y_1^{(j)} \\ \dots \\ y_n^{(j)} \end{bmatrix}, \quad (j=1, \dots, N, N=13). \quad (1)$$

Для построения профиля формируется выборка, удаляются стоп-слова и слова, встречающиеся во всей выборке только один раз, проводится лемматизация, рассчитывается вес термина как частота встречаемости информативного слова из выборки в публикациях автора. Размерность всех профилей преподавателей одинакова и равна n – числу информативных терминов в выборке ($n = 347$).

В качестве меры близости между профилями используется косинусное сходство (*Cosine similarity*), которое рассчитывается по формуле [1,3]:

$$\cos \alpha = \frac{(Y_j, Y_k)}{|Y_j| * |Y_k|} = \frac{\sum_{i=1}^n y_i^{(j)} * y_i^{(k)}}{\sqrt{\sum_{i=1}^n (y_i^{(j)})^2} * \sqrt{\sum_{i=1}^n (y_i^{(k)})^2}} \quad (2)$$

В вышеприведенной формуле использованы следующие обозначения: $y_i^{(j)}$ – частота встречаемости i -го слова в профиле j -ого преподавателя, $y_i^{(k)}$ – частота встречаемости i -го слова в профиле k -ого преподавателя ($j = 1, \dots, N, k = 1, \dots, N$).

Дальнейшие исследования проводятся на текстовом массиве публикаций преподавателей кафедры Управления и информатики (УиИ) НИУ «Московский энергетический институт». Выборка была получена из цифровой библиотеки eLibrary.ru. В качестве критерия для отбора было введено условие – в выборку включаются специалисты, которые имеют не менее пяти публикаций, проиндексированных в eLibrary.ru. Заданным критериям удовлетворяют 13 преподавателей кафедры.

К особенностям данной выборки относятся:

- используются только русскоязычные слова;
- используются слова, которые встречаются в выборке не менее двух раз.

Предварительный анализ выборки позволяет сделать следующие наблюдения:

- из суммарного числа научных трудов (387), содержащихся в eLibrary.ru за рассматриваемый период с 1991 по 2019 годы, наибольшее число статей было подготовлено сотрудниками кафедры в соавторстве (два автора опубликовало 104 статьи, три – 63, четыре – 43, пять – 29),
- пики публикационной активности (2007 г. – 24 статьи, 2010 г. – 30, 2017 г. – 39), как представляется, связаны с подготовкой и защитой на кафедре диссертаций, прежде всего работ на соискание степени доктора наук.

На сформированной выборке проводится построение терминологических профилей преподавателей и определяются тематики, по которым происходит группировка. Для этого составляется симметричная квадратная матрица «автор-автор» (размерность $N \times N$, $N=13$), на главной диагонали находятся единицы, остальные элементы показывают близость профилей преподавателей между собой. Расчет степени близости проводится по формуле косинусного

сходства. Номер строки – номер преподавателя, названия столбцов – инициалы преподавателя.

Таблица 1 - Значение косинусной меры между авторами

	А.Д.Н.	Б.А.В.	Б.А.А.	В.Д.В.	Е.В.Л.	К.О.С.	П.М.М.	С.Н.В.	Т.В.О.	Ф.Г.Ф.	Ф.Г.А.	Ш.В.А.	Я.Т.В.
1	1.0000	0.2427	0.0223	0.6194	0.1352	0.7374	0.2053	0.2222	0.1388	0.1356	0.1347	0.3804	0.2457
2	0.2427	1.0000	0.2490	0.2490	0.2673	0.2988	0.5885	0.2747	0.2481	0.2023	0.2273	0.3357	0.2544
3	0.0223	0.2490	1.0000	0.0299	0.0745	0.0291	0.1656	0.2155	0.3123	0.0620	0.0728	0.0445	0.0593
4	0.6194	0.2490	0.0299	1.0000	0.1655	0.6695	0.2209	0.1465	0.0861	0.1286	0.1233	0.3046	0.2199
5	0.1352	0.2673	0.0745	0.1655	1.0000	0.2301	0.2104	0.1983	0.0598	0.5442	0.1879	0.5164	0.3285
6	0.7374	0.2988	0.0291	0.6695	0.2301	1.0000	0.2988	0.2259	0.0970	0.2095	0.1655	0.3973	0.3204
7	0.2053	0.5885	0.1656	0.2209	0.2104	0.2988	1.0000	0.1372	0.1360	0.1280	0.2037	0.3967	0.2204
8	0.2222	0.2747	0.2155	0.1465	0.1983	0.2259	0.1372	1.0000	0.3576	0.2581	0.2029	0.1588	0.2493
9	0.1388	0.2481	0.3123	0.0861	0.0598	0.0970	0.1360	0.3576	1.0000	0.2059	0.1084	0.1717	0.0504
10	0.1356	0.2023	0.0620	0.1286	0.5442	0.2095	0.1280	0.2581	0.2059	1.0000	0.1759	0.3398	0.2473
11	0.1347	0.2273	0.0728	0.1233	0.1879	0.1655	0.2037	0.2029	0.1084	0.1759	1.0000	0.1300	0.2433
12	0.3804	0.3357	0.0445	0.3046	0.5164	0.3973	0.3967	0.1588	0.1717	0.3398	0.1300	1.0000	0.3180
13	0.2457	0.2544	0.0593	0.2199	0.3285	0.3204	0.2204	0.2493	0.0504	0.2473	0.2433	0.3180	1.0000

Анализ таблицы 1 показывает достаточно невысокую терминологическую близость между профилями преподавателей. Для выявления кластеров в качестве порога выберем значение 0,5. Из исходной выборки, состоящей из 13 преподавателей, выделяется три компактные группы сотрудников, хорошо интерпретируемых на основе имеющейся априорной информации.

Два полученных кластера можно отнести к категории «Научный руководитель – ученики». Терминологическая близость в этих группах сформировалась еще на стадии подготовки диссертаций и сотрудничество продолжилось в дальнейшем при выполнении совместных грантов. У всех авторов имеется значительное количество статей, написанных в соавторстве. Первый кластер содержит {А.Д.Н, В.Д.В. и К.О.С.} и охватывает достаточно широкий круг научных направлений: нечеткие системы, идентификация объектов управления, построение регуляторов, обработка электроретинограмм. Анализ терминологического состава группы позволяет выявить доминирующую тематику – «Нечеткие системы управления». Во втором кластере – {Е.В.Л. и Г.Ф.Ф.} основные исследования проводятся в области искусственных нейросетей применительно к системам управления, информационной безопасности и классификации. Таким образом, второй кластер соответствует тематике «Искусственные нейросети» (далее «Нейросети»).

Третий кластер, включающий {Б.А.В. и П.М.М.}, может быть отнесен к категории «Совместная НИР». Терминологическая близость (между профилями преподавателей) сложилась в ходе выполнения крупной НИР, осуществляемой в НИУ «Московский энергетический институт» по заданию Минобрнауки на долгосрочной основе. Публикации авторов относятся к тематике «Информационно-аналитические системы».

На рисунке 1 с помощью программы Word Cloud (<https://wordcloud.pro/ru>) иллюстрируется терминологический состав второго кластера ({Е.В.Л. и Г.Ф.Ф.}). Размер терминов определяется частотой их встречаемости в публикациях авторов, вошедших в данный кластер.

При использовании введенного порога возможно также расширить группу {Е.В.Л. и Г.Ф.Ф.} путем присоединения Ш.В.А. (за счет близости между терминологическими профилями Е.В.Л. и Ш.В.А.). Имеются аргументы за такое присоединение – часть работ Ш.В.А. затрагивает нейросетевую (нейро-нечеткую) тематику и может быть отнесена к кластеру «Нейросети». Однако, на наш взгляд, включать Ш.В.А. в данный кластер нецелесообразно. Во-первых, у Ш.В.А. не имеется совместных публикаций с Е.В.Л. и Г.Ф.Ф.

Во-вторых, публикации Ш.В.А. многодисциплинальны и соответствуют различным проблематикам (кибер-физические системы, интеллектуальные агенты, эксплуатационная надежность электрооборудования и др.). Это подтверждается достаточно высокими показателями близости (при пороге более 0,3) с восьмью из тринадцати сотрудников

кафедры (наиболее высокий показатель). Таким образом, статьи, связанные с нейросетевой тематикой, составляют лишь небольшую долю от общего количества публикаций автора (34 печатных труда). Принимая во внимание вышеизложенное, оставляем кластер «Нейросети» без изменений, состоящим из преподавателей, которые специализируются в этой области.

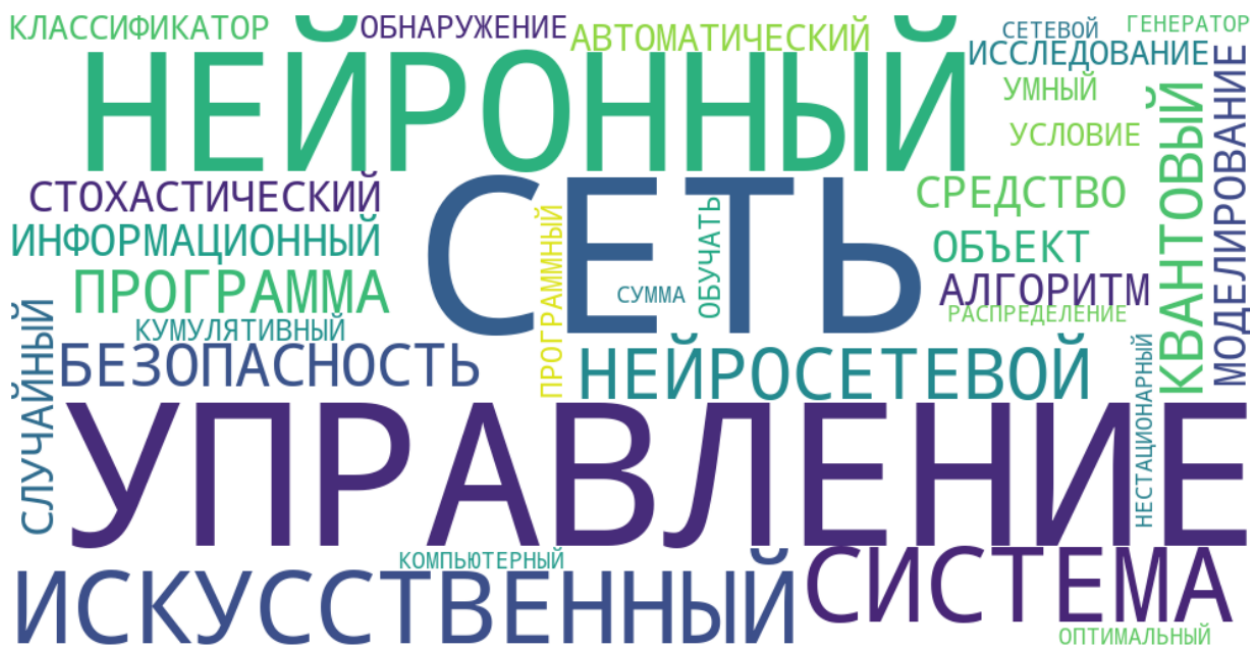


Рисунок 1 - Облако слов, характеризующих второй кластер

Для решения задачи исследования первоначально был использован язык C++, обеспечивающий высокую скорость работы программы и имеющий специализированную открытую библиотеку для проведения лемматизации слов (http://www.solarix.ru/for_developers/api/lemmatizator-api.shtml). Вместе с тем для расширения исследовательских возможностей, в частности применения мощных библиотек в области анализа текстовых данных и обучения классификаторов, окончательная версия программы была реализована с использованием языка Python 3 [4]. В нашей работе были задействованы следующие библиотеки: *rumorphy2* – лемматизатор слов, *wordcloud* – построение облаков слов, *matplotlib* – отображение построенных облаков слов, *numpy* – вычисления в списках. Результаты, получаемые с помощью программ на C++ и Python 3, практически совпадают, некоторые незначительные различия обусловлены по-разному работающими алгоритмами лемматизации.

На основе проведенного исследования можно сделать вывод о наличии на кафедре управления и информатики НИУ «Московский энергетический институт», как минимум, трех коллективов, проводящих исследования по общим тематикам: «Нечеткие системы управления», «Нейросети», «Информационно-аналитические системы». Вместе с тем полученные результаты свидетельствуют о достаточно широком спектре научных исследований сотрудников кафедры и определенной разрозненности работ (достаточно низкие показатели близости в таблице 1). Используя экспертные знания о специализации кафедральных преподавателей, следует отметить, что некоторые выявленные кластеры оказались меньше по своим размерам, чем предполагали эксперты. Эти несоответствия возникли из-за следующих причин:

- в наших экспериментах заданной системой признаков является множество ключевых слов, что ограничивает словарь задействованных терминов и может приводить к искажениям;

- количество статей у разных преподавателей существенно различается, как следствие, некоторые вектора (профили) оказываются сильно разреженными и не позволяют корректно рассчитать меру близости;
- многопрофильность ряда авторов затрудняет их отнесение к одному конкретному кластеру.

Для снижения числа несоответствий и возможных искажений в наших дальнейших исследованиях планируется расширить используемую систему признаков и использовать термины, встречающиеся в библиографических описаниях статей (т.е. во всех разделах кортежа $\langle T, A, K \rangle$). Кроме того, интерес представляет подтверждение полученных результатов с помощью применения «классических» методов кластерного анализа (например, метода k -средних, иерархического кластерного анализа) для получения групп тематически близких публикаций. Для решения проблемы многопрофильности автора (или многотематичности статей) имеет смысл использовать «мягкую» кластеризацию, предусматривающую отнесение авторов (или статей) сразу к нескольким кластерам.

Библиографический список

1. Флах П. Машинное обучение – Наука и искусство построения алгоритмов, которые извлекают знания из данных. М.: ДМК-пресс. 2015. – 400с.
2. Sebastiani F. Machine Learning in Automated Text Categorization. ACM Computing Survey. 34 (1), 2002. p. 1-47.
3. Корнеев В.В., Гареев А.Ф., Васютин С.В., Райх В.В. Базы данных. Интеллектуальная обработка информации. М.: Нолидж. 2000. – 352 с.
4. Мюллер А., Гвидо С. Введение в машинное обучение с помощью Python М.: 2017. – 393 с.

УДК 004.912; 519.254

С.Р. Маслихов, бакалавр

А.С. Мохов, канд. техн. наук, старший преподаватель

В.О. Толчеев, докт. техн. наук, профессор

Национальный исследовательский университет «Московский энергетический институт»

ПРИМЕНЕНИЕ ТЕХНОЛОГИЙ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДЛЯ ОЦЕНКИ СООТВЕТСТВИЯ НАУЧНОГО ПРОФИЛЯ КАФЕДРЫ И ТЕМАТИК ЛЕКЦИОННЫХ КУРСОВ

Аннотация: в статье предлагается метод выявления соответствия специализации (научного профиля) кафедры тематикам лекционных курсов, которые читаются преподавателями. Формируется выборка публикаций сотрудников из цифровой библиотеки eLibrary.ru и с помощью средств интеллектуального анализа текстовых данных изучается терминологическая близость между научными публикациями и программами учебных курсов. При обработке документов нами используется модель «мешок слов», проводится взвешивание терминов путем расчета частоты встречаемости, вычисляется косинусная мера близости между профилем кафедры и курсов. Предложенный метод иллюстрируется на примере кафедры управления и информатики НИУ «МЭИ».

Ключевые слова: обработка и анализ текстовых данных, модель «мешок слов», лемматизация, информативные признаки, дескрипторы, словарь терминов, косинусное сходство, облака слов.

Annotation: In this article we propose a method of identifying the compliance of specialization of the department with subjects of lecture courses. We form a sample of lecturer's publications from digital library (eLibrary.ru) and apply text mining tools to analyze the terminological proximity between scientific publications and lecture courses using the "bag of words" model and cosine similarity. Weight of each term equals to frequency of occurrence this term in document. Proposed method is illustrated on the example of the department "Control and Informatics" of Moscow Power Engineering Institute.

Key words: processing and analysis of text data, "bag of words", profile of department, profile of course, lemmatization, informative features, descriptors, cosine similarity, word clouds.

Технологии интеллектуального анализа текстовых данных (*Text Mining*) позволяют автоматизировать обработку неструктурированной документальной информации, выявляя важные закономерности, полезные для практического использования. В данной работе рассматриваются вопросы применения средств *Text Mining* для анализа деятельности кафедры, в частности, выявления связей между научной работой (публикациями сотрудников) и учебным процессом (лекционными курсами).

Сформулируем гипотезу исследования: на кафедре по тематикам читаемых дисциплин проводятся НИР, результаты которых публикуются в профильных журналах и трудах конференций. Таким образом, наша гипотеза предполагает наличие на кафедре взаимосвязи между научной и учебной работой.

Для определения силы связи будем использовать семантическую (терминологическую) близость между научным профилем кафедры и профилем учебного курса. Под профилями кафедры и учебного курса понимаются упорядоченные списки наиболее частотных терминов, которые встречаются в публикациях сотрудников кафедры (или учебных программах).

Конкретизируем постановку задачи: для кафедры задан список преподавателей $\{\bar{A}\}$ и перечень читаемых лекционных курсов $\{\bar{C}\}$. Необходимо сформировать две выборки, первая состоит из публикаций преподавателей, вторая – из электронных версий учебных программ курсов. Далее требуется описать текстовые документы в математическом виде и измерить степень близости между словарем терминов, который используется в научных публикациях, и информативными словами, наиболее часто встречающимися в учебных курсах.

Решение поставленной задачи состоит из следующих этапов.

1. Из электронной библиотеки eLibrary.ru формируется выборка размером N , которая включает научные работы преподавателей, имеющих не менее пяти статей в

eLibrary.ru. Каждая статья представляется в виде набора ключевых слов (такое представление реализовано в eLibrary.ru). Признаковое (терминологическое пространство), в котором анализируются документы, формируется из лемматизированных терминов (дескрипторов), которые остались в списке терминов после удаления «стоп-слов».

2. Из электронных версий учебных программ (множество $\{\bar{Y}\}$, размера T) составляется список терминов, которые входят в раздел 4.2. – "Краткое содержание программы". Признаковое пространство состоит из лемматизированных дескрипторов, которые остались в списке терминов после удаления «стоп-слов».

Для описания текстовых документов в данной работе используется модель «мешок слов» – неупорядоченный набор информативных терминов, не учитывающий наличие связей между ними. Вес термина в «мешке» определяется частотой слова (*term frequencies*):

$$x_{ij} = f_{ij}, \text{ где } f_{ij} - \text{частота слова } i \text{ в документе } j [1].$$

Профиль кафедры и профили курсов представляют собой векторы $X = \begin{bmatrix} x_1 \\ \dots \\ x_n \end{bmatrix}$ и $Y_t = \begin{bmatrix} y_1(t) \\ \dots \\ y_n(t) \end{bmatrix}$.

Здесь n – число всех уникальных дескрипторов в кафедральных публикациях и учебных программах курсов, $t = \overline{1, T}$. В большинстве случаев данные вектора сильно разрежены, ненулевые значения соответствуют только дескрипторам, встречающимся в профиле кафедры или конкретного курса.

3. Проводится оценка степени близости двух векторов (профилей). Для этого используется косинусная мера (*Cosine similarity*). Выбор косинусной меры обусловлен тем, что она является нормированной – изменяется в пределах $[0;1]$, где 1 соответствует абсолютно идентичным векторам (профилям). Таким образом, чем выше косинусная мера, тем больше тематика курса соответствует научной специализации кафедры. Кроме того, косинусная мера наиболее часто и эффективно используется для решения широкого круга задач обработки и анализа текстовых данных [1,2]. Расчет проводится по формуле [1,3]:

$$\cos(X, Y_t) = \frac{\sum_{i=1}^n x_i * y_t^i}{\sqrt{\sum_{i=1}^n (x_i)^2} * \sqrt{\sum_{i=1}^n (y_t^i)^2}} \quad (1)$$

где x_i – частота i -го термина в профиле кафедры, y_t^i – частота i -го термина в профиле t -го курса, $i = \overline{1, n}$, $t = \overline{1, T}$.

Проверим сформулированную гипотезу для кафедры управления и информатики (УиИ) НИУ «Московский энергетический институт».

Среди преподавателей кафедры 13 человек зарегистрированы в электронной библиотеке eLibrary.ru и имеют более пяти публикаций. Выборка для построения профиля кафедры содержит 387 печатных трудов, начиная с 1991 года. Полученный терминологический профиль кафедры иллюстрируется на рисунке 1 с помощью программы «Облако слов».

В полученном профиле имеются термины, представляющие общенаучную лексику (метод, модель, анализ), специализированные термины, отражающие направления научных исследований кафедры (управление, нечеткий, идентификация, регулятор, классификация). Отметим, что размер изображения термина на приведенном рисунке определяется его весом, т.е. частотой встречаемости в выборке, наибольшие веса оказались у терминов: «система», «управление», «модель», «нечеткий», «анализ».



В качестве курсов для построения терминологических профилей выбраны 9

- 1) Теория автоматического управления (ТАУ).
- 2) Методы оптимизации (МО).
- 3) Методы обработки данных (МОД).
- 4) Нейрокомпьютеры и их применение (НК).
- 5) Идентификация динамических объектов (ИДО).
- 6) Моделирование систем управления (МСУ).
- 7) Системы управления базами данных (СУБД).
- 8) Микроконтроллеры и микропроцессоры в системах управления (МК).
- 9) Информационные технологии (ИТ).
- 10) Нейро-нечеткие системы управления и диагностики (ННС).
- 11) Нечеткие алгоритмы управления (НАУ).
- 12) Системы поддержки принятия решений (СППР).

Таблица 1 - Косинусная мера между профилем кафедры УиИ и бакалаврскими курсами

	ТАУ	МО	МОД	НК	ИДО	МСУ	СУБД	МК	ИТ
УиИ	0.522	0.219	0.34	0.182	0.346	0.523	0.205	0.108	0.352

	ННС	НАУ	СППР
УИИ	0.465	0.356	0.3

косинусной меры близости более 0,7). Вместе с тем нами выявлено две дисциплины («Моделирование систем управления», «Теория автоматического управления»), для которых мера близости превосходит 0,5, т.е. имеется заметная связь между профилями. Для достаточно большой группы курсов обнаружена умеренная связь (не менее 0,3). К этим курсам относятся: «Нечеткие алгоритмы управления», «Нейро-нечеткие системы управления и диагностики», «Информационные технологии», «Идентификация динамических объектов», «Методы обработки данных», «Системы поддержки принятия решений». Ряд дисциплин («Системы управления базами данных», «Микроконтроллеры и микропроцессоры в системах управления», «Методы оптимизации», «Нейрокомпьютеры и их применение») оказались слабо связаны с научным профилем кафедры (значение косинусной меры меньше 0,3).

Отметим, что исследования, проведенные нами с помощью технологий *Text Mining*, достаточно хорошо согласуются с экспертными оценками. Однако, если некоторые из экспертных оценок в значительной степени совпадают с нашими результатами (специализация кафедры в области теории управления), то другие экспертные заключения не нашли существенного подтверждения (например, наличие заметной исследовательской и публикационной активности в области обработки и анализа данных).

Библиографический список

1. Маннинг К., Рагхаван П., Шютце Х. Введение в информационный поиск. – М.: Вильямс, 2014 – 528 с.
2. Мохов А.С., Исследование методов классификации двуязычных текстовых документов // Радиоэлектроника, электротехника и энергетика: Девятнадцатая Междунар. науч.-техн. конф. студентов и аспирантов: Тез. докл. в 4 т. Т. 2. М.: Издательский дом МЭИ, 2013. – 356с.
3. Елисеева И. И., Рукавишников В. О. Группировка, корреляция, распознавание образов: (статистические методы классификации и измерения связей). — М.: Статистика, 1977. — 143 с.
4. Мюллер А., Гвидо С. Введение в машинное обучение с помощью Python М.: 2017. – 393 с.

УДК 004.416

Ю.П. Николаева, старший преподаватель*Севастопольский государственный университет***ПРИМЕНЕНИЕ НЕЙРОННОЙ СЕТИ ДЛЯ АНАЛИЗА ПЕРСПЕКТИВ ПОСТУПЛЕНИЯ СТУДЕНТОВ В МАГИСТРАТУРУ**

Аннотация: в статье рассматривается возможность прогнозирования перспектив продолжения обучения студентов-бакалавров в магистратуре на основе решения задачи классификации при помощи нейронной сети. Проводится серия экспериментов для получения оптимальной структуры нейронной сети для повышения точности прогнозирования.

Ключевые слова: искусственные нейронные сети, машинное обучение, бинарная классификация, параметрическая оптимизация

Annotation: neuron network based method is proposed for students' educational prospective estimation. Series of experiments are conducted on network structure to provide optimal value of classification accuracy.

Key words: artificial neuron networks, machine learning, binary classification, parametric optimization

В настоящее время нейронные сети как инструмент анализа данных широко используются в различных областях, как в бизнесе, так и в социально-значимых проектах. Активно используются возможности нейронных сетей в том числе и в сфере образования. Решаются такие задачи, как анализ поведения студентов или абитуриентов в социальных сетях, прогнозирование успеваемости, формирование индивидуальных траекторий обучения, поддержка принятия решений в LMS (Learning Management Systems) и другие ([1], [2]).

Для выпускающей кафедры и учебного отдела может быть удобным иметь прогноз по перспективам поступления студентов в магистратуру. В данной статье будет рассмотрен метод решения задачи прогнозирования при помощи искусственной нейронной сети.

Постановка задачи. Рассмотрим задачу предсказания перспектив поступления студента в магистратуру. Подобная задача может быть рассмотрена как задача бинарной классификации на множестве студентов. Критерием оптимальности для классификации будем считать долю верно классифицированных данных. Будем рассматривать некоторое множество возможных классификаторов, порождаемых различными структурами используемой нейронной сети. Таким образом, возможно свести задачу оптимальной классификации к задаче параметрической оптимизации нейронной сети и поиску такого классификатора, который на заданном множестве исходных данных показывает максимальный результат в соответствии с критерием оптимальности. В качестве варьируемых параметров будут выступать характеристики представления исходных данных, структура элементов сети, параметры обучения сети.

Формат исходных данных: в качестве исходных данных для обучения нейронной сети взяты сведения о выпускниках, для которых имеется информация о результатах поступления в магистратуру.

Для каждого из 68 студентов есть данные об успеваемости за четыре года обучения в бакалавриате и о количестве научных публикаций. Общее количество предметов и оцениваемых видов работ – 75 (в соответствии с учебным планом по направлению 09.03.01 Информатика и вычислительная техника "Электронно-вычислительные машины, системы и сети"). Из этих предметов для обучения сети будем использовать подмножество из 37 предметов, для которых оценивание производилось по трёхбалльной системе, т.к. оценки по предметам, для которых программой предусмотрен недифференцируемый зачет, совпадают для всех выпускников. Для дисциплин по выбору в исходных данных будет указан 0 в соответствующей графе, если студент не изучал данный курс. В общем случае оценка представлена целым числом из множества $\{0, 3, 4, 5\}$

Для каждого студента имеются данные о количестве научных публикаций: некоторое целое число, равное 0 в том случае, если студент не занимался научной работой.

Факт поступления студента в магистратуру будет отражаться целым числом из множества $\{0, 1\}$, где 1 означает поступление.

Целевая функция. Будет использоваться обучение «с учителем» на предварительно размеченных данных. Зададим модель в терминах обучения с учителем.

Наблюдаемые величины – вектор x . В рассматриваемой модели вектор x – некоторое представление данных об успеваемости и научной работе студента.

Целевые переменные – вектор y . Вектор y содержит данные о фактическом поступлении студента.

Предсказания – $\tilde{y}$. Вектор $\tilde{y}$ содержит результат предсказания нейронной сетью факта поступления, $\tilde{y}$ можно рассматривать как вероятность принадлежности к одному из классов бинарного множества $\{0, 1\}$. Преобразуем вероятность в дискретные классы следующим образом. Будем считать, что для верно классифицированного элемента при некотором малом ε выполняется формула

$$|\tilde{y} - y| < \varepsilon \quad (1)$$

Пусть k – количество элементов выборки, для которого выполняется неравенство (1), K – объем всей выборки. Тогда целевой функцией будем считать долю верно классифицированных элементов выборки:

$$F = \frac{k}{K} \rightarrow \max, \quad (2)$$

ε далее будем называть «погрешностью классификации» или просто «погрешностью», а значение функции F – «точностью классификации» или «точностью».

Варьируемые параметры и группы параметров в рассматриваемой модели следующие:

- Представление исходных данных для модели (нормировка, сочетание различных групп смысловых входных данных);
- Структура сети (количество слоев, количество скрытых нейронов в каждом слое);
- Функции активации слоев нейронной сети (ReLU, Tanh, Sigmoid) [3];
- Оптимизатор нейронной сети (Adam, SGD и др.) [4];
- Схема использования данных при обучении сети (объем тестовой выборки, скорость обучения (learning rate)).

Варьирование всех уровней изменяемых параметров искусственной нейронной сети в рамках одного полнофакторного эксперимента достаточно трудоемко, поэтому разобьем эксперименты на несколько этапов. Далее этапы экспериментов и соответствующие факторы будем нумеровать римскими цифрами.

Выделим следующие этапы экспериментирования:

- I. Базовый эксперимент по обучению сети.
- II. Формирование представления модельных данных.
- III. Выбор структуры искусственной нейронной сети. Изменение количества слоев и скрытых нейронов.
- IV. Выбор параметров оптимизатора и функции активации.
- V. Выбор параметров обучающей выборки.

Для оценки качества обучения будет использована BCELoss – функция потерь на основе бинарной перекрестной энтропии (binary cross-entropy, BCE) с функцией активации выходного слоя типа Sigmoid.

I. Базовый эксперимент по обучению сети. В базовом эксперименте по обучению сети будем использовать полносвязную нейронную сеть с некоторой случайной комбинацией параметров.

Входные данные не нормированы, используются сведения только об успеваемости, без учета научной работы. Сеть имеет 37 входов по количеству предметов в исходных данных об успеваемости.

Глубина сети – 4 слоя, в каждом слое по 10 скрытых нейронов. Функция активации ReLU. Оптимизатор типа Adam. Количество эпох обучения – 1000, скорость обучения (learning rate) - 10^{-3} . Набор данных разбит на тестовую и обучающую выборку, объем тестовой – 30%.

Сеть реализована при помощи библиотеки PyTorch, использующей динамические графы вычислений.

При значении погрешности $\varepsilon = 0.1$, значение точности классификации составляет $F = 0.48$, что говорит о достаточно низком качестве этой модели со случайным подбором значений параметров.

На рисунке 1 проиллюстрирован результат обучения сети для рассмотренной модели. Видно, что модель гораздо лучше определяет принадлежность к классу $\{0\}$, чем к классу $\{1\}$. Это обусловлено, в том числе, и тем, что данные несбалансированы и студентов, поступивших в магистратуру, меньше, чем остальных.

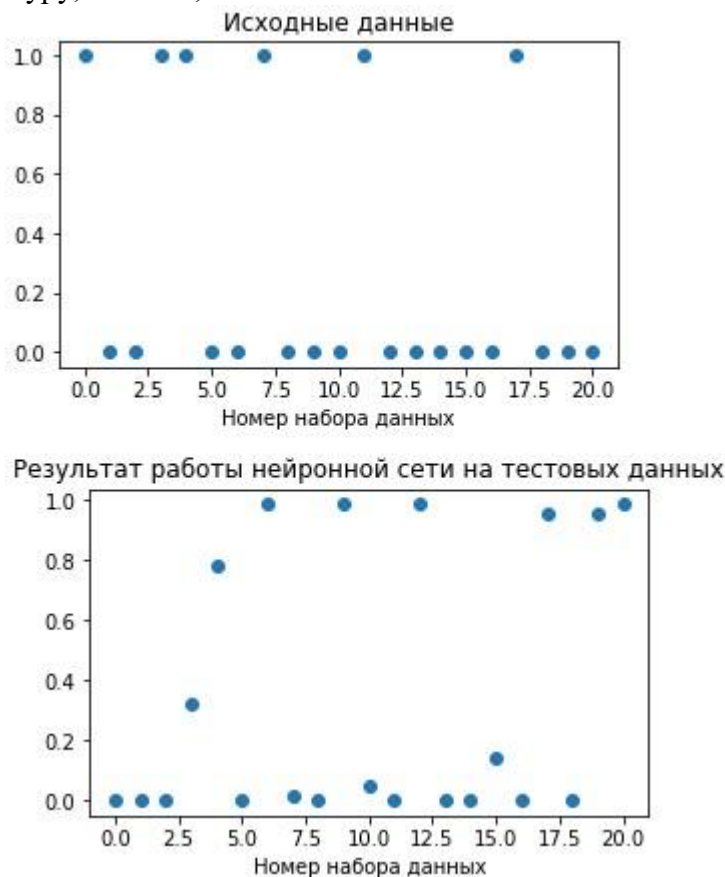


Рисунок 1 – Исходные данные (целевые переменные y) и результат работы сети (предсказания $\hat{y}$) для модели этапа I

II. Формирование представления модельных данных. В качестве факторов эксперимента выберем следующие параметры модели:

- Нормировка входных данных (ФИ.1).
- Состав данных (ФИ.2).
- Уровень погрешности (ФИ.3).

Для задания уровней фактора FII.1 используются формулы нормировки (3) и (4), где $\bar{p}$ и $\sigma(p)$ – математическое ожидание и среднее квадратичное отклонение выборки:

$$p_i^* = \frac{p_i - \min(p)}{\max(p) - \min(p)} \quad (3)$$

$$p_i^* = \frac{p_i - \bar{p}}{\sigma(p)} \quad (4)$$

Факторы и их уровни приведены в таблице 1. Объем данных позволяет провести полный факторный эксперимент.

Таблица 1 – Варьируемые уровни факторов для этапа II

	L1	L2	L3
Нормировка данных (FII.1)	Формула (3) с минимумом	Формула (4) со средним	Без нормировки
Состав данных (FII.2)	Оценки и публикации	Оценки	Публикации
Уровень погрешности (FII.3)	0.1	0.01	-

Результаты эксперимента проиллюстрированы на рисунках 2 и 3.

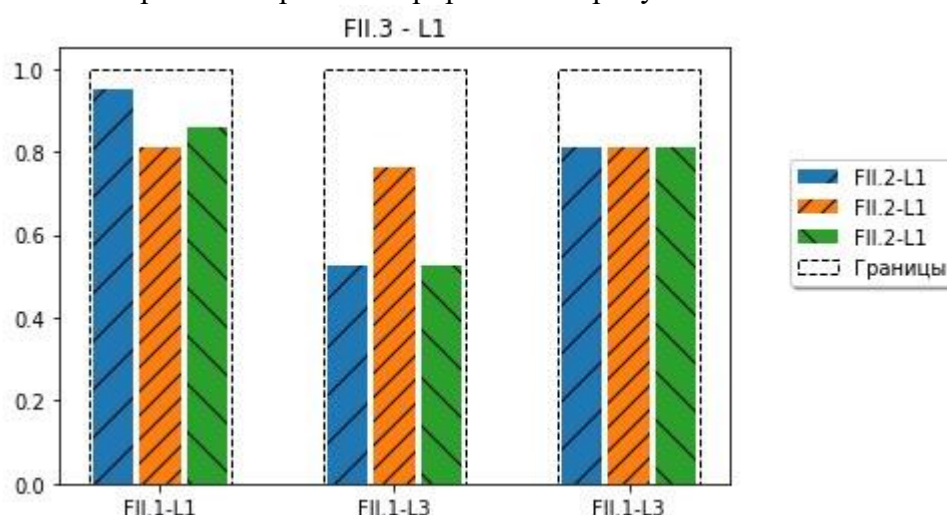


Рисунок 2 – Точность работы нейронной сети при уровне L1 фактора FII.3, т.е. при погрешности $\varepsilon=0.1$

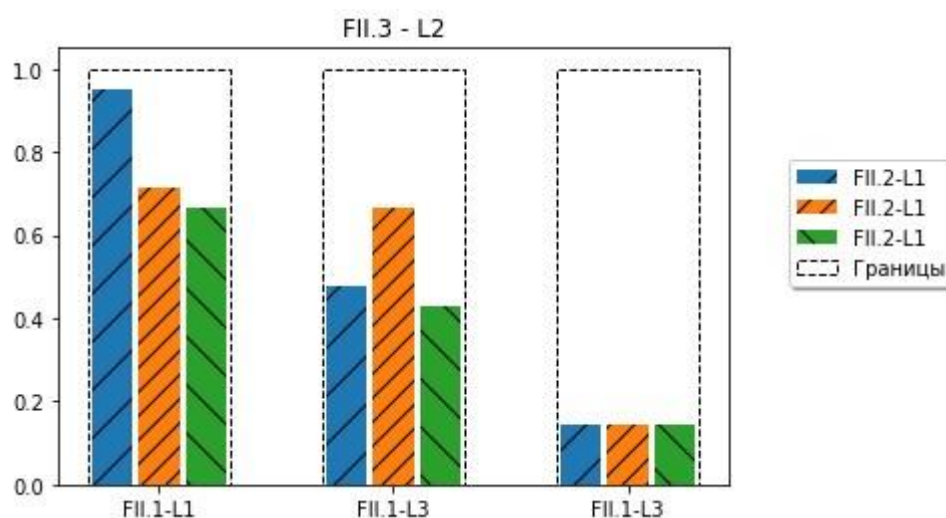


Рисунок 3 – Точность работы нейронной сети при уровне L2 фактора FII.3, т.е. при погрешности $\varepsilon=0.01$

В результате полного факторного эксперимента можно сделать следующие выводы:

- Уменьшение погрешности с $\varepsilon = 0.1$ (FII.3–L1, рисунок 2) до $\varepsilon = 0.01$ (FII.3–L2, рисунок 3) снижает точность работы сети для всех вариантов, кроме комбинации факторов (FII.1–L1 – FII.2–L1). Эта же комбинация дает самую высокую точность из представленных. В дальнейшем будем ее использовать в последующих экспериментах. То есть далее будем нормировать данные по формуле (3) и использовать таблицу-объединение 37 колонок с оценками по изученным дисциплинам и одной колонки с данными о количестве публикаций. Количество входов сети, таким образом, будет составлять 38.

Можно заметить также, что согласно рисунку 2 при FII.1–L3, т.е. без использования нормировки данных, сеть дает приемлемый результат при любом уровне фактора FII.2, т.е. при любом составе данных. Можно использовать этот вариант сети, когда не требуется низкое значение погрешности ε , но есть ограничение по вычислительным ресурсам, т.к. в этом случае можно ограничиться только данными о научных работах.

Далее будем использовать погрешность $\varepsilon = 0.1$, считая ее достаточной для большинства случаев использования сети.

III. Выбор структуры искусственной нейронной сети. Изменение количества слоев и скрытых нейронов.

Выберем следующие параметры в качестве факторов эксперимента для данного этапа:

- количество слоев нейронной сети (FIII.1). Зададим семь уровней фактора, L1 = 2, L3 = 3, ... L7 = 8. Максимальное количество рассматриваемых слоев зададим равным 8.
- количество нейронов в скрытом слое (FIII.2). Будем считать для простоты эксперимента, что все скрытые слои имеют равное количество нейронов. Зададим 18 уровней фактора, L1 = 3, L2 = 4, ... L18 = 20.

Результат полного факторного эксперимента представлен на рисунке 4. Правее и выше пунктирных линий находятся комбинации уровней факторов FIII.1 и FIII.2, при которых точность сети составляет более 0.9, что можно считать приемлемым результатом.

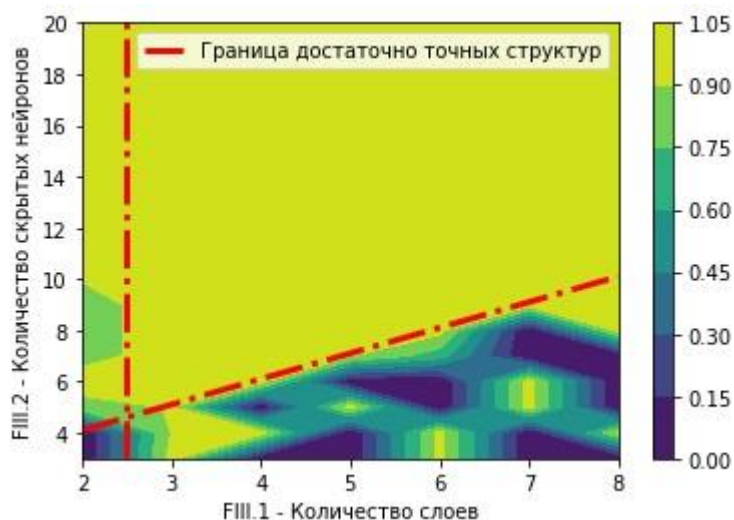


Рисунок 4 – Зависимость точности нейронной сети от количества слоев полносвязной сети (FIII.1) и количества скрытых нейронов (FIII.2)

Количество скрытых нейронов, равное 3 или 4, в большинстве случаев дает неприемлемый результат.

Далее будем использовать для экспериментов сеть с количеством слоев, равным 3 и 8 нейронами в скрытом слое. Для контроля используем еще одну точку – сеть с 8 слоями и 20

нейронами. Обе конфигурации дают точность 0.95, однако у первой меньше вычислительная сложность.

IV. Выбор параметров оптимизатора и функции активации. На данном этапе будем варьировать следующие факторы:

- FIV.1 – вариант оптимизатора сети.
- FIV.2 – функция активации нейронов скрытых слоев.

Возможные уровни факторов приведены в таблице 2. Представлены наиболее употребимые для обучения нейронных сетей варианты.

Результаты полнофакторного эксперимента представлены на рисунках 5 и 6.

Таблица 2 – Варьируемые уровни факторов для этапа IV

	L1	L2	L3	L4	L5
Оптимизатор (FIV.1)	Adam [5]	SGD, стохастический градиентный спуск	Adagrad [6]	RMSprop [7]	SGD-Momentum, стохастический градиентный спуск со значением момента 0.9 [8]
Функция активации (FIV.2)	ReLU, $ReLU(x) = \max(0, x)$	Sigmoid, сигмоидальная функция	Tanh, гиперболический тангенс	-	-

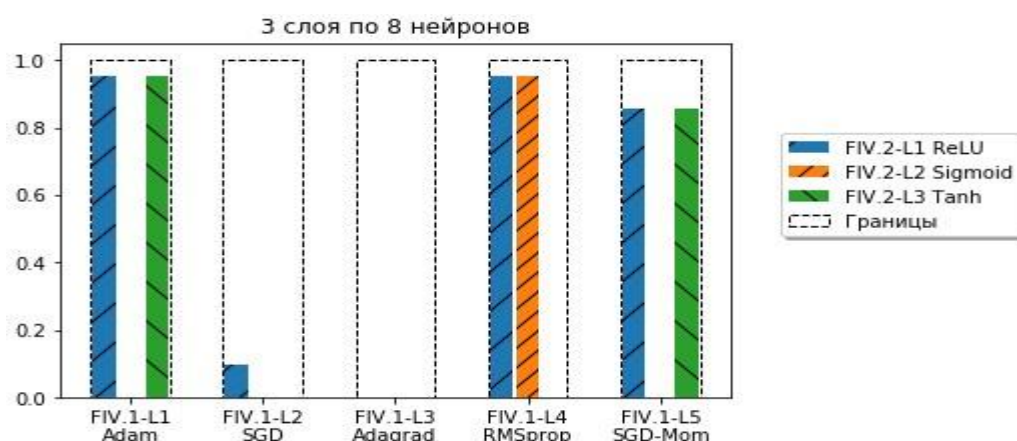


Рисунок 5 – Зависимость точности нейронной сети от типа оптимизатора (FIV.1) и функции активации (FIV.2). Сеть из 3 слоев по 8 нейронов

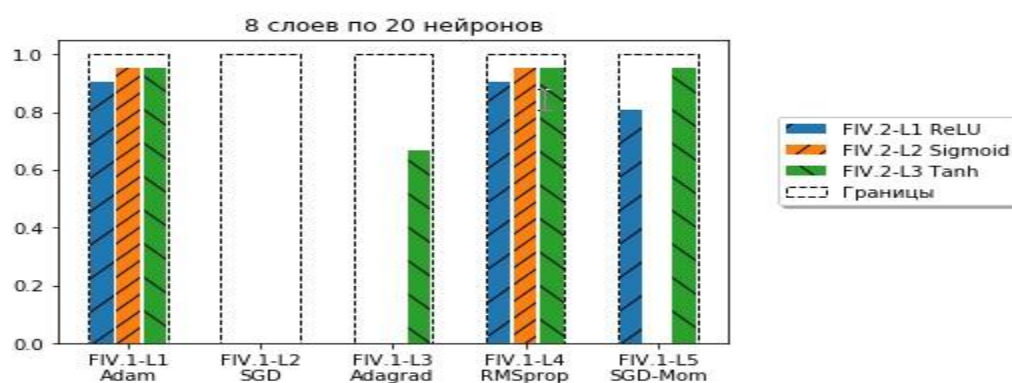


Рисунок 6 – Зависимость точности нейронной сети от типа оптимизатора (FIV.1) и функции активации (FIV.2). Сеть из 8 слоев по 20 нейронов

Выводы по эксперименту можно сделать следующие:

- оптимизатор SGD (FIV.1-L2) показывает неприемлемые (нулевые) результаты для двух выбранных структур сети. Использование SGD с инерцией (momentum) SGD (FIV.1-L2) существенно улучшает ситуацию, если использовать функцию активации ReLU (FIV.2-L1) или Tanh (FIV.2-L3).

- использование оптимизатора Adagrad (FIV.1-L3) также показывает неприемлемые результаты на имеющихся данных.

- для дальнейших экспериментов выберем вариант, показывающий хороший результат при меньшей сложности сети и не ухудшающий его с увеличением сложности. Из представленных комбинаций факторов этому требованию удовлетворяют две: (FIV.1-L1 - FIV.2-L3) Adam-Tanh, и (FIV.1-L4 - FIV.2-L2) RMSprop-Sigmoid. Отдадим предпочтение первому варианту, в соответствии с рекомендациями [9], где оптимизатор Adam указывается как наиболее универсальный.

V. Выбор параметров обучающей выборки. На заключительном этапе будем изменять параметры обучающих данных.

В качестве факторов эксперимента будут выбраны:

- скорости обучения learning rate (FV.1). Уровни фактора: L1 = 0.1, L2 = 0.01, L3 = 0.001, L4 = 0.0001

- доли тестовой выборки в общем объеме данных (FV.2). Уровни фактора: L1 = 0.1, L2 = 0.2, ... L9 = 0.9.

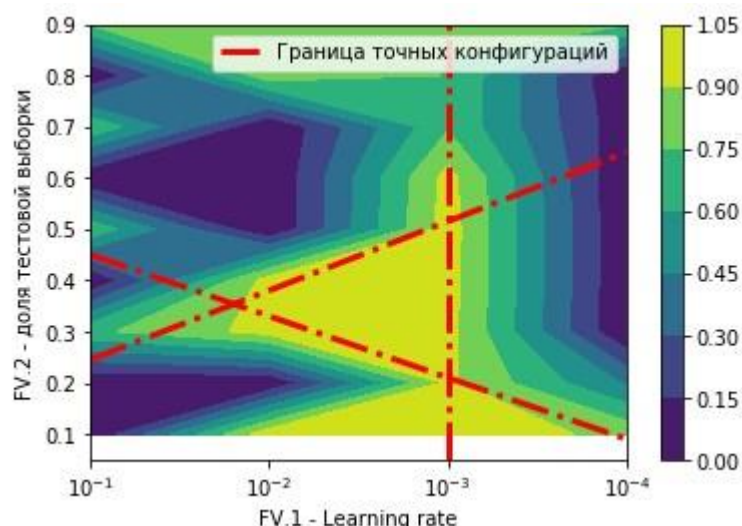


Рисунок 6 – Зависимость точности нейронной сети от скорости обучения learning rate (FV.1) и доли тестовой выборки в общем объеме данных (FV.2). Сеть из 3 слоев по 8 нейронов

На рисунке 6 проиллюстрированы результаты полнофакторного эксперимента. Пунктирные линии ограничивают область решений, для которых точность сети более 0.9. На имеющихся данных подтверждается общая рекомендация для искусственных нейронных сетей об использовании комбинации (FV.1-L3 – FV.2-L3), т.е. тестовой выборки объемом 0.3 от имеющихся данных при уровне скорости обучения (learning rate), равном 10^{-3} . Уровень точности при таком сочетании уровней факторов равен 0.95 на тестовой выборке.

Выводы. Проведенная серия экспериментов позволяет подобрать параметры искусственной нейронной сети для эффективного решения задачи о прогнозировании информации по поступлению студентов в магистратуру. В работе не были использованы библиотеки верхнего уровня, такие как ignite, так как исследовался процесс реагирования

модели на отдельные изменения в структуре нейронной сети. Дальнейшие направления исследований – использование других типов нейронных сетей и адаптивная оптимизация параметров в процессе обучения.

Библиографический список:

1. Nick Z. Zacharis. Predicting student academic performance in blended learning using artificial neural networks. URL: <http://www.aircconline.com/ijaia/V7N5/7516ijaia02.pdf>
2. Nurettin Yorek, Ilker Ugulu. A CFBN artificial neural network model for educational qualitative data analyses: Example of students' attitudes based on Kellerts' typologies. URL: <https://files.eric.ed.gov/fulltext/EJ1078385.pdf>
3. Vishnu Subramanian: Deep Learning with PyTorch. - Packt Publishing, 2018.
4. Eli Stevens, Luca Antiga. Deep Learning with PyTorch. Essential Excerpts. - Manning Publications, 2019.
5. Diederik P. Kingma, Jimmy Ba Adam: A Method for Stochastic Optimization URL: <https://arxiv.org/abs/1412.6980>
6. John Duchi, Elad Hazan, Yoram Singer Adaptive Subgradient Methods for Online Learning and Stochastic Optimization URL: <http://jmlr.org/papers/v12/duchi11a.html>
7. Alex Graves Generating Sequences With Recurrent Neural Networks URL: <https://arxiv.org/pdf/1308.0850v5.pdf>
8. Ilya Sutskever, James Martens, George Dahl, Geoffrey Hinton On the importance of initialization and momentum in deep learning URL: <https://www.cs.toronto.edu/~hinton/absps/momentum.pdf>
9. Макмахан Брайан, Рао Делип. Знакомство с PyTorch: глубокое обучение при обработке естественного языка. — СПб.: Питер, 2020.

УДК 004.558

Е.М.Шалимова, старший преподаватель*Севастопольский государственный университет***РАЗРАБОТКА ОБУЧАЮЩЕЙ СИСТЕМЫ ПО ДИСЦИПЛИНЕ «ОПЕРАЦИОННЫЕ СИСТЕМЫ»**

Аннотация: в настоящей работе рассматриваются принципы построения обучающих систем. Анализируются особенности систем для дисциплин программного цикла. Предлагается структура системы.

Ключевые слова: система, дисциплина, обучающая система, интерфейс, диалог.

Annotation: this paper discusses the principles of building learning systems. The features of the systems for the disciplines of the program cycle are analyzed. The structure of the system is proposed.

Keywords: system, discipline, training system, interface, dialogue.

В настоящее время учебный процесс в ВУЗах характеризуется унификацией и универсализацией учебных планов, что означает проведение занятий для студентов разных направлений подготовки по одинаковым учебным планам и, как следствие, обучение одинаковым дисциплинам, часто по одинаковым или схожим методикам.

Объединение потоков и уменьшение доли контактной работы при изучении дисциплин негативно влияет на студентов младших курсов, поскольку приводит к серьезным пробелам в усвоении материала и, следовательно, к низкой базовой подготовке по указанным дисциплинам, что, в свою очередь, в дальнейшем может привести к неудовлетворительному освоению и специальных дисциплин.

При укрупнении потоков преподаватель вынужден ориентироваться на средний уровень подготовки студентов и давать «усредненный материал», что негативно влияет на общий уровень преподавания дисциплины. Кроме того, концентрация на младших курсах общих дисциплин мешает студентам осознать особенности подготовки по выбранному направлению и приводит к снижению мотивации их обучения.

К примеру, дисциплина «Операционные системы» входит в учебные планы направлений подготовки 09.03.01–«Информатика и вычислительная техника», 09.03.02–«Информационные системы и технологии», 27.03.04–«Управление в технических системах». Однако для направлений 09.03.02 и 27.03.04 операционная система (ОС) должна рассматриваться как средство эффективной организации и управления информационными системами, а для направления 09.03.01 – как объект проектирования. С учетом этого, перечень вопросов, составляющих содержание дисциплины, и степень их детализации принципиально не могут быть одинаковыми. С другой стороны, в рассматриваемой дисциплине можно выделить общие для изучения вопросы и темы:

- понятие ОС, история развития, классификация, функции;
- основные семейства ОС ;
- архитектура ОС;
- управление процессами, методы взаимодействия и синхронизации процессов;
- тупики (deadlocks), методы предотвращения и обнаружения тупиков;
- управление памятью, организация виртуальной памяти;
- управление внешними устройствами;
- файловые системы;
- организация ввода-вывода.

Одним из способов решения этих проблем является использование обучающих систем, которые позволяли бы рассматривать один и тот же вопрос с разной степенью детализации с учетом направления подготовки.

Использование компьютерных систем поддержки учебного процесса позволяет в рамках традиционной аудиторной работы реализовать методики активизации

самостоятельной работы студентов. Например, в рамках часов, отводимых учебным планом на изучение материала, студент сам может распределять время на изучение теоретического материала и решение практических задач, проведение экспериментальных или лабораторных исследований. Результатом проработки теоретического материала могут быть ответы на контрольные вопросы, конспект и др. Кроме того, этап изучения основ теории должен заканчиваться собеседованием с преподавателем и/или проверкой полученных знаний с помощью обучающей системы.

Таким образом, современный компьютеризированный процесс обучения представляет собой форму сотрудничества преподавателя и студента, взаимодействующих друг с другом через обучающую систему. Схема взаимодействия «обучаемый-обучающая система» изображена на рисунке 1. Очевидно, что при разработке компьютерных систем поддержки учебного процесса особое внимание должно быть уделено разработке интерфейса. Интерфейс определяет общие принципы взаимодействия человека с компьютерной системой. При построении компьютерных систем учебного назначения используется диалоговый интерфейс, обеспечивающий двустороннюю передачу информации в реальном масштабе времени.

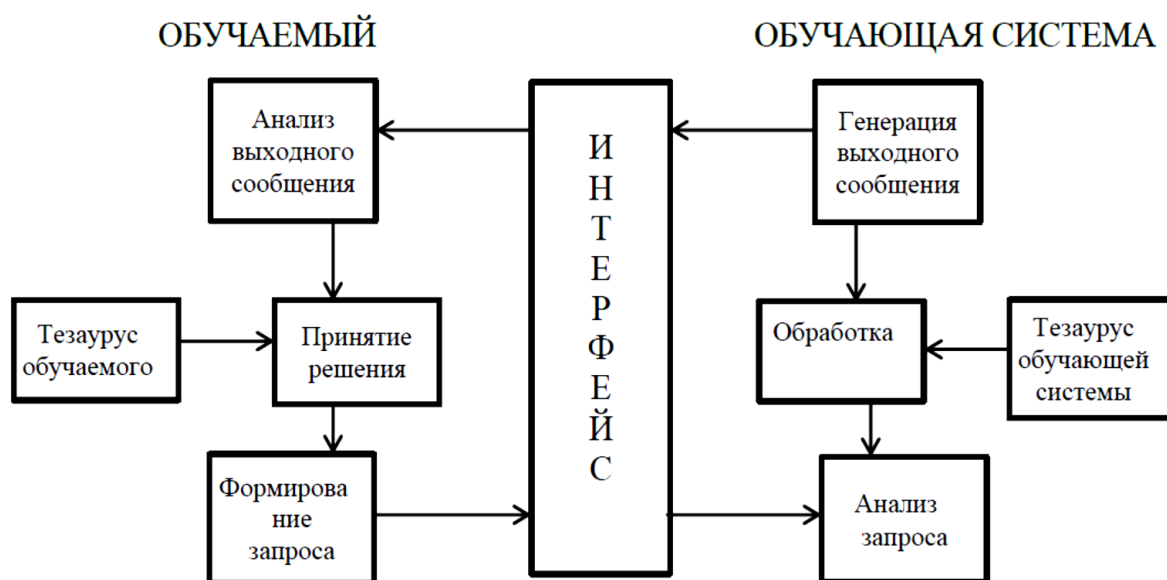


Рисунок 1 – Схема взаимодействия обучаемый-обучающая система

При проектировании диалога возникает целый ряд проблем, из которых можно особо отметить следующие: количество диалога, сфера его применения, педагогичность, завершенность, эффективность и симметричность [1].

Диалог между студентом и обучающей системой является важной составной частью организации контрольно-корректирующих воздействий преподавателя на качество и результат самостоятельного изучения студентом запланированного учебного материала. Как правило, указанные контрольно-корректирующие воздействия осуществляются с помощью тестов. С одной стороны, преподаватель получает информацию о степени усвоения материала и необходимой коррекции содержания и формы самостоятельной работы студента. С другой стороны, студент может критически оценить свои успехи и промахи в изучении данного материала и правильно организовать дальнейшую работу, обеспечив ее системность и систематичность.

Таким образом, диалог как основа самоконтроля, играет серьезную роль в формировании у обучаемого ответственности, дисциплинированности, сознательности, инициативности, стремления к результативности в познавательной деятельности.

Обучающие системы – взаимосвязанная совокупность иерархических информационно согласованных подсистем. Иерархическая организация позволяет при проектировании рассматривать обучающую систему с различной степенью подробности: на высшем уровне используется наименее детализированное представление, отражающее только общие черты и особенности проектируемой системы. На каждом следующем уровне степень подробности рассмотрения возрастает. Такой подход позволяет задачу большой размерности разделять на последовательно решаемые группы задач меньшей размерности.

Информационная согласованность означает, что подсистемы обмениваются потоками данных без участия управляющих воздействий со стороны внешней среды.

Кроме того, обучающая система должна быть открытой системой, то есть обладать свойством включения новых подсистем и замены использовавшихся ранее.

Любая обучающая система по своей сути уникальна. Ее уникальность в общем случае определяется той предметной областью, для которой она предназначена и, возможно, спецификой организации интерфейса. Однако, систему можно рассматривать как специализированную систему с максимальным использованием унифицированных модулей. В целом, при создании обучающих систем следует руководствоваться принципами системного единства, совместимости, типизации и развития; принцип системного единства обеспечивает создание и использование типовых элементов и решений; принцип развития обеспечивает пополнение, совершенствование и обновление составных частей системы.

Таким образом, современная обучающая система представляет собой комплекс программных модулей и данных, взаимодействующих друг с другом. Предлагается следующая структурная организация обучающей системы (рисунок 2). Стрелками на рисунке обозначены информационные и логические связи между отдельными компонентами системы.

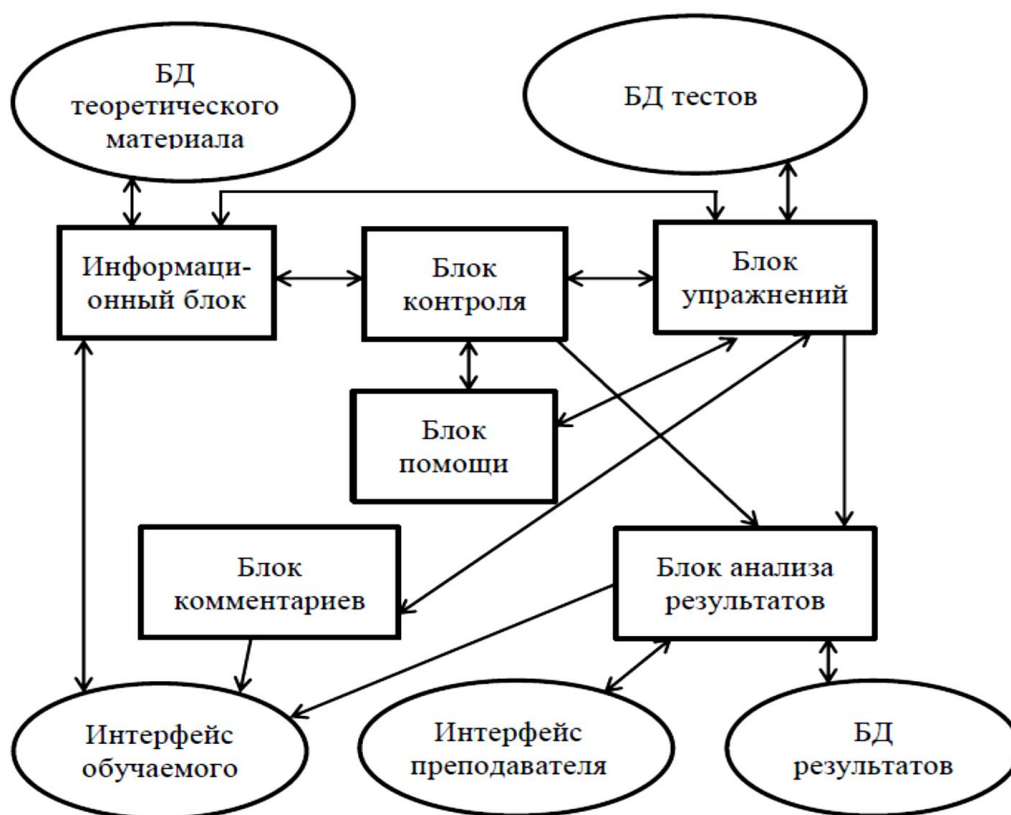


Рисунок2– Структура обучающей система

Представляется целесообразным отдельно реализовать интерфейс обучаемого и интерфейс преподавателя. Это позволит разделить учебную функцию системы от ее аналитической и контрольной функции.

Обучающая система должна включать информационный блок, содержащий теоретический материал. Наряду с информационным блоком (ИБ) обучающая система должна включать блок упражнений и блок комментариев. Подготовка упражнений – это наиболее трудоемкое дело в создании обучающих систем. Для каждой темы необходимо подобрать не только подходящие задания, но и выбрать форму упражнений, подготовить эталоны решений и предусмотреть типовые ошибки. При этом особое внимание следует уделять ситуации, когда задача имеет не единственное решение или оно может быть получено различными способами, и поэтому контролировать необходимо не только конечный результат, но и сам процесс решения. В этом случае реализация контрольной функции системы требует построения базы знаний и/или экспертной системы в соответствующей предметной области, другими словами, современная компьютерная система учебного назначения, ориентированная на решение трудно формализуемых задач, очень часто по своей сути является экспертной системой.

Блок комментариев может содержать различные виды внутренней обратной связи для реакций на действия студентов при выполнении упражнений – от простейших (верно, неверно, неточно и т.д.) до подробных разъяснений типовых ошибок.

Кроме уже рассмотренных блоков, в обучающей системе целесообразно выделить:

- блок помощи, отделив его от блока комментариев. Блок помощи должен при необходимости обеспечивать выдачу подсказки с заданной степенью детализации на каждом из этапов решения задачи;
- блок контроля, отделив его от блока упражнений. Блок контроля должен обеспечивать настройку системы упражнений. В качестве задаваемых параметров могут, например, выступать критерии оценки, уровень сложности заданий, ограничение на время ответа, степень детализации подсказок и т.д.;
- блок анализа и накопления статистики. Введение этого блока имеет целью сбор информации об использовании обучающей системы.

Результаты тестирования после статистической обработки и обобщения заносятся в специальную базу данных результатов, являющуюся электронным архивом, в котором хранятся протоколы сеансов работы с обучающей системой. Накопленная информация может быть полезна для дальнейшего развития и совершенствования всей системы и отдельных ее составляющих. Более того, на основе этой информации может быть построена экспертная система анализа обучения (САО). Такая система может использоваться для выявления трудностей, препятствующих усвоению материала, т.е. САО позволит локализовать конкретные пробелы в обучении, определить их диапазон, а также даст возможность преподавателю сделать окончательные выводы.

Таким образом, использование систем методической и компьютерной поддержки учебного процесса позволит: во-первых, сократить время доступа к необходимой информации; во-вторых, обеспечить оптимальный уровень взаимодействия каждого студента и преподавателя; в-третьих, повысить объективность оценки знаний студентов; в-четвертых, организовать изучение материала каждым студентом по индивидуальному графику.

В дальнейшем предполагается реализовать обучающую систему на платформе Moodle.

Библиографический список:

1. Дистанционные образовательные технологии: проектирование и реализация учебных курсов / Под общ. ред. М.Б. Лебедевой. СПб.: БХВ-Петербург, 2010. – 336 с.

УДК 004.415

С.Н. Широбокова, канд. экон. наук, доцент**М.Е. Диков, студент****О.Н. Сериков, студент***ФГБОУ ВО «Южно-Российский государственный политехнический университет (НПИ) имени М.И. Платова»***ОБ АВТОМАТИЗИРОВАННОМ ПОДХОДЕ К МОНИТОРИНГУ И АНАЛИЗУ ПРОФОРИЕНТАЦИОННОЙ РАБОТЫ КОНСУЛЬТАНТОВ ВУЗА В СОЦИАЛЬНОЙ СЕТИ «ВКОНТАКТЕ»**

Аннотация: Целью предлагаемого подхода и разработанного на его основе программного инструментария является автоматизированное формирование показателей активности взаимодействия консультантов факультетов вуза с потенциальными абитуриентами в социальной сети на основе динамики добавления пользователей социальной сети в друзья консультантов вуза и вступления в группу абитуриентов вуза. Данный подход дает возможность предварительно оценить количество лояльных к поступлению в вуз абитуриентов, проанализировать распределение их численности по факультетам, выявить и поощрить активных консультантов и предоставить информацию для принятия мер к активизации работающих менее эффективно консультантов.

Ключевые слова: социальная сеть «ВКонтакте», формализованная модель профилей абитуриентов и консультантов в социальной сети, запросы API, профориентационная деятельность, поиск пользователей по параметрам, коммуникации с абитуриентами.

Annotation: The purpose of the proposed approach and developed on the basis of its software tools is the automated formation of indicators of activity of University faculty consultants interaction with potential applicants in the social network based on the dynamics of adding users of the social network to the friends of University consultants and joining the group of University entrants. This approach makes it possible to pre-estimate the number of applicants loyal to the University, to analyze the distribution of their number by faculties, to identify and encourage active consultants and to provide information for measures to activate less effective consultants.

Key words: social network "Vkontakte", a formalized model of profiles of applicants and consultants in the social network, API requests, career guidance, search for users by parameters, communication with applicants.

Социальные сети являются в настоящее время одним из наиболее популярных сервисов коммуникаций, а также и инструментом рекламы, маркетинга, управления персоналом [1]. Для образовательных учреждений ресурсы социальных сетей могут служить эффективным средством коммуникации с потенциальными абитуриентами. В Южно-Российском государственном политехническом университете (НПИ) имени М.И. Платова для реализации этого направления профориентационной деятельности с 2013г. существует группа «Абитуриент ЮРГПУ(НПИ)» в социальной сети «ВКонтакте», в которой работают консультанты различных факультетов для предоставления помощи заинтересованным пользователям и потенциальным абитуриентам вуза.

Суть предлагаемого подхода состоит в автоматизированном мониторинге профориентационной деятельности в социальной сети путем сбора статистических данных о динамике добавления пользователей сети в друзья консультантов вуза и вступления в группу абитуриентов вуза. Консультанты факультетов, проводя агитационную работу с потенциальными абитуриентами (выпускниками школ, колледжей, лицеев и т.д.), рассказывают о направлениях подготовки в целом по университету, результаты ЕГЭ по каким предметам требуются на эти направления, будущем трудоустройстве, местах практики и т.п. Также более подробную информацию каждый консультант предоставляет о направлениях подготовки своего факультета. Чаще всего в случае лояльности к поступлению в университет абитуриент откликается на предложение добавиться в группу «Абитуриент ЮРГПУ(НПИ)», а также, если его заинтересовали направления факультета, консультант которого проводил с ним профориентационную работу, абитуриент добавляет консультанта в друзья – для коммуникаций по вопросам поступления в будущем. Таким образом,

количество добавленных в группу и количество добавившихся абитуриентов в друзья к консультанту можно считать не только показателем активности работы консультанта, но и показателями лояльности к поступлению в вуз в целом и на выбранный факультет в частности.

Для постоянного мониторинга деятельности консультантов в социальной сети на основе предложенного подхода разработан программный инструментарий, позволяющий формировать статистику по вышеуказанным показателям.

Представим в формализованном виде профили абитуриентов и консультантов как базовые компоненты и отношения между ними для дальнейшей алгоритмизации поставленной задачи [2].

Множество участников группы может быть представлено следующим образом:

$$Users = \{user_i\}, i = \overline{1, m},$$

где m – количество участников группы в сети.

Каждый элемент множества участников группы может быть представлен следующей моделью:

$$users_i = \langle Id_i, FN_i, LN_i, MN_i, ST_i, CT_i, CR_i, PH_i, DC_i, BD_i, FT_i \rangle,$$

где Id_i – идентификатор страницы в сети; FN_i – имя пользователя; LN_i – фамилия пользователя; MN_i – отчество пользователя; ST_i – пол пользователя; CT_i – город; CR_i – страна; PH_i – фото; DC_i – статус страницы; BD_i – дата рождения; FT_i – множество факультетов, консультанты которых находятся в друзьях (в начале работы алгоритма $FT_i = \emptyset$).

Множество консультантов может быть описано следующим образом:

$$Consultants = \{consultant_j\}, j = \overline{1, k},$$

где k – количество действующих консультантов.

Модель консультанта может быть представлена как:

$$consultant_j = \langle Id_j, FN_j, LN_j, MN_j, FT_j, FD_j \rangle,$$

где Id_j – идентификатор страницы в сети; FN_j – имя консультанта; LN_j – фамилия консультанта; MN_j – отчество консультанта; FT_j – факультет, от которого работает консультант; FD_j – множество друзей j -го консультанта в сети.

$$FD_j = \{fd_{jl}\}, l = \overline{1, b^j},$$

где b^j – количество друзей j -го консультанта.

Для контроля деятельности консультантов для каждого из них формируется множество участников группы, которых он добавил в друзья и пригласил в сообщество:

$$Intersect_j = Users \cap FD_j = \{inct_{jw}\}, j = \overline{1, k}, w = \overline{1, z^j},$$

где z^j – количество участников группы, которые находятся в друзьях у j -го консультанта.

Дополнительно формируется множество участников группы, которые находятся в друзьях у нескольких консультантов:

$$Mutual = Intersect_c \cap Intersect_v, c = \overline{1, k}, v = \overline{c+1, k}.$$

Затем для каждого элемента $inct_{jw} \in Intersect_j, j = \overline{1, k}, w = \overline{1, z^j}$ формируется множество факультетов, с которыми взаимодействовал пользователь через консультантов:

$$FTS'_{jw} = \begin{cases} FTS_{jw} \cup FT_v, & inct_{jw} \in Mutual, v = \overline{1, k}, \\ FTS_{jw} \cup FT_j, & inct_{jw} \notin Mutual \end{cases}$$

$$FTS_{jw} = FTS'_{jw}$$

Концептуальная модель разрабатываемого приложения представлена на рисунке 1 в стандарте *UML* в виде диаграммы классов.

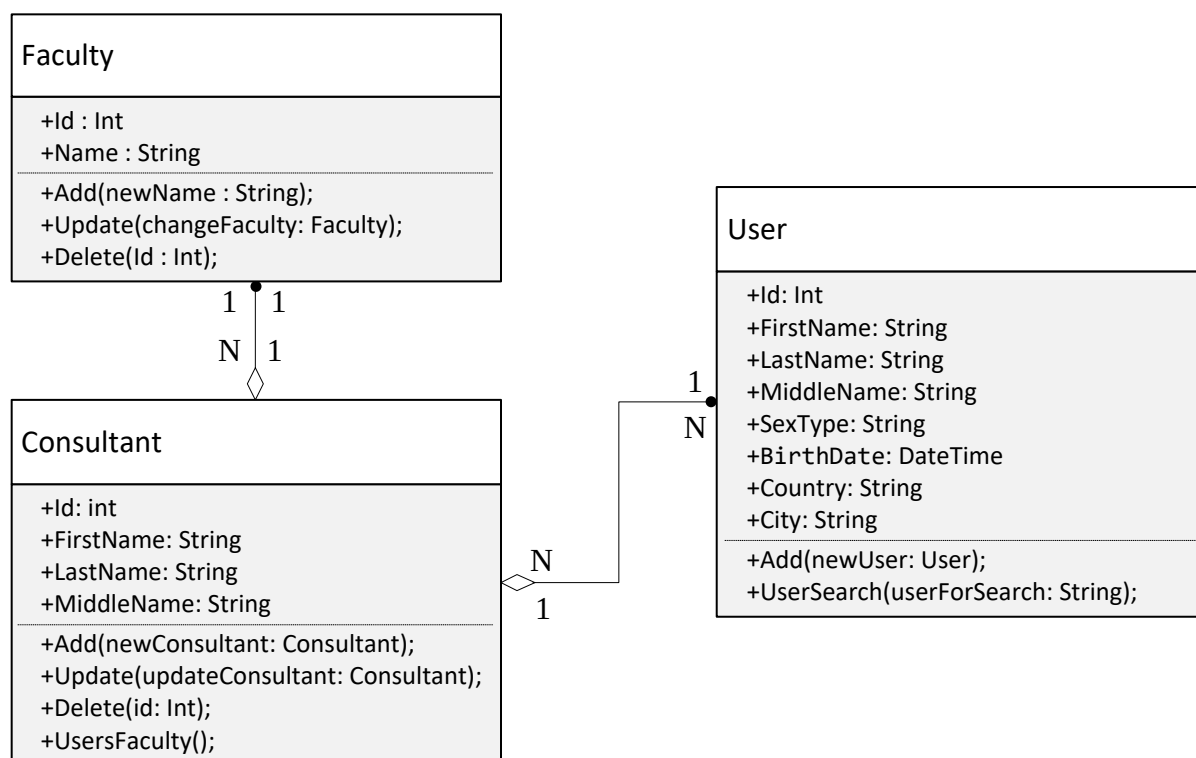


Рисунок 1 – Диаграмма классов разрабатываемого приложения

На рисунке 2 представлена логическая модель информационной базы.

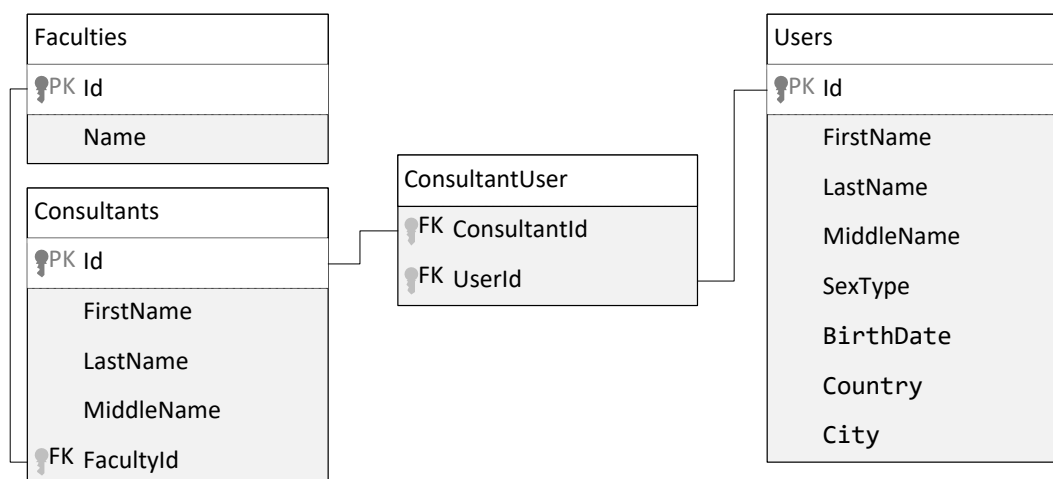


Рисунок 2 – Логическая модель информационной базы

Для создания приложения была использована технология *WPF* на языке *C#*, с применением библиотеки *Entity Framework Core* для работы с базой данных *SQLite*. Разработка проводилась в среде *Microsoft Visual Studio Community 2017*. Предложен авторский алгоритм (см. рисунок 3), основанный на выборке подмножеств участников группы, которые взаимодействовали с одним или несколькими консультантами, и последующей проверке на принадлежность к множествам друзей каждого консультанта для вычисления результативности, который значительно быстрее алгоритма полного перебора.

```

/// <summary>
/// Возвращает список пользователей, распределенных по факультетам.
/// </summary>
/// <param name="token">Токен авторизованного пользователя. </param>
/// <param name="fields">Поля в HTTP-запросе, которые необходимо вернуть. </param>
/// <param name="users">Список пользователей группы. </param>
/// <param name="listOfConsultant">Список всех консультантов. </param>
public async Task<List<User>> MainMethod
(string token, RequestFields fields, List<User> users, List<Consultant> listOfConsultant)
{
    //Заполнить список друзей каждого консультанта
    var consultantFriends = await GetAllFriendsOfConsultant(token, fields, listOfConsultant);
    //Найти пользователей группы, которые пересекаются с консультантами
    //(1 юзер : 1 консультант(факультет))
    var intersect = await GetIntersectFriendsWithGroup(users, listOfConsultant);
    //Найти пользователей группы, которые пересекаются с несколькими консультантами
    //(1 юзер : N консультантов(факультетов))
    var Mutual = await GetListOfMutualFriends(intersect);
    return await Task.Run(() =>
    {
        for (var i = 0; i < intersect.Count; i++)
        {
            for (var j = 0; j < intersect[i].Count; j++)
            {
                //Получаем пользователя из списка 1:1
                var User = users.FirstOrDefault(x => x.Id == intersect[i][j].Id);
                //Находим индекс полученного пользователя в общем списке пользователей группы
                var UserIndex = users.IndexOf(User);
                //Ищем пользователя в списке 1:N
                var UserFromMutual = Mutual.FirstOrDefault(x => x.Id == User.Id);
                //Создаем список факультетов для пользователя
                users[UserIndex].Faculties = new List<string>();
                if (UserFromMutual == null) //Если пользователь у 1 консультанта
                {
                    users[UserIndex].Faculties.Add(listOfConsultant[i].Faculty);
                }
                else
                {
                    //Иначе пользователь у N консультантов, находим у каких
                    //и добавляем их факультеты в список факультетов пользователя
                    for(var k = 0; k < intersect.Count; k++)
                    {
                        if (intersect[k].Contains(UserFromMutual))
                        {
                            users[UserIndex].Faculties.Add(listOfConsultant[k].Faculty);
                        }
                    }
                }
            }
        }
        return users;
    });
}

```

Рисунок 3 – Основной метод анализа деятельности консультантов

Входными параметрами основного метода анализа деятельности консультантов являются: *token* – ключ доступа авторизованного пользователя для работы с API методами сети; *fields* – поля об информации пользователей, которые необходимо вернуть в методе; *users* – участники группы; *listOfConsultant* – список действующих консультантов факультетов.

Алгоритм основан на исключении множества пользователей, которые не находятся в группе и в друзьях консультантов, за счет этого достигается быстроедействие поиска консультантов, с которыми взаимодействовал пользователь, что, в свою очередь, является показателем деятельности консультанта. Экспериментальное сравнение разработанного алгоритма с алгоритмом «полного перебора» показало большую эффективность по быстроедействию в 3,4 раза. Эксперимент проводился с помощью специального объекта, который измеряет скорость выполнения отдельных участков кода.

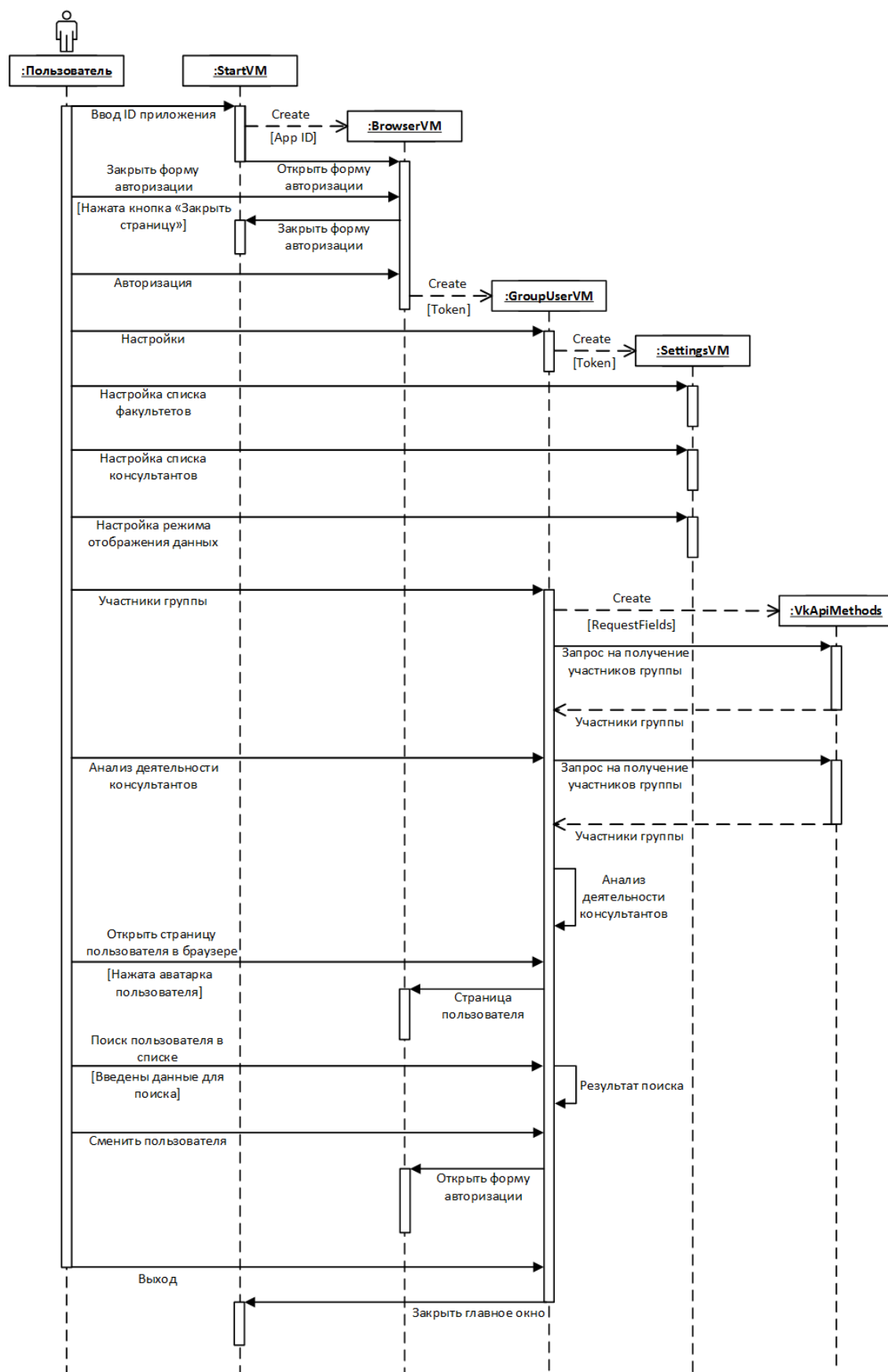


Рисунок 4 – Диаграмма последовательностей приложения анализа деятельности консультантов

На рисунке 4 представлена диаграмма последовательностей разработанного приложения анализа деятельности консультантов, которая отражает последовательность

действий администратора группы для корректной работы с приложением, а также взаимодействие методов и объектов различных подсистем.

На рисунке 5 приведены некоторые экранные формы приложения.

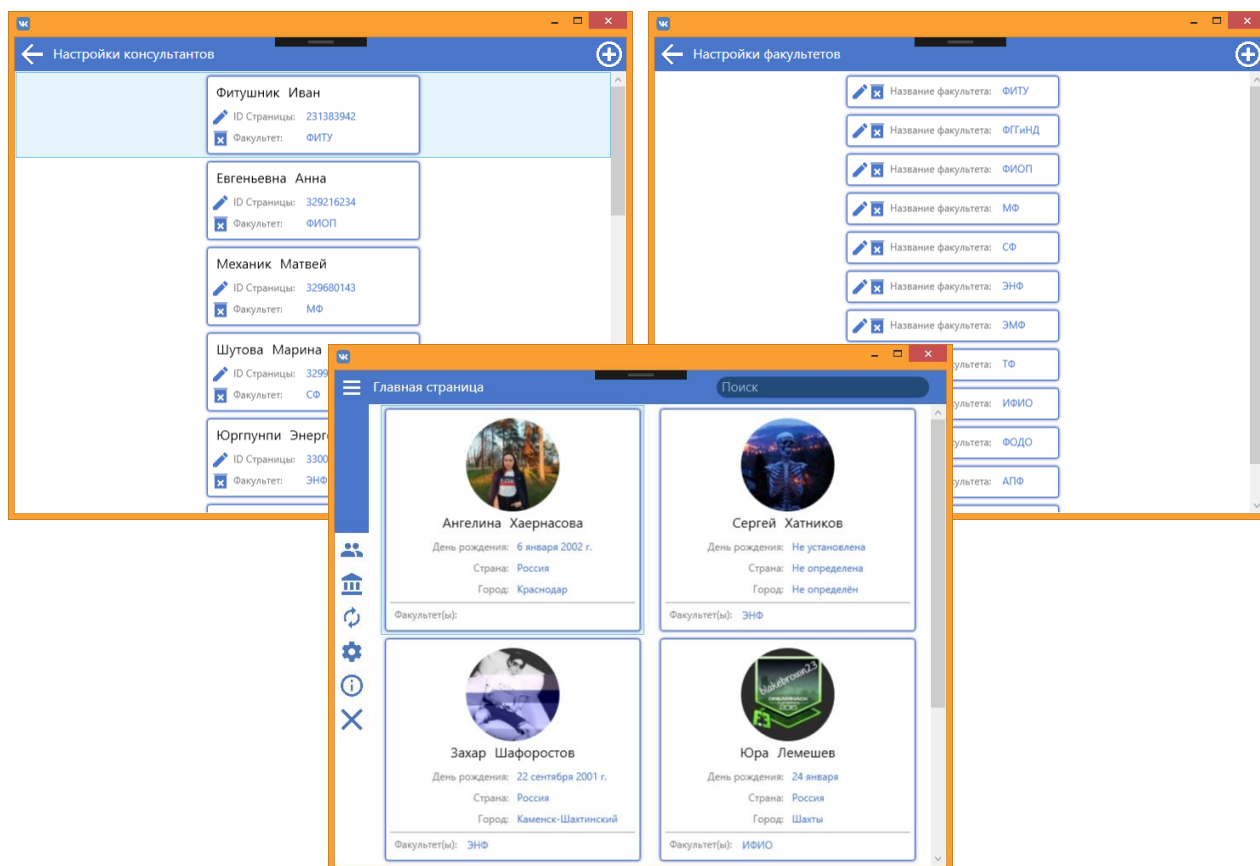


Рисунок 5 – Экранные формы приложения

Предложенный подход и основанный на нем инструментарий позволяют проводить автоматизированный мониторинг активности указанных в списке приложения консультантов, который основан на динамике добавления пользователей социальной сети в друзья консультантов и вступления в группу абитуриентов вуза. Указанные показатели статистики могут использоваться для предварительной оценки количества лояльных к поступлению в вуз абитуриентов, распределения их численности по факультетам, поощрения активных консультантов и принятия мер к активизации консультантов, работающих менее эффективно.

Библиографический список:

1. Диков М.Е., Жевакин Д.М., Перекрестова Т.И. Автоматизация мониторинга профориентационной деятельности консультантов вуза в социальной сети ВКонтакте // Фундаментальные исследования с применением компьютерных технологий в науке, производстве, социальных и экономических процессах: материалы 17-ой Национальной молод. Науч.-практ. конф., г.Новочеркасск, 20-23 декабря 2017г./ Юж.-Рос. гос. политехн. ун-т (НПИ) имени М.И. Платова. Новочеркасск: Лик, 2017. С.107-112.
2. Широбокова С.Н., Диков М.Е., Сериков О.Н., Жевакин Д.М., Перекрестова Т.И. Инструментарий мониторинга профориентационной деятельности в социальных сетях: формализованная модель и программная реализация // Наука и бизнес: пути развития. 2019. №3. С. 232-236.

УДК 004.415

С.Н. Широбокова, канд. экон. наук, доцент

О.Н. Сериков, студент

М.Е. Диков, студент

ФГБОУ ВО «Южно-Российский государственный политехнический университет (НПИ) имени М.И. Платова»

О ВАРИАНТЕ РЕАЛИЗАЦИИ НА ПЛАТФОРМЕ «1С:ПРЕДПРИЯТИЕ» ИНСТРУМЕНТАРИЯ АВТОМАТИЗИРОВАННОГО ФОРМИРОВАНИЯ УЧЕБНО-МЕТОДИЧЕСКОЙ ДОКУМЕНТАЦИИ НА ОСНОВЕ УЧЕБНОГО ПЛАНА

Аннотация: В статье описываются аспекты варианта реализации на платформе «1С:Предприятие» инструментария автоматизированного формирования шаблонов рабочих программ в формате DOCX на основе файла выгрузки учебного плана в формате Excel. Для алгоритмизации задачи разработано формализованное описание модели учебного плана направления подготовки в виде базовых компонентов и отношений между ними. Представлена объектная модель данных приложения в виде UML-диаграммы классов. Описаны аспекты программной реализации формирования файла-шаблона рабочей программы на основе Word-макета с установленными закладками.

Ключевые слова: формирование шаблонов рабочих программ, рабочий учебный план, рабочая программа дисциплины, автоматизированное формирование учебно-методической документации, формализованное описание модели учебного плана направления подготовки.

Annotation: The article describes the aspects of the implementation option on the platform "1C:Enterprise" tools for automated formation of templates of work programs in DOCX format based on the curriculum upload file in Excel format. For algorithmization of the problem, a formalized description of the curriculum model of the training direction in the form of basic components and relations between them is developed. The object model of application data is presented in the form of a UML class diagram. The aspects of the software implementation of the working program template file formation based on Word-layout with installed bookmarks are described.

Key words: formation of working program templates, working curriculum, working program of the discipline, automated formation of educational and methodical documentation, formal description of the model of the curriculum of the direction of training.

Комплект рабочих программ дисциплин является важнейшей составляющей основной образовательной программы направления подготовки. Рабочие программы разрабатываются на основе информации из учебного плана направления подготовки по определенному шаблону и содержат ряд формализуемых компонент (часы аудиторной и внеаудиторной работы, объем самостоятельной работы студента, семестры, наличие различных видов отчетности, формируемые дисциплинами компетенции и т.д.), которые могут быть сформированы автоматизированным способом, что позволит исключить ошибки и несоответствие данным из учебного плана, оптимизировать и ускорить процесс формирования учебно-методической документации [1]. Учебные планы по ФГОС 3++ разработаны в программном комплексе «Учебные планы ВО» (<https://www.mmis.ru/programs/plany>), позволяющем экспортировать их в файлы Excel. Из файла Excel далее осуществляется загрузка данных в инструментарий для формирования шаблонов файлов рабочих программ в формате Word.

Для алгоритмизации этапов автоматизированного формирования учебно-методической документации на основе учебного плана направления подготовки разработано формализованное описание модели учебного плана в виде базовых компонентов и отношений между ними. Формализованная модель учебного плана может быть представлена следующим образом:

MainClass = < *CodeDirection*, *NameDirection*, *NameOrientation*, *YearOfRecruitment*,
NumProtocolOfPlane, *DateProtocolOfPlane*, *TypeOfProgram*, *masDis*,
globalMasComp>,

где *CodeDirection* – шифр направления подготовки, *NameDirection* – наименование направления подготовки, *NameOrientation* – направленность (профиль), *YearOfRecruitment* – год набора, *NumProtocolOfPlane* – номер протокола утверждения учебного плана, *DateProtocolOfPlane* – дата протокола утверждения учебного плана, *TypeOfProgram* – вид программы подготовки (академический, прикладной) *masDis* – массив дисциплин, *globalMasComp* – массив компетенций, которые обеспечивает освоение направления подготовки.

Массив компетенций можно описать следующим образом:

$$globalMasComp = |Compenence^q|, q = \overline{1, w}.$$

Элемент массива, описывающий компетенцию, формально можно представить следующим образом:

$$Compenence^q = \langle CodeComp^q, NameComp^q, TypeComp^q, DescriptionComp^q \rangle,$$

где *CodeComp^q* – код *q*-й компетенции, *NameComp^q* – наименование *q*-й компетенции, *TypeComp^q* – тип *q*-й компетенции, *DescriptionComp^q* – описание *q*-й компетенции.

Массив дисциплин можно описать следующим образом:

$$masDis = |Discipline^i|, i = \overline{1, m}.$$

Элемент массива, описывающий дисциплину, формально можно представить следующим образом:

$$Discipline^i = \langle CodeDiscipline^i, NameDiscipline^i, CodeDepartment^i, sem_string^i, masComp^i, masSem^i, ZET^i, Hours^i, Lec^i, Labs^i, Practice^i, SR^i, Control^i \rangle,$$

где *CodeDisciplineⁱ* – код *i*-й дисциплины, *NameDisciplineⁱ* – наименование *i*-й дисциплины, *CodeDepartmentⁱ* – код обеспечивающей кафедры, *sem_Stringⁱ* – строка, обеспечивающая хранение всех семестров (как строку), в которых ведется обучение по *i*-й дисциплине, *masCompⁱ* – массив компетенций, формирование которых обеспечивает *i*-я дисциплина, *masSemⁱ* – массив семестров с распиской, *ZETⁱ* – объем *i*-й дисциплины в ЗЕТ, *Hoursⁱ* – объем *i*-й дисциплины в часах, *Lecⁱ* – количество часов лекций по дисциплине, *Labsⁱ* – количество часов лабораторных занятий по дисциплине, *Practiceⁱ* – количество часов практический занятий по дисциплине, *SRⁱ* – количество часов самостоятельной работы по дисциплине, *Controlⁱ* – количество часов на контроль (экзамен).

$$masSem = |VolumeSemesters^j|, j = \overline{1, s}.$$

Элемент массива, описывающий семестр дисциплины с распиской, формально можно представить следующим образом:

$$VolumeSemesters^j = \langle NumSemestr^j, ZE_Sem^j, isExam^j, isSmallExam^j, isExamWithRating^j, isCourseProject^j, isCourseWork^j, Lec_Sem^j, Labs_Sem^j, Practice_Sem^j, SR_Sem^j, Control_Sem^j, SR_KWUA_Sem^j, SR_SRS_Sem^j, SR_EX_KWUA_Sem^j, SR_EX_SRS_Sem^j \rangle$$

где *NumSemestr^j* – номер семестра, *ZE_Sem^j* – объем дисциплины в ЗЕТ в семестре, *isExam^j* – наличие в семестре экзамена по дисциплине, *isSmallExam^j* – наличие в семестре зачета по дисциплине, *isExamWithRating^j* – наличие в семестре зачёта с оценкой, *isCourseProject^j* – наличие в семестре курсового проекта, *isCourseWork^j* – наличие в семестре курсовой работы, *Lec_Sem^j* – количество часов лекций, *Labs_Sem^j* – количество часов лабораторных занятий, *Practice_Sem^j* – количество часов практический занятий, *SR_Sem^j* – количество часов самостоятельной работы студентов в семестре, *Control_Semⁱ* –

количество часов на контроль (экзамен) в семестре, $SR_KWUA_Sem^j$ – количество контактных часов самостоятельной работы в семестре (групповые консультации), $SR_SRS_Sem^j$ – количество часов самостоятельной работы студента в семестре, $SR_EX_KWUA_Sem^j$ – количество контактных часов консультаций перед экзаменом (групповые консультации перед экзаменом), $SR_EX_SRS_Sem^j$ – количество часов самостоятельной работы студента при подготовке к экзамену в период экзаменационной сессии.

Массив компетенций можно описать следующим образом:

$$globalMasComp = |Competention^k|, k = \overline{1, l}.$$

Элемент массива, описывающий дисциплину, формально можно представить следующим образом:

$$Competention^k = \langle CodeCompetention^k, NameComp^k, TypeComp^k, DescriptionComp^k \rangle$$

где $CodeCompetention^k$ – индекс компетенции, $NameComp^k$ – наименование компетенции, $TypeComp^k$ – тип компетенции (УК, ОПК, ПК), $DescriptionComp^k$ – описание компетенции.

Дополнительную информацию, необходимую для формирования шаблонов рабочих программ, можно описать следующим образом:

$$AdditionalData = \langle Ministry, MainHead, NameUni, NameProrector, PositionProrector, YearOfApproval, NameFaculty, Department, DepartmentMatching, NumProtocolOfDepartment, DateProtocolOfDepartment, DateOfFirstCertOfTheSpringSemester, DateOfTheFirstCertOfTheAutumnSemester, NormSRSContactLectures, NormSRSContactKP, NormSRSContactKR, NormSRSContactBeforeEX, NormSRSContactEX \rangle,$$

где *Ministry* – наименование министерства, *MainHead* – организационная форма высшего учебного заведения, *NameUni* – полное наименование вуза, *NameProrector* – фамилия и инициалы лица, утверждающего рабочую программу, *PositionProrector* – должность лица, утверждающего рабочую программу, *YearOfApproval* – год утверждения рабочих программ, *NameFaculty* – наименование факультета, *Department* – кафедра, утверждающая рабочую программу, *DepartmentMatching* – наименование согласующей кафедры, *NumProtocolOfDepartment* – номер протокола утверждения рабочей программы, *DateProtocolOfDepartment* – дата протокола утверждения рабочей программы, *DateOfFirstCertOfTheSpringSemester* – дата первой аттестации весеннего семестра, *DateOfTheFirstCertOfTheAutumnSemester* – дата первой аттестации осеннего семестра, *NormSRSContactLectures* – норма количества контактных часов групповых консультаций в течение семестра (процент от лекций), *NormSRSContactKP* – норма контактных часов консультаций и защиты курсового проекта, *NormSRSContactKR* – норма контактных часов консультаций и защиты курсовой работы, *NormSRSContactBeforeEX* – норма групповых консультаций перед экзаменом, *NormSRSContactEX* – норма времени сдачи экзамена.

Формализованная модель послужила основой разработки структуры классов программной реализации инструментария, представленной на рис. 1 в виде UML-диаграммы классов [2].

На первом этапе пользователь выбирает *Excel*-файл с данными учебного плана, на основании которого будут формироваться шаблоны рабочих программ. Помимо этого пользователю необходимо ввести в программу дополнительные данные, используемые при создании шаблонов рабочих программ и не включенные в исходный *Excel*-файл (например, название утверждающей кафедры, фамилия, имя, отчество и должность лица, утверждающего рабочие программы, наименование факультета и т.д.). На рис. 2 приведены

```

classDiagram
    class Дисциплины {
        <<справочник>>
        Ссылка: Строка
        Код: Строка, 20
        Наименование: Строка, 150
        ПорядковыйНомер: Строка, 30
        ЗЕТ: Число, 3, 0
        Часов: Число, 3, 0
        Лек: Число, 3, 0
        Лаб: Число, 3, 0
        ПР: Число, 3, 0
        СР: Число, 3, 0
        Контроль: Число, 3, 0
        НаименованиеВРодитПадеже: Число, 3, 0
        СеместрыТех: Строка, 30
        КодКафедры: Строка, 30
        СправочникСсылка.Кафедры
    }
    class ФормируемыеКомпетенции {
        Ссылка: Строка
        НомерСтроки: Число
        Компетенция: СправочникСсылка.Компетенции
    }
    class ТипКомпетенции {
        <<перечисление>>
        УК
        ОПК
        ПК
    }
    class КафедрныйСправочник {
        <<справочник>>
        Ссылка: Строка
        Код: Строка, 9
        Наименование: Строка, 150
        КраткоеОбозначение: Строка, 15
        ЗавКафедрой: Строка, 30
    }
    class ИндикаторДостижения {
        <<справочник>>
        Ссылка: Строка
        Код: Строка, 9
        Наименование: Строка, 0
    }
    class Семестры {
        Ссылка: Строка
        НомерСтроки: Число
        НомерСемистра: Число, 2,0
        зе_Сем: Число, 3, 0
        Экзамен: Булево
        Зачет: Булево
        ЗачетСОценкой: Булево
        КП: Булево
        КР: Булево
        Лек_Сем: Число, 3, 0
        Лаб_Сем: Число, 3, 0
        Пр_Сем: Число, 3, 0
        СР_Сем: Число, 3, 0
        Контроль_Сем: Число, 3, 0
        СРС_Конт_Сем: Число, 5, 2
        СРС_СамР_Сем: Число, 5, 2
        СРСэкз_Конт_Сем: Число, 5, 2
        СРСэкз_СамР_Сем: Число, 5, 2
    }
    class Загрузка {
        <<обработка>>
    }
    class ФормированиеРП {
        <<обработка>>
        УчебныйПлан: СправочникСсылка.УчебныйПлан
    }
    class УчебныйПлан {
        <<справочник>>
        Ссылка: Строка
        Код: Строка, 9
        Наименование: Строка, 100
        НаправлениеПодготовки: СправочникСсылка.НаправленияПодготовки
        Направленность: Строка, 50
        ГодНачалаПодготовки: Число, 4, 0
        ВидПрограммы: Строка, 30
        ПротоколУтвержденияНомер: Строка, 15
        ПротоколУтвержденияДата: Дата
    }
    class ДополнительныеДанные {
        <<справочник>>
        Ссылка: Строка
        Код: Строка, 9
        Наименование: Строка, 25
        Министерство: Строка, 150
        ОргФормаВуза: Строка, 150
        НаименованиеУниверситета: Строка, 150
        Полное: Строка, 150
        ДолжностьЛицаУтверждающегоРП: Строка, 30
        ФИОЛицаУтверждающегоРП: Строка, 25
        ГодУтвержденияРП: Строка, 4
        НаименованиеФакультета: Строка, 100
        КафедраУтверждающаяРП: СправочникСсылка.Кафедры
        КафедраСогласующаяРП: СправочникСсылка.Кафедры
        ПротоколУтвержденияРПНомер: Строка, 15
        ПротоколУтвержденияРПДата: Дата
        ДатаПервойАттестацииВесеннегоСеместра: Дата
        ДатаПервойАттестацииОсеннегоСеместра: Дата
        НормаСПС_Конт_Лек: Число, 5, 2
        НормаСПС_Конт_КР: Число, 5, 2
        НормаСПС_Конт_ПередЭкз: Число, 5, 2
        НормаСПСэкс_Конт_Сем: Число, 5, 2
    }
    class НаправленияПодготовки {
        <<справочник>>
        Ссылка: Строка
        Код: Строка, 9
        Наименование: Строка, 150
    }

    Дисциплины "1" -- "*" ФормируемыеКомпетенции
    Дисциплины "1" -- "*" Семестры
    ФормируемыеКомпетенции "1" -- "*" ТипКомпетенции
    ФормируемыеКомпетенции "1" -- "*" КафедрныйСправочник
    Семестры "1" -- "*" КафедрныйСправочник
    Семестры "1" -- "*" УчебныйПлан
    КафедрныйСправочник "1" -- "*" ИндикаторДостижения
    КафедрныйСправочник "1" -- "*" ДополнительныеДанные
    УчебныйПлан "1" -- "*" НаправленияПодготовки
    НаправленияПодготовки "1" -- "*" ДополнительныеДанные
    ФормированиеРП ..> * : УчебныйПлан
    
```

The diagram illustrates the following classes and their attributes:

- <<справочник>> Дисциплины**: Ссылка: Строка, Код: Строка, 20, Наименование: Строка, 150, ПорядковыйНомер: Строка, 30, ЗЕТ: Число, 3, 0, Часов: Число, 3, 0, Лек: Число, 3, 0, Лаб: Число, 3, 0, ПР: Число, 3, 0, СР: Число, 3, 0, Контроль: Число, 3, 0, НаименованиеВРодитПадеже: Число, 3, 0, СеместрыТех: Строка, 30, КодКафедры: Строка, 30, СправочникСсылка.Кафедры.
- ФормируемыеКомпетенции**: Ссылка: Строка, НомерСтроки: Число, Компетенция: СправочникСсылка.Компетенции.
- <<перечисление>> ТипКомпетенции**: УК, ОПК, ПК.
- <<справочник>> КафедрныйСправочник**: Ссылка: Строка, Код: Строка, 9, Наименование: Строка, 150, КраткоеОбозначение: Строка, 15, ЗавКафедрой: Строка, 30.
- <<справочник>> ИндикаторДостижения**: Ссылка: Строка, Код: Строка, 9, Наименование: Строка, 0.
- Семестры**: Ссылка: Строка, НомерСтроки: Число, НомерСемистра: Число, 2,0, зе_Сем: Число, 3, 0, Экзамен: Булево, Зачет: Булево, ЗачетСОценкой: Булево, КП: Булево, КР: Булево, Лек_Сем: Число, 3, 0, Лаб_Сем: Число, 3, 0, Пр_Сем: Число, 3, 0, СР_Сем: Число, 3, 0, Контроль_Сем: Число, 3, 0, СРС_Конт_Сем: Число, 5, 2, СРС_СамР_Сем: Число, 5, 2, СРСэкз_Конт_Сем: Число, 5, 2, СРСэкз_СамР_Сем: Число, 5, 2.
- <<обработка>> Загрузка**.
- <<обработка>> ФормированиеРП**: УчебныйПлан: СправочникСсылка.УчебныйПлан.
- <<справочник>> УчебныйПлан**: Ссылка: Строка, Код: Строка, 9, Наименование: Строка, 100, НаправлениеПодготовки: СправочникСсылка.НаправленияПодготовки, Направленность: Строка, 50, ГодНачалаПодготовки: Число, 4, 0, ВидПрограммы: Строка, 30, ПротоколУтвержденияНомер: Строка, 15, ПротоколУтвержденияДата: Дата.
- <<справочник>> ДополнительныеДанные**: Ссылка: Строка, Код: Строка, 9, Наименование: Строка, 25, Министерство: Строка, 150, ОргФормаВуза: Строка, 150, НаименованиеУниверситета: Строка, 150, Полное: Строка, 150, ДолжностьЛицаУтверждающегоРП: Строка, 30, ФИОЛицаУтверждающегоРП: Строка, 25, ГодУтвержденияРП: Строка, 4, НаименованиеФакультета: Строка, 100, КафедраУтверждающаяРП: СправочникСсылка.Кафедры, КафедраСогласующаяРП: СправочникСсылка.Кафедры, ПротоколУтвержденияРПНомер: Строка, 15, ПротоколУтвержденияРПДата: Дата, ДатаПервойАттестацииВесеннегоСеместра: Дата, ДатаПервойАттестацииОсеннегоСеместра: Дата, NormaSPC_Kont_Lek: Число, 5, 2, NormaSPC_Kont_KP: Число, 5, 2, NormaSPC_Kont_PerredEks: Число, 5, 2, NormaSPSeks_Kont_Sem: Число, 5, 2.
- <<справочник>> НаправленияПодготовки**: Ссылка: Строка, Код: Строка, 9, Наименование: Строка, 150.

Relationships are defined as follows:

- Disciplines (1) has many-to-many (*) relationships with FormableCompetences and Semesters.
- FormableCompetences (1) has one-to-many (*) relationships with TypeOfCompetence and FacultyReference.
- Semesters (1) has one-to-many (*) relationships with FacultyReference and AcademicPlan.
- FacultyReference (1) has one-to-many (*) relationships with AchievementIndicator and AdditionalData.
- AcademicPlan (1) has one-to-many (*) relationships with DirectionsOfPreparation and AdditionalData.
- FormationRP processes the AcademicPlan to generate it.

После импорта необходимых данных из учебного плана и ввода дополнительных данных должен быть подготовлен макет *Word*-документа с установленными «закладками». На рис 3 приведен пример страницы шаблона с «закладками». Закладками в шаблоне

обозначена та информация, которая формально может быть взята из учебного плана или дополнительных данных: информация обложки рабочей программы, таблица «Объем дисциплины (модуля) с распределением по семестрам», расписание самостоятельной работы студента (в том числе контактной работы), перечень компетенций, формируемых дисциплиной, и др.

Прикладная информатика (Учебный план)

Основное Дисциплины Компетенции

Дисциплины

Создать Создать группу

Поиск (Ctrl+F)

Наименование	Код	Владелец	Порядков...	ЗЕТ	Часов	Лек	Лаб	Пр	СР	Контроль	Код кафедры
Дисциплины											
Блок 1. Дисциплины (модули)	B1	Прикладная информатика	01								
Обязательная часть	B1.O	Прикладная информатика	01.01								
История (история России, всеобщая...)	B1.O.01	Прикладная информатика	01.01.B1.O.01	4	144	32		32	80		Юриспруденция, философия ...
Философия	B1.O.02	Прикладная информатика	01.01.B1.O.02	2	72	16		16	40		Юриспруденция, философия ...
Иностранный язык	B1.O.03	Прикладная информатика	01.01.B1.O.03	11	396			1...	200	36	Иностранные языки
Безопасность жизнедеятельности	B1.O.04	Прикладная информатика	01.01.B1.O.04	4	144	32	16	16	80		Общественные дисциплины
Физическая культура и спорт	B1.O.05	Прикладная информатика	01.01.B1.O.05	2	72	16		32	6	18	Физическое воспитание и спорт
Социология и психология	B1.O.06	Прикладная информатика	01.01.B1.O.06	3	108	16		16	76		Управление персоналом
Математика	B1.O.07	Прикладная информатика	01.01.B1.O.07	15	540	96		1...	156	144	Математика и математическо...
Физика	B1.O.08	Прикладная информатика	01.01.B1.O.08	10	360	64	32	32	142	90	Физика и фотоника
Введение в цифровые технологии	B1.O.09	Прикладная информатика	01.01.B1.O.09	4	144	16	32		60	36	Информационные и измерите...
Базы данных	B1.O.10	Прикладная информатика	01.01.B1.O.10	4	144	16	16	16	96		Информационные и измерите...

Дисциплины Дополнительные данные Кафедры Компетенции Направления подготовки Учебный план Сервис

Компетенции

Создать

Поиск (Ctrl+F)

Код	Владелец	Тип к...	Полное наименование
ОПК-1	Прикладная...	ОПК	Способен применять естественнонаучные и общетехнические знания, методы математического анализа и моделирования...
ОПК-2	Прикладная...	ОПК	Способен использовать современные информационные технологии и программные средства, в том числе отечественного...
ОПК-3	Прикладная...	ОПК	Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографическо...
ОПК-4	Прикладная...	ОПК	Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профес...
ОПК-5	Прикладная...	ОПК	Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем;
ОПК-6	Прикладная...	ОПК	Способен анализировать и разрабатывать организационно-технические и экономические процессы с применением метод...
ОПК-7	Прикладная...	ОПК	Способен разрабатывать алгоритмы и программы, пригодные для практического применения;
ОПК-8	Прикладная...	ОПК	Способен принимать участие в управлении проектами создания информационных систем на стадиях жизненного цикла;
ОПК-9	Прикладная...	ОПК	Способен принимать участие в реализации профессиональных коммуникаций с заинтересованными участниками проектн...
ПК-1	Прикладная...	ПК	Способен проводить предпроектное обследование объекта автоматизации и формирование концепции информационной с...

Формирование РП

Учебный план: Прикладная информатика

Путь к каталогу РП: N:\УЧЕБНЫЕ ПЛАНЫ 2019\

Сформировать шаблоны РП

Добавить Выделить все Снять все

Код	Дисциплина	Отметка
B1	Блок 1. Дисциплины (модули)	✓
B1.O	Обязательная часть	✓
B1.O.01	История (история России, всеобщая история)	✓
B1.O.02	Философия	✓
B1.O.03	Иностранный язык	✓
B1.O.04	Безопасность жизнедеятельности	✓
B1.O.05	Физическая культура и спорт	✓
B1.O.06	Социология и психология	✓

Рисунок 2– Экранные формы приложения

 [Ministry] [MainHead] [NameUni] УТВЕРЖДАЮ: [PositionProrector] [NameProrector] «__» _____ [Year] РАБОЧАЯ ПРОГРАММА [Disciplin] индекс и наименование дисциплины (модуля) (из учебного плана) [Direction] код и наименование направления подготовки (специальности), направленность (профиль) программа [IsBachOrMag] набор [YearRecuemen]г. Факультет [NameFaculcy] Кафедра [NameDepartamen] Курс [Course] Семестр [Semester] Итого по дисциплине [HourDis] (ЗЕ/час.) (с учетом ЗЕ/часов на экзамен) Новочеркасск: [YearDocument]г.	 [Ministry] [MainHead] [NameUni] УТВЕРЖДАЮ: Проректор по ОД Чеботарев С.Н. «__» _____ 2019г. РАБОЧАЯ ПРОГРАММА [B1.O.10 Базы данных] индекс и наименование дисциплины (модуля) (из учебного плана) «09.03.03 Прикладная информатика, направленность Прикладная информатика» код и наименование направления подготовки (специальности), направленность (профиль) программа академического бакалавриата набор 2019г. Факультет информационных технологий и управления Кафедра Информационные и измерительные системы и технологии Курс 2 Семестр 3 Итого по дисциплине 4/144 (ЗЕ/час.) (с учетом ЗЕ/часов на экзамен) Новочеркасск 2019г.
---	--

Рисунок 3– Пример страницы Word-макета с установленными «закладками» и страницы сформированного на его основе файла-шаблона рабочей программы дисциплины

На заключительном этапе для каждой из выбранных пользователем дисциплин инструментарий, выполняя запросы к необходимым объектам информационной базы данных, заполняет закладки макета рабочей программы данными из учебного плана и дополнительными данными, формирует шаблон рабочей программы, после чего производит сохранение файла в выбранную ранее пользователем папку. Далее преподаватель имеет возможность вручную дополнить Word-файл содержательной частью (наименование тем лекционных, лабораторных и практических занятий, темы для самостоятельного изучения и др.). Таким образом, инструментарий берет на себя формальные компоненты содержания рабочей программы, оставляя преподавателю больше времени на методическую и творческую работу над рабочей программой.

Созданный инструментарий позволяет работать с учебными планами разных направлений подготовки бакалавриата и магистратуры, исключить ошибки и несоответствие данным из учебного плана, оптимизировать и ускорить процесс формирования учебно-методической документации.

Библиографический список:

1. Широбокова С.Н., Щербакова Е.А., Кацупеев А.А., Евсин В.А. Математическая модель и программная реализация инструментария для автоматизированного формирования учебно-методической документации // Известия высших учебных заведений. Северо-Кавказский регион. Серия: Технические науки. 2016. № 3. С. 18-23.
2. Широбокова С.Н. Методика применения языка UML для построения моделей предметно-ориентированных экономических информационных систем на платформе «1С:Предприятие» // Наука и бизнес: пути развития. 2019. № 4. С. 200-210.

СЕКЦИЯ 8

Информационные технологии в ЭКОЛОГИИ



УДК 004.622

А.А. Вилинский, магистрант 2-го курса

Е.Н. Машенко канд. техн. наук, доцент

Севастопольский государственный университет

КЛАСТЕРНЫЙ АНАЛИЗ ДАННЫХ ЭКОЛОГИЧЕСКОГО МОНИТОРИНГА ПО ЗАГРЯЗНЕНИЮ АТМОСФЕРНОГО ВОЗДУХА

Аннотация: в данной статье рассматривается типовая задача анализа данных экологического мониторинга на примере показателей загрязнения атмосферного воздуха, решается задача кластеризации с целью выявления кластеров – уровней ПДК углекислого газа в воздухе, проводится анализ полученных результатов, делается вывод о целесообразности использования метода К-средних для решения такой задачи.

Ключевые слова: экологический мониторинг, загрязнение атмосферного воздуха, диоксид углерода, кластеризация, К-средних.

Annotation: this article deals with a common problem of analysing environmental monitoring data by solving clustering problem with the use of K-means method, further analysis is presented on the concentration norm of the carbon dioxide consistency in the air, conclusions are made on the matter of usage of the K-means method in this particular case.

Key words: environmental monitoring, air pollution, carbon dioxide, clustering algorithm, K-means.

В настоящее время значительной частью данных по экологическому мониторингу являются данные о концентрации определенных примесей в воздухе. Мобильные экологические станции делают замеры воздуха на постоянной основе, зачастую содержат детальные отчеты об абсолютных величинах определенных примесей в воздухе, а также их относительные доли предельной допустимой концентрации (ПДК). Типовая задача определения уровня опасности загрязнения воздуха (превышения ПДК) может решаться с помощью методов кластерного анализа.

Рассмотрим процесс анализа данных экологического мониторинга на примере реального набора данных из открытых источников. В качестве источника данных выберем портал открытых данных правительства Москвы [1], поскольку на данный момент он является наиболее объемным и по количеству данных, и по их вариативности [2].

В качестве конкретного набора данных исследуем набор о среднемесячных показателях загрязнения атмосферного воздуха [3].

Фрагмент набора данных выглядит следующим образом (Рисунок 1):

#	Период измерения	Название станции	Характеристика зоны наблю...	Адресный ориентир	Измеряемый параметр	Среднее значение в абсолют...	Среднее значение в долях П...
11	12.2018	Кокухово	Смешанные антропогенные источники загрязнений	город Москва, Лухмановская улица, дом 34	Сероводород	0.0021	
12	12.2018	Кокухово	Смешанные антропогенные источники загрязнений	город Москва, Лухмановская улица, дом 34	Диоксид серы	0.0009	0.02
13	12.2018	Новососино	Смешанные антропогенные источники загрязнений	город Москва, Саттыковская улица, дом 7, корпус 2	Сумма углеводородных соединений за вычетом метана	0.16	
14	12.2018	Новососино	Смешанные антропогенные источники загрязнений	город Москва, Саттыковская улица, дом 7, корпус 2	Сумма углеводородных соединений	1.54	
15	12.2018	Новососино	Смешанные антропогенные источники загрязнений	город Москва, Саттыковская улица, дом 7, корпус 2	Метан	1.39	

Рисунок 1 – Набор данных о загрязнении атмосферного воздуха

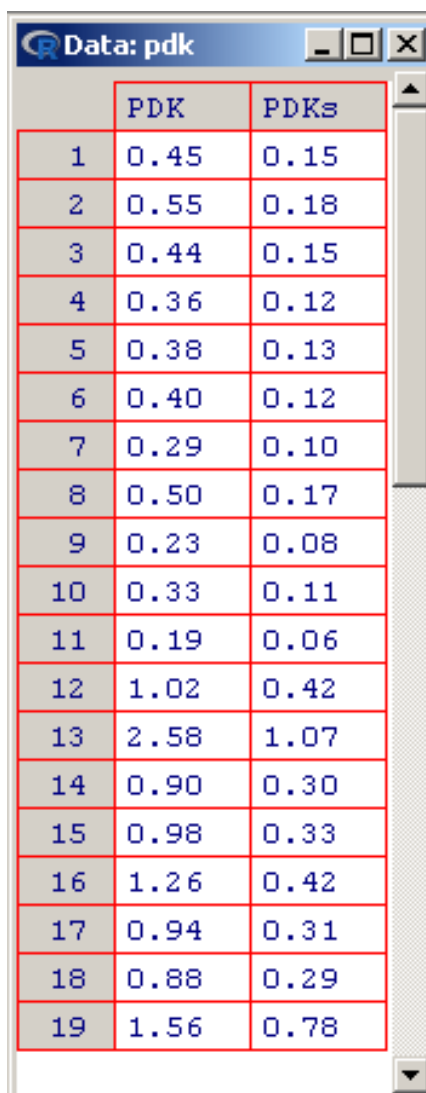
Для демонстрации процесса обработки таких данных возьмем наиболее часто встречаемое химическое соединение в таких данных – оксид углерода, который является чрезвычайно токсичным и опасным. Отсортируем выборку по дате замера, и возьмем 50 самых свежих значений.

В исследовании будут использоваться два столбца: абсолютное измеряемое значение и относительная доля ПДК для данного замера.

Государственная инспекционная служба выделяет четыре уровня опасности загрязнения атмосферного воздуха, каждый уровень соответствует определенным долям ПДК по измеряемому параметру. Зная это, мы можем предположить, что количество кластеров для такой выборки данных равно четырем.

Проведем кластеризацию набора данных с использованием языка статистической обработки R. Среда R Studio предоставляет возможность гибкого использования множества пакетов для статистической обработки данных, а также графического отображения результатов. Особое внимание следует уделить тому факту, что в R нет скаляров, любое число и любой объект интерпретируется средой как вектор, поэтому некоторые методы кластеризации и обработки данных в R работают гораздо быстрее чем в аналогичных средах статистической обработки.

Импортируем наш .csv набор данных в R (Рисунок 2):



	PDK	PDKs
1	0.45	0.15
2	0.55	0.18
3	0.44	0.15
4	0.36	0.12
5	0.38	0.13
6	0.40	0.12
7	0.29	0.10
8	0.50	0.17
9	0.23	0.08
10	0.33	0.11
11	0.19	0.06
12	1.02	0.42
13	2.58	1.07
14	0.90	0.30
15	0.98	0.33
16	1.26	0.42
17	0.94	0.31
18	0.88	0.29
19	1.56	0.78

Рисунок 2 – Набор данных в представлении R Studio

Первый столбец соответствует абсолютным величинам, второй – их относительным долям ПДК.

Для проведения кластеризации воспользуемся классическим методом К-средних. Данный метод отлично показывает себя на данных, в которых прослеживается линейная зависимость переменных.

Основным «подводным камнем» данного алгоритма является выбор количества кластеров, которое является входным параметром алгоритма, однако с количеством кластеров мы уже определились, выдвинув предположение о том, что их должно быть четыре.

Проведем кластеризацию с использованием пакета ClusteR (Рисунок 3):

```
> result<-kmeans(pdk,4)
> result
K-means clustering with 4 clusters of sizes 22, 21, 5, 1

Cluster means:
      PDK      PDKs
1 0.3527273 0.1168182
2 0.7928571 0.2700000
3 1.3380000 0.5440000
4 2.5800000 1.0700000

Clustering vector:
[1] 1 1 1 1 1 1 1 1 1 1 1 2 4 2 2 3 2 2 3 3 2 2 2 2 2 2 1 1 1 1 2 2 3 2 3 2 1 1
[39] 1 2 2 1 1 1 1 2 2 2 2

Within cluster sum of squares by cluster:
[1] 0.4799136 0.3696286 0.2222000 0.0000000
(between_SS / total_SS = 90.2 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"       "
```

Рисунок 3 – Кластеризация исходного набора данных в R

Вектор кластеров показывает, к какому из кластеров принадлежит объект набора данных (измерение оксида углерода). Выведем результат в виде двумерного графика (Рисунок 4).

На графике измерения, принадлежащие к разным кластерам, отмечены разным цветом. Можно заметить, что прослеживается четкая тенденция: измерения с относительным ПДК от 0 до 0.2 относятся к первому кластеру, от 0.2 до 0.4 ко второму, от 0.4 до 0.8 к третьему, и оставшееся измерение, которое превышает ПДК, относится к четвертому кластеру. Исключением является одно измерение, которое должно принадлежать к третьему кластеру, но алгоритм определил его во второй кластер.

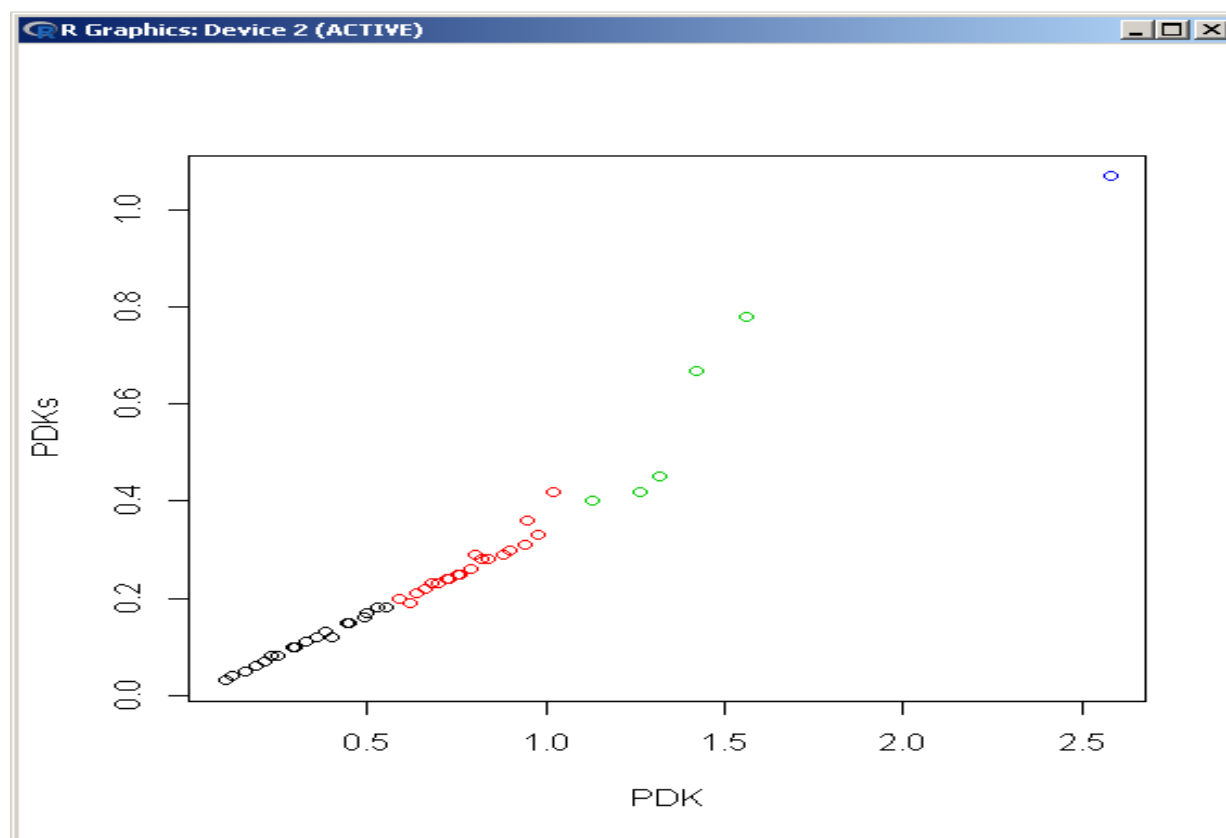


Рисунок 4 – Распределение измерений по кластерам

Для того чтобы понять, правильно ли алгоритм определил кластеры, добавим к исходному набору данных информацию об уровне опасности.

Затем проведем кросс-сравнение таблиц по построенным кластерам и колонке с уровнем опасности, полученным из открытого набора данных (Рисунок 5).

```
> table(pdk1$DangerLevel, result$cluster)
```

	1	2	3	4
1	22	0	0	0
2	0	20	0	0
3	0	1	5	0
4	0	0	0	1

Рисунок 5 – Сравнение кластеров двух наборов

Как видно из рисунка 5, в третьем кластере находится измерение, которое алгоритм отнес ко второму кластеру, хотя в реальности данное измерение относится к третьему кластеру, так как его абсолютная величина равна 0.42, что означает, что его следует отнести к третьему уровню опасности («Умеренное загрязнение атмосферного воздуха»).

Таким образом реальное распределение объектов по кластерам должно было выглядеть так (Рисунок 6):

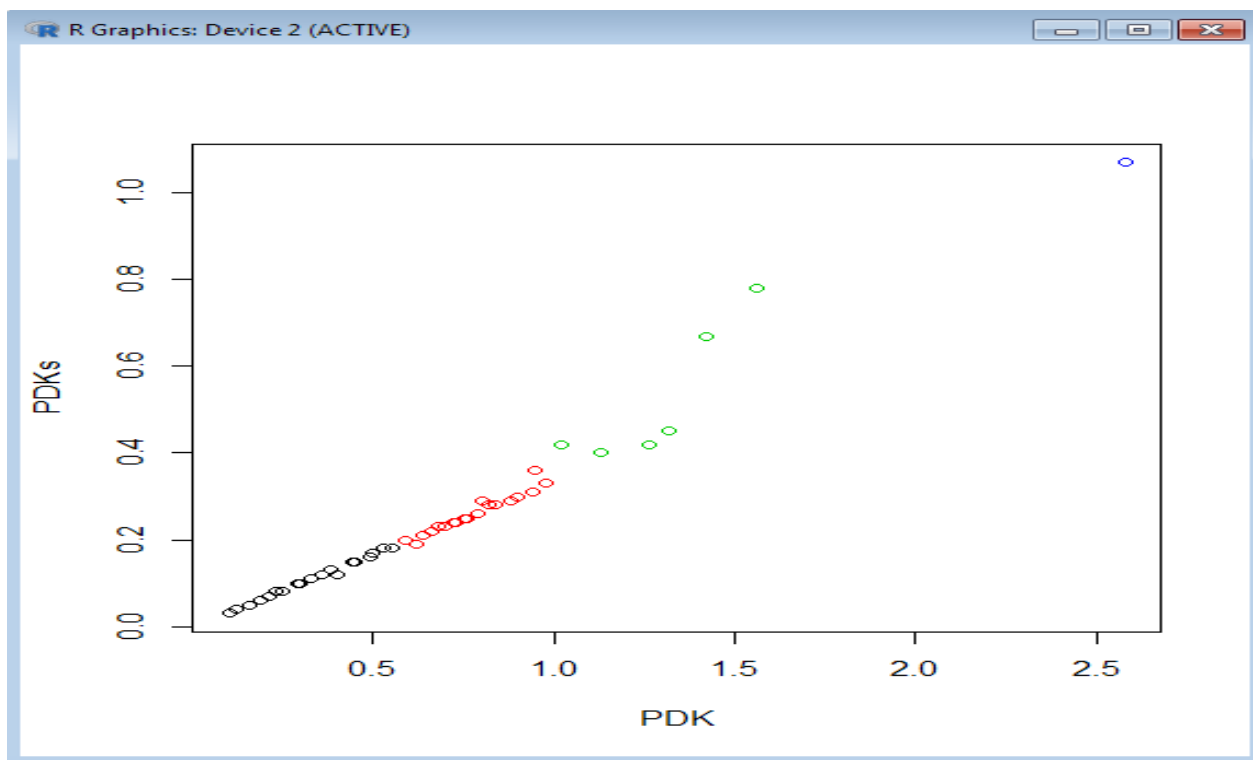


Рисунок 6 – Итоговое распределение измерений по кластерам

Закключение. Исходя из проведенных исследований, можно сделать вывод, что на этом наборе данных алгоритм К-средних показал хороший результат, с относительной погрешностью в 2.04%, однако следует отметить, что это не всегда так. Как уже отмечалось ранее, слабая сторона данного алгоритма заключается в том, что нам необходимо самим задавать начальное число кластеров, что не всегда возможно сделать, так как необходимо более детальное исследование данных перед тем, как можно будет сделать вывод о числе кластеров. Также минусом алгоритма является сильная чувствительность к шумам, которые могут сильно сдвигать центры кластеров, из-за чего процент неправильных занесений объектов в кластеры будет увеличиваться.

Исследование выполнено при финансовой поддержке РФФИ и города Севастополь в рамках научного проекта № 18-47-920005\19.

Библиографический список:

1. Портал открытых данных правительства Москвы [Электронный ресурс]. 2019. URL: <https://data.mos.ru/> (дата обращения 14.04.19).
2. А.А. Вилинский, Е.Н. Машенко. Сравнительный анализ открытых источников данных по экологическому мониторингу. // Мир компьютерных технологий: Сборник статей всероссийской научно-технической конференции студентов, аспирантов и молодых ученых, г. Севастополь, 2 – 5 апреля 2019 г/ М-во образования и науки РФ, Севастопольский государственный университет; науч. ред. Е.Н. Машенко – г. Севастополь: СевГУ, 2019.
3. Среднемесячные показатели загрязнения атмосферного воздуха [Электронный ресурс]. 2019. URL: <https://data.mos.ru/opendata/2453? pageNumber=1> (дата обращения 16.04.19).

СЕКЦИЯ 9

Программное и
аппаратное обеспечение
прикладных областей



УДК 004.4'273

Д.В. Бутов, магистрант

О.В. Ченгарь, канд. техн. наук, доцент

Севастопольский государственный университет

КОНЦЕПТУАЛЬНАЯ 3D-МОДЕЛЬ АВТОНОМНОЙ БЕСЕДКИ С АЛЬТЕРНАТИВНЫМ ИСТОЧНИКОМ ЭНЕРГИИ

Аннотация: В статье рассмотрено создание концептуальной модели в среде формирования трёхмерных объектов 3Ds MAX. Сформирован полный трёхмерный концепт автономной беседки с альтернативным источником энергии, проанализированы аспекты построения подобных моделей и их целесообразность.

Ключевые слова: концептуальная модель, 3Ds MAX, трёхмерные объекты, концептуальные схемы.

Annotation This article will consider the creation of a conceptual modeling environment for the formation of three-dimensional objects 3Ds MAX. A complete three-dimensional concept of an Autonomous gazebo with an alternative energy source will be fully formed. Also will be considered the advantages of forming such models, and their feasibility. *Key words:* conceptual model Creation, 3Ds MAX, three-dimensional objects, conceptual schemes.

На сегодняшний день возникает необходимость создания новых эргономичных систем, различных форм, размеров и функциональных возможностей. Формируется огромное количество запросов общества, которые необходимо использовать и учитывать в различных разработках. Для представления таких проектов и их одобрения заказчиком простого текстового описания и формирования технического задания недостаточно, необходима визуализация проекта, т.е. разработка графической концептуальной модели, на примере которой можно рассмотреть архитектурные решения, возможности системы. В таких случаях зачастую используются различные среды трехмерного графического моделирования, такие как: 3Ds MAX, MAYA, Cinema 4D и т.д. Эти платформы позволяют создать концептуальную трёхмерную модель, которая покажет внешний вид и, частично, функциональные возможности разрабатываемой системы [1]. Также на основе построенной модели можно точно понять, какие модули необходимо добавить, а какие исключить, ввиду архитектурных сложностей или нецелесообразности.

В данной статье предложена концептуальная модель автономной беседки с альтернативным источником энергии, построенная в среде 3Ds MAX, просчитывается возможность установки в данную конструкцию дополнительных модулей.

Из анализа, выполненного в предыдущей работе [2], в качестве среды разработки для данного концепта была выбрана среда моделирования 3Ds Max, так как именно эта платформа обладает необходимым набором библиотек и инструментов для разработки проектов подобного типа [3,4].

При создании систем данного образца необходимо построить схему работы всей установки, а также создать первичный вариант всей конструкции, предварительно разбив всю модель на стационарные и дискретные части, так как при создании концептуальной модели необходимо чётко понимать, какие функции она будет выполнять. Была создана структурная схема проектируемого объекта (рисунок 1), которая подробно описывает все элементы конструкции.

На схеме чётко видны составные части проектируемого объекта. В начале работы необходимо определиться с источником питания. В данной конструкции в качестве питающего элемента использованы солнечные батареи.

Солнечная батарея – фотоэлектрический преобразователь (фотоэлемент) полупроводниковых устройств, которые способны преобразовывать солнечную энергию в постоянный ток [5,6]. В настоящий момент максимальный КПД солнечных панелей составляет 66%. Так мощность солнечного потока на квадратный метр составляет около 1366 Ватт, с современными аккумуляторами получится снять в среднем 902 Ватт.

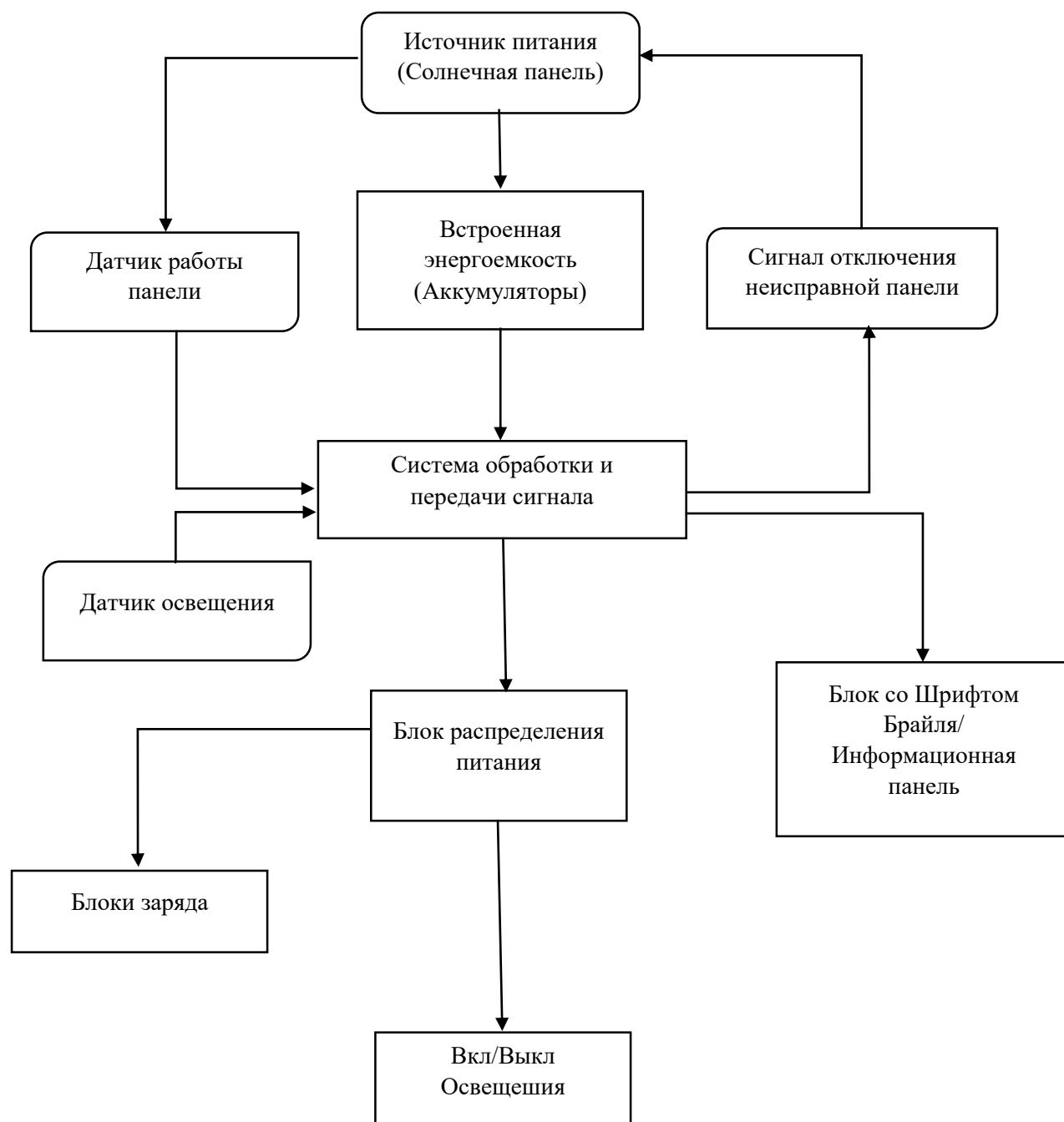


Рисунок 1 - Структурная схема проектируемого объекта

Данный блок будет питать беседку, давая возможность использовать её автономно от городских сетей электроснабжения. Также такое решение даёт возможность установки подобных конструкций в места, где затруднительно или экономически не выгодно проводить линии электроснабжения. Т.е. такой подход предоставит возможность устанавливать подобные конструкции, не затрагивая исторические объекты или памятники.

При разработке формы панелей необходимо было использовать дополнительное программное обеспечение, а именно среду CorelDraw. На данной платформе были подготовлены геометрические схемы (рисунок 2 и рисунок 3) элемента питания, что дало четкое представление формы всей панели.

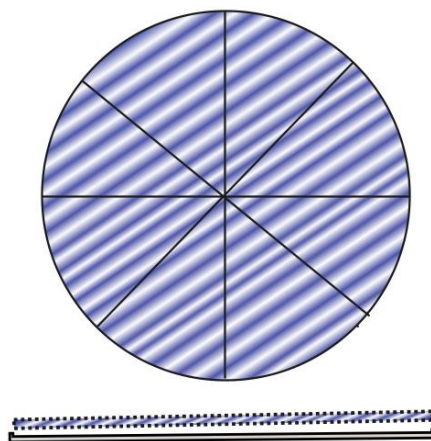


Рисунок 2 – Фронтальный и горизонтальный вид верхней солнечной панели меньшего диаметра

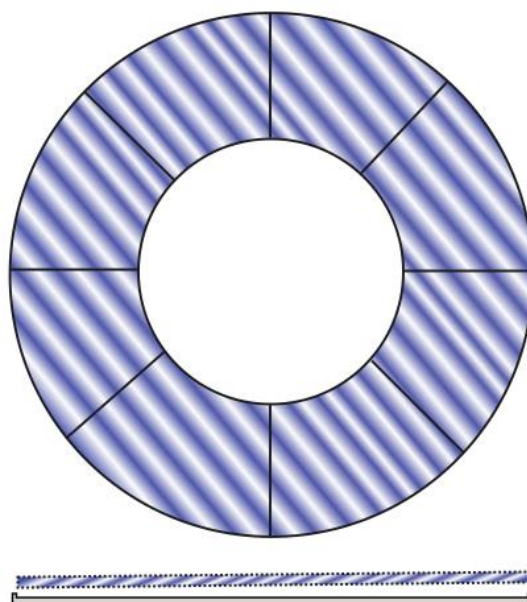


Рисунок 3 – Фронтальный и горизонтальный вид нижней солнечной панели большего диаметра

Как видно из рисунка 2 и 3, данные панели должны быть разделены на сегменты, что даст возможность устройству функционировать даже в том случае, если какая-то из частей будет повреждена или перекрыта тенью, листвой и т.д. Данное решение немного уменьшит рабочую площадь панели, однако придаст ей большую стойкость к повреждениям, а также увеличит срок возможной бесперебойной работы. Такой подход облегчит проблему ремонта данных установок, а именно даст возможность снять повреждённый сегмент и поставить новый, не подвергая разборке всю конструкцию.

Образец сегмента солнечной панели также был сформирован в среде трёхмерной разработки 3Ds MAX (рисунок 4).

Для разрабатываемой конструкции можно использовать световые панели военного образца, которые способны заряжаться от любого источника света, в том числе и от лунного. Такие панели используются на беспилотных летательных аппаратах [7].

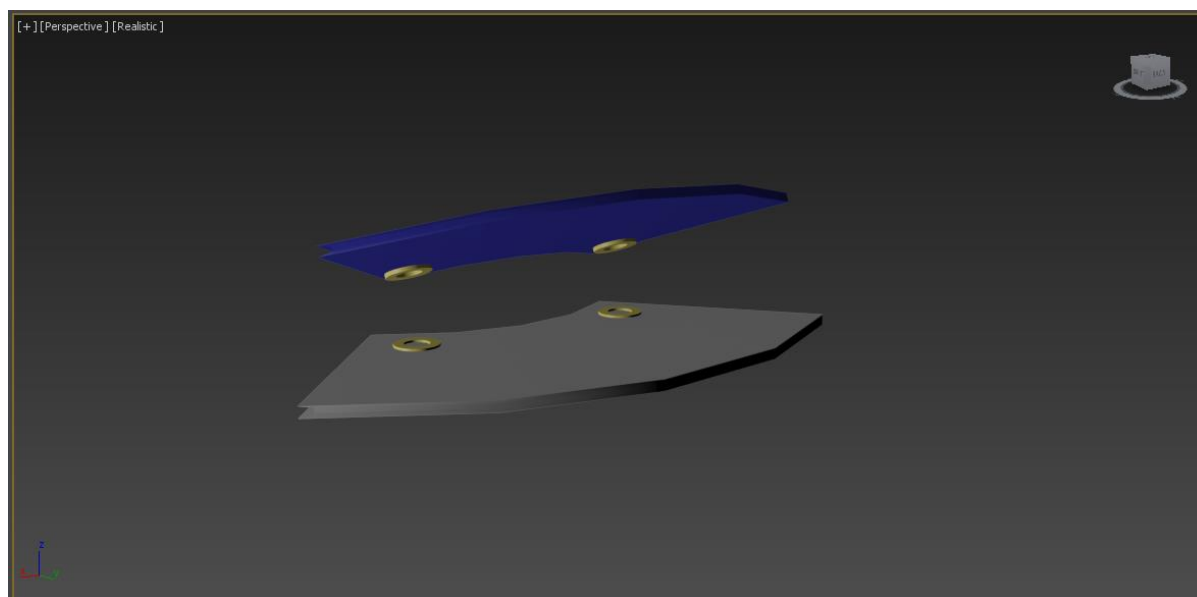


Рисунок 4 – Концептуальная трёхмерная модель сегмента солнечной панели

После определения концепции и формирования схематического представления источника питания было продолжено формирование модели всей конструкции, опираясь на ранее сформированную схему концептуальной модели. Так на рисунках 5 и 6 представлено схематическое представление фронтальной и боковой проекции разрабатываемой модели соответственно.

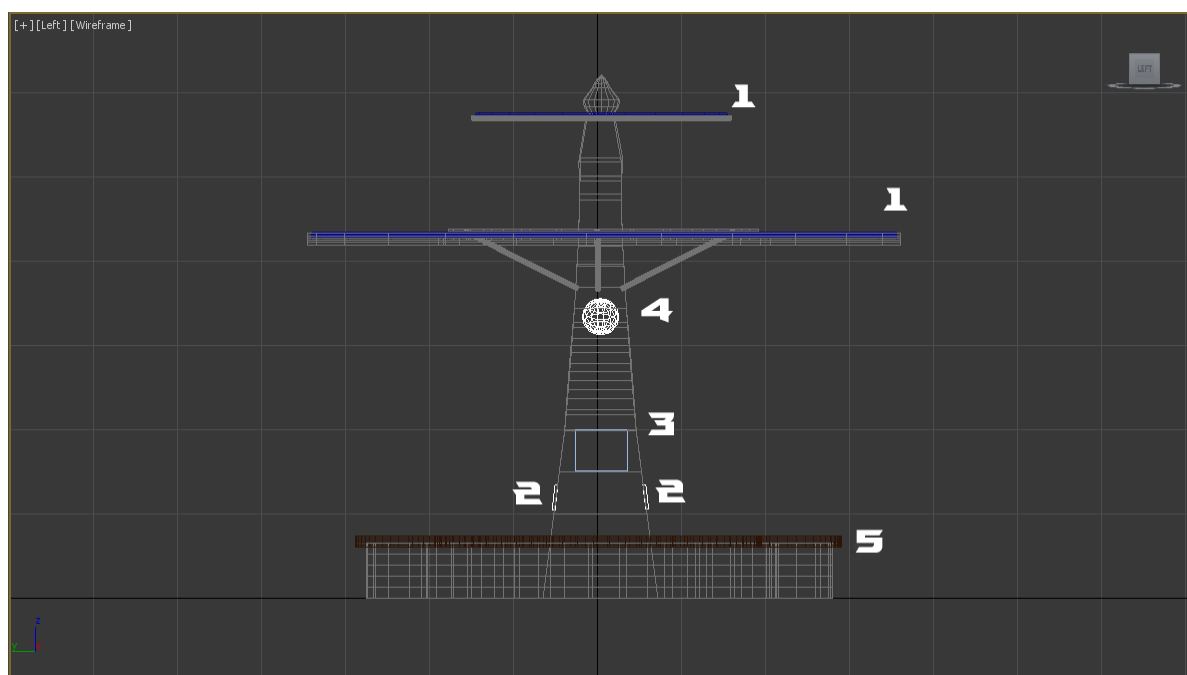


Рисунок 5 – Фронтальная проекция концептуальной модели

Визуальное решение было выбрано с учётом современных архитектурных тенденций, а именно перехода на футуристический дизайн, а также из-за особенностей предложенного источника питания данной конструкции [8].

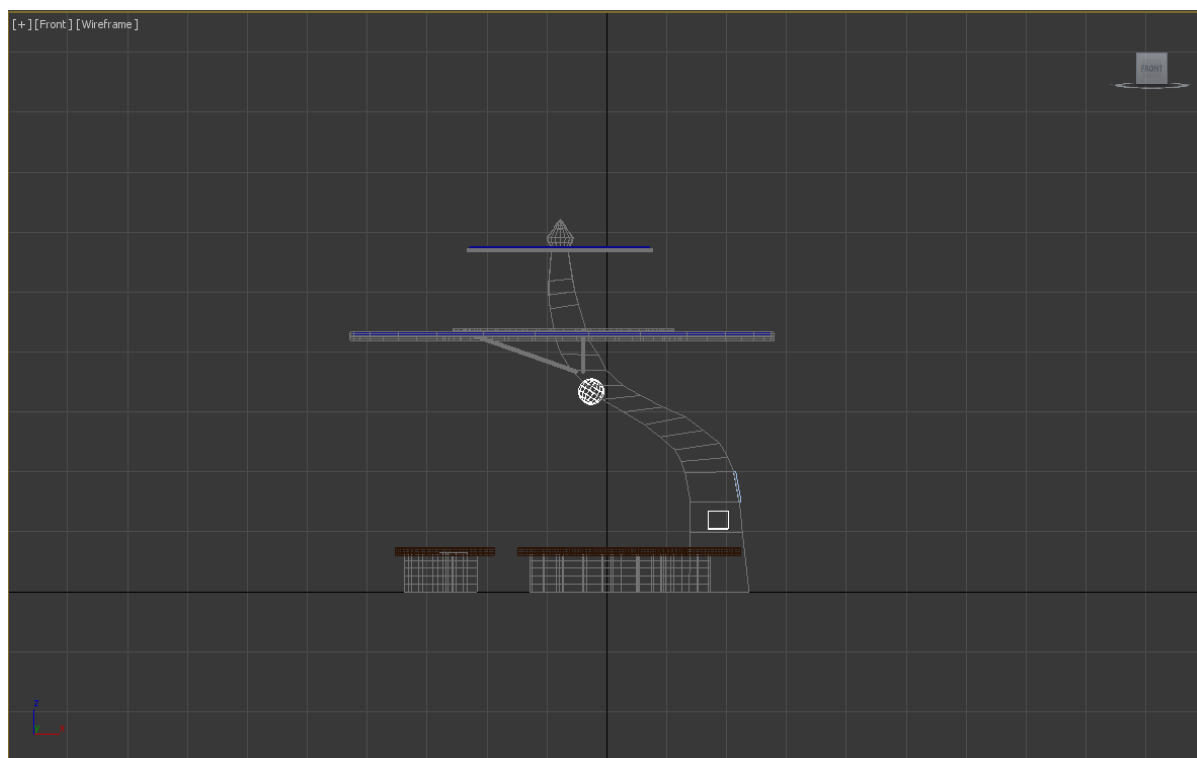


Рисунок 6 – Схематическое представление боковой проекции конструкции

Создание схематического представления конструкции сопровождается построением трёхмерного представления всей модели, из которого можно формировать дискретные элементы модели. Вся модель была построена с помощью Standard Primitives [3] а также Extended Primitives [9]. Все использованные примитивы подвергались векторному изменению формы, а также полигональной подгонке размеров и форм, после чего был реализован процесс сглаживания и группирования фигур. Так, на рисунке 7 представлено схематическое отображение верхней проекции беседки.

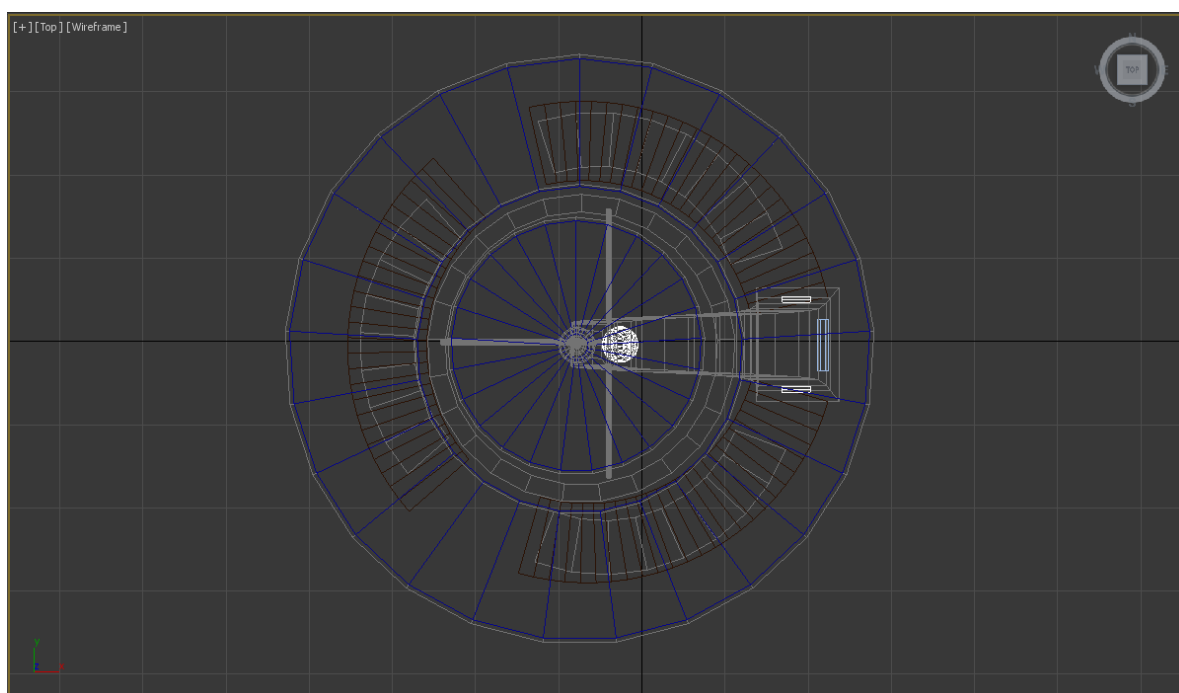


Рисунок 7 – Схематическое представление верхней проекции беседки

Согласно представленной на рисунке 1 структурной схеме можно проследить, что в конструкции предусматривается также и установка освещения.

Для тёмного времени суток беседку можно оснащать фонарями или диодными линиями. Также такие модели можно снабдить фонарями, которые будут освещать площадь по периметру конструкции. Это даст возможность не устанавливать стандартные фонари, которые работают от общей сети электроснабжения. Такое решение позволит экономить электроэнергию города в среднем на 5-7% в год, что уменьшит потребление ресурсов на ТЭЦ в среднем на 10% в год. На рисунках 8 и 9 представлена визуализация конструкции в темное время суток без использования освещения и с дополнительными фонарями и диодной подсветкой.

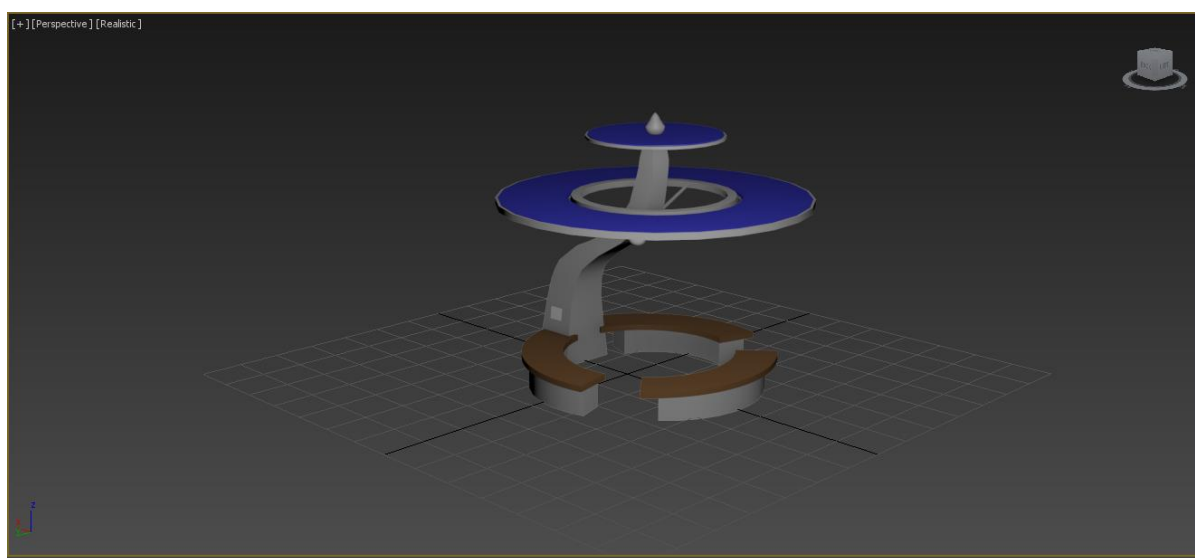


Рисунок 8 – Модель беседки без освещения в темное время суток

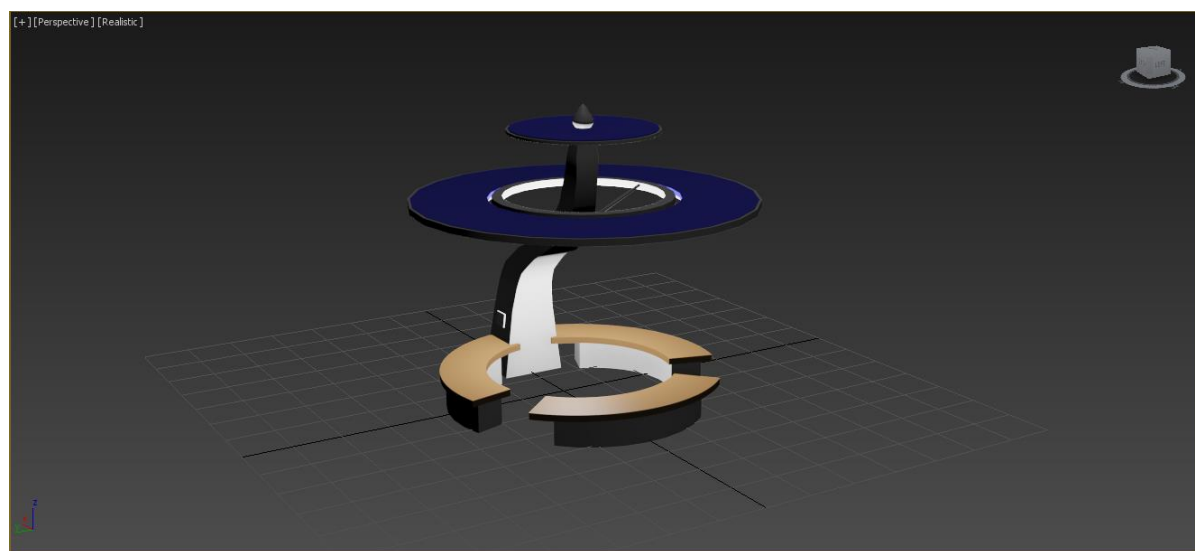


Рисунок 9 – Модель беседки с освещением в тёмное время суток

Также, согласно структурной схеме (рисунок 1), разработанная конструкция предполагает установку некоторых дополнительных элементов, представленных на рисунке 10.

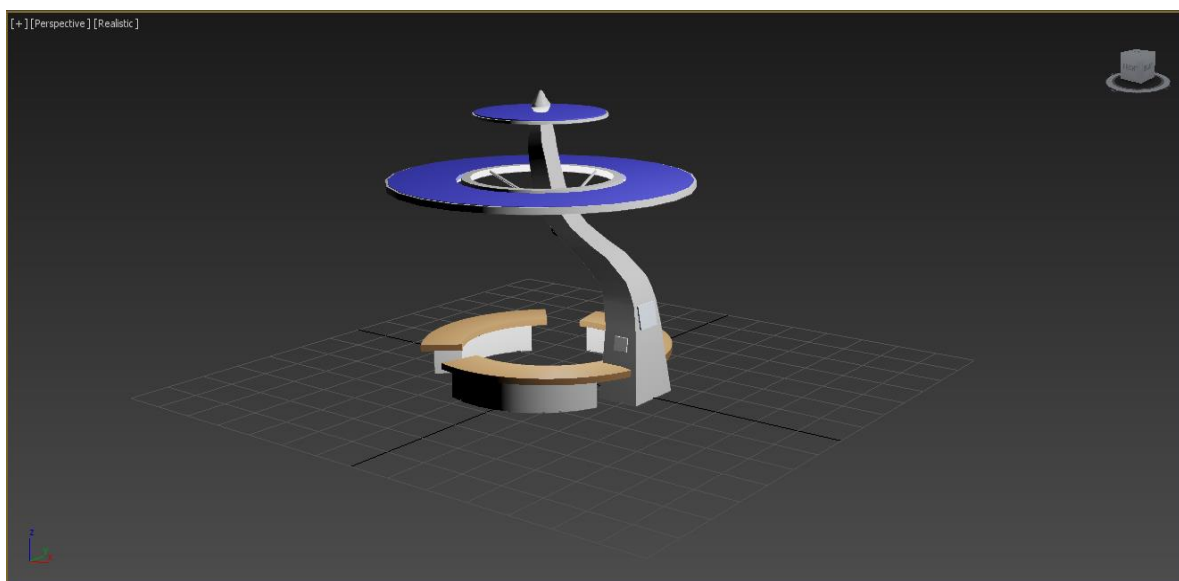


Рисунок 10 – Размещение дополнительных модулей на конструкции

К таким дополнительным элементам относятся следующие:

- 1) Блоки заряда гаджетов (выходы USB).

Данный пункт решает проблему постоянного использования гаджетов современным обществом. В наше время быть постоянно «на связи» или «в сети» очень важно, и в момент, когда заряд аккумулятора иссякает, создаются определённые трудности для пользователя. Данная установка позволит избежать подобной проблемы. Для этого предлагается использование блоков из пары разъемов (2х - USB), установленных на стенках мест для сидения. Также в конструкции предусмотрены различные виды входов и разъёмов, включая «европейскую вилку».

- 2) Панель со шрифтом Брайля, также можно установить экран «городского проводника».

Шрифт Брайля, или «язык слепых», крайне необходим в данной конструкции, так как в современном мире обязательно должно уделяться внимание людям с ограниченными возможностями и данный проект не стал исключением. На данную конструкцию может быть установлена отчеканенная пластина с описанием места, в котором находится человек, или объёмное отображение окружения (возможна совместная установка), которое видит человек с данного ракурса. Это даст возможность людям с ограниченными возможностями увидеть достопримечательности города, и «рассмотреть» каждую деталь места, в котором он находится.

К сожалению, в настоящий момент не существует устройства, которое бы автоматически формировало надписи данным шрифтом, однако ведутся активные работы в этом направлении [10]. Поэтому сейчас Европейский Союз финансирует разработку устройства «Anagraphs», которое можно будет подключать к электронной книге или компьютеру. Устройство будет формировать шрифт Брайля и выводить на экран до 6 000 осязаемых точек.

Кроме того, совместно с предыдущим решением возможна установка цифрового экрана (планшета), с информацией о месте, в котором человек находится, а также с путеводителем по городу в виде карты или специально разработанного приложения. Таким образом, можно обеспечить человека необходимой информацией о месте, в котором он находится, а также о городе в целом. Также рядом можно установить аудио разъёмы, что

позволит прослушать всю возможную экскурсионную информацию либо подробное описание данного места, опять же для людей с ограниченными возможностями.

3) Сидения.

Так как разрабатываемая конструкция позиционируется как беседка, её необходимо снабдить местами для сидения, а именно блоками из полимерных материалов с лакированными деревянными брусками на верхней плоскости. Это позволит людям комфортно сидеть как в холодное, так и в тёплое время года.

4) Крепления.

Так как конструкция предполагается автономной и переносной, для нее необходимо предусмотреть небольшие «крючки», для крепления к плоскости, для предотвращения переворотов от сильного ветра или неадекватных посетителей.

5) Wi-Fi роутер.

Также необходимо предусмотреть установку точки беспроводной передачи данных (роутера), что позволит людям без мобильного интернета быть в сети. Это увеличит покрытие сети в населённом пункте и его отдалённых частях.

6) Дополнительные устройства.

Также на данную беседку можно добавить систему оповещения, или дисплей, на котором будет отражаться время и температура воздуха.

7) Система самотестирования.

Подобное решение упростит поиск неисправностей или поломок во время эксплуатации беседки и ускорит возможный ремонт всей системы.

Для полного описания разработки концепта также обязательно следует обосновать выбор материалов необходимых для реализации данной модели.

Для создания полноценной конструкции на основе построенной трёхмерной модели следует учитывать расчёты, полученные в ходе предыдущих тестов [8]. Для самой модели следует использовать лёгкие и легкосплавные материалы, что обеспечит прочность и одновременно лёгкость всей конструкции. К таким материалам относится большинство сплавов алюминия. Например, в данном случае, предлагается использование алюминий-цинкового сплава. Такой сплав даст необходимую прочность изделию, а цинковое покрытие даст возможность конструкции противостоять коррозии. Так же такое решение удешевит всю конструкцию, и обеспечит её лёгкость и ремонтпригодность. На местах для сидения лучше использовать лакированные бруса хвойных деревьев, так как данные породы дерева славятся своим долгим циклом службы и дешевизной. Основы мест для сидений, а также места для установки аккумуляторов, стоит собирать из полимерных материалов, например, пластика. Такой слой пластика обеспечит прочность конструкции, а дополнительный слой обеспечит герметичность батарей. Аккумулирующую составляющую обеспечат литий-ионные аккумуляторы, благодаря своим ёмкостным качествам, а также большим циклам перезаряда, и относительно небольшой стоимостью.

В заключение следует отметить, что при разработке подобных конструкций трёхмерная модель является вполне обоснованной необходимостью, т.к. она позволяет учесть множество различных факторов, которые становятся очевидными только при полноценной визуализации разработки. Также на подобных моделях возможно производить экспериментальное тестирование, за счет переноса всей конструкции в среду физического движка (игрового движка). Благодаря концептуальной модели, построенной в трёхмерной среде моделирования, можно также понять направления развития всего проекта в целом. Так, например, можно «улучшить» панель со шрифтом Брайля, дав ей возможность менять

надписи путем смены пластин, или сделать всю таблицу состоящей из маленьких частей, подобно пикселям, которые будут выдвигаться согласно алфавиту шрифта.

Библиографический список:

1. Большаков, В. 3D-моделирование в AutoCAD, КОМПАС-3D, SolidWorks, Inventor, T-Flex / В. Большаков, А. Бочков, А. Сергеев. - М.: Книга по Требованию, 2010. - 336 с.
2. Бутов Д.В., Ченгарь О.В. Анализ современного состояния и тенденций развития игровых движков и 3D-редакторов // Сборник статей всероссийской студенческой НТК "Мир компьютерных технологий". – Севастополь: Федеральное государственное автономное образовательное учреждение высшего образования "Севастопольский государственный университет". 2019. С.219-223.
3. Пекарев Л. Самоучитель 3DS Max 5. СПб.: БХВ, 2003.
4. Полевой 3D Studio MAX 3 для профессионалов (+CD) / Полевой, Роб. - М.: СПб: Питер, 2001. - 848 с.
5. Солнечная энергетика: учебное пособие для вузов /под ред. Виссарионова В.И., М.: изд. дом МЭИ, 2008.
6. Челяев В. Ф. Солнечная энергетика - энергетика будущего. «Энергия: экономика, техника, экология», № 10, 2008.
7. Миролубов И.Н. и др. Пособие по решению задач по сопротивлению материалов: учебное пособие для технических вузов. – М. : Высшая школа, 2007. – 399 с.
8. Феодосьев В.И. Сопротивление материалов: Учебник для студентов высших технических учебных заведений – 10-е изд., переработанное и дополненное – М.: Изд-во МГТУ им. Н.Э. Баумана, 2008. – 588 с.
9. Алямовский, А.А. SolidWorks 2007/2008. Компьютерное моделирование в инженерной практике / А.А. Алямовский. - М.: СПб: БХВ-Петербург, 2008. - 192 с.
10. Зиновьевич М.В., Дудок Т.Г. Исследование процесса нанесения шрифта Брайля методом вакуумного формования // Труды БГТУ. Серия 4: Принт- и медиатехнологии. 2016. №9 (191). URL: <https://cyberleninka.ru/article/n/issledovanie-protssessa-naneseniya-shrifta-braylya-metodom-vakuumnogo-formirovaniya> (дата обращения: 23.12.2019).

УДК 681.3

С.Н. Дзюба, магистрант

С.Н. Фисун, канд. техн. наук, доцент

Севастопольский государственный университет

СРАВНЕНИЕ ЭФФЕКТИВНОСТИ АЛГОРИТМОВ ВОСХОДЯЩЕГО СИНТАКСИЧЕСКОГО АНАЛИЗА

Аннотация: статья посвящена исследованию эффективности алгоритмов восходящего синтаксического анализа, используемых для разработки современных информационных систем. Описаны экспериментальные исследования двух алгоритмов восходящего синтаксического анализа в зависимости от длины входной строки и объема исходной грамматики.

Ключевые слова: синтаксический анализ, разработка компиляторов, восходящий синтаксический анализ, LR(1), SLR(1).

Annotation: the article is devoted to the study of the effectiveness of the algorithms for upward syntactic analysis used to develop information systems in machine-building. Experimental studies of two algorithms for upward syntax analysis are described depending on the length of the input string and the volume of the original grammar.

Key words: parsing, compilers development, upward parsing, LR(1), SLR(1).

Тенденции современного мира заключаются в автоматизации любой выполняемой человеком ручной работы. Построение блока синтаксического анализа для компилятора и не только включает в себя значительное количество ручной работы, подчиняющейся строгим алгоритмам. Потому разработка программной системы автоматизации построения синтаксических анализаторов весьма актуальна.

Эффективная работа программной системы автоматизации построения синтаксических анализаторов позволит сократить время, затрачиваемое на разработку блока синтаксического анализа, что неизбежно ведет в экономической выгоде.

Целью исследования в данной статье является сравнение эффективности работы восходящих синтаксических анализаторов путем измерения зависимости времени обработки восходящим синтаксическим анализатором входной строки от ее длины и от объема исходной грамматики.

Программная система автоматизации построения синтаксических анализаторов осуществляет генерацию восходящего синтаксического анализатора SLR(1) или LR(1).

В ходе проведения экспериментов будут также сравниваться эффективности разбора входной строки этих алгоритмов между собой.

В обязательном порядке в ходе проведения экспериментов должен учитываться «прогрев» JVM [1].

В ходе проведения экспериментов на вход программы будет подана строка, соответствующая фрагменту грамматики учебного языка программирования ASPLE-3, который представляет оператор цикла с предусловием while. В сравнении с ранее проведенными экспериментами для нисходящих анализаторов LL(1)[2], грамматика претерпела незначительные изменения и выглядит следующим образом:

1. $S \rightarrow w v d \text{ Seq } e$
2. $\text{Seq} \rightarrow \text{Op} \mid \text{Op} ; \text{Seq}$
3. $\text{Op} \rightarrow o \mid \text{Op} \rightarrow S$

Входная строка будет изменяться в длине за счет добавления в нее вложенных операторов цикла от 10 до 196 618 терминальных символов. Для исключения влияния «прогрева» JVM, распознавание входной строки будет проводить 500 000 раз для каждого из выбранных значений длины входной цепочки в вышеупомянутом диапазоне, из которых будут приниматься последние 50 000 результатов, от которых будет взято среднее арифметическое.

В качестве грамматики с большим количеством правил взята грамматика учебного языка программирования StudPL, которая представляет описание класса с переменными, условными операторами и операторами цикла:

1. classDeclaration -> className BEGIN classBody END
2. className -> qualifiedName
3. classBody -> sequenceStatement
4. sequenceStatement -> statement | statement sequenceStatement
5. statement -> variableDeclaration | assignment | whileStatement | ifStatement | expression
6. variableDeclaration -> VARIABLE name EQUALS expression
7. assignment -> name EQUALS expression
8. ifStatement -> IF (expression) BEGIN sequenceStatement END | IF (expression (BEGIN sequenceStatement END ELSE BEGIN sequenceStatement END
9. whileStatement -> WHILE (expression) BEGIN sequenceStatement END
10. name -> ID
11. expression -> factor | expression + factor | expression - factor
12. factor -> primary | factor * primary | factor / primary
13. primary -> ID | value | (expression) | (comparison)
14. comparison -> expression EQU expression | expression NEQ expression
15. variableReference -> ID
16. value -> -NUMBER | NUMBER | BOOL | STRING
17. qualifiedName -> ID | . ID | epsilon

Пример кода, соответствующий данной грамматике, выглядит следующим образом:

```
test{
    var t="t"
    if (c!=3){
        t="adsf"
        while(af){
            var a=34
            while (asd){
                c=3
            }
        }
    }
    else {
        var c=5
    }
}
```

Подход к проведению эксперимента не изменился: учитывается прогрев JVM, на вход синтаксического анализатора подается строка вида “ ID BEGIN [WHILE (ID) BEGIN VARIABLE ID EQUALS ID * ID END] [IF (ID) BEGIN VARIABLE ID EQUALS ID + ID END] END”, длина которой увеличивается за счет добавления вложенных операторов условия и цикла (заклучены в квадратные скобки для удобства восприятия).

Сравним результаты экспериментов, проведенных для восходящих синтаксических анализаторов SLR(1) и LR(1) над фрагментом грамматики учебного языка программирования ASPLE-3 и грамматики StudPL. Сводный график по четырем экспериментам представлен на рисунке 1.

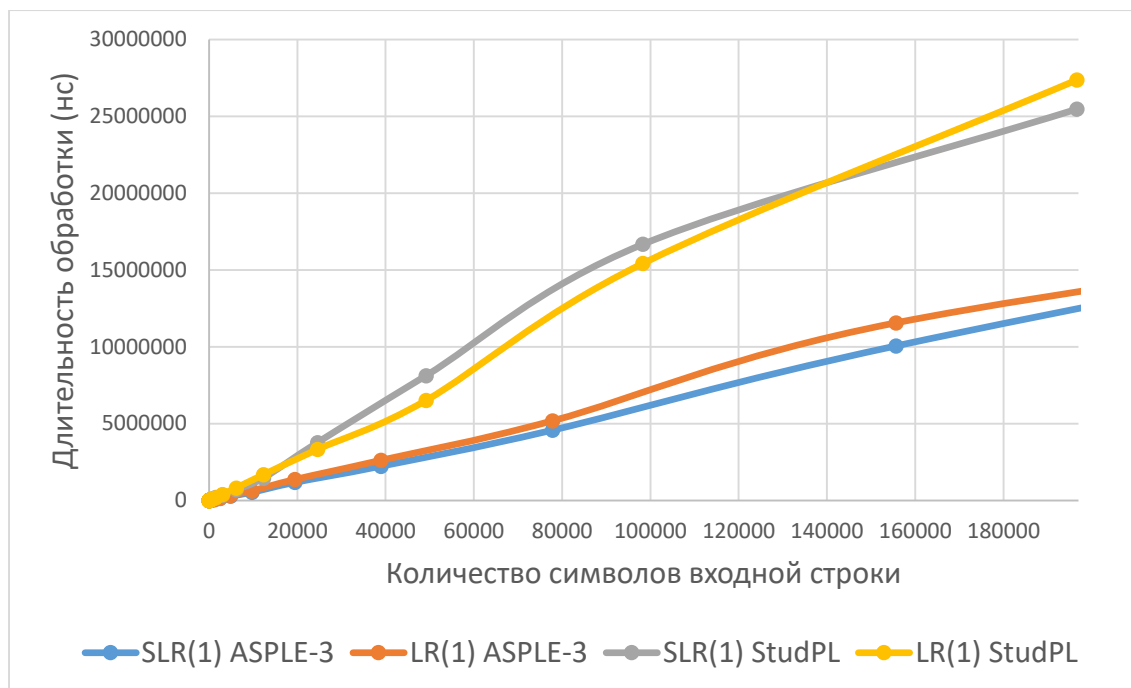


Рисунок 1 – Сводный график результатов экспериментов над восходящими синтаксическими анализаторами

На рисунке 1 кривые под названием «SLR(1) APSLE-3» и «LR(1) ASPLE-3» соответствуют результатам экспериментов с грамматикой ASPLE-3, а кривые «SLR(1) StudPL» и «LR(1) StudPL» соответствуют результатам измерений для учебной грамматики StudPL, приближенной к реальной. Наблюдается влияние объема исходной грамматики, а также длины входной цепочки на длительность обработки синтаксическими анализаторами входной строки, что связано с тем, что размер управляющих таблиц тем больше, чем больше правил в грамматике и, соответственно, больше элементов в управляющих таблицах синтаксического анализа влечет за собой увеличение количества выполняемых действий и переходов. Также стоит отметить, что эффективность алгоритмов восходящего синтаксического разбора LR(1) и SLR(1) находится на приблизительно одинаковом уровне, несмотря на отличающиеся алгоритмы.

Программный комплекс автоматизации построения восходящих синтаксических анализаторов позволяет генерировать синтаксические анализаторы для значительного объема контекстно-свободных грамматик, принадлежащих классу LR(1) – грамматик. При разработке исходной грамматики следует помнить, что ее объем влияет на эффективную работу синтаксического анализатора.

Библиографический список:

1. Дзюба С.Н., Фисун С.Н. Исследование влияния виртуальной Java-машины на оценки производительности информационных систем в машиноприборостроении // Научный журнал «Автоматизация и измерения в машино-приборостроении. Выпуск №1». - Севастополь, 2018. – С. 90 - 94.
2. Дзюба С.Н., Фисун С.Н. Исследование методов автоматизации построения нисходящих синтаксических анализаторов // Сборник статей Всероссийской студенческой научно-технической конференции «Мир компьютерных технологий». – Севастополь, 2018. – С. 149 – 152.

УДК 681.3

С.Н. Фисун, канд. техн. наук, доцент

Д. Ю. Омельченко, магистрант

Севастопольский государственный университет

ПРОГРАММНАЯ СИСТЕМА ДЛЯ АВТОМАТИЗАЦИИ РАЗРАБОТКИ НИСХОДЯЩИХ СИНТАКСИЧЕСКИХ АНАЛИЗАТОРОВ

Аннотация: в статье рассматриваются особенности построения программной системы автоматизации разработки нисходящих синтаксических анализаторов. Рассматриваются алгоритм функционирования универсального синтаксического анализатора в рамках программного комплекса, а также вопросы ограничения входных данных и выбора методик выполнения поставленной задачи проектирования системы.

Ключевые слова: трансляция, преобразования, синтаксический анализ, контекстно-свободная грамматика, факторизация, управляющая таблица.

Annotation: The article discusses the features of building a software system for automating the development of top-down parsers. We consider the functioning algorithm of a universal parser in the framework of the software package, as well as the issues of limiting input data and choosing methods for fulfilling the task of designing a system.

Key words: translation, transformations, parsing, context-free grammar, factorization, control table.

Развитие информационных технологий привело к тому, что во многих специализированных областях применяются различные программные решения. К таким решениям относятся методы автоматизации разработки специального программного обеспечения на базе специфических языков программирования, разрабатываемых под ряд определенных задач.

Процесс предварительных преобразований грамматических структур формальных языков является сложным алгоритмическим процессом со значительным количеством различных преобразований и ограничений. Проектирование автономных систем преобразования исходных грамматик в рамках одного множества формальных языков является важным этапом построения автономных систем разработки программного обеспечения.

Целью разработки рассматриваемой автономной системы синтаксического анализа с предварительным преобразованием исходных грамматик является получение эффективного метода построения нисходящих синтаксических анализаторов с минимальным участием пользователя системы.

Для достижения поставленной цели по разработке программной системы автоматизации построения нисходящих синтаксических анализаторов, поставленная задача проектирования была разделена на несколько специальных подзадач:

- разработка и реализация алгоритмов предварительного преобразования исходной грамматики;
- моделирование разработанных алгоритмических решений на тестовых наборах;
- проектирование программной системы преобразования грамматических структур к заданному типу языковых форм;
- разработка синтаксического анализатора для типовых грамматических форм языков;
- выполнение тестирования разработанного программного продукта на тестовых наборах;
- оценка эффективности использования программной системы для применения в системах автоматизации разработки специального программного обеспечения.

В процессе выполнения поставленной задачи были рассмотрены различные методы анализа и преобразования грамматик контекстно-свободных языков. Были выделены три основных подхода реализации синтаксического анализа грамматик:

- Рекурсивный спуск;

В процессе работы системы сохраняется возможность автономного доступа к отдельным модулям программы. Обмен информацией между ними происходит посредством файловой системы IBM PC. Входными данными разрабатываемой системы являются исходная грамматика в формальном представлении и анализируемая строка лексем. Выходные данные системы: автономный синтаксический анализатор с рядом управляющих переменных, преобразованная грамматика, управляющая таблица нисходящего анализатора и заключение о грамматике и анализируемой строке входных данных.

На основании проведенного системного анализа разрабатываемой программной системы спроектирована схема взаимодействия подсистем программы, в процессе передачи данных и управления программным комплексом. Рассматриваемая программная система основывается на децентрализованном принципе управления без прямого вмешательства пользователя программы. Схема взаимодействия подсистем программной системы представлена на рисунке 2. Она отображает порядок работы отдельных элементов системы на основании циклического алгоритма функционирования программы.

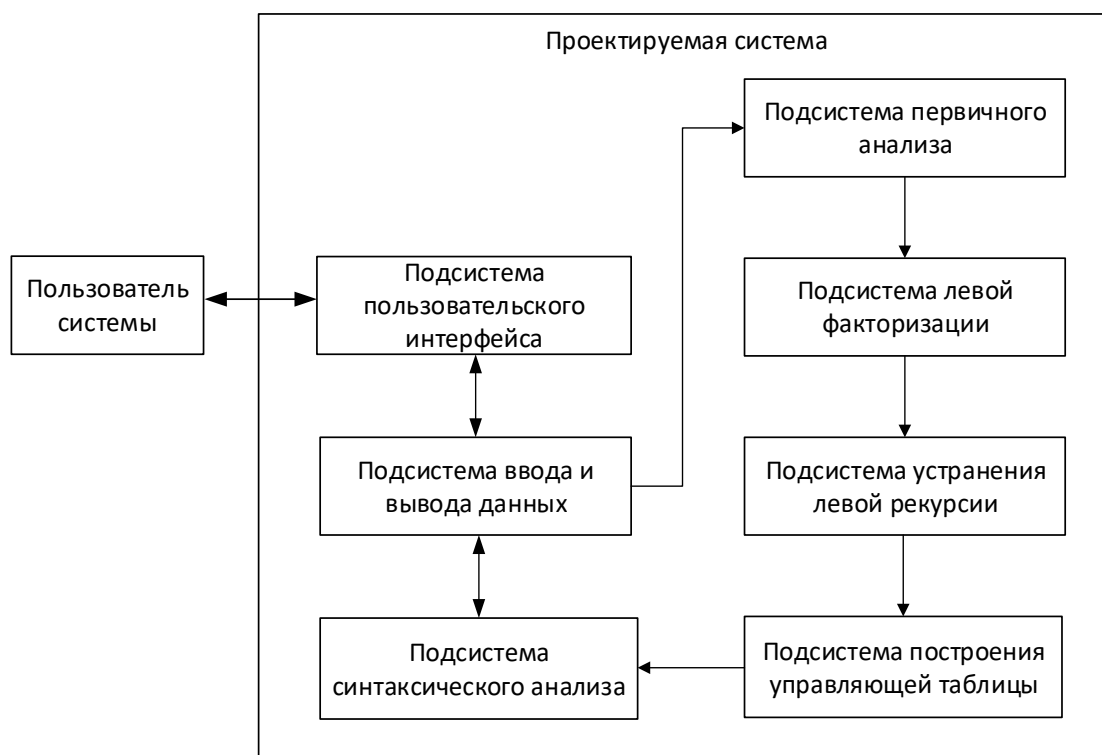


Рисунок 2 – Схема взаимосвязи подсистем программы

Программная система синтаксического анализа входной строки является отдельной подсистемой программного продукта, в рамках которого реализуется алгоритм разбора входной строки лексем с использованием управляющих таблиц грамматики. В процессе выполнения анализа выполняется проверка соответствия рассматриваемого символа рабочего стека ожидаемому правилу грамматики.

Процесс анализа выполняется путем поэтапного разбора символов входной строки лексем для очередного предложения или оператора с использованием двух стеков данных: рабочего и выходного. В результате функционирования программы формируется дерево разбора или выходной стек и заключение о принадлежности строки лексем анализируемой грамматике.

Если строка лексем не принадлежит грамматике, то подсистема синтаксического анализа формирует информацию об ошибке в наглядном виде. Затем анализ продолжается, начиная с первого символа следующего оператора. Схема алгоритма программной системы

для автоматизации построения нисходящих синтаксических анализаторов представлена на рисунке 3. Программная система разработана с использованием современного объектно-ориентированного языка программирования Java и реализована на ПЭВМ типа IBM PC, работающей под управлением операционной системы Windows.

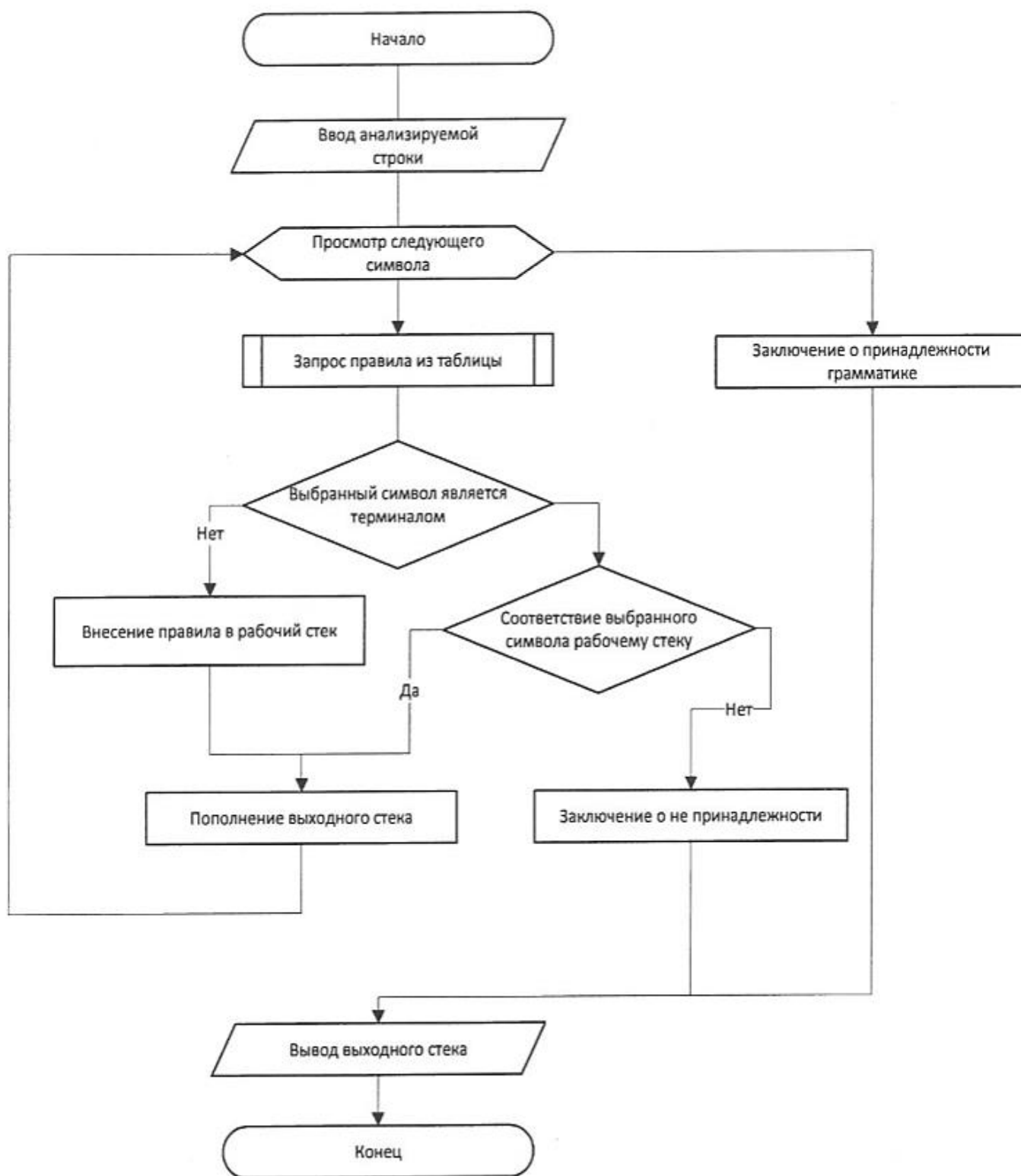


Рисунок 3 – Алгоритм выполнения синтаксического анализа

В процессе разработки, отладки и тестирования программной системы проведено исследование методов нисходящего синтаксического анализа контекстно-свободных грамматик при разработке специального программного обеспечения. Основными особенностями рассматриваемой программной системы являются:

- высокая степень автоматизации построения синтаксических анализаторов;
- значительное сокращение временных затрат на разработку синтаксического анализатора в сравнении с ручной разработкой;
- более высокая эффективность по сравнению с моделями конечного автомата с магазинной памятью.

Разработанная программная система испытана на реальных тестовых примерах и использована в учебном процессе при выполнении лабораторных работ по дисциплине «Системное программное обеспечение».

Библиографический список:

1. Ахо А. Компиляторы: принципы, технологии и инструментарий // А. Ахо, М. Лам, Р. Сети, Дж. Ульман. – М.: ООО «Вильямс», 2008. - 1184 с.
2. Омельченко Д.Ю., Фисун С.Н. Автоматизация преобразования грамматик для детерминированных нисходящих синтаксических анализаторов // Сборник статей Всероссийской студенческой научно-технической конференции «Мир компьютерных технологий». – Севастополь, 2019. - С. 255 - 259.
3. Дзюба С.Н., Фисун С.Н. Исследование методов автоматизации построения нисходящих синтаксических анализаторов // Сборник статей Всероссийской студенческой научно-технической конференции «Мир компьютерных технологий». – Севастополь, 2018. – С. 149 – 152.

УДК 004.

Ю. А. Цокало, магистр 2 курса¹

Е. Н. Машенко, канд. техн. наук, доцент¹

И. Е. Цокало, канд. фарм. наук, старший преподаватель²

¹Севастопольский государственный университет

²Крымский федеральный университет имени В. И. Вернадского, Медицинская академия имени С.И. Георгиевского

ЭКСПЕРТНАЯ СИСТЕМА ОБРАБОТКИ РЕЗУЛЬТАТОВ ХИМИЧЕСКИХ ЭКСПЕРИМЕНТОВ

Аннотация: Рассматривается экспертная система проверки однородности выборки и исключения выпадающих значений для химических экспериментов, построен алгоритм экспертной системы, который основан на общей фармакопейной статье.

Ключевые слова: экспертная система, однородность выборки, исключение выпадающих значений, обработка результатов эксперимента, метрологические измерения

Annotation: a expert system for checking the uniformity of samples and eliminating outliers for chemical experiments, the construction of an algorithm for an expert system based on a general pharmacocopy article are considered

Key words: expert system, sample uniformity, processing experiment results, dropout exception, metrological measurements.

Метрологические измерения применяются для определения погрешностей измерений, а методы используются для достижения их единства и требуемой точности. А применяя экспертную систему для расчета теории погрешности измерений основанной на использовании аппарата теории вероятности и математической статистики.

В фармацевтической практике в экспериментальных исследованиях очень широко используются аналитические и технологические методики, которые позволяют проводить оценку качества сырья для фармацевтического производства [1]. Однако объем проводимых исследований требует большого количества расчетов для обработки данных эксперимента.

Целью данного исследования является построение экспертной системы для автоматизации процесса расчетов при проверке однородности выборки, а также исключения выпадающих значений (выбросов) в данных при обработке результатов химических экспериментов в задаче оценки качества сырья для фармацевтической промышленности.

В связи с этим возникает необходимость в применении специализированных методов обработки данных [1], которые позволят провести оценку и сравнение методик аналитического эксперимента, а также проанализировать исследуемые при таком эксперименте объекты, решить ряд прикладных задач.

В данной статье приняты следующие обозначения:

f – число степеней свободы;

i – порядковый номер варианты;

n – объемы выборки;

$\bar{P}$, P – доверительная вероятность соответственно при дву- и односторонней постановке задачи;

Q_1 , Q_n – контрольные критерии идентификации грубых ошибок;

R – размах варьирования;

r – коэффициент корреляции;

s – стандартное отклонение;

s^2 – дисперсия;

$s_{\bar{x}}$ – стандартное отклонение среднего результата;

μ – истинное значение измеряемой величины.

Опишем алгоритм работы экспертной системы для проверки однородности выборки и исключения выпадающих значений варианта.

Предположим, что на вход мы подаем отдельные значения варианта выборки объема n . Обозначим выборку как x_i ($1 \leq i \leq n$) и расположены в порядке возрастания. Результаты, полученные при статистической обработке, будут верны только при условии, если эта выборка однородна, т. е. если в выборку не будут входить варианты грубых ошибок допущенные при измерении или расчете. Подобные варианты необходимо исключить из выборки перед вычислением статистических характеристик. На рисунке 1 приведен алгоритм для выборки небольшого объема ($n < 10$). Алгоритм составлен на основе общей фармацевтической статьи 42-0111-09 [1, 2].

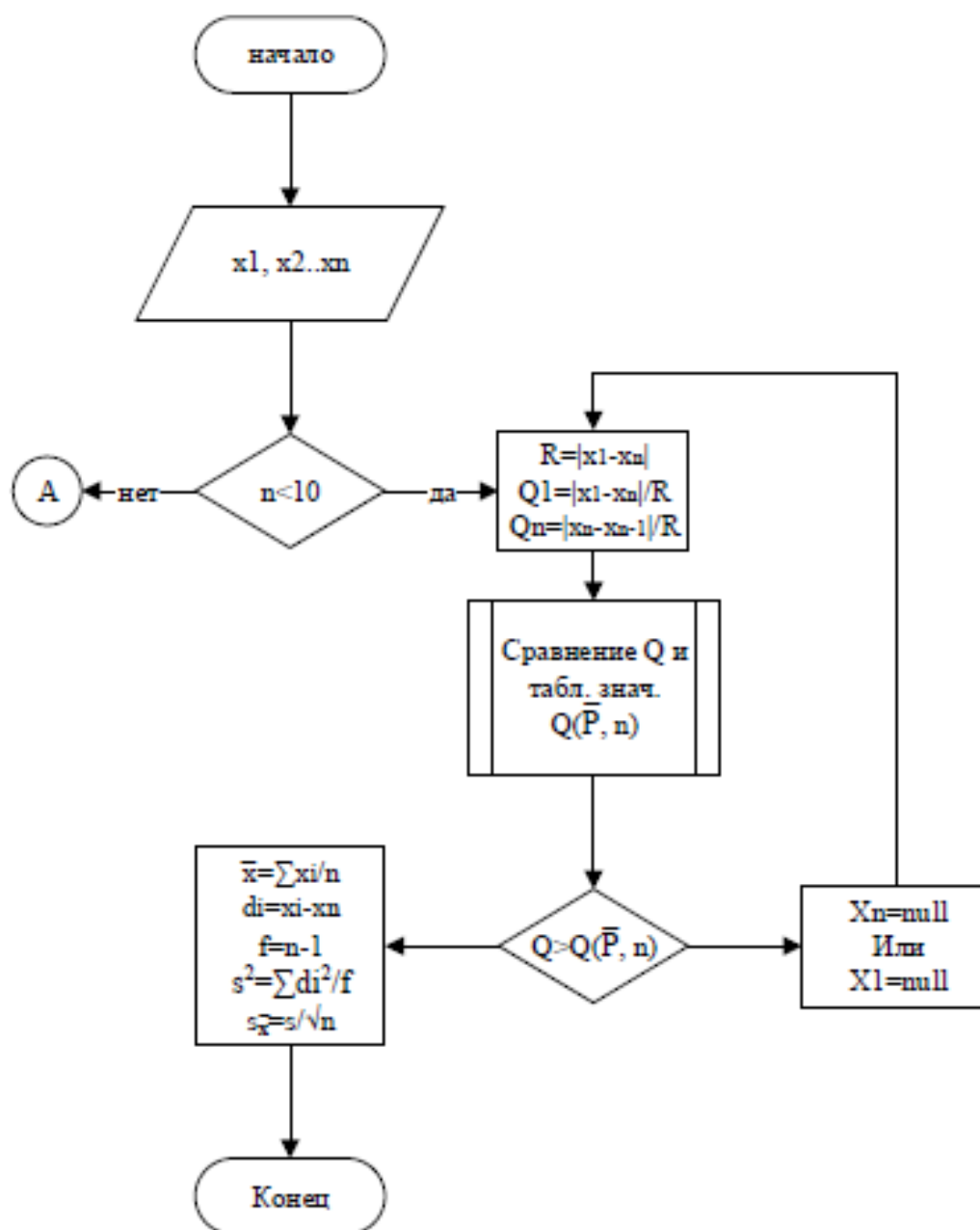


Рисунок 1 – Алгоритм проверки однородности для выборки малого объема ($n < 10$).

На вход подается выборка $n < 10$. Для данной выборки проводится проверка на величину ошибки, исходя из величины размаха варьирования R , а также расчет значений контрольного критерия Q .

Выборка признается неоднородной, если хотя бы одно из вычисленных значений Q превышает табличное значение $Q(\bar{P}, n)$, найденное для доверительной вероятности $\bar{P}$ (см. таблицу 1).

Таблица 1 – Критические значения контрольного критерия $Q(P, n)$

n	Q		
	P=90%	P=95%	P=99%
3	0.89	0.94	0.99
4	0.68	0.77	0.89
5	0.56	0.64	0.76
6	0.48	0.56	0.70
7	0.43	0.51	0.64
8	0.40	0.48	0.58
9	0.38	0.46	0.55

Варианты x_1 или x_n , для которых соответствующее значение $Q > Q(\bar{P}, n)$, отбрасываются, и для полученной выборки уменьшенного объема выполняют новый цикл вычисления с целью проверки ее однородности. Полученная в конечном счете однородная выборка используется для вычисления $\bar{x}$, s^2 , s и $s_{\bar{x}}$ [1].

Для идентификации выборки большого объема ($n \geq 10$) необходимо проводить предварительную статистическую обработку всей выборки, полагая ее однородной, и уже затем, на основании найденных статистических характеристик, решать вопрос о справедливости сделанного предположения об однородности. Алгоритм процесса идентификации приведен на рисунке 2.

Статистическая обработка для нахождения среднего значения выборки. В большинстве случаев среднее по выборке $\bar{x}$ является наилучшей оценкой истинного значения измеряемой величиной μ , если его вычисляют как среднее арифметическое всех вариантов. При этом разброс вариантов x_i вокруг среднего $\bar{x}$ характеризуется величиной стандартного отклонения s . В количественном химическом анализе s часто рассматривается как оценка случайной ошибки, свойственной данному методу анализа. Величина дисперсии s^2 может рассматриваться как мера воспроизводимости результатов, представленных в данной выборке [1].

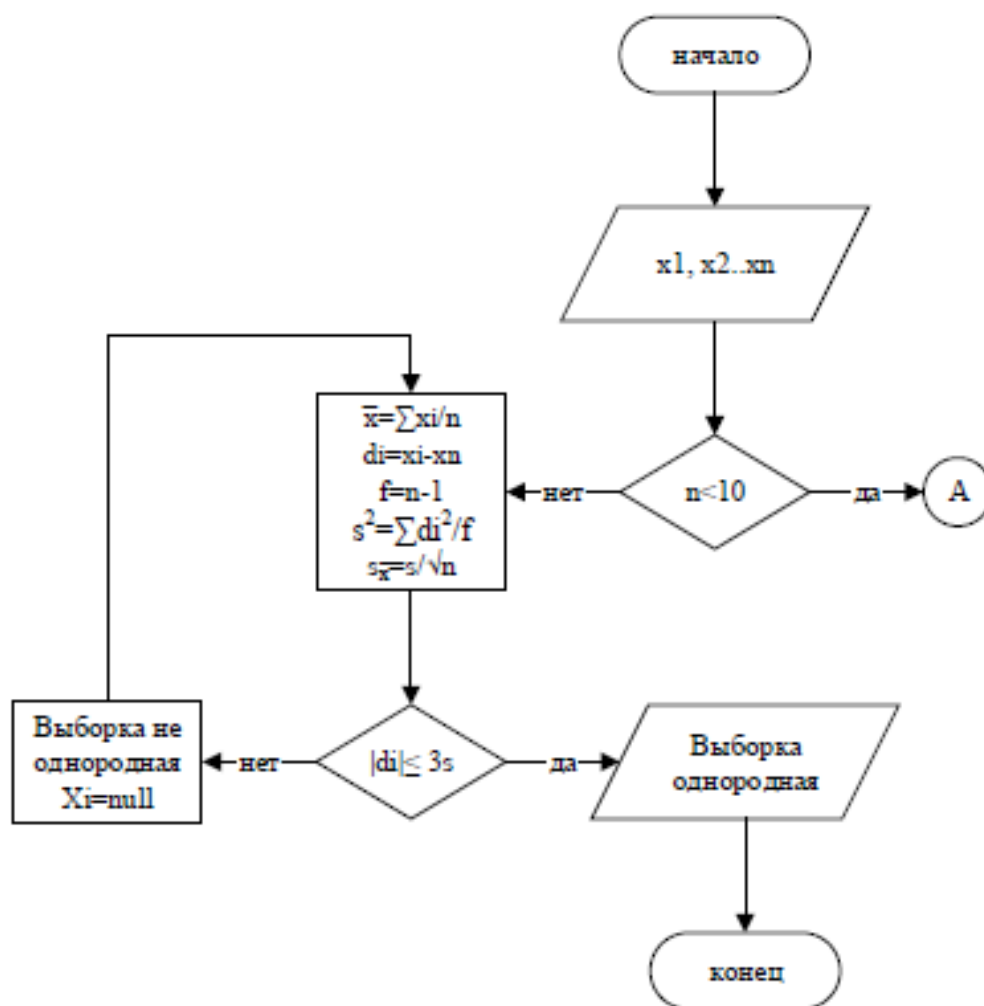


Рисунок 2 – Алгоритм проверки однородности для выборки большого объема $n \geq 10$.

После статистической обработки для выборок большого объема проверяется условие $|d_i| \leq 3s$. Если условие выполняется, выборка считается однородной, иначе варианты, для которых $|d_i| > 3s$, отбрасываются, как отягощенные грубыми ошибками с доверительной вероятностью $P > 99,0\%$, затем для полученной выборки повторяется цикл вычисления статистических характеристик и снова проводится проверка однородности. Вычисление статистических характеристик считают законченным, когда выборка сокращенного объема оказывается однородной.

Объединив описанные выше алгоритмы, сформируем алгоритм функционирования экспертной системы, реализующей процедуры проверки однородности выборки и исключения выпадающих значений вариантов (см. рисунок 3).

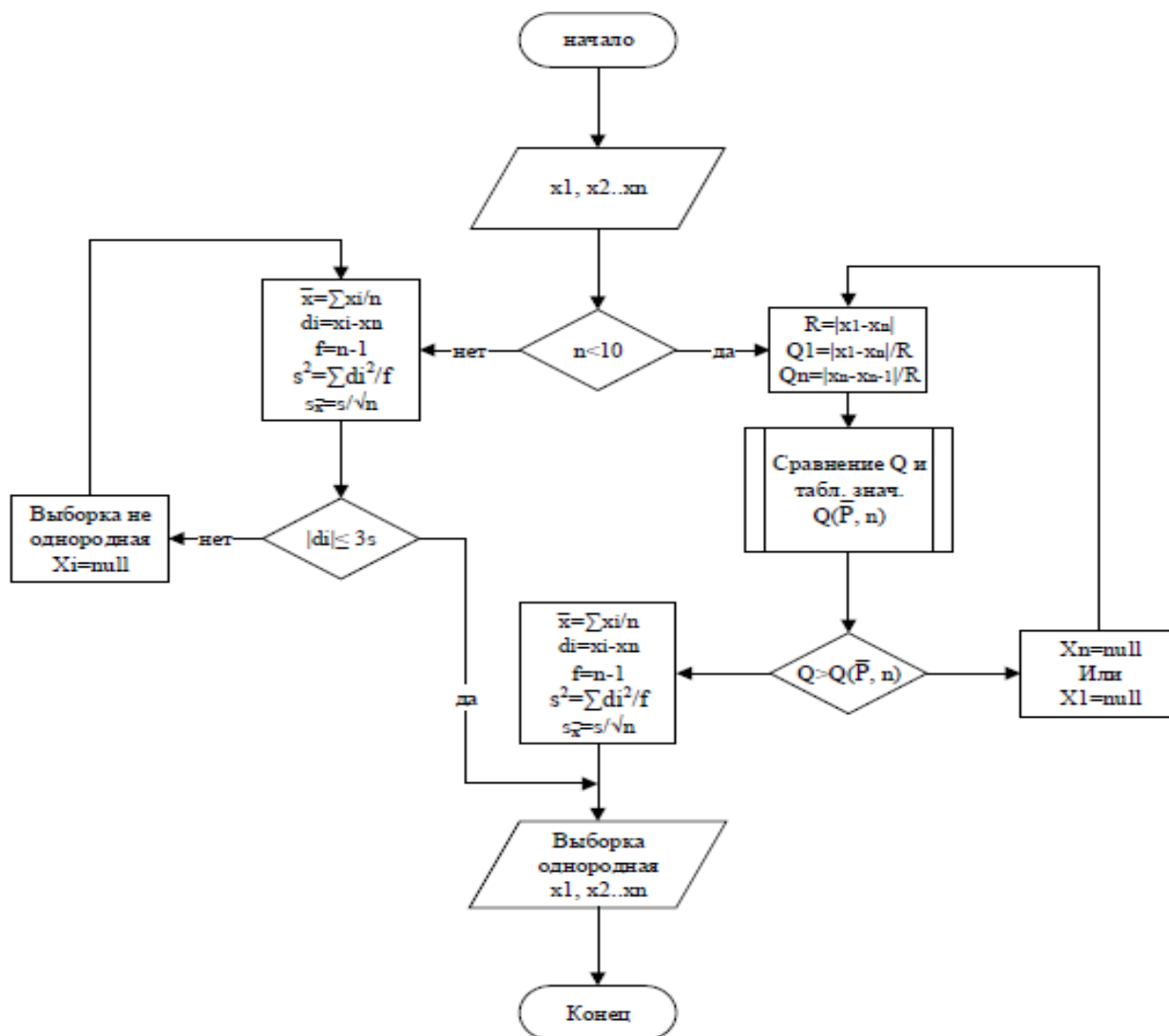


Рисунок 3 – Алгоритм проверки однородности и исключения значений выбросов.

Закключение. Итогом данного исследования является алгоритмическая реализация экспертной системы для автоматизации процесса расчетов при проверке однородности выборки, а также исключения выпадающих значений (выбросов) в данных при обработке результатов химических экспериментов в задаче оценки качества сырья для фармацевтической промышленности.

Перспективами дальнейших исследований является создание прототипа экспертной системы и проведение экспериментов с различными объемами выборок и типами данных для задач оценки качества сырья для фармацевтической промышленности.

Библиографический список:

1. Юргель Н. В., Багирова В. Л. XII государственная фармакопея Российской федерации (часть 2) // Москва 2010. с. 440
2. Тюменский государственный университет. Метрологические характеристики методов анализа // Методические указания ТГУ. 2015. URL: <https://studfiles.net/preview/3239894/>
3. Джексон П. Введение в экспертные системы // Вильямс 2001 с. 393

УДК 004.652.4

А.Г. Шлемко, магистрант

О.В. Ченгарь, канд. техн. наук, доцент

Севастопольский государственный университет

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ЭФФЕКТИВНОСТИ ORM ПРИ ПРОЕКТИРОВАНИИ РЕЛЯЦИОННЫХ БАЗ ДАННЫХ ДЛЯ СЛОЖНЫХ ПРИЛОЖЕНИЙ

Аннотация: В статье демонстрируется значение ORM при проектировании и разработке реляционных баз данных. Рассматриваются методы построения схем таблиц и способов взаимодействия между ними, а именно: CRUD-операции, функции выборки данных из нескольких таблиц и их сортировки, сообщение между разными базами данных. Основное внимание уделяется подходу проектирования баз данных с использованием ORM и исследованию эффективности последней.

Ключевые слова: object-relational mapping, базы данных, реляционные базы данных, EntityFrameworkCore, SQL Manager, MS SqlServer, SQL, AutoMapper.

Annotation: The article demonstrates the importance of ORM in the design and development of relational databases. We consider methods for constructing schemas of tables and ways of interacting with them, namely: CRUD operations, functions for fetching data from several tables and their sorting, communication between different databases. The focus is on the approach using ORM and the study of the effectiveness of the latter.

Key words: (object-relational mapping, databases, relational databases, EntityFrameworkCore, SQL Manager, MS SqlServer, SQL, AutoMapper.

Время монолитных архитектур при разработке приложений давно исчерпало себя. На смену этой технологии пришла новая, так называемая луковая архитектура, в которой программный комплекс жестко разделяется на отдельные независимые друг от друга модули [1]. Такие разделения происходят как на логическом уровне (разнесение по разным проектам или даже сборкам), так и на физическом уровне (развертывание разных модулей на разных машинах).

Теперь разработчику необходимо учитывать изменчивость требований к функционалу разрабатываемого продукта и быть готовым к постоянному рефакторингу уже существующего исходного кода. В связи с этим появился новый этап на стадии проектирования базы данных: выбор инструмента для взаимодействия со структурой хранения. Очевидно, что чистый SQL не подойдет, так как он жестко привязывает разработчиков к его реализации и в случае, если необходимо будет перейти на PostgreSQL, придется с нуля переписывать слой, отвечающий за сообщение с базой данных. Такой переход можно упростить благодаря ORM (Object-Relational Mapping), который позволяет абстрагироваться от конкретных баз данных и описать общую структуру доменных сущностей.

Onion-архитектура представляет собой разделение приложения на уровни. При чем есть один независимый уровень, который находится в центре архитектуры. От этого уровня зависит второй уровень, от второго - третий и так далее. То есть получается, что вокруг первого независимого уровня наслаивается второй-зависимый. Вокруг второго наслаивается третий, который также может зависеть и от первого (рисунок 1).

Количество уровней может отличаться, но в центре всегда находится модель домена (Domain Model), то есть те классы моделей, которые используются в приложении и объекты которых хранятся в базе данных:

Первый уровень вокруг модели домена образуют интерфейсы, которые управляют работой с моделью домена. Обычно это интерфейсы репозитория, через которые происходит взаимодействие с базой данных.

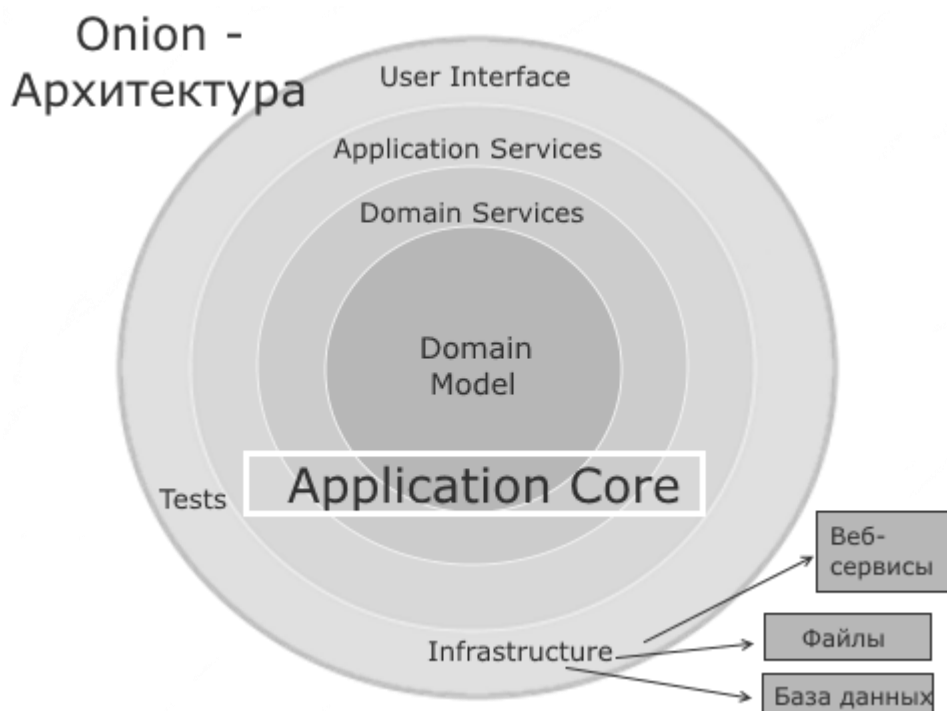


Рисунок 1 - Onion-архитектура

Внешний уровень представляет такие компоненты, которые очень часто изменяются. Обычно внешний уровень образуют пользовательский интерфейс, тесты и вспомогательные классы инфраструктуры приложения. К этому уровню также относятся конкретные реализации интерфейсов, объявленных на нижележащих уровнях. Например, реализация интерфейса репозитория, который объявлен на уровне Domain Services. Вообще все внутренние уровни, которые можно объединить в Application Core, определяют только интерфейсы, а конкретная реализация этих интерфейсов располагается на внешнем уровне.

Также стоит отметить, что все внешние хранилища, как базы данных, файлы, внешние веб-сервисы, от которых мы можем получать данные, - все это является внешним по отношению к архитектуре.

При создании архитектуры приложения надо понимать, что реальное количество уровней здесь весьма условно. В зависимости от масштаба задач уровней может быть и больше, и меньше. Однако важно понимать сам принцип, что в центре у нас модели домена, а все остальное зависит от них. Каждый внешний уровень может зависеть от внутреннего, но не наоборот.

Основной задачей ORM является обеспечение взаимодействия с данными в терминах классов, а не таблиц данных и напротив, преобразовать термины и данные классов в данные, пригодные для хранения в СУБД. Необходимо также обеспечить интерфейс для CRUD-операций над данными. В общем, необходимо избавиться от необходимости писать SQL-код для взаимодействия в СУБД.

В настоящее время разработано множество инструментов, таких как Spring, NHibernate, EntityFramework, SubSonic, которые устраняющих необходимость в преобразовании объектов для хранения в реляционных базах данных путем того, что предоставляют библиотеки классов, способных выполнять преобразования автоматически [2]. Имея список таблиц в базе данных и объектов в программе, они автоматически преобразуют запросы из одного вида в другой.

В объектно-ориентированном программировании объекты в программе представляют объекты из реального мира. В качестве примера можно рассмотреть адресную книгу, которая

содержит список людей с нулём или более телефонов и нулём или более адресов. В терминах объектно-ориентированного программирования они будут представляться объектами класса «Человек», которые будут содержать следующий список полей: имя, список (или массив) телефонов и список адресов. Суть задачи состоит в преобразовании таких объектов в форму, в которой они могут быть сохранены в файлах или базах данных, и которые легко могут быть извлечены в последующем, с сохранением свойств объектов и отношений между ними. Эти объекты называют «хранимыми» (англ. persistent) [3].

При внутреннем обращении к объекту типа «человек», ORM самостоятельно сгенерирует соответствующий SQL-запрос и выполнит его. Далее ORM преобразует полученную выборку из СУБД в виде объекта (или списка объектов, если таких было несколько в БД). С точки зрения программиста система должна выглядеть как постоянное хранилище объектов. Он может просто создавать объекты и работать с ними как обычно, а они автоматически будут сохраняться в реляционной базе данных. На практике всё не так просто и очевидно. Все системы ORM обычно проявляют себя в том или ином виде, уменьшая в некотором роде возможность игнорирования базы данных. Более того, слой транзакций может быть медленным и неэффективным (особенно в терминах сгенерированного SQL). Все это может привести к тому, что программы будут работать медленнее и использовать больше памяти, чем программы, написанные с использованием классического Bare SQL, но ORM избавляет программиста от написания большого количества кода, часто однообразного и подверженного ошибкам, тем самым значительно повышая скорость разработки. Кроме того, большинство современных реализаций ORM позволяют программисту при необходимости самому жёстко задать код SQL-запросов, который будет использоваться при тех или иных действиях (сохранение в базу данных, загрузка, поиск и т. д.) с постоянным объектом.

По своей сути ORM работает следующим образом: доменные сущности описываются в виде обычных классов, которые содержат набор полей, определяющих саму доменную сущность. В дальнейшем разработчику больше не нужно обращаться к БД — это сделает за него ORM. Для работы с хранимыми данными достаточно взаимодействовать с их представлениями в виде классов. После всех правок ORM самостоятельно сгенерирует query-запрос (или серию запросов) для используемой БД, который и произведет в конечном счете манипуляции над данными на стороне БД. Сообщение между ORM и БД происходит по средством маппинга – определения соответствия данных между потенциально различными семантиками одного объекта или разных объектов [4]. Очевидно, дублирование данных увеличивает требования к вычислительным системам, а также повышает время обработки запросов. Более того стоит отметить, что запросы, генерируемые ORM, не всегда эффективны.

Для примера рассмотрим логику работы EntityFrameworkCore [5]. EF Core - представляет собой объектно-ориентированный фреймворк компании Microsoft. Данный фреймворк является по функциональному назначению ORM-инструментом. То есть EF Core позволяет работать с БД, представляет собой более высокий уровень абстракции: EF Core позволяет абстрагироваться от самой БД и ее таблиц, дает возможность работать с данными независимо от конкретно реализации сервера хранения. Если на физическом уровне мы оперируем таблицами, индексами, первичными и внешними ключами, но на концептуальном уровне, который нам предлагает Entity Framework, мы уже работаем с объектными представлениями.

При создании программного обеспечения нам нужно фокусироваться на минимизации сложности, насколько это возможно. Object-relational mapping — не что иное, как слой дополнительной сложности, которая существует только потому, что мы хотим, изменять объекты и хранить их персистентно в БД. Одна из основных идей ООП это использование абстракций, которые представляют концепции из реальной предметной области и проецируют их на доменные сущности, представленные в виде обычных объектов (классов).

Сложность, которая стоит за ORM, вызвана тем, что мы называем несоответствием объектной и реляционной модели. Это задача крайне сложная, потому что существует множество различных моделей баз данных и много различных способов представления объектов. Представляются два варианта: маппинг 1:1 между моделью базы данных и объектами либо использование вспомогательной системы, которая будет обеспечивать сопоставление представления объектов в базе данных с представлениями объектов в памяти.

Оба варианта имеют узкие места. Первый особенно трудоемкий, потому что маппинг 1:1 плотно связывает уровень приложений с базой данных, что существенно усложняет алгоритм выполнения программы. Во втором случае все сводится к громозкости исходного кода, к тому же требует ведения дополнительных файлов – MapperProfiler, в котором описывается логика маппинга полей. Любая неточность описания либо банальное различие в написании имен свойств чревато потерями со стороны целостности данных БД.

Сравнение эффективности не ограничивается одним замером времени обработки запроса в миллисекундах. Стоит отметить, что простые CRUD-операции (Create, Read, Update, Delete) ORM обрабатывает в разы быстрее, более того, разработчику даже не следует заморачиваться о типах данных – все приведения типов выполнит сама ORM. Проблемы начинаются, когда необходимо сделать выборку данных из нескольких разных таблиц (или еще хуже баз данных) и упаковать ответ в одно представление.

Имеем следующую структуру базы данных:

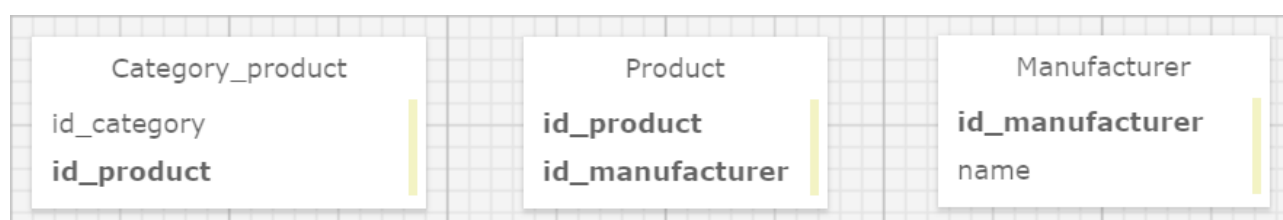


Рисунок 2 - Структура базы данных

Составляем SQL-запрос, время выполнения которого будет рассматривать. Запрос включает в себя объединение нескольких таблиц с помощью оператора INNER JOIN:

```
SELECT m.name, cp.id_category
FROM manufacturer as m
INNER JOIN product as p
    ON m.id_manufacturer = p.id_manufacturer
INNER JOIN category_product as cp
    ON p.id_product = cp.id_product
WHERE cp.id_category = 'some value'
```

На рисунке 2 ниже представлены результаты замеров времени выполнения таких запросов. В качестве операции соединения использовался оператор JOIN. Показатели времени являются усредненными [6].

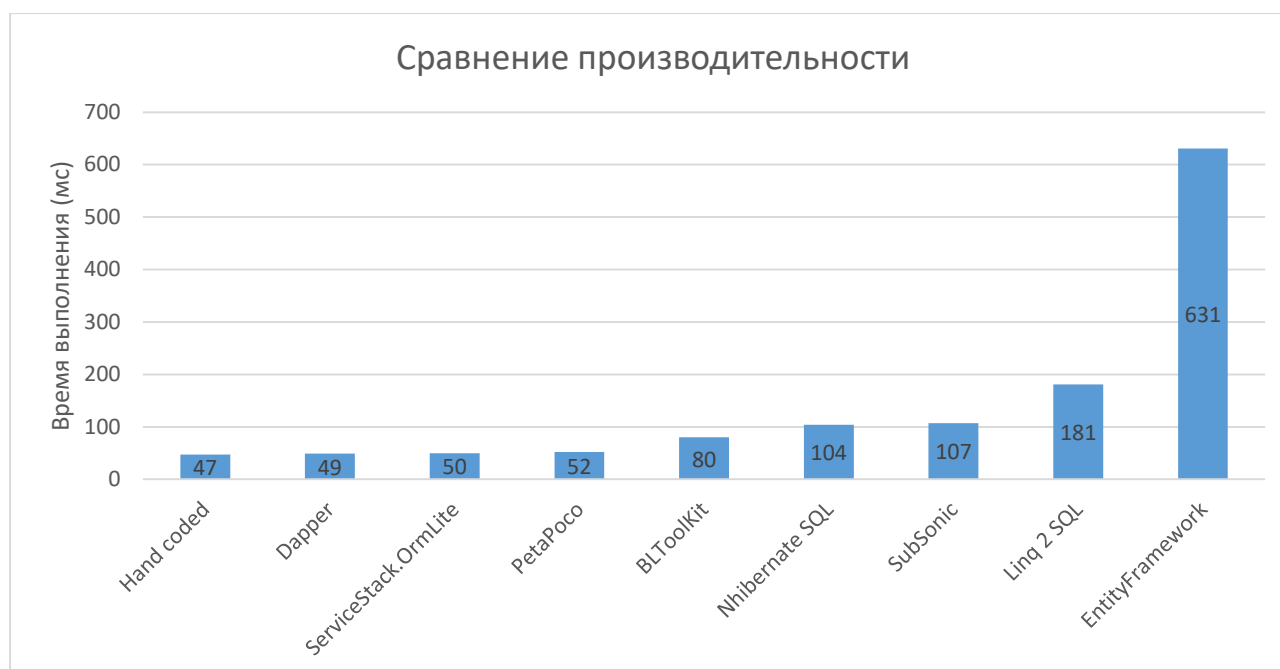


Рисунок 3 - Замеры времени обработки запросов выборки

Из рисунка 3 видно, что в случае со сложным запросом самым эффективным подходом оказался чистый SQL. Очевидно, что в бизнес-проектах Bare SQL использовать не целесообразно ввиду больших затрат на написание самих скриптов, сложности, а порой и невозможности отладки таких запросов и отсутствия какого-либо экрана защиты БД от так называемых инъекций. SQL инъекция — это один из самых доступных способов взлома сайта.

Суть таких инъекций – внедрение в данные (передаваемые через GET, POST запросы или значения Cookie) произвольного SQL кода. Если сайт уязвим и выполняет такие инъекции, то, по сути, есть возможность обращаться к БД напрямую, минуя все ограничения [7]. Различные ORM, как правило, имеют уже встроенных механизм защиты от подобных уязвимостей. В частности, EF Core является самым надежным среди прочих остальных. Также стоит обратить внимание на то, каким образом реализованы транзакции в различных фреймворках. Транзакция – это атомарный набор запросов к базе данных. Даже если ваша программа аварийно завершится, база данных гарантирует, что будут сохранены все изменения, или ни одно из них [8]. В EF Core механизм транзакций реализован лучше, чем в NHibernate, потому что первый более эффективно отрабатывает при одновременной попытке записи в одну и ту же таблицу разными подключениями [9]. Таким образом, получается, что недостаточно сравнить различные ORM-фреймворков по быстродействию. Необходимо учитывать уровень изолированности атомарных операций, экран защиты от внешних вмешательств, логику сопоставления данных в БД с внутренними объектами, сложность внедрения самой ORM в систему и тд.

В настоящий момент разработчики ORM-фреймворков значительно улучшили качество генерируемого SQL-кода, повысили уровень безопасности и обеспечили должный уровень консистентности БД. Качество поддержки и обслуживания предоставляемых программных инструментов не вызывает нареканий, однако для эффективного использования того или иного фреймворка вначале необходимо ознакомиться с его сильными и слабыми сторонами.

Библиографический список

1. Архитектура приложений. Onion-архитектура. [Электронный ресурс]. Дата обновления: 30.10.2015. URL: <https://metanit.com/sharp/mvc5/23.1.php> - 27.03.2019
2. Введение в ORM (Object Relational Mapping). [Электронный ресурс]. Дата обновления: 26.09.2012. URL: <http://internetka.in.ua/orm-intro/> - 30.03.2019
3. Базы данных, знаний и хранилища данных. Big data, СУБД и SQL и noSQL. [Электронный ресурс]. Дата обновления: 17.07.2017. URL: <https://intellect.icu/orm-4701/> - 01.04.2019
4. Data Mapping. [Электронный ресурс]. Дата обновления: 18.06.2018. URL: https://en.wikipedia.org/wiki/Data_mapping/ - 01.04.2019
5. Руководство по EF Core: сайт Евгения Попова. [Электронный ресурс]. Дата обновления: 26.08.2018. URL: <https://metanit.com/sharp/entityframeworkcore/> - 03.04.2019
6. Когда не нужен ORM, а нужно что-то поменьше. Dapper micro-ORM. [Электронный ресурс]. Дата обновления: 16.01.2018. URL: <http://regfordev.blogspot.com/2016/01/orm-dapper-micro-orm.html/> - 03.04.2019
7. SQL инъекции. Проверка, взлом, защита. [Электронный ресурс]. Дата обновления: 20.10.2014. URL: <https://habr.com/ru/post/130826/> - 05.04.2019
8. Основные принципы работы с сессиями и транзакциями. [Электронный ресурс]. Дата обновления: 18.10.2015. URL: <https://habr.com/ru/post/271115/> - 13.04.2019
9. Best ORM to use. [Электронный ресурс]. Дата обновления: 27.08.2017. URL: <https://stackoverflow.com/questions/3187928/best-orm-to-use-with-c-sharp-4-0/> - 17.04.2019
10. Фаулер М. Шаблоны корпоративных приложений / М. Фаулер, Д. Райс, М. Фоммел, Э. Хайет, Р. Ми, Р. Стаффорд. – М.: Вильямс, 2016.
11. Фаулер М. Чистый код / М. Фаулер. – М.: Питер, 2010.
12. Бек К. Рефакторинг. Улучшение проекта существующего кода / К. Бек, М. Фаулер, Дж. Брант. – М.: Вильямс, 2017.



Сборник статей всероссийской научно-технической конференции
Издано в авторской редакции

Ответственность за содержание статей несут авторы

Ответственный за выпуск: О.В.Ченгарь

Для создания электронного издания использовано:
Microsoft Word 2013, Adobe Acrobat Pro
Подписано к использованию 18.09.2019 г.
Уч.-изд. л. 22,23. Объем данных 8,3 Мб

ФГАОУ ВО «СевГУ», кафедра ИТиКС
299053, Севастополь, ул. Университетская, 33.
Тел.: (8692) 43-50-08
E-mail: itiks@sevsu.ru

г. Севастополь

2019г.