

Министерство образования Республики Беларусь  
Учреждение образования  
«Белорусский государственный университет  
информатики и радиоэлектроники»

## **КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ**

**55-я юбилейная научная конференция  
аспирантов, магистрантов и студентов**

Сборник тезисов докладов

22–26 апреля 2019 года  
Минск, БГУИР

УДК 004.021+004.2+004.4+004.6+004.7+004.9  
ББК 32.97+32.81

55-я юбилейная конференция аспирантов, магистрантов и студентов учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», 22-26 апреля 2019 г., БГУИР, Минск, Беларусь: тезисы докладов. – Мн. – 2019. – 287 с.; ил.

В сборнике опубликованы тезисы докладов, представленных на 55-й юбилейной научной конференции аспирантов, магистрантов и студентов БГУИР. Материалы одобрены оргкомитетом и публикуются в авторской редакции.

Для научных и инженерно-технических работников, преподавателей, аспирантов, магистрантов и студентов вузов.

## СОДЕРЖАНИЕ

<b>СЕКЦИЯ «ЭЛЕКТРОННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ»</b>	<b>11</b>
1. МЕТОД РАСПОЗНАВАНИЯ ОТПЕЧАТКОВ ПАЛЬЦЕВ, ОСНОВАННЫЙ НА ПРИМЕНЕНИИ ВЕЙВЛЕТ-ПРЕОБРАЗОВАНИЯ	12
2. СВЕРТОЧНАЯ НЕЙРОННАЯ СЕТЬ ГЛУБОКОГО ОБУЧЕНИЯ ДЛЯ МОНИТОРИНГА ИЗМЕНЕНИЯ РАКОВОЙ ОПУХОЛИ	13
3. МИКРОСЕРВИСЫ И ОБЛАКА В РАЗРЕЗЕ IoT	14
4. ПРИМЕНЕНИЕ АЛГОРИТМА ОБРАТНОГО РАСПРОСТРАНЕНИЯ ОШИБКИ ДЛЯ КОЛОРИЗАЦИИ ИЗОБРАЖЕНИЙ	15
5. КЛАССИФИКАЦИЯ ГАЛАКТИК С ПОМОЩЬЮ КАПСУЛЬНЫХ НЕЙРОННЫХ СЕТЕЙ	16
6. ЗНАЕТЕ ЛИ ВЫ, ЧТО ПРОИСХОДИТ, КОГДА ГОВОРИТЕ МИРУ “ПРИВЕТ!”?	18
7. ВЫЯВЛЕНИЕ DNS ТУННЕЛИРОВАНИЯ В КОМПЬЮТЕРНОЙ СЕТИ МЕТОДОМ ROMDP	20
8. СВЕРТОЧНАЯ НЕЙРОННАЯ СЕТЬ ДЛЯ РАСПОЗНАВАНИЯ РУКОПИСНЫХ СИМВОЛОВ	22
9. ПРИНЦИПЫ РАБОТЫ ДИНАМИЧЕСКОЙ ДИСПЕТЧЕРИЗАЦИИ МЕТОДОВ В РАЗЛИЧНЫХ ЯЗЫКАХ ПРОГРАММИРОВАНИЯ	24
10. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ СИСТЕМЫ ХРАНЕНИЯ ОБЪЕКТОВ АВТОРСКИХ ПРАВ НА ОСНОВЕ ТЕХНОЛОГИИ БЛОКЧЕЙН	26
11. ОБРАБОТКА ПОСТОЯННОГО ПОТОКА ДАННЫХ С АППАРАТНЫХ И ВИРТУАЛЬНЫХ ПЛАТФОРМ CISCO	28
12. СПОСОБЫ МИГРАЦИИ СХЕМ ДАННЫХ С СОБЛЮДЕНИЕМ ПРИНЦИПОВ ИДЕМПОТЕНТНОСТИ И КОНВЕРГЕНТНОСТИ	29
13. ПРОГРАММНАЯ СИСТЕМА СИМУЛЯЦИИ РАЗРУШАЕМОСТИ НА ОСНОВЕ UNREAL ENGINE 4	30
14. ПРОГРАММА ЛОЯЛЬНОСТИ ДЛЯ КЛИЕНТОВ СПОРТИВНЫХ КЛУБОВ НА БАЗЕ ТЕХНОЛОГИИ БЛОКЧЕЙН	32
15. МИКРОСЕРВИС ДЛЯ УДАЛЁННОГО УПРАВЛЕНИЯ ЧПУ СТАНКОМ	33
16. ВНУТРЕННЯЯ И ВНЕШНЯЯ ОЦЕНКА КОНВЕРСИИ СЛОВА В ВЕКТОР	34
17. ПРИМЕНЕНИЕ АЛГОРИТМА СЕГМЕНТАЦИИ ИЗОБРАЖЕНИЙ НА ОСНОВЕ ГРАФОВ ДЛЯ АНАЛИЗА ФОТОСНИМКОВ ТОПОЛОГИЙ ИНТЕГРАЛЬНЫХ МИКРОСХЕМ	35
18. ВИД РАСПРЕДЕЛЕНИЯ ВЕКТОРОВ ЗНАЧЕНИЙ НЕЙРОНОВ НА МНОГОСЛОЙНОМ ПЕРСЕПТРОНЕ	37
19. РЕАЛИЗАЦИЯ MVC КОНЦЕПЦИИ НА ПРИМЕРЕ ВЕБ-ПРИЛОЖЕНИЯ С ТЕСТИРОВАНИЕМ	38
20. РАСПОЗНАВАНИЕ ЭМОЦИЙ ЧЕЛОВЕКА ПО ФОТО	39
21. ПРОГРАММНОЕ СРЕДСТВО КЛАССИФИКАЦИИ ОБРАЗОВ С ИСПОЛЬЗОВАНИЕМ СВЕРТОЧНОЙ НЕЙРОННОЙ СЕТИ ГЛУБОКОГО ОБУЧЕНИЯ	40
22. МЕТОДИКА РАСПОЗНАВАНИЯ ПОКАЗАНИЙ СТРЕЛОЧНЫХ ПРИБОРОВ НА ОСНОВЕ АНАЛИЗА ИЗОБРАЖЕНИЙ	41

23.	РОБОТ С РАСПОЗНАВАНИЕМ ЛИЦА ПОЛЬЗОВАТЕЛЯ	42
24.	ИДЕНТИФИКАЦИЯ ПОЛЬЗОВАТЕЛЯ ПО ЕГО РУКОПИСНОЙ ПОДПИСИ	44
25.	КОРРЕКТИРОВКА ВЕКТОРА ДВИЖЕНИЯ В РЕАЛЬНОМ ВРЕМЕНИ	45
26.	СИСТЕМА УПРАВЛЕНИЯ ДОКУМЕНТООБОРОТОМ КАФЕДРЫ	47
27.	MICROSOFT ORLEANS КАК РЕАЛИЗАЦИЯ МИКРОСЕРВИСНОЙ АРХИТЕКТУРЫ	49
28.	СТРУКТУРЫ ДАННЫХ ДЛЯ РАБОТЫ СО СДЕЛКАМИ НА БИРЖЕ	50
29.	УСТРАНЕНИЕ ДИСТОРСИИ НА ШИРОКОУГОЛЬНЫХ ИЗОБРАЖЕНИЯХ	52
30.	СИСТЕМА ПРОФИЛИРОВАНИЯ ДЛЯ ОПЕРАЦИОННОЙ СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ	53
31.	ОБНАРУЖЕНИЕ И РАСПОЗНАВАНИЕ ОБЪЕКТОВ В ВИДЕОПОТОКЕ НА ОСНОВЕ R-CNN ДЕТЕКТОРОВ	55
32.	СИСТЕМА АВТОМАТИЧЕСКОГО ПОЛИВА РАСТЕНИЙ	56
33.	АЛГОРИТМ ВЗАИМОДЕЙСТВИЯ МОБИЛЬНЫХ РОБОТОВ В СОСТАВЕ РОБОТОТЕХНИЧЕСКОГО КОМПЛЕКСА	57
34.	УМНОЖЕНИЕ КВАТЕРНИОНОВ С ИСПОЛЬЗОВАНИЕМ ЛОГАРИФМИЧЕСКОЙ СИСТЕМЫ СЧИСЛЕНИЯ	59
35.	МЕТОДИКА ПОИСКА ОПТИМАЛЬНОГО МАРШРУТА НА ОСНОВЕ АНАЛИЗА ДАННЫХ ОТ КАРТОГРАФИЧЕСКИХ СЕРВИСОВ	62
36.	СОВРЕМЕННЫЕ АЛГОРИТМЫ МОДЕЛИРОВАНИЯ АМОРФНЫХ ОБЪЕКТОВ	63
37.	СРЕДСТВА ЗАЩИТЫ ДАННЫХ ПРИ ОБЛАЧНЫХ ВЫЧИСЛЕНИЯХ	64
38.	ИДЕНТИФИКАЦИЯ ПОЛЬЗОВАТЕЛЯ МЕТОДОМ АНАЛИЗА ДИНАМИКИ ВВОДА ТЕКСТА	65
39.	ТМОФ УЯЗВИМОСТЬ	66
40.	ЗАЩИТА ПЕРЕДАВАЕМОЙ ИНФОРМАЦИИ В АВТОМОБИЛЬНОЙ СИГНАЛИЗАЦИИ	67
41.	СИСТЕМЫ УПРАВЛЕНИЯ ПРОЦЕССАМИ ПРОЕКТИРОВАНИЯ ПРОГРАММ	68
42.	ОСНОВНЫЕ ПОДХОДЫ ОПРЕДЕЛЕНИЯ НЕВНИМАТЕЛЬНОСТИ ВОДИТЕЛЯ ЗА РУЛЕМ	70
43.	ПЕРЕНОС СТИЛЯ МЕЖДУ ИЗОБРАЖЕНИЯМИ	71
44.	ПРИМЕНЕНИЕ НЕЙРОННЫХ СЕТЕЙ ДЛЯ РЕДАКТИРОВАНИЯ ИЗОБРАЖЕНИЙ	72
45.	ПРИМЕНЕНИЕ АЛГОРИТМОВ КОНТУРНОГО АНАЛИЗА ПРИ СОЗДАНИИ ПАНОРАМНЫХ ИЗОБРАЖЕНИЙ	73
46.	АУДИТ БЕЗОПАСНОСТИ ТРАФИКА В СИСТЕМЕ IP-ТЕЛЕФОНИИ	74
47.	ОСНОВНЫЕ ЭТАПЫ 3D РАСПОЗНАВАНИЯ ЛИЦ	76
	<b>СЕКЦИЯ «ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»</b>	<b>78</b>
48.	ПРОГРАММНОЕ СРЕДСТВО УПРАВЛЕНИЯ ЗАДАЧАМИ ПО МЕТОДОЛОГИИ КАНБАН	79
49.	WEB APPLICATION “ELECTRONIC CALENDAR”	80
50.	ПРИЛОЖЕНИЕ ДЛЯ АВТОМАТИЗАЦИИ ОРГАНИЗАЦИИ ЗАДАЧ И РАБОЧЕГО ГРАФИКА ПРЕПОДАВАТЕЛЯ	82

51.	ПРОГРАММНОЕ СРЕДСТВО «KASISKI-BABBAGE FREQUENCY ANALYSIS»	83
52.	ПРОГРАММНОЕ СРЕДСТВО МОНИТОРИНГА РАБОТЫ СЕРВИСОВ В РАСПРЕДЕЛЕННОЙ СИСТЕМЕ	84
53.	ЗАДАЧИ АВТОМАТИЗАЦИИ БАНКОВСКИХ ПРОЦЕССОВ В КРЕДИТОВАНИИ	85
54.	ИГРОВОЕ ПРИЛОЖЕНИЕ «МОМО»	86
55.	СПОСОБЫ АВТОМАТИЧЕСКОГО РАСПОЗНАВАНИЯ НОМЕРОВ ВАГОНОВ ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА	87
56.	ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ ПЕРСОНАЛИЗИРОВАННОГО ПОДБОРА ПОДАРКОВ	89
57.	ПРИМЕНЕНИЕ МОБИЛЬНЫХ УСТРОЙСТВ ДЛЯ КРИПТОГРАФИЧЕСКИХ ВЫЧИСЛЕНИЙ	91
58.	PLACE – ЛЕГКИЙ ПОИСК ПАРКОВОЧНЫХ МЕСТ	92
59.	ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИИ БЛОКЧЕЙН ДЛЯ АВТОМАТИЗАЦИИ УПРАВЛЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫМИ АКТИВАМИ	93
60.	АЛГОРИТМ АВТОМАТИЧЕСКОГО ВЗВЕШИВАНИЯ ЖЕЛЕЗНОДОРОЖНЫХ ВАГОНОВ	94
61.	МЕТОД РАСПОЗНАВАНИЯ ЭМОЦИЙ ПО ИЗОБРАЖЕНИЮ НА ОСНОВЕ МАШИННОГО ОБУЧЕНИЯ	96
62.	ГРУППОВАЯ РОБОТОТЕХНИКА И ИДЕИ ЕЕ ПРИМЕНЕНИЯ	97
63.	ОДНОПЛАТНЫЕ КОМПЬЮТЕРЫ В СИСТЕМАХ ПОЛУЧЕНИЯ ПОТОКОВЫХ АУДИОДАНЫХ	98
64.	ВЕБ-ПРИЛОЖЕНИЕ СОЗДАНИЯ И ПРОВЕДЕНИЯ ОПРОСОВ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИЙ REACT, REDUX, ASP.NET WEBAPI 2	99
65.	ИСПОЛЬЗОВАНИЕ АЛГОРИТМА PATCHWORK ДЛЯ РЕАЛИЗАЦИИ СИСТЕМЫ ВОДЯНЫХ ЗНАКОВ	100
66.	МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ ПРОСМОТРА РАСПИСАНИЯ ОБЩЕСТВЕННОГО ТРАНСПОРТА НА ПЛАТФОРМЕ XAMARIN С ИСПОЛЬЗОВАНИЕМ УСТРОЙСТВА ТИПА «УМНЫЕ ЧАСЫ»	101
67.	ОБЗОР РЕШЕНИЙ СТАТИСТИЧЕСКОГО АНАЛИЗА КОДА	102
68.	ОСОБЕННОСТИ РАЗРАБОТКИ WEB-ПРИЛОЖЕНИЙ ПРИ НИЗКОМ УРОВНЕ КОМПЬЮТЕРНОЙ ГРАМОТНОСТИ ПОЛЬЗОВАТЕЛЕЙ	103
69.	ХРАНИЛИЩЕ ДАННЫХ КАК ЭЛЕМЕНТ ПРОЦЕССА ПРИНЯТИЯ РЕШЕНИЙ	104
70.	ВЛИЯНИЕ ФИЗИЧЕСКОГО СОСТОЯНИЯ СПОРТСМЕНА НА РЕЗУЛЬТАТ ТЕННИСНОГО МАТЧА	105
71.	ИСПОЛЬЗОВАНИЕ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ ДЛЯ ПРОГНОЗИРОВАНИЯ ПОГОДНЫХ УСЛОВИЙ	107
72.	РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА ГЕНЕРАЦИИ UML-ДИАГРАММ НА ОСНОВЕ ОПИСАНИЯ НА МЕТАЯЗЫКЕ	108
73.	ВЕБ-СЕРВИСЫ ВОПРОСОВ И ОТВЕТОВ	109
74.	АВТОМАТИЗАЦИЯ ДЕЯТЕЛЬНОСТИ МЕДИЦИНСКИХ УЧРЕЖДЕНИЙ	110
75.	ПРОГРАММНОЕ СРЕДСТВО СБОРА И АНАЛИЗА ТЕХНОЛОГИЧЕСКИХ ПАРАМЕТРОВ СТРЕЛЫ ФРОНТАЛЬНОГО ПОГРУЗЧИКА	112

76.	ПРОГРАММНОЕ СРЕДСТВО УПРАВЛЕНИЯ ПРОЦЕССОМ ПОДАЧИ ДОКУМЕНТОВ В ВЫСШЕЕ УЧЕБНОЕ ЗАВЕДЕНИЕ	114
77.	СРАВНИТЕЛЬНЫЙ АНАЛИЗ МОДЕЛЕЙ ПРОГНОЗИРОВАНИЯ ПОГОДЫ	115
78.	ИСПОЛЬЗОВАНИЕ МНОГОКРАТНЫХ ТЕСТОВ С ИЗМЕНЯЕМЫМ НАЧАЛЬНЫМ СОСТОЯНИЕМ ДЛЯ ПСЕВДОИЩЕРПЫВАЮЩЕГО ТЕСТИРОВАНИЯ ОЗУ	117
79.	ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА ЗАЩИТЫ ПРИВАТНОСТИ И ФИЛЬТРАЦИИ НЕЖЕЛАТЕЛЬНЫХ ПИСЕМ В ЭЛЕКТРОННОЙ ПОЧТЕ	119
80.	ОБЗОР РЕШЕНИЙ ПО МОДЕЛИРОВАНИЮ ДВИЖЕНИЯ ПОТОКОВ ЛЮДЕЙ ПРИ ЭВАКУАЦИИ ИЗ ПОМЕЩЕНИЙ	121
81.	УНИФИЦИРОВАННЫЙ ИНТЕРФЕЙС ДЛЯ СОЗДАНИЯ КРОССПЛАТФОРМЕННЫХ ВЕБ-ОРИЕНТИРОВАННЫХ SMART TV ПРИЛОЖЕНИЙ	122
82.	ПРОГРАММНОЕ СРЕДСТВО «ОБЩЕРЕСПУБЛИКАНСКИЙ БАНК ДАННЫХ УЧАСТНИКОВ РЕПЕТИЦИОННОГО ТЕСТИРОВАНИЯ» С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ ASP .NET CORE	123
83.	ИНФОРМАЦИОННАЯ СИСТЕМА СБОРА И ОБРАБОТКИ ТЕЛЕМЕТРИЧЕСКИХ ДАННЫХ	125
84.	ПРОБЛЕМА БАЛАНСИРОВКИ НАГРУЗКИ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ	127
85.	SAFE МЕТОДОЛОГИЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	128
86.	СТЕГАНОГРАФИЧЕСКАЯ СИСТЕМА, ОСНОВАННАЯ НА ИСПОЛЬЗОВАНИИ МЕРЫ СЛОЖНОСТИ ГРАФИЧЕСКОГО ОБРАЗА	131
87.	СХЕМА ГЕНЕРАЦИИ И ВОССТАНОВЛЕНИЯ ДАННЫХ ПРИ ЛОКАЛИЗАЦИИ ДЕФЕКТА	132
88.	ПОСТРОЕНИЕ ИСКУССТВЕННОЙ НЕЙРОННОЙ СЕТИ ДЛЯ РАСПОЗНАВАНИЯ ГРАФИЧЕСКИХ ФИГУР ТЕХНИЧЕСКОГО АНАЛИЗА	134
89.	ПОДХОДЫ К РЕАЛИЗАЦИИ КОНТРОЛЯ ДОСТУПА К ФАЙЛОВОЙ СИСТЕМЕ В ОС СЕМЕЙСТВА WINDOWS NT	136
90.	АЛГОРИТМЫ ПОСТРОЕНИЯ ПЕРСОНАЛИЗИРОВАННЫХ МУЗЫКАЛЬНЫХ РЕКОМЕНДАТЕЛЬНЫХ СИСТЕМ	137
91.	КРИПТОАНАЛИЗ ОПТИМИЗАЦИЯ ТЕСТА КАСИСКИ ПОД ПРОГРЕССИВНЫЙ КЛЮЧ	139
92.	ПРЕДСТАВЛЕНИЕ МНОГОМЕРНОЙ ИНФОРМАЦИИ НА КАРТЕ	141
93.	ОПТИМИЗАЦИЯ МЕТОДА УДАЛЕНИЯ ШУМА В ЦИФРОВЫХ ЗАПИСЯХ ЗВУКА	143
94.	ИСПОЛЬЗОВАНИЕ ЦИФРОВЫХ ФИЛЬТРОВ ДЛЯ ОБРАБОТКИ ИЗОБРАЖЕНИЙ	144
95.	ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИИ UNITY 3D В СИМУЛЯЦИИ РАБОТЫ ПОДЪЕМНЫХ БАШЕННЫХ КРАНОВ	145
96.	АЛГОРИТМ ПОСТРОЕНИЯ СЕМАНТИЧЕСКОЙ СЕТИ С ОБРАБОТКОЙ МЕСТОИМЕНЕЙ ТЕКСТА	147
97.	ВЕБ-ПРИЛОЖЕНИЕ УЧЕТА РАБОЧЕГО ВРЕМЕНИ ПРЕПОДАВАТЕЛЯ	148
98.	АНАЛИЗ СПОСОБОВ ОРГАНИЗАЦИИ ИЕРАРХИЧЕСКИХ СТРУКТУР В РЕЛЯЦИОННЫХ БАЗАХ ДАННЫХ	150

99.	ПРОГРАММНОЕ СРЕДСТВО СРАВНИТЕЛЬНОЙ ОЦЕНКИ МАРШЕВЫХ ТЕСТОВ ОЗУ	152
100.	ЗАДАЧА ПОСТРОЕНИЯ МОДЕЛИ ЛИЧНОСТИ НА ОСНОВЕ АНАЛИЗА ТЕКСТА	154
101.	СОВРЕМЕННОЕ СОСТОЯНИЕ СИСТЕМ ИНФОРМАЦИОННОЙ ПОДДЕРЖКИ В СФЕРЕ ТУРИЗМА	155
102.	ПРОГРАММНОЕ СРЕДСТВО ПРОЕКТИРОВАНИЯ РЕКЛАМНЫХ ВЕБ-СТРАНИЦ	156
103.	ОБЗОР РЕШЕНИЙ ПО АВТОМАТИЗАЦИИ ПРОВЕДЕНИЯ БУМАЖНОГО ТЕСТИРОВАНИЯ	157
104.	АВТОМАТИЗИРОВАННОЕ ТЕСТИРОВАНИЕ, ОСНОВАННОЕ НА КОНЦЕПЦИИ ТЕСТОВОЙ ПИРАМИДЫ	159
105.	СИМУЛЯТОР БИРЖИ КАК ПОДХОД К ОБУЧЕНИЮ ТОРГОВЛИ КРИПТОВАЛЮТАМИ	160
106.	АВТОМАТИЗИРОВАННАЯ ВЕБ-СИСТЕМА УПРАВЛЕНИЯ ПРОЦЕССОМ РАСПРЕДЕЛЕНИЯ СТУДЕНТОВ НА ПРАКТИКУ	161
107.	МЕТОД ПРЕДОСТАВЛЕНИЯ ТУРИСТИЧЕСКИХ УСЛУГ С ИСПОЛЬЗОВАНИЕМ НЕЙРОСЕТЕВЫХ ТЕХНОЛОГИЙ	163
108.	ПРИМЕНЕНИЕ ХРАНИЛИЩ ДАННЫХ ДЛЯ СРЕДНЕГО И КРУПНОГО БИЗНЕСА	165
109.	МОДЕЛИ И АЛГОРИТМЫ РАСПРОСТРАНЕНИЯ РАДИОАКТИВНЫХ ВЕЩЕСТВ В ОКРУЖАЮЩЕЙ СРЕДЕ	166
110.	ПОДГОТОВКА И ОБРАБОТКА МАРКИРОВАННЫХ ТЕСТОВЫХ ЗАДАНИЙ	167
111.	ВЫЯВЛЕНИЕ КЛЮЧЕВЫХ ПРИЗНАКОВ ПРИ ИНДЕКСИРОВАНИИ ТЕКСТОВЫХ ДОКУМЕНТОВ	169
112.	КОМПИЛЯТОР ЯЗЫКА ОБЕРОН, РЕАЛИЗУЮЩИЙ ПАРАДИГМУ РАСШИРЯЕМОГО ПРОГРАММИРОВАНИЯ	170
113.	ПРИМЕНЕНИЕ ТЕОРИИ МАССОВОГО ОБСЛУЖИВАНИЯ В НАТУРНОМ ЭКСПЕРИМЕНТЕ	171
114.	СИСТЕМА МОНИТОРИНГА ПАРАМЕТРОВ USB-ИНТЕРФЕЙСА	172
115.	ИССЛЕДОВАНИЕ И СРАВНЕНИЕ НЕЙРОННЫХ СЕТЕЙ ДЛЯ КЛАССИФИКАЦИИ ТЕКСТА	173
116.	ВЕБ-ОРИЕНТИРОВАННОЕ ПС УДАЛЕННОГО УПРАВЛЕНИЯ ОСВЕЩЕНИЕМ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ PHP	175
	<b>СЕКЦИЯ «ВЫСШАЯ МАТЕМАТИКА»</b>	<b>177</b>
117.	МАТЕМАТИЧЕСКИЕ МОДЕЛИ ТЕОРИИ ИГР	178
118.	ТАЙНЫ СУДОКУ И НЕ ТОЛЬКО	180
	<b>СЕКЦИЯ «ИНФОРМАТИКА»</b>	<b>183</b>
119.	ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИЙ BIG DATA В ТРАНСПОРТНОЙ СФЕРЕ	184
120.	ПРОБЛЕМЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ОС WINDOWS	186
121.	СРАВНЕНИЕ АЛГОРИТМОВ РЕШЕНИЯ ЗАДАЧИ КОММИВОЯЖЁРА	187
122.	МОБИЛЬНОЕ ПРИЛОЖЕНИЯ ДЛЯ ИНДИВИДУАЛЬНЫХ ЗАНЯТИЙ СПОРТОМ	189

123. БАЛАНСИРОВКА НАГРУЗКИ МЕЖДУ УЗЛАМИ ПРИ ПОСТРОЕНИИ РАСПРЕДЕЛЕННЫХ СИСТЕМ	190
124. ИСПОЛЬЗОВАНИЕ ЛЕКСИЧЕСКОЙ ОБФУСКАЦИИ VHDL-ОПИСАНИЙ ДЛЯ ВНЕДРЕНИЯ ВОДЯНЫХ ЗНАКОВ.	191
125. СРАВНЕНИЕ СВЕРТОЧНЫХ И РЕКУРРЕНТНЫХ НЕЙРОННЫХ СЕТЕЙ В ЗАДАЧЕ АНАЛИЗА ТОНАЛЬНОСТИ ТЕКСТА	193
126. ХРАНИЛИЩЕ ДАННЫХ СИСТЕМЫ КОМПЛЕКСНОГО АНАЛИЗА ДАННЫХ ИНТЕРНЕТ ИСТОЧНИКОВ	194
127. СРАВНЕНИЕ ВАРИАЦИЙ ГРАДИЕНТНОГО СПУСКА НА ПРИМЕРЕ ЗАДАЧИ РАСПОЗНАВАНИЯ СИМВОЛОВ	195
128. ОБЛАЧНАЯ ИНФРАСТРУКТУРА КАК КОД НА ПРИМЕРЕ СИСТЕМЫ ДЛЯ АВТОМАТИЗАЦИИ КОНТРОЛЯ ДОСТУПА К SAP	197
129. ИСПОЛЬЗОВАНИЕ АЛГОРИТМА КЛАСТЕРИЗАЦИИ DBSCAN ДЛЯ ФИЛЬТРАЦИИ ВЫБРОСОВ В ДАННЫХ	198
130. БАЙЕСОВСКАЯ ОПТИМИЗАЦИЯ	200
131. АВТОМАТИЗИРОВАННАЯ СИСТЕМА УПРАВЛЕНИЯ ВЗАИМООТНОШЕНИЯМИ С КЛИЕНТАМИ С ПРИМЕНЕНИЕМ ОБЛАЧНЫХ CRM-СИСТЕМ	202
132. СЕРВИС ПО АРЕНДЕ АВТОМОБИЛЕЙ НА ОСНОВЕ МОБИЛЬНОЙ ПЛАТФОРМЫ ANDROID	204
133. МЕТОДИКА ПРОЕКТИРОВАНИЯ КОМБИНАЦИОННОГО УСТРОЙСТВА ВСЕВОЗМОЖНЫХ ПЕРЕСТАНОВОК	205
134. СРАВНЕНИЕ АЛГОРИТМОВ ТЕСТИРОВАНИЯ ЧИСЕЛ НА ПРОСТОТУ. ПОСТРОЕНИЕ ОПТИМАЛЬНОГО АЛГОРИТМА	208
135. ВИЗУАЛИЗАЦИЯ РЕШЕНИЙ НЕКОТОРЫХ МАТЕМАТИЧЕСКИХ ЗАДАЧ В MAPLE	209
136. НЕЙРОННЫЕ СЕТИ ДЛЯ ПРОГНОЗИРОВАНИЯ РЕЗУЛЬТАТОВ СОРЕВНОВАНИЙ ПО КИБЕРСПОРТИВНЫМ ДИСЦИПЛИНАМ	211
137. ВИЗУАЛИЗАТОР ПРОСТРАНСТВЕННО-ВРЕМЕННЫХ ФУНКЦИЙ	213
138. ФАРМАКОЛОГИЧЕСКАЯ СИСТЕМА ОЦЕНКИ ТЕРАПЕВТИЧЕСКОЙ АКТИВНОСТИ ВЕЩЕСТВ	214
139. АВТОМАТИЗАЦИЯ МЕДИЦИНСКОЙ РЕАБИЛИТАЦИИ ПАЦИЕНТОВ	215
140. АВТОМАТИЗАЦИЯ ПОЛЬЗОВАТЕЛЬСКИХ СЦЕНАРИЕВ КАК УНИВЕРСАЛЬНЫЙ ПОДХОД К АВТОМАТИЗАЦИИ ПРОЦЕССОВ	216
141. БЛОКЧЕЙН-ПЛАТФОРМА HYPERLEDGER В СФЕРЕ ЗДРАВООХРАНЕНИЯ	218
142. ЭФФЕКТИВНОЕ ВСТРОЕННОЕ АНСАМБЛИРОВАНИЕ НЕЙРОННЫХ СЕТЕЙ	220
143. ОЦЕНКА АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ. ДИЛЕММА СМЕЩЕНИЯ-ДИСПЕРСИИ	221
144. АНАЛИЗ ЭФФЕКТИВНОСТИ РЕКЛАМЫ КАК ETL ПРОЦЕСС	223
145. ГИБРИДНАЯ СТРАТЕГИЯ ОТОБРАЖЕНИЯ ЛОГИЧЕСКИХ АДРЕСОВ В ФИЗИЧЕСКИЕ В НАКОПИТЕЛЯХ ИНФОРМАЦИИ НА ОСНОВЕ ФЛЭШ-ПАМЯТИ	225
146. ПРИЛОЖЕНИЕ ДЛЯ УВЕЛИЧЕНИЯ ПРОИЗВОДИТЕЛЬНОСТИ ТРУДА НА ОСНОВЕ ТЕХНИКИ POMODORO	227

<b>СЕКЦИЯ «ФИЛОСОФИЯ И ЖИЗНЬ»</b>	<b>230</b>
147. ВИДЕОИГРЫ КАК НОВЫЙ ВИД ИСКУССТВА	231
148. ПРОБЛЕМА БЕЗРАБОТИЦЫ В КОНТЕКСТЕ ИНФОРМАТИЗАЦИИ ОБЩЕСТВА	232
149. ТРАНСФОРМАЦИЯ ГУМАНИЗМА В ВЫСОКОТЕХНОЛОГИЧЕСКОМ ОБЩЕСТВЕ	233
150. ОБРАЗОВАНИЕ И УСПЕХ	234
151. ТРАНСГУМАНИЗМ. НА ПУТИ К ПОСТЧЕЛОВЕКУ	236
152. ИДЕЯ БОГА В МИРОВОЗЗРЕНИИ И ТВОРЧЕСТВЕ Л.Н. ТОЛСТОГО	237
153. ВНУТРЕННИЕ КОНФЛИКТЫ. СПОСОБЫ ИХ РАЗРЕШЕНИЯ	238
<b>СЕКЦИЯ «ФИЗИКА»</b>	<b>240</b>
154. ОПЫТ ИСПОЛЬЗОВАНИЯ ЭВРИСТИЧЕСКИХ ТЕХНОЛОГИЙ ПРИ ОРГАНИЗАЦИИ ЛЕКЦИОННЫХ ЗАНЯТИЙ ПО ФИЗИКЕ НА ПОТОКЕ ПОИТ В БГУИР	241
155. МОДЕЛИРОВАНИЕ ДИНАМИЧЕСКИХ ПРОЦЕССОВ ЭФФЕКТА ХОЛЛА В СРЕДЕ MATLAB	243
<b>СЕКЦИЯ «ВСТРАИВАЕМЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ»</b>	<b>245</b>
156. РЕАЛИЗАЦИЯ ФИЛЬТРОВ В КОНЕЧНЫХ ПОЛЯХ	246
157. МЕТОДЫ ОПТИМИЗАЦИИ АЛГОРИТМОВ ДЛЯ ОБРАБОТКИ ЗВУКОВЫХ СИГНАЛОВ НА МОБИЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ ПЛАТФОРМАХ	247
158. ОЦЕНКА ПОКАЗАТЕЛЕЙ ПЕРТУРБАЦИИ КВАЗИПЕРИОДИЧЕСКОГО СИГНАЛА В ЗАДАЧАХ ДИАГНОСТИКИ ГОЛОСА	248
159. УПРАВЛЕНИЕ ЧАСТОТОЙ ОСНОВНОГО ТОНА ГОЛОСА В ЗАДАЧАХ СИНТЕЗА РЕЧИ ПО ТЕКСТУ	250
160. АНАЛИЗ КЛАССИФИКАТОРОВ ДЛЯ ДИАГНОСТИРОВАНИЯ НЕВРОЛОГИЧЕСКОГО ЗАБОЛЕВАНИЯ БАС	251
161. СИСТЕМА ОБНАРУЖЕНИЯ ПЕРЕДНЕГО ПЛАНА НА ЕСТЕСТВЕННЫХ ИЗОБРАЖЕНИЯХ С ИСПОЛЬЗОВАНИЕМ НЕЙРОННОЙ СЕТИ	253
162. ВЫСОКОСКОРОСТНАЯ РЕАЛИЗАЦИЯ КРИПТОГРАФИЧЕСКИХ ХЭШ-ФУНКЦИЙ НА БАЗЕ FPGA	255
163. ЛОГИЧЕСКАЯ ОПТИМИЗАЦИЯ НЕПОЛНОСТЬЮ ОПРЕДЕЛЁННЫХ ФУНКЦИЙ	256
164. ИССЛЕДОВАНИЕ КРИТЕРИЕВ ОЦЕНКИ КОРРЕКТНОСТИ ПРОИЗНОШЕНИЯ ФРАЗ ДЛЯ СИСТЕМЫ ИСПРАВЛЕНИЯ РЕЧЕВЫХ ДЕФЕКТОВ	257
165. ИНТЕЛЛЕКТУАЛИЗИРОВАННАЯ СИСТЕМА УПРАВЛЕНИЯ СВЕТОВЫМИ ЭФФЕКТАМИ	259
166. ТЕХНОЛОГИЯ МОНИТОРИНГА ПЕРЕМЕЩЕНИЯ ГРУЗОВ ИСПОЛЬЗУЯ РАДИОЧАСТОТНУЮ ИДЕНТИФИКАЦИЮ	261
167. АВТОМАТИЧЕСКАЯ СИСТЕМА РАДИОЧАСТОТНОЙ ИДЕНТИФИКАЦИИ ГРУЗОВ	262
168. ИТЕРАТИВНО-КОНВЕЙЕРНЫЙ ПРОЦЕССОР ХЭШ-ФУНКЦИИ SHA-256 НА БАЗЕ ПЛИС	264
169. АППАРАТНАЯ ПЛАТФОРМА ДЛЯ ПРОТОТИПИРОВАНИЯ АЛГОРИТМОВ И АРХИТЕКТУР СЛУХОВЫХ АППАРАТОВ НА ОСНОВЕ СНК ZYNQ	265

170. ПОСТРОЕНИЕ ИНФРАКРАСНЫХ МУЛЬТИТАЧ СЕНСОРНЫХ ПАНЕЛЕЙ	267
171. ЭЛЕКТРОННАЯ ЦИФРОВАЯ ПОДПИСЬ НА БАЗЕ ЭЛЛИПТИЧЕСКИХ КРИВЫХ	269
172. УСТРОЙСТВО ДЛЯ НЕПРЕРЫВНОГО НЕИНВАЗИВНОГО ИЗМЕНЕНИЯ АРТЕРИАЛЬНОГО ДАВЛЕНИЯ	270
173. ПРИМЕНЕНИЕ ДЕТЕКТОРА ГОЛОСА В СЛУХОВОМ АППАРАТЕ	271
174. МЕТОДЫ ОБРАБОТКИ СИГНАЛОВ ДЛЯ КОРРЕКЦИИ ДЕФЕКТОВ РЕЧИ С ПОМОЩЬЮ МЕХАНИЗМА ЗВУКОВОЙ ОБРАТНОЙ СВЯЗИ	273
175. БРЕЙН-СИСТЕМА	274
176. СЖАТИЕ ИЗОБРАЖЕНИЙ НА ОСНОВЕ НЕЙРОННЫХ СЕТЕЙ	275
177. МЕТОД КОМПЕНСАЦИИ УРОВНЯ ГРОМКОСТИ ПРИ ПЕРЕНОСЕ ЧАСТОТ В РЕЧЕВОМ СИГНАЛЕ	278
178. НЕЙРОИНТЕРФЕЙС НА ОСНОВЕ МИКРОКОНТРОЛЛЕРА ДЛЯ КОНТРОЛЯ ОБЪЕКТОВ В СРЕДЕ СИМУЛЯЦИИ.	280
179. МЕТОД ПОВЫШЕНИЯ РАЗБОРЧИВОСТИ РЕЧИ НА ОСНОВЕ ЧАСТИЧНОГО ПЕРЕНОСА ЧАСТОТНОГО ДИАПАЗОНА	281
180. ВЫЯВЛЕНИЕ ПАТОЛОГИЙ В ГОЛОСЕ НА ОСНОВЕ РАЗЛОЖЕНИЯ СИГНАЛА НА ЭМПИРИЧЕСКИЕ МОДЫ	283
181. ГЛУБОКАЯ НЕЙРОННАЯ СЕТЬ ДЛЯ ДЕТЕКТИРОВАНИЯ ЛИЦ В УСЛОВИЯХ ОГРАНИЧЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ РЕСУРСОВ	285

**СЕКЦИЯ «ЭЛЕКТРОННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ»**

# МЕТОД РАСПОЗНАВАНИЯ ОТПЕЧАТКОВ ПАЛЬЦЕВ, ОСНОВАННЫЙ НА ПРИМЕНЕНИИ ВЕЙВЛЕТ-ПРЕОБРАЗОВАНИЯ

Абрамович Н.А.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Насуро Е.В. – к.т.н., доцент

В статье представлен метод распознавания отпечатков пальцев, основанный на информации, извлеченной из Вейвлет-преобразования дискретного изображения. Метод имеет преимущества в скорости и вычислительной сложности перед классическими за счёт применения непосредственно к изображению, без предварительной фильтрации.

Опознавание по отпечаткам пальцев – один из фундаментальных методов идентификации личности [1]. Причины его популярности хорошо известны: отпечаток пальца обладает двумя важными для идентификации свойствами.

- 1) Узор каждого отпечатка пальцев остаётся неизменным до конца жизни (неизменность) [2];
- 2) Отпечатки различаются даже у близнецов (уникальность) [2].

В литературных источниках предлагается множество различных подходов к автоматическому распознаванию. Самыми известными из них являются следующие: подход, основанный на распознавании мелких деталей узора и подход, основанный на распознавании изображения.

Подход, основанный на мелких деталях на сегодняшний день самый популярный, и используется почти во всех системах автоматической идентификации и верификации. Хотя подходы, основанные на мелких деталях, могут сильно отличаться друг от друга, все они требуют дорогих операций предварительной обработки для извлечения деталей и поэтому работают относительно медленно и требуют большого количества вычислительных ресурсов.

Данную проблему позволяют решить подходы, основанные на распознавании изображения. Подобные подходы обычно не нуждаются в операциях предварительной обработки и основаны на применении различных относительно недорогих преобразований непосредственно к исходному черно-белому изображению, потребляя таким образом значительно меньше вычислительных ресурсов [4].

В докладе будет изложен метод распознавания отпечатка пальца, основанный на распознавании изображения с использованием Вейвлет-преобразования, применяющемуся к исходному изображению, который выполняется достаточно быстро. После чего вычисляется признаковое описание, основанное на среднеквадратичном отклонении подизображений, полученных в результате Вейвлет-декомпозиции. Для улучшения результатов метода Вейвлет-преобразование применяется не сразу к целому исходному изображению, а к нескольким квадратным неперекрывающимся участкам, выделенным из исходного изображения и располагающимся в пределах зон интереса, которые строятся вокруг ключевых точек с использованием специального алгоритма.

Таким образом, разработанный алгоритм выполняется в следующей последовательности:

- 1) Определяются ключевые точки для изображения отпечатка пальца;
- 2) Выделяется прямоугольная область, расположенная внутри изображения отпечатка пальца с центром в ключевой точке;
- 3) Область разбивается на неперекрывающиеся квадратные блоки;
- 4) Для каждого из блоков выполняется Вейвлет-декомпозиция на  $n$  октав, из которой получается признаковое описание длиной  $3n$ ;
- 5) Полученные признаковые описания комбинируются в одно, которое и используется в классификаторе.

Информация для Вейвлет-преобразования извлекается непосредственно из чёрно-белого изображения и, следовательно, предложенный метод требует меньше вычислительных ресурсов, чем методы, основанные на мелких деталях. Метод был сравнен с одним из предложенных в литературных источниках и показал положительные результаты. Высокий процент точности и низкая вычислительная сложность показывает, что метод может быть эффективно использован в системах, содержащих небольшое количество отпечатков пальцев.

#### Список использованных источников:

1. A.K. Jain. R. Bolle. and S. Pankanti. Eds., Biometrics - Personal Identification in New York Society. Kluwer Academic Publishers. 1999.
2. H.C.Lee. R.E.Gaenssleh. Eds., Advances in Fingerprint Technology. Elsevier. New-York. 199 1.
3. Nalini K. Ratha, Shaoyun Chen. Anil K.Jain, "Adaptive Flow Orientation-Based Feature Extraction in Fingerprint Images", Pattern Recognition, vol.28, no.11, pp.1657-1672. 1995.
4. Nalini K. Ratha. Kalle Karu. Shaoyun Chen. Anil K.Jain. "A Real-Time Matching System for Large Fingerprint Databases". IEEE Trails. 011 Patteix Rccogriitiori arid Mrcliirc liitclligeiice. vol. 18. no.8. pp.799- 813. 1996.

# СВЕРТОЧНАЯ НЕЙРОННАЯ СЕТЬ ГЛУБОКОГО ОБУЧЕНИЯ ДЛЯ МОНИТОРИНГА ИЗМЕНЕНИЯ РАКОВОЙ ОПУХОЛИ

Альромх А.М.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Иванов Н.Н. – к.ф.-м.н., доцент

Описана структура приложения, анализирующее изменение формы и размера глиомы головного мозга на основе полученных в течение некоторого периода времени, от 3 месяцев до одного года на сканнере МРТ трехмерных изображений различных модальностей.

В октябре 1971 г. рентгенологи госпиталя Св. Джорджа в Лондоне поместили пациента в новую сканирующую машину и получили первое в мире медицинское магнитно-резонансное изображение. Так началось практическое использование нового неинвазивного метода исследования. За 45 с лишним лет развития физической теории слабых взаимодействий было создано массовое производство сканнеров МРТ, разработаны новые модальности сканирования, созданы надежные программные продукты автоматизированного распознавания магнитно-резонансных томограмм [1]. Тем не менее, продолжают широкие научные теоретические и экспериментальные исследования в этой области.

В этом сообщении описывается один метод автоматического отслеживания изменения глиомы пациента – самой распространенной опухоли головного мозга. Имеется достаточно большое количество публикаций по этой теме [2], [3], [4]. Здесь сравниваются 2 алгоритма мониторинга головного мозга, в котором уже была достоверно обнаружена злокачественная опухоль. Предполагается, что от момента обнаружения заболевания до текущего момента постоянно проводятся лечебные мероприятия с контролем МРТ. Предлагаемые программные приложения дают оценку хода лечения и классифицируют развитие опухоли на основе применения сверточных глубоких нейронных сетей.

Для анализа текущего состояния опухоли рассматривается применение двух приложений, одно из них классическое, сравнивающее стек трехмерных МРИ в модальностях, которые использовали рентгенологи. Второе приложение содержит элемент новизны, стек трехмерных изображений регистрируется и рассматривается как единое четырехмерное изображение. Это изображение обрабатывается свертками с четырехмерными ядрами с целью выявления четырехмерных характеристик изменения опухоли по переменной времени. При этом четырехмерная модель строится из стека трехмерных моделей в количестве не менее четырех стеков, выполненных в последовательные моменты времени в интервале от четырех месяцев до 2 лет. Это позволяет сделать вывод о скорости изменения опухоли и дает необходимую информацию лечащим врачам.

Оба приложения выполняют следующие шаги:

- 1) нормализация интенсивности трехмерных стеков изображения;
- 2) размывание изображения гауссовой пирамидой с целью удаления шумов и несущественных деталей;
- 3) регистрация стеков изображений пациента; особенно этот шаг важен для четырехмерного варианта алгоритма;
- 4) обучение нейронной сети на имеющемся размеченном материале;
- 5) Тестовая проверка работы алгоритмов.
- 6) Анализ новых изображений.

Как обычно в нейронных сетях, основные усилия после отладки и проверки работоспособности приложений прилагаются к пониманию того, по каким признакам построенная нейронная сеть классифицирует объекты. Для этого приходится строить искусственные изображения и изучать промежуточные результаты на скрытых слоях сети. Кроме того, приходится экспериментировать с выбором количества скрытых слоев и количества нейронов в каждом слое.

#### Список использованных источников:

1. Filler A. G. The History, Development and Impact of Computed Imaging in Neurological Diagnosis and Neurosurgery: CT, MRI, and DTI / A. G. Filler // Nature Precedings. – 2010. – Vol. 7, №1. P. 1–85.
2. Zhou J. Practical Data Acquisition Method for Human Brain Tumor Amide Proton Transfer (APT) Imaging / J. Zhou, et al. // Magnetic Resonance in Medicine. – 2008. – Vol. 60. – P. 842–849.
3. Angulakshmi M. Automated Brain Tumour Segmentation Techniques – A Review / M. Angulakshmi, G.G. Lakshmi Priya // International Journal of Imaging Systems and Technology. – 2017. – Vol. 27, P. 66–77.
4. Athency A. / Brain Tumor Detection and Classification in MRI Images Antony Athency, M.A. Ancy Brigit, K.A. Fathima, Raju Dilin, M.C. Binish // International Journal of Innovative Research in Science, Engineering and Technology. – 2017. – Vol. 6, № 5, P. 84–89.

## МИКРОСЕРВИСЫ И ОБЛАКА В РАЗРЕЗЕ IoT

Басов Д.А., Жук Д.А.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Селезнёв И.Л. – к.т.н., доцент

Устройства, составляющие инфраструктуру IoT управляют и собирают всевозможные форматы данных, что существенно повышает сложность разработки, поддержки и расширяемости решений в данной области. Возможным вариантом выхода из данной ситуации является преобразование данных IoT в базу данных пользователей и обеспечить выполнение запросов к ней с помощью микросервисов. Это решение позволяет эффективно проводить проверку подлинности данных от контроллеров IoT и проводить аналитику в формате больших данных.

Интернет вещей (IoT), который соединяет и обеспечивает связь между обширной сетью датчиков, приборов и объектов, является сегодня одной из самых перспективных ИТ концепций. IoT на настоящее время производит революцию во многих сферах, от автоматизации вождения автомобилей до личных помощников. Бизнес рассматривает IoT как возможность подключить миллиарды новых устройств к Интернету и генерировать большие суммы доходов.

Эти аргументы заставляют задать фундаментальный вопрос, что же на самом деле означает IoT? Большинство принимает этот термин дословно, полагая, что он просто подразумевает подключение датчиков на различных устройствах и сами машины непосредственно к сети Интернет. Тем не менее, эта модель создает потенциальные проблемы. Если IoT датчики и элементы управления находятся в сети Интернет, они могут быть подвержены атакам из любой точки всемирной паутины.

В то же время, сегодня уже миллиарды датчиков и элементов управления размещены как в домах, так и в офисах – и они не просто подключены к Интернет. Линии питания и подключения большинства современных датчиков и элементов управления обеспечивают их связь с единым контроллером, который интерпретирует полученные данные и действует на основе результатов их обработки. В последнее время эти контроллеры получили возможность подключения к сети Интернет, и большинство систем «Умный дом» использует именно эту модель.

В модели контроллера владелец дома или бизнеса устанавливает датчики и контрольные точки, обеспечивающие связь с контроллером. Контрольная точка осуществляет основные функции управления, а также может отправлять данные на мобильное устройство для удаленного управления. Контроллер должен быть подключен к сети Интернет и обеспечивать гарантии безопасности. Данная модель контроллера может решить многие вопросы безопасности IoT для пользователей. Тем не менее, другие проблемы остаются, например, неопределенность с протоколированием и сертификацией форматов данных, задача поиска и аутентификации датчика. Решение этих вопросов может привести к внедрению значительных изменений для облачной среды организации.

**Переосмысление подхода к IoT.** Сенсорные устройства управляют и собирают различные форматы данных. Пользователи, подключающиеся к датчику IoT, вряд ли знают формат данных обмена, кроме случая, когда они стандартизированы. Не зная формата данных, интерпретировать их и осуществлять управление невозможно. Развитая инфраструктура IoT предполагает большое количество датчиков и множество их мест размещения. Данные, поступающие от одного типа сенсоров ситуативны и отражают условия непосредственно в зоне своего охвата.

Одним из решений является преобразование данных IoT в базу данных пользователей, обеспечение выполнения запроса к которой можно осуществлять с помощью микросервисов. Аналитики могут извлечь данные по запросу -- не только на текущий период, но также и за прошлые периоды, если проверку подлинности данных от контроллеров IoT собирать в удобном формате больших данных. Процесс хранения может помочь согласовать форматы данных от разных датчиков в общий слой и идентифицировать данные по местоположению или множеству других факторов, которые облегчают к ним доступ и их использование.

Предлагаемая модель базы данных переносит IoT из локальной сети в облако (рисунок 1). Пользователям по-прежнему необходимо собирать данные от устройств IoT, но также идентифицировать и хранить их, чтобы обеспечить легкость доступа. Эта модель также требует, чтобы решения безопасности IoT организации перешли на уровень облачных сред, а не оставались на уровне сети. Облачные активы, развивающиеся для использования через приложения без непосредственного участия приложений (как и активы IoT, поскольку датчики не являются частью пользовательских приложений) также требуют специального планирования для решения вопросов форматов данных и поддержки синхронизированного анализа данных от нескольких источников IoT. Если архитектура на основе контроллеров, может решить эту проблему, то масштабируемость приложений IoT может оказаться сложной задачей.

IoT подход на основе баз данных и микросервисов также предлагает улучшение поддержки конфиденциальности и политики безопасности. Так как шаблоны запросов непосредственно видны, IoT системы, основанные на микросервисах и запросах, легче обнаружить при попытке отслеживать местоположение человека.

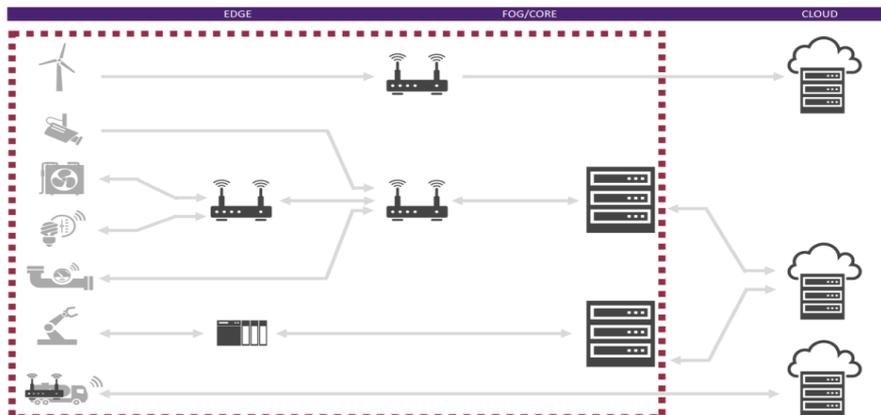


Рисунок 1 – Перенос IoT из локальной сети в облако

Миссия IoT состоит в подключении активов, обеспечивающих поступление ценной информации и влияющих на выполнение важных задач, но поток информации от этих активов является основой в инфраструктуре интернета вещей. Принятие подхода к IoT, основанного на базах данных, может предоставить всю необходимую нам информацию без потенциальных рисков и затрат.

**Список использованных источников:**

1. Building Microservices: Designing Fine-Grained Systems / Sam Newman // O'Reilly Media 2015 -- 280 s
2. Интернет вещей. Исследования и область применения / Е. Зараменских, И. Артемьев // Инфра-М 2017 – 188 с.
3. Архитектура интернета вещей / Ли Перри // ДМК-Пресс, 2019 – 454 с.

## ПРИМЕНЕНИЕ АЛГОРИТМА ОБРАТНОГО РАСПРОСТРАНЕНИЯ ОШИБКИ ДЛЯ КОЛОРИЗАЦИИ ИЗОБРАЖЕНИЙ

*Беликова Т.О., Евсаев П.В.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Лукашевич М.М. – к.т.н., доцент*

Колоризация – любой процесс, в результате которого в последовательность изображений или в монохромное изображение добавляется цвет.

Алгоритм обратного распространения ошибки представляет собой распространение сигналов ошибки от выходов сети к её входам, применяя градиентные методы оптимизации и осуществляя подбор весов для многослойной сети [1]. Ключевым фактором здесь является целевая функция, которая формируется в виде квадратичной суммы разностей между полученными и требуемыми значениями выходных сигналов.

Для обучения сети нужны обучающая выборка и расчёт соответствующих значений сигналов нейронной сети, а также необходимо минимизировать значения целевой функции. Выборка будет состоять из цветных изображений, их полутоновых копий изображений локальных признаков, построенные для каждого полутонового изображения. Одним из способов обучения для непрерывной целевой функции является градиентные методы оптимизации.

Каждый цикл обучения состоит из следующих этапов [1]:

- 1) сгенерировать входные сигналы и вычислить производные значения функции активации каждого слоя, значения выходных слоёв (включая скрытый слой);
- 2) изменить направления сигналов и заменить функции активации на производные от них, подать разность между ожидаемым и полученным значением на новый вход сети и произвести расчёт значений требуемых обратных разностей;
- 3) на основе результатов, полученных в п.1 и 2, обучить оригинальную сеть и сеть обратного распространения ошибки;

4) для каждой обучающей выборки необходимо повторить пункты 1,2 и 3. Процесс остановится, когда норма градиента упадет ниже значения точности процесса обучения.

По описанному алгоритму нейронная сеть обучится восстанавливать только одну цветовую компоненту. Посредством того, что любой цвет представляется в виде трёх компонент: красный, зеленый и синий (RGB) [2] можно сделать вывод, что для колоризации изображения понадобится три нейронных сети.

Таким образом для каждой составляющей цвета будет своя нейронная сеть (рисунок 1).

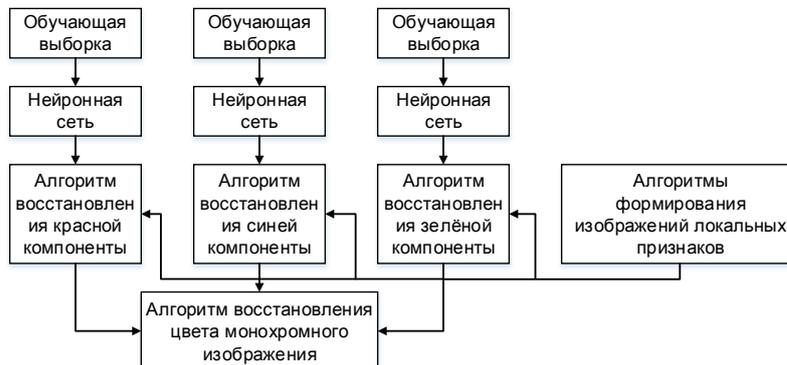


Рисунок 1 – Схема получения восстановления цвета монохромного изображения

После обучения, применяется алгоритм раскрашивания изображений:

- 1) выбирается тип сцены соответствующий содержанию исходного монохромного изображения;
- 2) происходит построение набора локальных признаков изображения;
- 3) формируется синяя составляющая выходного цветного изображения. Нейронная сеть проходит по каждой точке полутонового изображения;
- 4) пункт 3 повторяется для красной и зеленой составляющей выходного изображения;
- 5) из трёх полученных компонент собирается цветное изображение.

**Список использованных источников:**

1. Haykin S. Neural networks, a comprehensive foundation. – N.Y.: Macmillan College Publishing Company, 1994  
2. Гонсалес Р., Вудс Р. Цифровая обработка изображений, 3-е издание, исправленное и дополненное. — М.: Техносфера, 2012. — 1104 с.

## КЛАССИФИКАЦИЯ ГАЛАКТИК С ПОМОЩЬЮ КАПСУЛЬНЫХ НЕЙРОННЫХ СЕТЕЙ

*Береснев Д.В.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Самаль Д.И. – к.т.н., доцент*

Рассмотрена проблематика вопроса классификации галактик. Дано описание данных, а также их специфика. Представлена характеристика капсульных нейронных сетей. Рассмотрены особенности архитектуры капсульных нейронных сетей. Рассмотрено применение капсульных нейронных сетей для решения задачи классификации галактик. Приведены результаты работы капсульной нейронной сети на тренировочном и тестовом наборе данных. Сделаны выводы касательно применимости капсульных нейронных сетей для решения задачи морфологической классификации галактик, а также отмечены перспективы дальнейших исследований.

Очень быстрые темпы прироста объемов информации, улучшение характеристик собранных данных позволяют осваивать новые направления исследований в астрономии и астрофизике. Однако задачи хранения и структуризации очень больших массивов собранных данных требуют внедрения совершенно новых подходов в обработке и разработки новых методов анализа. Таким образом, возникает острая необходимость применения новых методов к последним собранным данным, и астрономические снимки галактик не являются исключением. Поэтому решение задачи классификации галактик крайне актуально сегодня в астрономии и астрофизике.

**Галактика** — это массивная связанная гравитацией система, которая состоит из звезд, звездных остатков, межзвездного вещества и большого количества некоторой субстанции, которая в

современной астрономической и астрофизической литературе условно называется темной материей.[1] *Решение задачи морфологической классификации галактик* — это процесс разработки системы разделения галактик на группы по визуальным признакам, используемая в астрономии и астрофизике. В современной астрономии есть несколько схем для разделения галактик на морфологические типы и одна из наиболее популярных схем была выдвинута Эдвином Хабблом, и впоследствии доработана другими исследователями. На рис. 1 изображена последовательность Хаббла, которая отображает схему классификации на три основных типа галактик: *эллиптические* (пример эллиптической галактики изображен на рис. 2), *спиральные* (пример спиральной галактики изображен на рис. 3) и *иррегулярные* (пример иррегулярной галактики изображен на рис. 4).

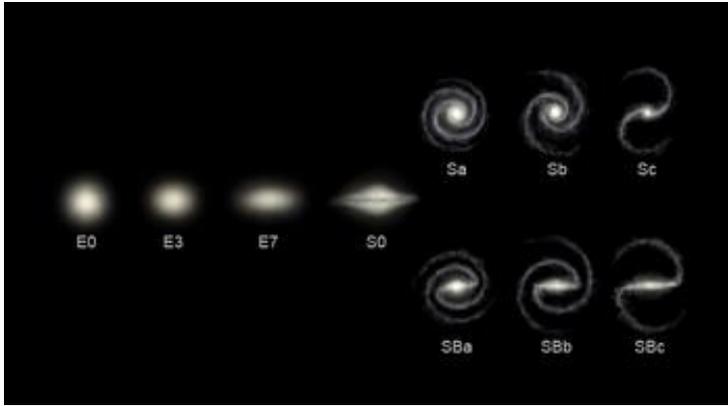


Рисунок 1 – Последовательность Хаббла

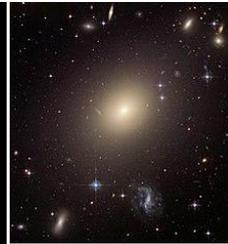


Рисунок 2 – Эллиптическая галактика



Рисунок 3 – Спиральная галактика



Рисунок 4 – Иррегулярная галактика

Данные, которые использовались для анализа в рамках данной работы были взяты из проекта **GalaxyZoo**. GalaxyZoo — это международный проект краудсорсинговой (гражданской) науки, одной из задач которого является классификация изображений галактик, собранных телескопами проекта SDSS. *Sloan Digital Sky Survey* (англ. «Слоановский цифровой небесный обзор») — проект крупномасштабного многоспектрального исследования изображений и спектров красного смещения звёзд и галактик при помощи телескопа, который расположен в обсерватории Апаچی-Пойнт в штате Нью-Мексико. Стоит отметить, что большой объем данных в проекте GalaxyZoo классифицировался вручную волонтерами всего мира, что безусловно очень медленно. Таким образом, чтобы повысить эффективность работы с данными, крайне необходимо автоматизировать процесс посредством разработки интеллектуальной системы на базе технологий глубинного обучения. Поэтому, в рамках данной работы рассматривается модель капсульной нейронной сети для решения задачи морфологической классификации галактик. Также, для решения данной задачи использовался практико-ориентированный метод, на базе которого производилось построение программного средства для классификации, подробно описан в [2]. Данный подход использует «дерево принятия решения» из проекта GalaxyZoo.

**Капсульная нейронная сеть** (*capsule neural network, CapsNet*) — это архитектура искусственных нейронных сетей, которая разработана для решения задач распознавания изображений. Положительными особенностями данной нейросетевой архитектуры являются значительно меньшее количество данных, которое требуется для процесса обучения, а также точность распознавания, которая выше по сравнению со сверточными нейронными сетями. Важнейшим новшеством капсульных нейросетей являются капсулы — это структурные единицы капсульной нейронной сети, являющиеся промежуточными между слоями и нейронами. Капсулы — это группы специализированных нейронов, которые отслеживают не только отдельные детали и фрагменты изображения, но и расположение этих фрагментов друг относительно друга. Данная архитектура была придумана Джеффри Хинтоном в 1979 году, сформулирована в 2011 году и опубликована в 2017 году. Архитектура капсульной нейронной сети представлена на рис. 5 [3].

На вход нейронной сети CapsNet подается изображение, которое, стоит отметить, имеет трехмерное представление ввиду наличия цветового канала. Первая часть нейросети CapsNet — сверточный слой. Целью данного слоя является извлечение из входного изображения базовых признаков. ReLU — это активационная функция, аргументом которой является значение. Если это значение отрицательно, ReLU зануляется, если положительное — принимает значение аргумента. Целью данного активационного слоя является добавление нелинейности, которая позволяет описывать все виды функций. Далее следует слой PrimaryCaps, который состоит из множества *капсул*. *Капсула* — это группа нейронов в некотором слое капсульной ИНС, которые являются выходным представлением свойств некоторой сущности на изображении. Затем следует процесс

сжатия, который преобразует значения выходного вектора капсул таким образом, что меняется только его длина, а углы остаются постоянными. Таким образом, получается вектор в диапазоне значений от 0 до 1, что соответствует вероятности. Следующий шаг — это применение алгоритма динамической маршрутизации. В CNN используется операция *max pooling* как способ уменьшить размер. Однако в капсульных сетях используется другой метод — направление по соглашению (*routing by agreement*).

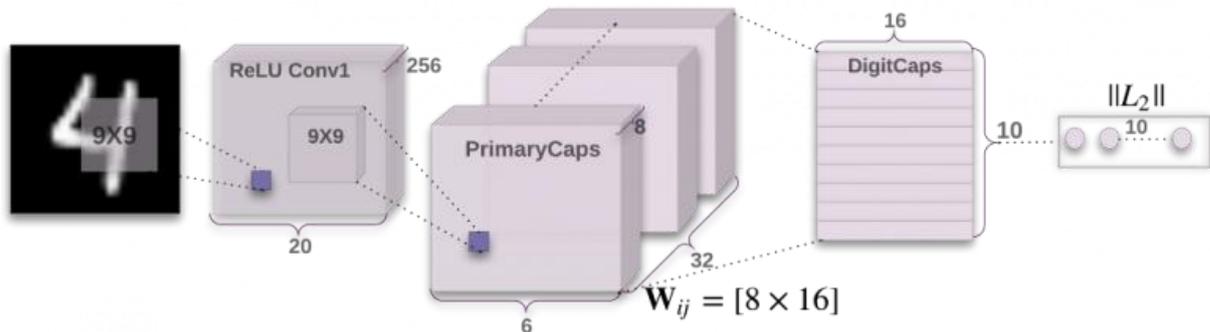


Рисунок 5 – Архитектура капсульной нейронной сети

Капсульные нейронные сети — крайне перспективная архитектура нейронных сетей для решения задачи классификации, которая улучшает распознавание изображений при изменяющихся ракурсах и иерархической структурой. Процесс обучения капсульных нейронных сетей осуществляется с помощью динамической маршрутизации между капсулами. Капсульные сети снижают ошибку распознавания объекта в другом ракурсе в сравнении со сверточными неросетями [4,5]. В настоящее время данная архитектура модифицируется и дорабатывается с целью получения результатов на сравнительно меньшем объеме данных, чем этого требуют CNN [6, 7].

**Список использованных источников:**

1. Сурдин, В.Г. Галактики / В.Г. Сурдин. – М.: ФИЗМАТЛИТ, 2013. – 432 с.
2. Willet, K. W. Galaxy Zoo 2: detailed morphological classifications for 304,122 galaxies from the Sloan Digital Sky Survey / K. W. Willet, Chris J. Lintott, Steven P. Bamford. – USA : Coornell University, 2013. – 30 p. – Mode of access: <https://arxiv.org/pdf/1308.3496.pdf>. – Date of access: 22.03.2019.
3. Sabour, S. Dynamic Routing Between Capsules / S. Sabour, N. Frosst, G. E. Hinton. – Toronto: Google Brain, 2017. – 11 p. – Mode of access: <https://arxiv.org/pdf/1710.09829.pdf>. – Date of access: 22.03.2019.
4. Mukhometzianov, R. CapsNet comparative performance evaluation for image classification / R. Mukhometzianov, J. Carrillo. – Canada: University of Waterloo, 2018. – 14 p. – Mode of access: <https://arxiv.org/ftp/arxiv/papers/1805/1805.11195.pdf>. – Date of access: 22.03.2019.
5. Xi, Ed. Capsule Network Performance on Complex Data / Ed. Xi, S. Bing, Y. Jin. – Pittsburgh: Carnegie Mellon University, 2017. – 7 p. – Mode of access: <https://arxiv.org/pdf/1712.03480.pdf>. – Date of access: 22.03.2019.
6. Hinton, G. E. Matrix capsules with EM routing / G. E. Hinton, S. Sabour, N. Frosst. – Toronto: Google Brain, 2018. – 15 p. – Mode of access: <https://openreview.net/pdf?id=HJWlfgWRb>. – Date of access: 22.03.2019.
7. Lin, A. On Learning and Learned Representation with Dynamic Routing in Capsule Networks / A. Lin, J. Li, Z. Ma. – Guangdong: Guangdong Polytechnic Normal University, 2018. – 12 p. – Mode of access: <https://arxiv.org/pdf/1810.04041.pdf>. – Date of access: 22.03.2019.

## ЗНАЕТЕ ЛИ ВЫ, ЧТО ПРОИСХОДИТ, КОГДА ГОВОРите МИРУ “ПРИВЕТ!”?

Борушко А.Н.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Киклевич У.С. – ассистент кафедры ЭВМ

В данной статье на примере простейшей программы показывается, что в программировании за простыми вещами стоят более сложные абстракции.

При нынешних скоростях развития языков программирования, появления новых техник, фреймворков и технологий мы начинаем забывать то, с чего все начинали.

Каждый, кто хоть когда-нибудь пробовал написать свою первую строчку кода, писал именно “Hello, World!”.

Все знают, что это такое и что мы получим в консоли, но далеко не каждый знает то, как работает программа “Привет, мир!”

Когда люди пишут такие незамысловатые программы, они даже не подозревают, что стоит за несколькими строками кода.

На примере языка программирования Си и операционной системы Windows я более подробно рассмотрю принцип работы простейшей программы.

Первое, что происходит в процессе компиляции – над программой начинает работать препроцессор. Он буквально копирует в наш `main.c` файл все, что задано с помощью **#include**. Таким же образом мы подключаем к нашей программе библиотеку **stdio.h**. Препроцессор же копирует в нашу программу все стандартные функции, которые объявлены в заголовочном файле **standard input/output**.

После чего компилятор начнет переводить наш код в машинный.

Компоновщику (**linker**) передается параметр **/subsystem**, который задает тип приложения, например **/SUBSYSTEM:WINDOWS** для оконных программ, или **/SUBSYSTEM:CONSOLE** для консольных программ. Это нужно операционной системе. По типу приложения, который прописывается в заголовке **.exe** файла, операционная система понимает, какое приложение ей нужно запускать - оконное или консольное.

На строчке **int main** мы обозначаем входную точку (**entry point**) для нашей программы.

На самом деле точкой входа является некоторая функция, которая будет вызвана операционной системой при старте приложения. В C/C++ это функция рантайма (**CRT**), которая проводит инициализацию **CRT**, вызывает конструкторы глобальных объектов. Для каждой платформы эта функция имеет свое название. Для Windows это **mainCRTStartup**. Если же этой стартовой точки не будет у вас программе, то компилятор выдаст ошибку:

**"Entry point must be defined"**.

Стартовые точки могут быть абсолютно разными. По умолчанию стартовой точкой является функция **int main**, которая возвращает число родителю процессу. В нашем случае родительским процессом является терминал. Терминалом определено, что дочерние процессы должны возвращать 8-ми битные целые числа, которые служат индикатором успешного завершения программы или завершения с ошибкой.

Внутри этой функции мы определяем поведение программы. В нашем случае мы вызываем функцию **printf** из стандартной библиотеки, которая может выводить данные разных типов в терминал. В конце мы возвращаем число ноль родителю процессу, чтобы обозначить успешное завершение программы.

Стандартный компилятор Windows позволяет скомпилировать Си код, но не проводит компоновку, что дает нам возможность просмотреть **.obj** файл нашей программы. В нем хранится машинный код программы. В заголовке объектного файла имеются три основных сегмента: **.data**, **.drectve**, **.text\$mn**.

В секции **.data** хранятся инициализированные глобальные и статические поля.

В **.drectve** хранятся команды для компоновщика. Данная секция удаляется после компоновки объектного файла.

**.text\$mn** или **text segment** содержит в себе скомпилированные под платформу X машинные инструкции. Во время выполнения **.exe** файла загрузчик программ наполнит определенный сектор процесса нашей программы этими инструкциями.

После компиляции генерируется символьная таблица и таблица перенаправлений.

Символьная таблица (Рисунок 1) содержит всю информацию по использованным символам, на которые ссылается подаваемый на вход компилятора исходный текст.

В первой колонке таблицы содержится индекс символа или его номер. Во второй виден адрес символа перед перемещением.

Если в третьей колонке содержится **SECT** и номер – это означает, что символ определен в данной секции объектного файла, если **UNDEF**, то символ находится в глубине **CRT**. Также в символьной таблице можно увидеть является ли символ **External** или **Static**. Если он **Static**, то символ виден только в исходном тексте нашей программы (**translation unit**), если **External**, то он виден всем остальным файлам.

В таблице перенаправлений, в колонке под названием **Offset**, указывается адрес, куда мы перенаправляем символ. А в колонке **Applied To** указан адрес символа в той или иной секции объектного файла.

С помощью системной утилиты **dumpbin** мы можем просмотреть **assembly** код. Так как **assembly** имеет одинаковые маппинги с машинным кодом, то в нем есть: адрес памяти, машинный код и эквивалентные ему команды **assembly**. В нем содержится функция **\_main**, которая делает вызов **\_printf**. Сам **\_printf** делает еще два вызова, один прямоком в **CRT** чтобы добавить возможность стандартного ввода/вывода, и **\_vfprintf\_I**, которая используется для работы самой функции **printf**. В версии компилятора начиная с 2015 года, части функции **printf** уже содержатся в заголовочном файле.

Для того, чтобы получить .exe файл, нужно провести компоновку объектного файла. В заголовке .exe файла можно увидеть дополнительную секцию .rdata. Именно в этой секции находится строка со значением "Hello, World!"

Таким образом я продемонстрировал, что в программировании ничего не работает просто так. Даже за самой минимальной программой лежит большое количество "прослоек", которые работают не идеально и требуют постоянных доработок и оптимизации.

```
Dump of file Source.obj
File Type: COFF OBJECT

COFF SYMBOL TABLE
000 01046992 ABS      notype      Static      | @comp.id
001 80000191 ABS      notype      Static      | @feat.00
002 00000000 SECT1   notype      Static      | .drectve
  Section length  2F, #relocs  0, #linenums  0, checksum    0
004 00000000 SECT2   notype      Static      | .debug$$
  Section length  78, #relocs  0, #linenums  0, checksum    0
006 00000000 SECT3   notype      Static      | .text$mn
  Section length  14, #relocs  2, #linenums  0, checksum F829D91C
008 00000000 SECT4   notype      Static      | .text$mn
  Section length   A, #relocs  1, #linenums  0, checksum 71A05264, selection  2 (pick any)
00A 00000000 SECT5   notype      Static      | .text$mn
  Section length  29, #relocs  2, #linenums  0, checksum 2B25B17F, selection  2 (pick any)
00C 00000000 SECT6   notype      Static      | .text$mn
  Section length  3A, #relocs  2, #linenums  0, checksum CAE6D625, selection  2 (pick any)
00E 00000000 SECT4   notype      External    | __local_stdio_printf_options
00F 00000000 UNDEF   notype      External    | __acrt_iob_func
010 00000000 UNDEF   notype      External    | __stdio_common_vfprintf
011 00000000 SECT5   notype      External    | __vfprintf_l
012 00000000 SECT6   notype      External    | _printf
013 00000000 SECT3   notype      External    | _main
014 00000008 UNDEF   notype      External    | ?_OptionsStorage@?1??_local_stdio_printf_options@@@9@9
printf_options'::'2'::_OptionsStorage)
015 00000000 SECT7   notype      Static      | .data
  Section length   D, #relocs  0, #linenums  0, checksum AF57950C
017 00000000 SECT7   notype      Static      | $SG7440
018 00000000 SECT8   notype      Static      | .chks64
  Section length  40, #relocs  0, #linenums  0, checksum    0
```

Рисунок 1 – Символьная таблица

## ВЫЯВЛЕНИЕ DNS ТУННЕЛИРОВАНИЯ В КОМПЬЮТЕРНОЙ СЕТИ МЕТОДОМ POMDP

*Бубнов Я.В.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Иванов Н.Н. – к.физ-м.н., доцент*

В статье предлагается способ выявления целевой кибератаки, в частности DNS туннелирования в компьютерной сети. Предложенный метод базируется на использовании параметризованного частично-наблюдаемого марковского процесса принятия решений.

DNS туннелирование — это техника инкапсуляции произвольных данных через протокол системы доменных имен. В течение последних лет атаки, связанные с кражей информации из платежных систем, так или иначе эксплуатировали данную уязвимость протокола. Ярким примером может служить программа FrameworkPOS [1], с помощью которой в 2014-2015 годах осуществлялась масштабная кража информации о кредитных картах с платежных систем. Основная трудность, связанная с выявлением DNS туннелирования, заключается в сложности дифференциации туннелируемого трафика от обычного.

В виду того, что отключение системы DNS в компьютерных сетях представляется принципиально невозможным, требуются более интеллектуальные способы решения задачи — обнаружения DNS туннелирования.

За последние годы исследования проблемы обнаружения DNS туннелирования в компьютерных сетях предложено некоторое количество подходов в построении детекторов. Например, в статье [2] Надлер, Аминов и Шабтай решают проблему обнаружения туннелирования

методом бинарной классификации сетевого трафика, добившись в своих экспериментах 99-процентной чувствительности детектора. В статье [3] автор предлагает использовать многослойную feedforward нейронную сеть для классификации трафика по различным методикам туннелирования, что предоставляет возможность дифференциации политик блокировки в зависимости от подвида атаки. Автор демонстрирует 85-процентную точность детектора на тестовом наборе данных.

На практике, эффективность вышеупомянутых способов обнаружения DNS туннелирования в значительной степени зависят от исходного, обучающего набора данных. Процесс обучения модели, в свою очередь, становится зависимым от типов туннелирования, масштабов и конфигурации сети.

С целью решения поставленной проблемы: детектирования DNS туннелирования при использовании неидеальных детекторов, в работе [4] МакКарти предлагает использовать модификацию POMDP, а именно - виртуально-распределенный POMDP. В своей работе МакКарти строит механизм принятия решения о блокировке доменного имени исходя из объема и ценности трафика, направляемого в данный домен.

Очевидные трудности, которые возникают при использовании подхода МакКарти, связаны с необходимостью анализа совокупного трафика сети, что при возрастании масштабов сети, становится трудновыполнимым. Вместо анализа совокупного трафика должен производиться локальный анализ непосредственно DNS запроса.

Для эффективного использования POMDP при обнаружении DNS туннелирования, функция вознаграждения должна быть сформулирована следующим образом. Пусть  $L \in [0, 1]$  — это требуемый уровень обеспечения безопасности сети. Тогда, при  $L = 0$  достигается наименьшая степень защищенности сети, и в то же время наибольший уровень доступности системы, так как вероятность заблокировать важный для функционирования системы домен сводится к нулю. При  $L = 1$  достигается наибольшая степень защищенности, однако вероятность ложно позитивных блокировок доменов возрастает, что сказывается на функционировании сети.

Исходя из этого, L-параметризованную POMDP модель для обнаружения DNS туннелирования можно задать следующим образом:

$S = \{\text{Blocking, Passing}\}$   
 $A = \{\text{Analyze, Block, Pass}\}$   
 $O = \{\text{OK, Bad}\}$

Где  $S$  — совокупность состояний системы: DNS запрос блокируется или пропускается системой.  $A$  — совокупность возможных действий системы: опросить детектор, заблокировать запрос или пропустить запрос.  $O$  — совокупность возможных наблюдений: детектор определил запрос как безопасный, детектор отнес запрос к классу DNS туннелей.

Путь  $Q$  — точность детектора, а  $L$  — уровень безопасности, а  $K$  — коэффициент сложности доступа к детектору, тогда функция вознаграждения  $R(O, A)$  описывается в таблице 1.

Таблица 1 - Функция вознаграждения

O	A	R(O, A)
Bad	Block	L
OK	Block	-(1-L)
Bad	Pass	1-L
OK	Pass	L
Bad	Analyze	L*K
OK	Analyze	L*K

Для проведения эксперимента использовалась тестовая установка, состоящая из клиентской и серверной станций, между которыми установлено SSH соединение через DNS туннель с помощью пакета iodine [5]. На клиентской станции установлен детектор, предложенный в [3], а также L-параметризованная POMDP система, принимающая решение о блокировке исходящих DNS запросов.

На описанной установке поставлены эксперименты со значением параметра  $L \in [0,5; 1,0]$ . В данном эксперименте основной задачей является поиск оптимальной политики для принятия решения о блокировке запроса. Поиск оптимальной политики осуществляется с помощью алгоритма «инкрементальной чистки», описанной в [6].

Пусть детектор с вероятностью  $Q = 0,85$  корректно относит рассматриваемый DNS запрос к нужному классу. Тогда точность распознавания DNS туннелей в зависимости от  $L$  представлена на рисунке 1а, где штриховой линией отмечена точность распознавания исходного детектора. Качество классификации DNS туннелирования оценивается по построенным ROC-кривым, представленными на рисунке 1б.

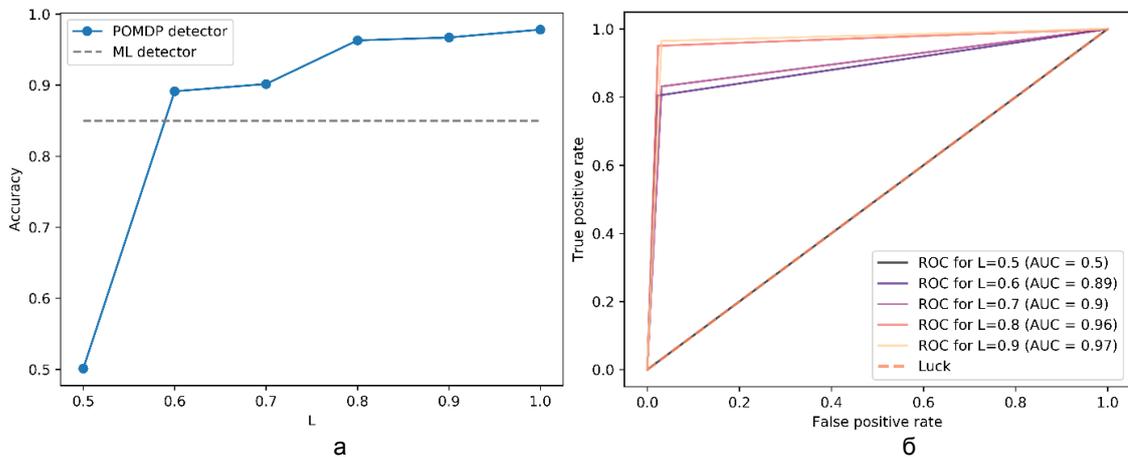


Рисунок 1а – зависимость точности модели от параметра  $L$ ,  
1б – ROC-кривые для различных значений параметра  $L$

Из полученных результатов видно, что рост параметра  $L$  положительно влияет на точность распознавания DNS туннелирования. В то же время, при  $L < 0,6$  качество распознавания резко снижается, и при  $L = 0,5$  носит характер случайного гадания. Стоит отметить, что в этом случае, система производит меньшее количество ложно положительных ошибок, но большее количество ложно отрицательных ошибок. Это объясняется тем, что система пропускает больше подозрительного трафика с целью обеспечения доступности сети в целом.

Таким образом, в поставленном эксперименте, с помощью  $L$ -параметризованного POMDP удалось добиться увеличения точности распознавания DNS туннелирования на 10% в сравнении с детектором, представленным в статье [3].

**Список использованных источников:**

1. Valenzuela, I. Game Changer: Identifying and Defending Against Data Exfiltration Attempts. – Boston : Sans Institute, 2015.
2. Nadler, A. Detection of Malicious and Low Throughput Data Exfiltration Over the DNS protocol / A. Nadler, A. Aminov, A. Shabtai – Negev : Ben Gurion University of the Negev, 2017.
3. Bubnov, Y. DNS Tunneling Detection Using Feedforward Neural Network / Y. Bubnov // European Journal of Engineering Research & Science. – 2018. – Vol. 3, № 11. – P. 16-19.
4. McCarthy, S. Data Exfiltration Detection and Prevention: Virtually Distributed POMDPs for Practically Safer Networks / S. McCarthy, A. Sinha, M. Tambe, P. Manadhata – Los Angeles : University of Southern California, 2016.
5. Iodine [Электронный ресурс]. – Режим доступа: <http://code.kryo.se/iodine>. – Дата доступа: 03.22.2019.
6. Cassandra, A. Incremental Pruning: A Simple, Fast, Exact Method for Partially Observable Markov Decision Process / A. Cassandra, M. Littman, N. Zhang // Proceedings of the Thirteen Conference on Uncertainty in Artificial Intelligence. – 1997. – P. 54-61.

## СВЕРТОЧНАЯ НЕЙРОННАЯ СЕТЬ ДЛЯ РАСПОЗНАВАНИЯ РУКОПИСНЫХ СИМВОЛОВ

Гаджиев С. С.

Белорусский государственный университет информатики и радиоэлектроники,  
г. Минск, Республика Беларусь

Воронов А. А. – к.т.н., доцент

Рассмотрены вопросы построения сверточной нейронной сети для распознавания рукописных символов. Определена основная модель, приведены примеры слоев нейронной сети.

В этой статье опишем структуру сверточной нейронной сети для распознавания рукописных символов. Классическая архитектура CNN представлена на рисунке 1:

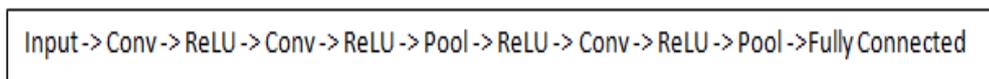


Рис. 1- Архитектура CNN

Рассмотрим, какие действия сеть будет выполнять на высоком уровне. Все фильтры можем рассмотреть, как идентификаторы свойства. Свойством могут быть кривые, прямые границы,

значение цвета. Например, первый фильтр будет иметь размер  $9 \times 9$ , и он будет являться детектором прямых границ. Фильтр имеет пиксельную структуру, численные значения которой выше вдоль области, идентифицирующей форму прямой линии. Когда в левом верхнем углу начального изображения начинает проходить фильтр, он умножает значение фильтра на значения пикселей выделенной зоны.

Выше было упомянуто, что умеют делать фильтры первого свёрточного слоя. Они идентифицируют такие свойства базового уровня, как границы и кривые. Как можно себе представить, чтобы представить какой тип объекта присутствует на картинке, нам нужна нейронная сеть, которая может распознать свойства более высокого уровня, как к примеру глаза, лицо или уши. Нужно подумать, какая картинка будет приходиться на первый слой. К примеру его размер будет  $32 \times 32$ . Когда рисунок пройдет через первый свёрточный слой, выход этого слоя станет входным параметром второго слоя. Сейчас представить это немного сложнее в визуальном виде. Когда описывали первый слой, входом были только данные исходного изображения. Но когда мы перешли ко второму слою, входным параметром для него будет одна или же несколько карт свойств — исход работы первого слоя. Все наборы входных данных определяют места, где на исходном образе находятся требуемые базовые признаки.

Сейчас, когда вы используете набор фильтров поверх этого на выходе активизируются фильтры, которые имеют более высокий уровень. Чем больше количество свёрточных слоёв обрабатывает картинку и чем глубже оно движется по сети, тем более сложные свойства будут входить в карты активации. В конце сети могут быть фильтры, которые активизируются, когда на рисунке есть рукописный текст, есть розовые объекты и т.д.

Еще один увлекательный момент. Когда вы передвигаетесь дальше внутрь сети, фильтры отрабатывают со все большим полем восприятия, и значит, они могут обрабатывать данные с большей площадью начального рисунка.

Теперь, когда мы имеем возможность обнаружить высокоуровневые свойства, мы можем начинать привязывать полносвязный слой в конец сети. Полносвязный слой берёт входную информацию и выводит  $N$ -пространственный вектор, где  $N$  — количество классов, из которых будет выбран нужный. К примеру, если требуется нейронная сеть по распознаванию символов латинского алфавита, и  $N$  будет иметь значение 12, то есть возьмем для примера первые 12 символов из алфавита. Все символы в этом  $N$ -пространственном векторе будут представлять собой вероятность определенного класса. К примеру, если результирующим вектором для сети, которая распознает символы является вектор  $(0, 0,2, 0,15, 0,8, 0, 0, 0,02, 0,01, 0, 0, 0,02)$ , это означает что есть 20% вероятность, что на рисунке "B", 15% вероятность, что на рисунке "C", 80% вероятность — "D", и 2% вероятность — "I".

Метод, благодаря которому работает полносвязный слой — это обращение к выходу предыдущего слоя и нахождение параметров, которые более характеризуют определенный класс. К примеру, если сеть идентифицирует на каком-либо образе кота, у набора свойств, которые отражают высокоуровневые характеристики, такие как лапы или морду кота, должны иметь высокие параметры. Точно так же, если сеть идентифицирует, что на изображении черепаха — у неё будут высокие значения в наборах свойств, представленных высокоуровневыми характеристиками такими как размер и форма панциря или лапы.

Полносвязный слой ссылается на то, что функции более высокого уровня сильно привязаны к определенному классу и имеют некоторые веса, в таком случае, когда высчитываются произведения весов с предыдущим слоем, то получается верные вероятности для различных классов.

Обучение — это один из аспектов нейронных сетей, о котором до сих пор не упоминалось. Вероятно, это самая важная часть. Сейчас мы задаемся вопросами. Как фильтры первого свёрточного слоя узнают, что нужно найти границы и кривые? Как полносвязный слой узнает, что должна найти набор свойств? Как фильтры каждого слоя узнают, что именно хранить? Метод, с которым машина способна подбирать значения фильтра — это обучающий процесс, который известен как метод обратного распространения ошибки.

Метод обратного распространения ошибки можно поделить на 4 различных блока:

прямое распространение;

функцию потерь;

обновление веса;

обратное распространение.

Когда происходит прямое распространение, берётся тренировочный образ — к примеру, это матрица  $28 \times 28$  — и образ пропускают через всю сеть. В первом обучающем примере, так как все веса или значения фильтра были инициализированы случайным образом, выходным значением будет что-то вроде  $[.11, .15, .01, .1, .07, .09, .13, .11, .11, .07]$ , то есть такое значение, которое не даст предпочтения какому-то определенному числу. Нейронная сеть с такими весами не сможет найти параметры базового уровня и не может обоснованно идентифицировать класс образа. Это приводит к функции потерь. Для обучения требуются обучающие данные. У таких данных есть и изображение, и

ярлык. Допустим, первое обучающее изображение — это цифра 2. Ярлыком изображения будет [0 0 1 0 0 0 0 0 0]. Функция потери может быть выражена по-разному. В нашем случае применяется среднеквадратическая ошибка, это 0.5 умножить на (реальность — предсказание) в квадрате.

Примем этот параметр за переменную  $L$ . Как можно догадаться, потеря будет иметь очень высокое значение для первых двух обучающих образов. Теперь давайте подумаем об этом интуитивно. Мы хотим добиться того, чтобы предсказанный класс был таким же, как класс обучающего изображения. Чтобы добиться этого, нужно привести к минимуму количество потерь, которое имеется.

Скорость обучения — это параметр, который выбирается при разработке сети. Высокая скорость обучения означает, что в обновлениях весов выполнялись большие шаги, поэтому образу может понадобиться меньше времени, для того чтобы набрался верный набор весов. Но слишком высокая скорость обучения может привести к очень крупным и неточным шагам, которые могут помешают набрать оптимальные показатели. Процесс прямого распространения, функцию потерь, обратное распространение и обновление весов, обычно называют одной эпохой. Программа будет повторять эпоху определенное количество раз для каждого обучающего образа. После того, как завершится обновление свойств на последнем обучающем образе, сеть предположительно должна быть достаточно хорошо обучена и веса слоёв настроены верно.

Для построения модели нейронной сети было использована библиотека Tensor Flow. Было протестировано на базе данных MNIST. Результат зависел от количества эпох для обучения нейронной сети. К примеру, если использовать 10 эпох, то процент распознавания будет 60%. С увеличением количества эпох до 100 к примеру процент распознавания поднялся до 85%. Так же стоит учитывать количество классов для обучения. Чем больше классов, тем меньше процент распознавания.

#### Список использованных источников

1. Шапиро, Л. Компьютерное зрение / Л. Шапиро, Дж. Стокман. — М. // БИНОМ. Лаборатория знаний, 2006. — 762 с.
2. Гонсалес, Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. // М. : Техносфера, 2005. — 1073 с.

## ПРИНЦИПЫ РАБОТЫ ДИНАМИЧЕСКОЙ ДИСПЕТЧЕРИЗАЦИИ МЕТОДОВ В РАЗЛИЧНЫХ ЯЗЫКАХ ПРОГРАММИРОВАНИЯ

Ганнусенко Н.Н.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Иванов Н.Н. — к.ф.-м.н., доцент

При разработке программного обеспечения можно использовать различные парадигмы программирования. И часто для их применения используются полиморфные объекты — это объекты, которые меняют своё поведение в зависимости от обстоятельств. И динамическая диспетчеризация методов — один из способов употребления таких объектов. Динамическая диспетчеризация методов — это механизм выбора реализации полиморфной операции на этапе времени выполнения программы. А механизмы их реализации уже представлены по-разному в каждом из языков программирования.

В ходе изучения способов реализации динамической диспетчеризации методов в разных языках программирования эти механизмы можно разделить на подходы:

- применение виртуальных таблиц методов;
- применение толстых указателей;
- обращение через таблицы поиска;
- поиск посредством пересылки сообщений;
- переходы через указатели на функции.

В таких языках программирования, как C++, Java, C#, Swift, используются виртуальные таблицы методов. Хоть их реализации и различаются в зависимости от наличия множественного наследования [1], от наличия интерфейсов [2] или протоколов [3, 4], но у реализаций есть общее — выделение вызовов динамического поведения в отдельные виртуальные таблицы методов [5], а доступ к ним осуществляется посредством обращения к указателям на связанные таблицы, ссылки которых хранятся у каждого объекта класса. Пример виртуальной таблицы методов представлен на рисунке 1.

В таких языках программирования, как Rust или Go, используется концепция толстых указателей. Принцип её работы схож с подходом использования виртуальных таблиц методов, но имеется разница в том, где конкретно хранится указатель на виртуальную таблицу методов. В толстых указателях ссылки на соответствующие виртуальные таблицы методов хранятся вместе

указателем на сам объект, а не внутри его данных [6, 7]. Пример толстого указателя представлен на рисунке 2.



Рисунок 1 – Пример виртуальной таблицы методов для “Cat”

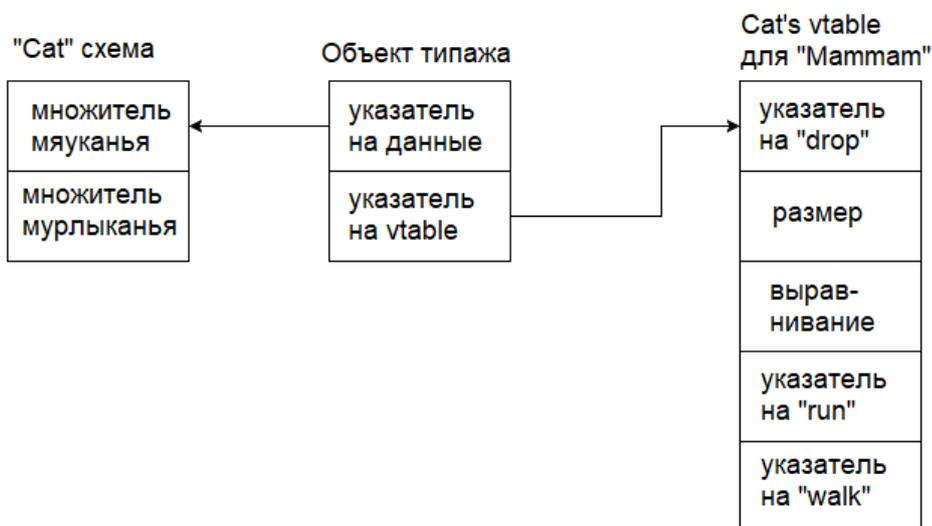


Рисунок 2 – Пример толстого указателя для “Cat”

В интерпретируемых языках программирования, таких как Python, Ruby, Javascript, разрешается поиск методов посредством обращения к ним через таблицы поиска [8]. При таком подходе в каждом объекте хранится ссылка на свою таблицу поиска методов и поиск конкретной реализации происходит через обход таблицы поиска в текущем объекте и в иных связанных местах в зависимости от определенного порядка разрешения методов в конкретном языке программирования.

В языке программирования Objective-C, а также в рамках обратной совместимости и в Swift, методы разрешаются посредством механизма пересылки сообщений [9]. В реализации этого механизма в данных языках программирования в каждом объекте хранится ссылка на метаданные, в которых есть указатель на все реализованные методы. И поиск метода происходит путем конвертирования имени методов и соответствующие индексы методов класса, по которым потом идет обращение в список методов метакласса объекта. Пример заполнения методов метаклассов при использовании механизма пересылки сообщений представлен на рисунке 3.

В языке программирования C также имеется механизм динамической диспетчеризации методов, и достигается он посредством манипулирования указателями на функции [10]. Для того, чтобы этот механизм использовать, определяется структура данных со всеми связанными указателями на функции и далее заполняется ссылками на необходимые указатели на функции в зависимости от требуемого поведения в конкретной задаче.

Таким образом, были выделены различные подходы к реализации механизма динамической диспетчеризации методов, а также кратко рассмотрены их конкретные реализации в различных языках программирования.

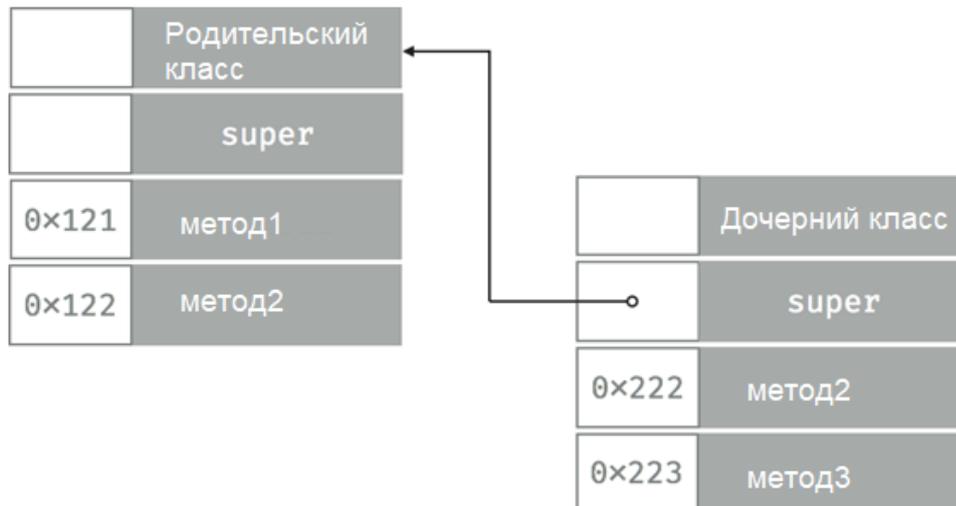


Рисунок 3 – Пример реализации механизма пересылки сообщений

**Список использованных источников:**

1. Understanding Virtual Tables in C++ [Электронный ресурс]. – Режим доступа: <https://pabloariasal.github.io/2017/06/10/understanding-virtual-tables/> – Дата доступа: 19.03.2019.
2. The Black Magic of (Java) Method Dispatch [Электронный ресурс]. – Режим доступа: <https://shipilev.net/blog/2015/black-magic-method-dispatch/> – Дата доступа: 19.03.2019.
3. Method Dispatch in Swift [Электронный ресурс]. – Режим доступа: <https://www.raizlabs.com/dev/2016/12/swift-method-dispatch/> – Дата доступа: 20.03.2019.
4. Secrets of Swift's Speed [Электронный ресурс]. – Режим доступа: <https://www.mikeash.com/pyblog/friday-qa-2014-07-04-secrets-of-swifts-speed.html> – Дата доступа: 20.03.2019.
5. Interface dispatch [Электронный ресурс]. – Режим доступа: <https://lukasatkinson.de/2018/interface-dispatch/> – Дата доступа: 20.03.2019.
6. Exploring Dynamic Dispatch in Rust [Электронный ресурс]. – Режим доступа: <https://alschwalm.com/blog/static/2017/03/07/exploring-dynamic-dispatch-in-rust/> – Дата доступа: 21.03.2019.
7. Блэнди Дж., Орендорф Дж., Программирование на языке Rust / пер. с англ. А. А. Слинкина. – М.: ДМК, Пресс, 2018. – 550 с.: ил, стр. 227 – 228.
8. Method resolution order in Python Inheritance [Электронный ресурс]. – Режим доступа: <https://www.geeksforgeeks.org/method-resolution-order-in-python-inheritance/> – Дата доступа: 21.03.2019.
9. Objective-C vs Swift messages dispatch [Электронный ресурс]. – Режим доступа: <https://blog.untitledkingdom.com/objective-c-vs-swift-messages-dispatch-9d5b7fd58327> – Дата доступа: 21.03.2019.
10. Мартин Р., Чистая архитектура. Искусство разработки программного обеспечения. – СПб.: Питер, 2018. – 352 с.: ил., стр. 60 – 63.

## ПРОГРАММНАЯ РЕАЛИЗАЦИЯ СИСТЕМЫ ХРАНЕНИЯ ОБЪЕКТОВ АВТОРСКИХ ПРАВ НА ОСНОВЕ ТЕХНОЛОГИИ БЛОКЧЕЙН

Глоба А.А.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Ганжа В.А. – к.физ.-мат.н., доцент

Разрабатываемая система служит для хранения объектов авторских прав с использованием технологии блокчейн. Блокчейн в общем виде – это выстроенная в соответствии с определенными правилами цепочка блоков, содержащих какую-либо информацию. Обычно копии цепочек хранятся распределённо на разных устройствах в пределах одной сети. С этой базы данных невозможно ничего удалить или провести замену/подмену блока, доступно только чтение.

В данный момент разрабатываемая БД представляет собой структуру, представленную на рис.1:

В ходе разработки системы для хранения авторских прав, было принято решение разделить БД на несколько цепочек, где каждому типу данных (текстовые, аудио-, изображения) соответствовала своя цепочка. Более того, на каждую отдельную цепочку накладывались свои правила для прохождения валидации. Такой подход обеспечивает гибкость и относительно легкую

расширяемость. Каждой цепочке соответствует свой файл с данными, в котором связь осуществляется через поле id, а контроль целостности – с помощью хэша MD5.

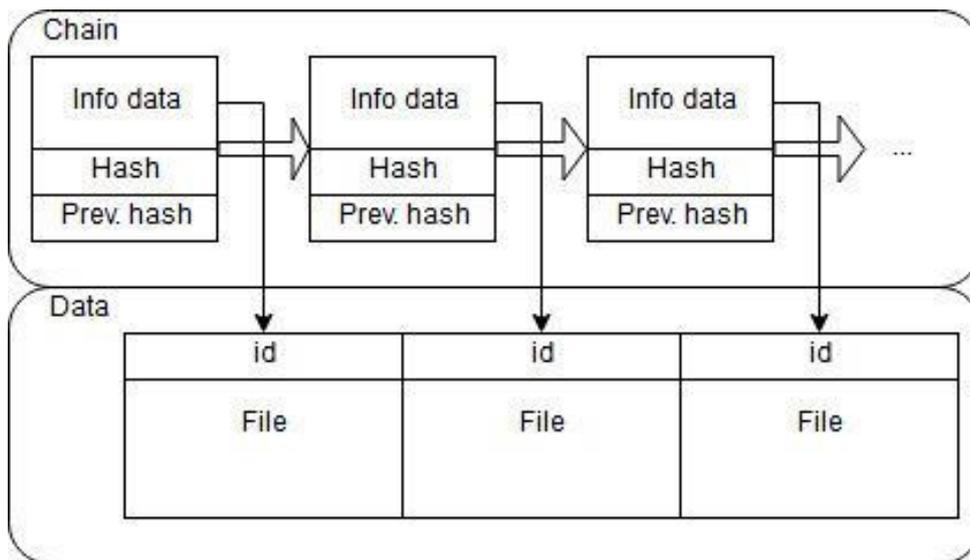


Рис.1 – структура данных в разрабатываемой системе

Сама разрабатываемая система разделена на 3 уровня:

1. Уровень данных (цепочки блокчейн).
2. Уровень API – HTTP-интерфейс для взаимодействия с цепочками.
3. Уровень представления – графический интерфейс пользователя.

Важным условием работоспособности разрабатываемой системы является возможность обмена данными и обновлениями между узлами сети, а также разрешение возможных конфликтов при этом. Для этого используется следующий алгоритм (пример для одного типа цепочки):

1. Узел запрашивает цепочки от всех известных ему узлов.
2. Если получает цепочку, длина которой больше, чем длина цепочки данного узла ( $L_{new} > L_{current}$ ), то новая цепочка проходит валидацию. Так же при этом от удаленного узла скачивается блок данных с файлами, которые также проходят валидацию.
3. Если проверка новой цепочки и новых данных прошла успешно, то узел обновляет замещает свою цепочку новой, а также дополняет данные новыми файлами, если нет – продолжает поиск.
4. Поиск обновлений для каждого типа данных (текст, изображения и т.п.) проходит отдельно и независимо друг от друга.

Валидация при этом проходит в несколько этапов и считается успешной, если выполнены следующие условия:

1. Хэш-суммы блоков корректны.
2. Хэш-сумма блока N и значение поля previousHash блока N+1 совпадают для всех блоков.
3. Данные (текст, изображения, аудиозаписи) соответствуют описанным в системе требованиям, в том числе и на повтор/плагиат.
4. Оценка текстового файла на плагиат, в частности, совершается путём попарного сравнения нового и старых текстов с помощью функций TF-IDF и косинуса векторов.

TF-IDF — статистическая мера, используемая для оценки важности слова в контексте документа. Вес некоторого слова прямо пропорционален частоте употребления этого слова в строке и обратно пропорционален частоте употребления слова во всех строках коллекции.

TF-IDF состоит из:

1. **TF** (*term frequency* — частота слова) — соотношение количества вхождений некоторого слова к общей сумме числа слов документа. По формуле (1) оценивается важность слова в пределах отдельного документа, где в числителе находится число вхождений слова в документ, а в знаменателе — общее число слов в данном документе.

$$tf(t, d) = \frac{n_t}{\sum_k n_k}, \quad (1)$$

где  $n_t$  – число вхождений слова  $t$  в документ.

2. **IDF** (*inverse document frequency* — обратная частота документа) — инверсия частоты, с которой некоторое слово встречается в документах коллекции.

$$idf(t, D) = \frac{|D|}{|\{t \in d_i\}|}, \quad (2)$$

где  $|D|$  - количество документов в коллекции,  $|\{t \in d_i\}|$  – число документов из коллекции  $D$ , в которых встречается слово  $t$ .

Как результат выполнения функции TF-IDF получаем связный список типа «ключ-значение», где каждому слову присвоен определенный коэффициент. Полученные значения используются для определения сходства текстов с помощью косинус-функции:  $\cos(a, b) = \frac{(a, b)}{|a| \times |b|}$  (3). Полученное значение и определяет сходство текстов, если оно слишком велико, новый файл не проходит валидацию.

Аналогичные проверки на плагиат планируется реализовать и для остальных типов данных.

**Список использованных источников:**

1. Jones K. S., A statistical interpretation of term specificity and its application in retrieval.
2. Saul Schleimer, Daniel S. Wilkerson, Alex Aiken Winnowing: Local Algorithms for Document Fingerprinting.

## ОБРАБОТКА ПОСТОЯННОГО ПОТОКА ДАННЫХ С АППАРАТНЫХ И ВИРТУАЛЬНЫХ ПЛАТФОРМ CISCO

Губин И.Ю.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Иванов Н.Н. – к.т.н., доцент

В настоящее время потребности абонентов к пропускной способности сети растут с каждым днём, так стараясь удовлетворить эти потребности, операторы связи ведут активное внедрение новых аппаратных и виртуальных решений, которые могли бы обеспечить обработку большого объема трафика. А при большом количестве оборудования, как аппаратного, так и виртуального, за ним надо наблюдать, получать статистические данные, обрабатывать их и строить прогнозы для лучшей балансировки и реакции на проблемы в моменты пиковой нагрузки. Все это вытекает в сбор, обработку и хранение огромного количества данных.

В ходе доклада будет рассказано, каким образом и какие данные собираются, как с аппаратных, так и виртуальных платформ. Как происходит предобработка и обработка статистических данных оборудования. Как хранятся эти данные и каким образом получается к ним доступ.

Так же будет изложено о пользе сбора и анализа данных, какие выводы можно сделать на основе полученных отчетов, как это может повлиять на коммерческую составляющую, а также как статистические данные помогают избежать аварий и вовремя среагировать при достижении определенных лимитов по ресурсам и много другое.

Будут рассмотрены варианты визуализации данных, посредством Grafana – открытой платформы для анализа и мониторинга, на которой отображаются нужные графики с использованием обработанных данных из баз данных.

Кроме вариантов визуализации, так же будут рассмотрены алгоритмы получения данных напрямую с аппаратной платформы, а также сбор данных после предобработки специализированным ПО.

Будет затронута тема обработки данных на языке программирования Python 3, а также библиотека, предназначенная для работы с большими потоками и объемами данных pandas – это программная библиотека на языке Python для обработки и анализа данных. Работа pandas с данными строится поверх библиотеки NumPy, являющейся инструментом более низкого уровня. Предоставляет специальные структуры данных и операции для манипулирования числовыми таблицами и временными рядами.

В итоге будут сделаны выводы о проделанной работе, полученном опыте при разработке и полезности в применении данного подхода для продакшн проекта. Будет рассказано о плане будущего развития проекта, какие модули будут добавлены по возможности в будущем и простоте расширения.

И в заключении будет сказано об аналитике состоянии сети в целом, о возможности прогнозирования ситуаций и нагрузок на сеть благодаря наличию большого количества данных, применении современных алгоритмов из обработки и машинного обучения. И какую пользу все это может принести как владельцу сервиса, так и пользователям.

**Список использованных источников:**

1. Маккинни У., Python и анализ данных, 2015. – 482с
2. Cisco ASR 5x00 System Administration Guide, StarOS Release 2x.x, 2018. – 682s
3. Вандер Плас Дж., Python для сложных задач. Наука о данных и машинное обучение, 2018. – 336с
4. Daniel Lee, Visualize Your Data With Grafana, 2017. – 57s

## **СПОСОБЫ МИГРАЦИИ СХЕМ ДАННЫХ С СОБЛЮДЕНИЕМ ПРИНЦИПОВ ИДЕМПОТЕНТНОСТИ И КОНВЕРГЕНТНОСТИ**

*Данильчик В.В.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Прытков В.А. – проректор по уч. работе, к.т.н., доцент*

Основным направлением изучения способов миграции схем данных из одной структуры в другую является определение оптимальных принципов модификации структур и алгоритма перестроения объектов с учетом существующих зависимостей.

Актуальность анализа способов миграции схем данных обусловлена необходимостью постоянной поддержки структуры базы данных и ее последующей оптимизации при выходе новой версии программного продукта. В настоящее время процессу миграции данных часто сопутствуют проблемы их дублирования и возникновения ошибок, связанных с изменением структуры данных.

При разработке программного продукта и, соответственно, написании кода разработчики пользуются системами контроля версий, такими как Git или Subversion. Данные системы позволяют вести совместный процесс разработки несколькими сотрудниками, хранить только разницу между версиями кода, отслеживать и, при необходимости, возвращаться к конкретной версии исходного кода. В то же время применение систем контроля версий при работе со структурами схем данных довольно ограничено.

В результате в процессе развития программного продукта отследить изменения структуры данных весьма проблематично ввиду того, что DDL, язык определения схем данных, не предусматривает документирование изменения структуры на каждом шаге [1].

Одним из методов решения данной задачи является создание в каталоге проекта пронумерованных скриптов модификации структуры базы данных при этом, не прибегая к использованию дополнительных инструментов. В таком случае все изменения будут накапливаться в проекте при условии соблюдения заранее определенных правил участниками проекта.

Одним из наиболее современных и практичных способов контроля структуры схем данных является использование дополнительного инструмента Liquibase. Основной задачей системы является контроль номера последнего выполненного скрипта изменения структуры данных и обеспечение единовременного линейного выполнения скриптов по изменению структуры данных. Однако, использование инструментов, в основе которых лежит принцип накопления change-скриптов, не позволяет решить задачу так же легко, как написать программный код приложения. Главным минусом данного подхода является необходимость хранения огромного количества неактуальных change-скриптов, что затрудняет получение представления о фактически используемой структуре схемы данных.

Для решения поставленных задач исследованы такие принципы, как конвергентность и идемпотентность. Соблюдение приведенных принципов в разработанном модуле миграции позволяет наиболее эффективно контролировать состояние схемы данных и не допускать повторного выполнения скрипта миграции данных при его успешном запуске ранее.

Наиболее оптимальным решением задачи миграции схем данных и является следование данным принципам: идемпотентности и конвергентности [2]. Отличительной особенностью такого подхода является то, что скрипт, который удовлетворяет условиям принципов, описывает состояние, к которому объект должен быть приведен, а не действия, которые требуется произвести над ним.

Идемпотентность обеспечивает отсутствие изменений объекта при выполнении скрипта повторно в случае, если скрипт был выполнен успешно при первом его запуске. Соблюдение данного принципа позволяет избежать дублирования и искажения данных. Однако, в случае невыполнения скрипта или неудачного завершения его выполнения, система, при повторном выполнении данного

скрипта, будет пытаться прийти к тому состоянию, которое было задано в качестве желаемого. Такое поведение системы обусловлено принципом конвергентности, который направлен на приведение системы в нужное состояние при каждом следующем запуске одного и того же скрипта без выполнения операций, успешно выполненных ранее. Т.е. изменения, не выполненные при одной попытке, система будет пытаться выполнить еще раз при повторном запуске.

**Список использованных источников:**

1. Клеппман. М. Высоконагруженные приложения. Программирование, масштабирование, поддержка/М. Клеппман // СПб: Питер, 2018.
2. Database Migration: What It Is and How to Do It [Электронный ресурс] - Режим доступа: <https://rollout.io/blog/database-migration> - Дата доступа: 22.03.2019

## ПРОГРАММНАЯ СИСТЕМА СИМУЛЯЦИИ РАЗРУШАЕМОСТИ НА ОСНОВЕ UNREAL ENGINE 4

*Дмитриев А.С.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Золоторевич Л.А., доцент*

В данной статье рассматривается одна из возможных реализаций системы симуляции разрушаемости в Unreal Engine 4. Для реализации отдельных компонентов используется внешний плагин Apex Destruction для Unreal Engine и отдельная программа PhysX Lab.

В основе разрабатываемой системы симуляции применяется Unreal Engine 4 как один из наиболее стабильных и производительных движков. При проектировании системы в первую очередь материалы объектов разрушения были разбиты на четыре группы:

- 1) породы – различные каменные образования и структуры;
- 2) металл – тугоплавкие материалы и металлоконструкции;
- 3) дерево – древесина и изделия деревообрабатывающей промышленности;
- 4) стекло – изделия стекольной промышленности и бьющиеся материалы.

Для моделирования разрушаемости объектов первой группы используется алгоритм Воронова, основанный на одноименной диаграмме или так называемом разбиении Дирихле. В качестве его программной реализации используется внешний плагин Apex Destruction для Unreal Engine 4, который заранее делит объекты на сцене на определенное количество элементов, называемых чанками. Все чанки являются параллелепипедами различных размеров, содержащими один из будущих обломков разрушаемого объекта. Так как разделение объекта запекается заранее, разрушаемость является довольно предсказуемой, но с другой стороны позволяет обеспечить высокую скорость моделирования в режиме реального времени (рисунок 1).

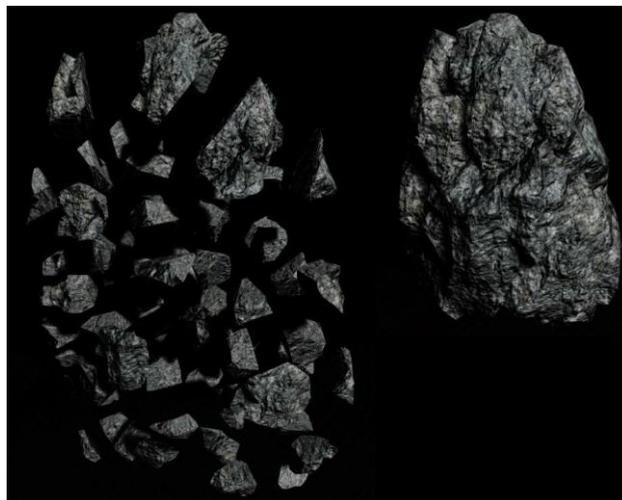


Рисунок 1 – Пример деления объектов в Apex Destruction

Вторая группа основана на трёхмерной разметке модели и её компонентов или так называемом скелетал меше. Каждая такая точка в разметке называется нодами. Положение каждого нода может

быть изменено динамически во время работы программы. Таким образом можно симулировать плавление металлических конструкций путем постепенного смещения узлов и, собственно, сжатия самой модели. Кроме того, при ударах и давлении на объект можно смещением узлов симулировать искривление самой конструкции.

Для моделирования разрушения объекта из дерева используется внешняя программа PhysX Lab, позволяющая делать разметку разрушаемых объектов схожим образом с плагином Apex Destruction, однако используя другой алгоритм. В этом случае задается количество параллельных разрезов по каждой из осей, после чего уже на каждый из кусков накладывается зашумление. Таким образом, получается набор чанков одинакового размера. Все данные по частям декомпозиции хранятся в формате, пригодном для импорта в Unreal Engine 4. При таком делении обломки получаются более вытянутыми и больше похожими на те, которые образуются при разрушении деревянных конструкций (рисунок 2).



Рисунок 2 – Пример деления объектов в PhysX Lab

Моделирование процессов разрушения объектов четвертой группы является наиболее сложным по следующей причине - стекло всегда бьется радиально от точки попадания по всей своей поверхности. Так как точка попадания может быть любой, предусмотреть такое количество заготовленных заранее вариантов невозможно или нерационально. Следовательно, необходимо разработать алгоритм, который способен в реальном времени радиально разбивать модель от точки попадания. Данная задача пока остается нерешенной.

Следующим этапом является объединение предыдущих групп в единые конструкции. Делается это при помощи компонентов в Blueprint-класссах. Каждый Blueprint-класс представляет собой набор визуальных компонентов со встроенной логикой, которая может быть вызвана извне действиями других Blueprint-класссов или событиями на сцене.

Рассмотрим это на примере разрушения железобетонной балки. Если опустить все тонкости, то в центре такой конструкции будет находиться металлический каркас, вокруг которого будут находиться бетонные пластины. При столкновении массивного объекта с объектом данного класса сначала будут разрушаться внешние пластины, а затем уже станет гнуться внутренняя конструкция, причем при сильном изгибе, так как коллизия внутренней конструкции пересечется с коллизией пластин с другой стороны, они также будут разрушены, как это было бы в реальности.

Далее комбинированным компонентам присваивается определенная прочность. Сделано это для того, чтобы симулировать разрушение объектов во времени. Функция изменения прочности имеет вид прямолинейной функции, переходящей в гиперболу. Функция отражает зависимость прочности объекта от времени. Точка перехода между функциями является границей ветхого состояния.

При моделировании разрушения сложных конструкций, например зданий, масса, давящая на объект, извлекается из физической части движка. Оттуда же извлекается и скорость объектов, если она не равна нулю. Также учитывается и ускорение свободного падения. Затем берутся крайние точки соприкосновения объектов, определяется площадь соприкосновения и сила давления на объект.

Когда сила давления немного превышает текущую прочность, объект начинает разрушаться быстрее. При превышении лимита объект теряет прочность и моментально осыпается.

Для визуального отражения старения объектов используются инстансы материалов. Это копии материалов, позволяющие менять их отдельные параметры. Таким образом, создав инстансы материалов грязи, плесени и ржавчины, мы можем наложить их поверх текстур объектов сцены и постепенно изменять их степень отрисовки. Выглядит это как постепенное появление грязи, плесени или ржавчины на соответствующих объектах.

Разрабатываемая система может быть использована не только в компьютерных играх, но и при моделировании процессов разрушаемости объектов инженерного типа. Упор в данном подходе сделан на производительность и быстродействие, а не на точность моделей. Однако система может масштабироваться в зависимости от мощности применяемой ЭВМ.

**Список использованных источников:**

1. Physics Simulation [Электронный ресурс]. — Режим доступа : <https://docs.unrealengine.com/en-us/Engine/Physics> – Дата доступа: 14.03.2019г.
2. Apex [Электронный ресурс]. — Режим доступа : <https://docs.unrealengine.com/en-us/Engine/Physics/Apex> – Дата доступа: 14.03.2019г.
3. Н.Г.Бураго – Институт проблем механики им. А.Ю. Ишлинского РАН, Москва, 119526, Россия – Вычислительная механика сплошных сред. – 2008. – Т. 4, № 4. – С.5-20

## **ПРОГРАММА ЛОЯЛЬНОСТИ ДЛЯ КЛИЕНТОВ СПОРТИВНЫХ КЛУБОВ НА БАЗЕ ТЕХНОЛОГИИ БЛОКЧЕЙН**

*Евсаев П.В., Беликова Т.О.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Фролов И.И. – к.т.н., доцент*

Исследования рынка клиентской лояльности показывают, что эффективность программ лояльности падает. Это связывают с неудобствами, связанными с хранением большого количества дисконтных карт/установкой большого количества мобильных приложений. Разработка системы, являющейся единой точкой управления программами лояльности для организаций и их клиентов видится своевременным и актуальным.

По прогнозам экспертов, объем рынка клиентской лояльности к 2020 году подойдет к отметке в 48 миллиардов долларов США. Несмотря на внушительный оборот, исследования показывают, что эффективность программ лояльности снижается. Это связывают с тем, что каждая программа лояльности требует использования карты либо мобильного приложения, что снижает привлекательность каждой следующей программы для потребителя. Создание единой системы для запуска программ лояльности снизит поставщикам товаров и услуг издержки на обслуживание собственной CRM системы и позволит иметь единый канал связи с аудиторией потребителей всех организаций-участников сети. С точки зрения пользователя, наличие подобной системы позволит иметь единую точку управления всеми бонусными баллами и акционными предложениями.

При выборе инструментов для разработки проекта, выбор был сделан в пользу стека Hyperledger. Стек Hyperledger позволяет создавать блокчейны с ограничением прав доступа для участников сети, определять собственную модель взаимодействия между участниками сети через строго определенный набор транзакций. Возможность обновления чейнкода дает возможность проводить разработку в итерационном формате, постепенно расширяя функциональности системы. Использование привилегированного блокчейна и обязательный процесс идентификации пользователя снижает риск атаки на сеть и снимает необходимость взимать плату за проведение транзакции.

В соответствии с бизнес-целями, в сети необходимо и достаточно наличие двух типов участников: организация и потребитель. Организации должны иметь право управлять программами лояльности, участники - иметь возможность вступать/выходить из программ лояльности. Оба типа участников должны иметь право осуществлять переводы токенов программ лояльности другим участникам сети.

Процесс запуска программы лояльности может выглядеть следующим образом: при создании организацией программы лояльности, у нее создается кошелек, на который поступает весь объем эмитированных в рамках программы лояльности токенов. Далее, организация высылает клиентам приглашения. Когда потребитель принимает приглашение, у него создается привязанный к этой программе лояльности кошелек, на который/с которого ему могут поступать токены. Токены являются уникальными для программы лояльности и могут быть реализованы либо переданы другому клиенту-участнику программы.

Для доступа к управлению своими ресурсами участникам сети необходимо пройти процесс идентификации. При регистрации, новому участнику сети ему выдается персональный сертификат, выпущенный доверенным для проекта центром выдачи сертификатов. Соответствующая сертификату пара ключей сохраняется локально и будет использована при следующем входе в систему.

В качестве направлений для развития создаваемой системы можно рассмотреть возможность запуска совместных программ лояльности для нескольких организаций, реализовать интеграцию с действующими CRM системами. Наличие гибкого, функционирующего по описанному алгоритму проекта поможет решить существующие проблемы на рынке клиентской лояльности, а использование распределенного реестра для хранения данных об операциях дает хороший фундамент для развития проекта с точки зрения финансового аудита.

**Список использованных источников:**

1. Mastering Bitcoin. Programming the open blockchain / Andreas M. Antonopoulos // O'Reilly Media 2017 – 416 p.
2. Mastering Blockchain: Distributed ledger technology, decentralization, and smart contracts explained, 2<sup>nd</sup> Edition / Imran Bashir // Packt Publishing 2018 – 656 p.
3. Blockchain: A Practical Guide to Developing Business, Law and Technology Solutions интернета вещей / Joseph J. Bambara // McGraw-Hill Education, 2018 – 320 p.

## **МИКРОСЕРВИС ДЛЯ УДАЛЁННОГО УПРАВЛЕНИЯ ЧПУ СТАНКОМ**

*Жук Д.А., Басов Д.А.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Селезнев И.Л. – к.т.н., доцент*

Удаленный доступ позволяет пользователю, находящемуся на одном компьютере, взаимодействовать с удаленной машиной и выполнять на ней интерактивный сеанс работы. Удаленный доступ позволяет создать впечатление, что терминал пользователя или его рабочая станция присоединены напрямую к удаленной машине, посылая каждый символ, нажатый на клавиатуре пользователя на удаленную машину и отображая каждый символ, возвращенный с удаленной машины, на экране терминала пользователя.

Удаленный доступ — функция, которая даёт пользователю возможность подключаться к компьютеру при помощи другого устройства через интернет. Пользователь работает с файлами и программами точно так же, как если бы он находился возле этого компьютера. Эта функция пригодится тем компаниям, в которых большинство сотрудников находится за пределами офиса, аутсорсинге или в командировках. Им не нужно будет скачивать все необходимые для работы данные на внешний носитель или отправлять их по почте — достаточно связаться с офисным компьютером. Удаленный доступ используется системным администратором для управления работой системы и устранения ошибок, а также руководителями, желающими проконтролировать процесс выполнения работы своими сотрудниками. Применяется он и для дистанционной формы обучения в образовательных учреждениях.

Есть два способа организации удаленного доступа

- виртуальный рабочий стол - DaaS (desktop as a service);
- установка специализированного ПО.

Первый способ - desktop as a service. Этот способ предоставляет пользователю доступ к готовому удаленному рабочему столу. В него может входить весь необходимый пользователю функционал, который при необходимости возможно расширить. Работать можно с любого устройства - рабочий стол везде будет один и тот же. Все действия выполняются не на ПК пользователя, а на сервере поставщика услуг, нагрузка на клиентскую машину минимальна, поэтому и требования к характеристикам не существенны. Для работы данные пользователя и программы размещаются в сети интернет, а не на локальном сервере. Воспользоваться ими можно из любого удобного места, где есть интернет соединение. Примерами таких программ являются VMware ESXi, Microsoft Hyper-V, Citrix XenServer.

Во втором способе для создания удаленного подключения используется специальное программное обеспечение. Важной частью является возможность постоянного доступа в интернет, сервера, рабочих машин, обладающих необходимыми параметрами. Через интернет два компьютера связываются удалённым подключением. Если рабочий ПК находится в локальной сети, то получить доступ к нему извне возможно только с помощью специальных программ. Такое программное обеспечение делает возможным подключение к другому компьютеру из любого удобного места. Программы дают возможность работать так, как если бы пользователь работал непосредственно за компьютером. Возможно видеть рабочий стол и выполнять все действия на удаленном устройстве, изменять настройки ПО, обмениваться файлами, шифровать передаваемые данные, подключать веб-камеры, удаленные проекторы и прочие сетевые устройства. Примерами такого ПО являются Team Viewer, RAdmin, Ammy Admin.

Минусом этого способа является необходимость в настройке сетевого оборудования. Помимо этого, необходимо понимать, что уменьшается безопасность сети, появляется шанс проникновения сторонних лиц в локальную сеть.

Большинство ЧПУ станков, как правило оснащено ПК с устаревшими комплектующими, малопроизводительными, способными лишь запускать программу для управления станком. Иногда возникают ситуации, в которых нужно изменить программу, и для этого необходим ещё один полноценный рабочий компьютер.

Выходом из данной ситуации служит терминальный доступ с рабочего ПК к компьютеру со всеми необходимыми программами. Для обеспечения такого доступа необходимо лишь наличие клиентской части ПО Citrix и данные пользователя для подключения к серверу Citrix.

Плюсами данного способа является возможность использования одного и того же приложения несколькими пользователями одновременно, независимо друг от друга.

**Список использованных источников:**

1. Харун Даниил, Создание бездисковых терминалов Windows RDP (Citrix ICA) на базе FreeBSD.
2. Информационные технологии и управление : материалы 49 науч. конф. аспирантов, магистрантов и студентов, Минск, 6–10 мая 2013 г. / Белорус. гос. ун-т информатики и радиоэлектроники ; редкол.: Л. Ю. Шилин [и др.]. – Минск : БГУИР, 2013. – 103 с.
3. William Manning, Citrix Certified Administrator for Citrix XenApp 5 for Windows Server 2008 Certification Exam Preparation Course, London, UK, UK ©2010

## **ВНУТРЕННЯЯ И ВНЕШНЯЯ ОЦЕНКА КОНВЕРСИИ СЛОВА В ВЕКТОР**

*Игнатъев Е.Д.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Иванов Н.Н. – к.т.н., доцент*

В последнее время количество исследований в сфере семантики неуклонно растёт. Для проведения таких исследований языка, зачастую прибегают к использованию конвертированных в вектор слов. Но при проведении конвертации могут встречаться различные ошибки которые в конечном итоге повлияют на точность и качество исследования. Поэтому необходимо проводить оценку такой конвертации.

В 2013 году компания состоялась релиз инструмента word2vec предлагая новый подход в представлении слова [1]. В 2015 году была проведена систематизация всех методов оценки и их разделение на *внутренние* и *внешние* [2].

Методы внутренней оценки — это эксперименты, в которых представления слов сравниваются с человеческими суждениями о словесных отношениях. Созданные вручную наборы слов часто используются для получения оценок человека, а затем эти оценки сравниваются с вложениями слов (этот метод сбора суждений называется абсолютной внутренней оценкой). Методы внутренней оценки-это эксперименты, в которых вложения слов сравниваются с человеческими суждениями о словесных отношениях. Созданные вручную наборы слов часто используются для получения оценок от человека, а затем эти оценки сравниваются с представлениями слов (этот метод сбора суждений называется абсолютной внутренней оценкой).

Иногда людей просят оценить качество вложений слов непосредственно, например, когда разные модели производят разные суждения о словесных отношениях, и задача человека — сказать, какая модель работает лучше (такой метод называется сравнительной внутренней оценкой). Сравнительная внутренняя оценка позволяет не оценивать абсолютное качество вложений, а находить наиболее адекватные вложения в заданном множестве.

Большинство методов абсолютной внутренней оценки предназначены для сбора данных, являющихся результатом сознательных процессов в человеческом мозге (другими словами, у оценщиков есть время подумать над ответами). Следовательно, существует вероятность того, что такие ответы предвзяты определенными субъективными факторами (например, из-за отсутствия четкого определения значения каждый человек интерпретирует отношения слов по-своему, внося вариативность в оценки). И неясно, действительно ли сознательные оценки способны сообщить структуру семантики на естественном языке.

Методы внешней оценки основаны на возможности представления слов в качестве векторов признаков алгоритмов контролируемого машинного обучения (например, модели максимальной энтропии), используемых в одной из различных задач обработки естественного языка. Производительность контролируемой модели (измеряемая в наборе данных для задачи обработки естественного языка) функционирует как мера качества представления слов. Некоторые исследования предполагают, что представление слов, показывающее хороший результат по одной задаче, покажет хороший результат по другим, а результаты встраивания слов по разным задачам коррелируют, определяя какой-то глобальный оценочный балл для дистрибутивной семантики [3].

Представление слов, вероятно, может быть использовано практически в любой задаче обработки естественного языка, и исследователи (описывая возможные варианты использования представлений слов в других последующих задачах) не упоминают задачу оценки представлений

слов. Тем не менее, согласно определению внешней оценки, приведенному выше, любая последующая задача может рассматриваться в качестве метода оценки.

Конечно, внешняя оценка имеет определенные преимущества, и в тех случаях, когда встраивание слов предполагается использовать только для решения конкретной задачи, оценка производительности контролируемой модели по этой задаче даст наиболее адекватную оценку производительности встраивания слов. Но внешняя оценка терпит неудачу, если вложения, которые вы хотите оценить, обучены служить в широком диапазоне различных задач, так как оценки производительности вложений word в различных задачах не коррелируют между собой.

**Список использованных источников:**

1. Mikolov et al., 2013a. Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space.
2. Schnabel et al., 2015. Schnabel, T., Labutov, I., Mimno, D. M., and Joachims, T. (2015). Evaluation methods for unsupervised word embeddings. // EMNLP, 2015, 298–307.
3. Bakarov A. A Survey of Word Embeddings Evaluation Methods // CoRR, vol. abs/1801.09536, 2018. Режим доступа: <http://arxiv.org/abs/1801.09536>.

## ПРИМЕНЕНИЕ АЛГОРИТМА СЕГМЕНТАЦИИ ИЗОБРАЖЕНИЙ НА ОСНОВЕ ГРАФОВ ДЛЯ АНАЛИЗА ФОТОСНИМКОВ ТОПОЛОГИЙ ИНТЕГРАЛЬНЫХ МИКРОСХЕМ

Канаш В.Н.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Воронов А.А. – к.т.н., доцент

Описан алгоритм сегментации изображений на основе графов, а также рассмотрена возможность применения алгоритма для решения задач предварительной обработки изображений интегральных микросхем для последующего анализа с применением нейронных сетей.

Изображения топологий интегральных микросхем, полученные при увеличении отличаются большим разрешением, а также наличием множества мелких элементов, затрудняющих процесс анализа и контроля. Для обработки таких изображений в настоящее время используются различные методы интеллектуального анализа данных, в том числе нейронные сети. Однако для того, чтобы эффективно использовать нейронные сети для анализа изображений большого размера, необходимо выделять на них некоторые “точки интереса” - отдельные фрагменты, которые содержат в себе больше полезной информации, чем другие части изображения. Для поиска таких фрагментов в настоящее время обычно применяется метод скользящего окна. Он заключается в том, что в контексте решаемой задачи задается конфигурация «окна обработки» – двумерной области, охватывающей конечное множество пикселей входного изображения. В процессе обработки это окно смещается по изображению, последовательно занимая все возможные положения. Фрагменты изображения, покрываемые окном на текущем шаге используются в качестве входных данных для более сложных методов анализа. Недостатком данного подхода является то, что для прохождения окна по всему изображению и параллельного применения нейронной сети для каждого положения окна требуется дополнительное время. Также требуется решить задачу выбора оптимального масштаба окна.

В данном докладе рассматривается возможность применения алгоритма сегментации изображения на основе графов (*Graph-Based Segmentation*) для выделения отдельных фрагментов на изображениях топологий интегральных микросхем в целях подготовки изображения для последующего анализа с применением сверточных нейронных сетей.

Алгоритм сегментации изображения на основе графов (*Efficient Graph-Based Image Segmentation*) был впервые описан в 2004 году [1]. Он широко используется в задачах анализа изображений из-за простоты реализации и высокой производительности. Алгоритм использует представление изображения в виде взвешенного связного неориентированного графа, где каждый пиксель изображения является отдельной вершиной. Ребра графа связывают соседние пиксели между собой. Возможно 2 варианта построения такого графа: каждый пиксель связывается с 4 ближайшими соседями, либо с 8. В случае полутоновых изображений, вес (длина) ребра, соединяющего соседние вершины, выражается формулой (1)

$$w(v_i, v_j) = |I(p_i) - I(p_j)| \quad (1)$$

где  $I(p_i)$  – интенсивность (яркость) пикселя  $p_i$

К этому графу применяется алгоритм Краскала [2], который предназначен для нахождения минимального остовного дерева графа. Данный алгоритм предполагает сортировку всех ребер графа с последующим построением минимального дерева за один проход. В случае алгоритма сегментации изображений в алгоритм Краскала вносятся изменения. Так, если внутри существующего поддерева величина максимального перепада интенсивностей больше, чем перепад интенсивностей между двумя деревьями, то эти два сегмента подлежат объединению. В противном случае эти сегменты останутся разделенными.

Для повышения эффективности алгоритма, перед обработкой к изображению применяется фильтр Гаусса. Это необходимо для того, чтобы исключить возможный шум и артефакты. Для управления величиной размытия в алгоритм вводится параметр  $\sigma$ .

В начале работы алгоритма между соседними пикселями может наблюдаться существенный перепад интенсивностей, мешающий объединению их в один сегмент. Так как соседние пиксели с большой вероятностью являются составляющими частями одного сегмента, при вычислении перепада интенсивности внутри существующих сегментов, к полученному значению добавляют величину  $T$ , зависящую от размера построенного сегмента (2)

$$T(C) = \frac{k}{|C|} \quad (2)$$

где  $|C|$  – мощность рассматриваемого сегмента а  $k$  – параметр сегментации, задаваемый вручную. Это позволяет объединять находящиеся рядом пиксели в сегменты на начальных этапах работы алгоритма.

Для тестирования модели использовались полутонные фотоснимки ИС, полученные при увеличении (см. рисунок 1). Результатом работы алгоритма с параметрами  $k=500$ ,  $\sigma=1$  является карта сегментов изображения (см. рисунок 2), в целях визуализации каждый сегмент выделен отдельным цветом.

После того как была проведена сегментация изображения, полученную информацию можно использовать для дальнейшего анализа – к примеру, выделить сегменты определенного размера (см. рисунок 3) и использовать сверточную нейронную сеть для классификации объектов на изображении.

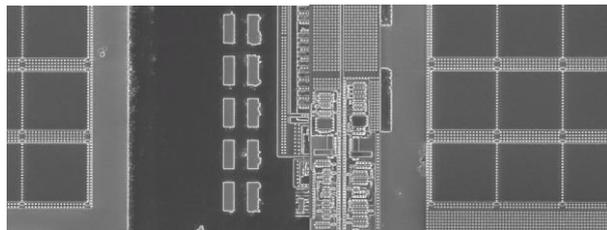


Рисунок 1 – Исходные изображения

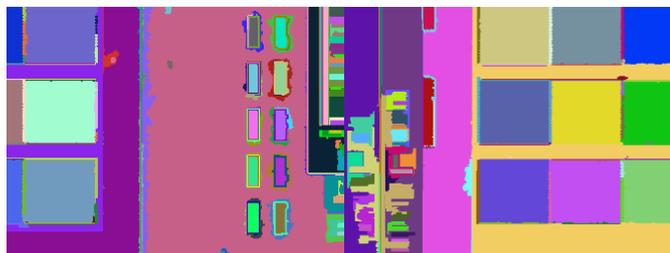


Рисунок 2 – Сегментированные изображения

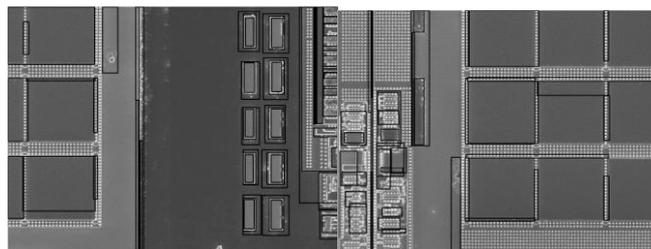


Рисунок 3 – Исходные изображения с выделенными границами части сегментов

Особенностями задачи анализа топологий микросхем является большой размер исходных изображений с множеством мелких деталей. При этом использование нейронных сетей при анализе изображений предполагает, что на вход сети будут подаваться участки изображения небольших размеров. Использование метода скользящего окна вынуждает пропускать через нейронную сеть большое количество фрагментов изображения, которые не содержат в себе полезных данных. Поэтому предварительная обработка изображения, позволяющая выделить на нем определенные точки интереса приведет к экономии времени и вычислительных мощностей. Алгоритм сегментации изображений на основе графов позволяет разделить изображение на множество сегментов относительно перепадов яркости соседних пикселей. Пользуясь полученной картой сегментов возможно построение более эффективных алгоритмов для поиска и идентификации объектов с применением нейронных сетей. К примеру, использование метода скользящего окна может быть ограничено областью изображения, содержащую сегменты определенного размера.

**Список использованных источников:**

1. Felzenszwalb P. F., Huttenlocher D. P. Efficient graph-based image segmentation // International journal of computer vision. – 2004. – Т. 59. – №. 2. – С. 167-181.
2. Joseph. B. Kruskal. On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. // Proc. AMS. 1956. Vol 7, No. 1. С. 48-50

## **ВИД РАСПРЕДЕЛЕНИЯ ВЕКТОРОВ ЗНАЧЕНИЙ НЕЙРОНОВ НА МНОГОСЛОЙНОМ ПЕРСЕПТРОНЕ**

*Караки Ю.Ш.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Иванов Н.Н. – к.ф.-м.н., доцент*

Построение эффективных алгоритмов необходимо для минимизации затрат, требуемых для решения задач. В последние годы остро стоит вопрос о параметрах нейронных сетей, широко используемых в информационных технологиях. В сообщении рассматриваются методы оптимизации параметров сверточных нейронных сетей, применяемых для обработки изображений.

В последние годы в информационных технологиях широко применяются искусственные нейронные сети. Однако вычислительно трудные задачи требуют построения нейронных сетей, содержащих миллионы и миллиарды узлов и связывающих их дуг. В 1996 г. была поставлена задача оптимизации таких параметров нейронной сети как количество скрытых слоев, количества узлов в слоях, сокращения связей между слоями и пр. [1]. Существенных достижений за прошедшие 23 года достигнуто не было.

При анализе изображений подавляющее число современных публикаций посвящается применению Сверточных Глубоких Нейронных Сетей (СГНС). Для извлечения ребер объектов на изображении на этих сети применяют свертки с ядрами размерности от 2x2 до 5x5. Обычно на следующем после свертки шаге выполняется операция подвыборки, то есть ячейки сетка, накладываемой на анализируемое изображение с размером ячейки 2x2 заменяются одним из пикселей ячейки, что уменьшает размер изображения в 4 раза. Эта операция сокращает время обработки. Обычно при обработке изображения свертка с последующей подвыборкой применяется несколько раз. Нейроны каждого скрытого слоя СГНС активизируются с целью уменьшения разброса их числовых значений. Активизация выполняется некоторой нелинейной функцией, которая часто содержит экспоненту.

Применение сверток и других операций выделяет некоторые черты объектов на изображении, но пользователю эти особенности изображений не известны. СГНС определяет эти черты в процессе обучения, на котором нейронная сеть без участия пользователя задает конкретные веса дугам, связывающим нейроны. После окончания автоматического конструирования параметров СГНС пользователь подает на вход СГНС исследуемое множество изображений и получает его разбиение на классы (например, на женские и мужские лица), что и является целью исследования.

Две основные задачи оптимизации гиперпараметров СГНС – это вычисление оптимального числа скрытых слоев нейросети и нахождение оптимального количества нейронов с каждым слоем. Эти задачи поставил еще Кевин Свингер [1], он же без математического обоснования предложил оценки для этих параметров. Сложность этих задач кроме всего прочего обусловлена двумя факторами: 1) отсутствия понимания того, какие черты изображений принимает во внимание СГНС при классификации; 2) насколько репрезентативно обучающее множество. Главный подход к задаче оптимизации количества скрытых слоев сети состоит в апостериорном оценивании вероятности

правильной классификации исследуемого множества изображений. Его идея заключается в том, что для используемого количества скрытых слоев вычисляется частота правильно распознанных объектов и затем эта частота интерполируется на последующие один-два слоя. Эксперт оценивает, следует ли добавить скрытые слои. При этом для упрощения оценок без каких-либо обоснований предполагается, что векторы текущих значений нейронов на слоях имеют распределение Гаусса [2].

Для решения задачи об оптимальном количестве нейронов в слоях используется генетический алгоритм [3].

Интенсивное исследование гиперпараметров нейронных сетей началось с 2014 г. и до настоящего времени не принесло существенных результатов.

**Список использованных источников:**

1. Swingler, K. Applying Neural Networks. A practical Guide / K. Swingler. – Burlington, Massachusetts: Morgan Kaufmann, 1996. 303 p.
2. Hinz T. Speeding up the Hyperparameter Optimization of Deep Convolutional Neural Networks / T. Hinz, N.as Navarro-Guerrero, S. Magg, S. Wermteraman // International Journal of Computational Intelligence and Applications – 2018. – Vol. 17, No. 2. – P. 1-15.
3. S. Safarik, J. Genetic algorithm for automatic tuning of neural network hyperparameters / J. Safarik, J. Jalowiczor, E. Gresak, J. Rozhon // Proc SPIE-Intl Soc Optical Eng, Orlando, Florida. – May 2018. – Vol. 10643. – P. 1-7.

## РЕАЛИЗАЦИЯ MVC КОНЦЕПЦИИ НА ПРИМЕРЕ ВЕБ-ПРИЛОЖЕНИЯ С ТЕСТИРОВАНИЕМ

*Карнющенко В.В.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Золоторевич Л.А., – к.т.н., доцент*

Концепция MVC легла в создание фреймворка, с помощью которого было реализовано данное веб-приложение. Реализация в форме веб-приложения позволит обучаемому работать с материалом дистанционно, а обучающему не тратить времени на проверку тестов. Данное программное средство разработано с целью сделать процесс обучения интереснее и продуктивнее.

Для реализации фреймворка, и как следствие, веб-приложения была выбрана одна из наиболее популярных в среде веб-разработчиков концепция разработки программного обеспечения и это – MVC концепция. Суть данной идеи состоит в разделении данных программного средства: работы с базой данных(**Models**), пользовательского интерфейса(**Views**) и логики приложения(**Controllers**)[1]. Таким образом, модификация каждого компонента может осуществляться независимо. Принцип концепции MVC представлен на рисунке 1.

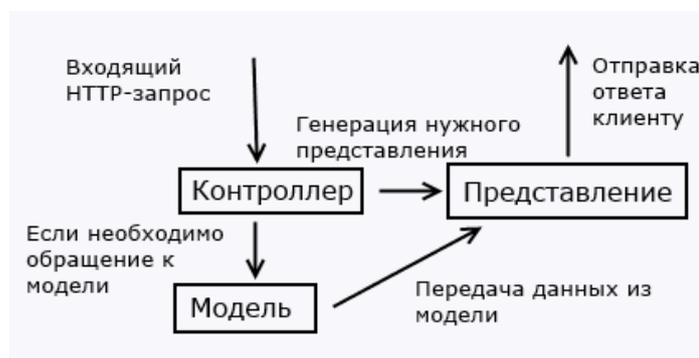


Рисунок 1 – Основная схема концепции MVC

Одним из основных плюсов данной концепции является то, что у разработчика нет привязки к какому-либо конкретному языку программирования. Вы можете писать основную логику приложения на любом языке приложения. Решения веб-приложений реализованных с помощью фреймворков зачастую работают значительно быстрее и выдерживают большую нагрузку. С этим связан большой процент в сети интернет-магазинов написанных с помощью фреймворков. Однако для успешной разработки на фреймворке требуется не только наличие хорошего опыта программирования, но и понимание предстоящих бизнес-процессов. К примеру, фреймворки в отличие от коробочных решений имеет только необходимый функционал для последующей разработки программистом.

Для написания фреймворка использовался один из наиболее популярных, а также хорошо себя зарекомендовавших в разработках веб-приложений скриптовый язык программирования PHP. Используя PHP и ООП(объектно-ориентированное программирование) был написан ряд классов(контроллеров), отвечающих за логику приложения, таким образом исполнив всю необходимую нагрузку с точки зрения серверных операций. PHP показал себя в процессе разработки весьма достойно, это касается относительной простоты написания кода и масштабируемости проекта.

Все необходимые операции с базой данных, а именно CRUD(create-read-update-delete) велись с помощью СУБД(система управления базой данных) MySQL. В качестве основного движка для построения таблиц был сделан выбор в пользу InnoDB, т.к. в отличии от своих конкурентов обладает полезными свойствами необходимым при реализации и использовании эффективных запросов: транзакции и внешние ключи. При создании простых запросов использовался Propel(объектно-реляционное отображение), данная технология также неплохо себя зарекомендовала в тех случаях, когда была необходимость простой выборки с таблицы базы данных, в случаях, когда была необходима более сложная фильтрация данных использовался чистый SQL.

За процесс построения страниц (представлений) веб-приложения, а также их оптимизацию отвечал шаблонизатор TWIG. Это компилирующий обработчик с открытым исходным кодом помогал избавиться от большого количества повторяющейся информации на страницах веб-приложения.

**Список использованных источников:**

1. Пьюривал С. Основы разработки веб-приложений. СПб: Питер, 2015 – 272 с.

## РАСПОЗНАВАНИЕ ЭМОЦИЙ ЧЕЛОВЕКА ПО ФОТО

*Кессо П.И.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Искра Н.А. – старший преподаватель*

Информация об эмоциях может использоваться в различных сферах деятельности современного общества. Врачи могут использовать эту информацию для отслеживания эмоционального здоровья пациентов. Исходя из эмоционального состояния возможно оценивать производительность сотрудников. Эмоции могут использоваться в маркетинговых целях для улучшения качества индивидуальной рекламы. Хорошие результаты в задачах анализа изображений, достигнутые при использовании сверточных нейронных сетей, говорят о возможности использования алгоритмов глубокого машинного обучения для решения задачи распознавания эмоций человека по фото.

В 1970 году психолог Пол Экман выделил 6 базовых эмоций присутствующих, по его мнению, во всем человеческом культурном наследии [1]. Он выделил шесть основных человеческих эмоций: счастье, печаль, отвращение, страх, удивление, злость. Человеческие эмоции выражаются во многих аспектах: выражение лица, жесты, речь (как устная, так и письменная).

На данный момент можно выделить два подхода к распознаванию эмоций человека, основанных на использовании нейронных сетей.

Классический подход к задаче классификации эмоций основан на классификации ключевых точек человеческого лица. Расположение ключевых точек фиксирует жесткие и нежесткие деформации лица из-за движений головы и мимики. Для получения ключевых точек человеческого лица могут использоваться такие алгоритмы как PDM, CML, AAM, DPM или CNN [2]. Следующий этап распознавания при классическом подходе, это классификация ключевых точек. Для классификации ключевых точек хорошо подходит метод опорных векторов.

На данный момент проблема поиска ключевых точек достаточно хорошо изучена и имеется большое количество алгоритмов, позволяющих получить ключевые точки с точностью, достаточной для дальнейшей классификации по этим точкам эмоций человека. Но, для использования классического подхода необходимо чтобы положение лица на изображении было выровнено.

Альтернативой использованию классического подхода является подход, основанный на сверточных нейронных сетях.

Сверточная нейронная сеть – это архитектура искусственных нейронных сетей, нацеленная на эффективное распознавание образов. Сверточные сети являются хорошим базовым решением для классификации различных визуальных данных.

Подход основанный на использовании машинного обучения может быть разделен на две основные категории: статические и динамические методы. В статических методах, в качестве информации для распознавания, используется единичное изображение человеческого лица. Тогда

как в динамических методах используется последовательность изображений. Кроме информации о каждом изображении в последовательности, динамические методы учитывают временную связь между смежными изображениями [3].

Основной проблемой использования сверточных нейронных сетей при решении задачи распознавания человеческих эмоций по фото является отсутствия достаточной обучающей выборки, из-за чего возникает проблема переобучения сети. Также проблемой является наличие избыточной информации на изображениях, используемых для обучения сети таких как позиция головы и неравномерное освещение [4].

Среди доступных в настоящее время наборов данных для анализа выражения человеческого лица можно выделить SFEW/AFEW [5], который в отличие от большинства доступных наборов данных, не был создан в строго контролируемых лабораторных условиях.

Для набора данных SFEW/AFEW реализован SPI (Strictly Person Independent) протокол, позволяющий проводить оценку алгоритмов распознавания человеческих эмоций. Метриками для измерения производительности систем распознавания эмоций являются точность, полнота и специфичность [6].

**Список использованных источников:**

1. Paul Ekman - Basic Emotions (1970)
2. Yue Wu and Qiang Ji - Facial Landmark Detection: a Literature Survey (2016)
3. Xiangyun Zhao - Peak-Piloted Deep Network for Facial Expression Recognition (2017)
4. Shan Li and Weihong Deng - Deep Facial Expression Recognition: A Survey (2018)
5. Abhinav Dhall - Acted Facial Expressions In The Wild Database (2011)
6. Abhinav Dhall - Static Facial Expression Analysis in Tough Conditions: Data, Evaluation Protocol and Benchmark(2011)

## **ПРОГРАММНОЕ СРЕДСТВО КЛАССИФИКАЦИИ ОБРАЗОВ С ИСПОЛЬЗОВАНИЕМ СВЕРТОЧНОЙ НЕЙРОННОЙ СЕТИ ГЛУБОКОГО ОБУЧЕНИЯ**

*Кирвель А.И.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Иванов Н.Н. – кандидат физ-мат наук, доцент*

В сообщении дано краткое описание приложения, разделяющего множество цветных фотографий лиц людей на 2 класса: мужские и женские лица. Предполагается, что фотографии имеют обычное качество студийной фотосъемки с размером 9x9 см., фото в фас или 3/4. Для классификации построена сеть глубокого обучения элементами инструмента Deep Learning Toolbox пакета прикладных программ MATLAB.

В основу архитектуры построенной нейронной сети положена предобученная сверточная нейронная сеть глубокого обучения AlexNet. Сверточная нейронная сеть AlexNet была разработана Алексом Крижевским и опубликована совместно с Ильей Суцкевером и Джеффри Хинтоном [1]. В 2012 году данная сверточная нейронная сеть одержала победу в конкурсе по распознаванию изображений ImageNet Large Scale Visual Recognition Challenge с отрывом в 10,8% ошибок от ближайшей сети-конкурента. В оригинале сеть имела выходной слой из 1000 нейронов, подразумевала разделение образов на 1000 классов. Для поставленной задачи количество выходных нейронов сети было уменьшено до 2 в соответствии с поставленной задачей. Построенная сеть была обучена на основе базы образов Caltech 101, содержащей изображения лиц людей размером 300x300 пикселей [2]. Предварительно размеры фотографий были приведены к 227x227 пикселей согласно числу входных нейронов. Исходный набор изображений был разделен на обучающую (80%) и тестовую (20%) выборки. Для процесса обучения с алгоритмом Левенберга-Марквардта оказалось достаточно 10 эпох.

Обученная сверточная нейронная сеть извлекает черты лиц, применяя стандартные ядра свертки, такие как фильтры Собеля, Роджерса, Превитта. Для уменьшения размерности данных и сокращения времени работы приложения применялись операции подвыборки. Далее приложение обрабатывает полученные черты по методу опорных векторов (SVM), принадлежащему семейству линейных классификаторов, который также может рассматриваться как специальный случай регуляризации по Тихонову, получая окончательный результат. Анализ работы алгоритма выполнялся на случайной выборке фотографий из всемирной паутины. Эти фотографии лиц не использовались при обучении нейронной сети.

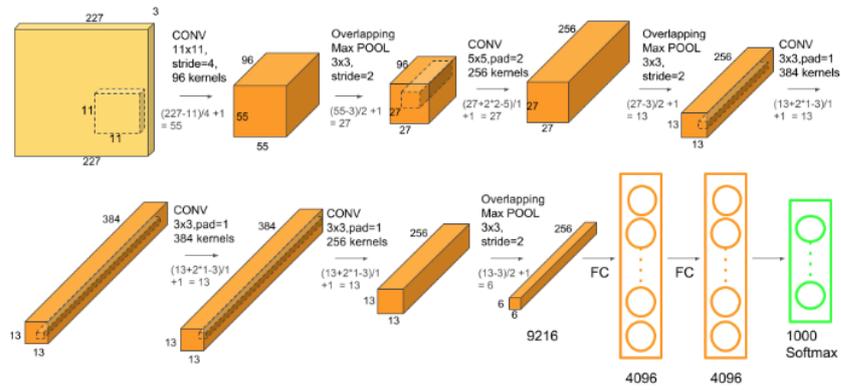


Рисунок 1 – Структура сверточной нейронной сети AlexNet

Нейронная сеть имеет 8 скрытых слоев нейронной обработки данных, а именно: 5 сверточных слоев, за тремя из которых следуют слои подвыборки (иначе слои пулинга либо слои субдискретизации), и 3 полносвязных слоя, которые завершаются реализацией алгоритма опорных векторов. Сеть использует функцию активации ReLU (rectified linear unit). Все 40 тестовых фотографий были распознаны правильно. Таким образом, получено программное средство для классификации лиц мужчин и женщин.

**Список использованных источников:**

1. Krizhevsky, A. ImageNet classification with deep convolutional neural networks / A. Krizhevsky, I. Sutskever, G. E. Hinton // Communications of the ACM. – 2017. – Vol. 60, № 6. – P. 84–90.
2. Fei-Fei, L. Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories / L. Fei-Fei, R. Fergus, P. Perona // Computer Vision and Image Understanding. – 2007. – Vol. 106, № 1. – P. 59–70.

## МЕТОДИКА РАСПОЗНАВАНИЯ ПОКАЗАНИЙ СТРЕЛОЧНЫХ ПРИБОРОВ НА ОСНОВЕ АНАЛИЗА ИЗОБРАЖЕНИЙ

*Коваленко Г.О.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Одинец Д.Н. – к.т.н., доцент*

Системы автоматизации, внедряемые в технологические процессы на предприятии, зачастую подразумевают замену или дублирование операторов, которые следят за состоянием технологического процесса (стрелочным приборам). Предлагается вариант специализированного программного средства, которое позволяет без участия оператора выполнять распознавание показаний стрелочного прибора.

В последние десятилетия двадцатого века массовое производство стрелочных измерительных приборов достигло значительных объемов. К трудностям, возникающим при работе с аналоговыми стрелочными приборами, можно отнести отсутствие электрического выхода для индикации результатов измерений. Одним из вариантов решения проблемы распознавания показаний прибора при отсутствии данного выхода является применение методов на основе анализа изображения.

Использование систем компьютерного зрения помогает при решении множество практических задач. Одной из классических задач в компьютерном зрении, обработке изображений и машинном зрении является распознавание объектов на изображении. В данный момент существует множество методов для решения этой задачи. На выбор конкретного метода влияет тип объекта, который необходимо обнаружить на изображении, а также определенные условия, к которым относятся освещение, фон и положение объекта относительно камеры [1].

Методика определения показания стрелочного прибора, разработанная в рамках диссертационного исследования, основана на определении угла отклонения стрелочного указателя с помощью следующего алгоритма:

- 1) получение изображения;
- 2) приведение изображения к градациям серого;
- 3) применение фильтра Гаусса;
- 4) применение медианного фильтра;
- 5) применение преобразования Хафа для нахождения окружности;

- 6) нахождение стрелочного указателя;
- 7) определение угла отклонения стрелки прибора.

Преобразование Хафа – это алгоритм, который позволяет извлекать элементы из изображения. В нашем случае он предназначен для поиска объектов, принадлежащих определённому классу фигур (окружность). Для этого используется процедура голосования, которая применяется к пространству параметров. Пространство параметров используется для получения объектов, принадлежащих к определенному классу фигур, по локальному максимуму в так называемом накопительном пространстве. Построение данного пространства выполняется при вычислении трансформации Хафа [2]. Изначально метод был предназначен для поиска прямых линий на изображении, однако в настоящее время появились различные модификации преобразования Хафа, которые позволяют выделить на изображении эллипсы и другие известные виды аналитических кривых [3].

Стрелочный указатель в общем виде представляет из себя прямую линию. Для нахождения прямых линий на изображении используется метод Line Segment Dector [4]. Из найденного массива линий производится исключение тех линий, которые не входят в окружность, полученную на предыдущем шаге. Далее осуществляются анализ полученного преобразованного массива линий для нахождения стрелочного указателя.

Определение угла отклонения стрелочного указателя сводится к нахождению угла пересечения стрелки с горизонталью и последующей корректировке значения в зависимости от квадранта, в котором находится стрелка.

**Список использованных источников:**

1. Forsyth D.A., Ponce J. Computer Vision: A Modern Approach / Forsyth D.A., Ponce J. - Pearson Education, Prentice Hall. – 2012. - 793 p.
2. Hough transform [Электронный ресурс]. Режим доступа: - [https://en.wikipedia.org/wiki/Hough\\_transform](https://en.wikipedia.org/wiki/Hough_transform)
3. Преобразование Хафа, его обобщения и модификации [Электронный ресурс]. Режим доступа: – [http://wiki.technicalvision.ru/index.php/Преобразование\\_Хафа,\\_его\\_обобщения\\_и\\_модификации](http://wiki.technicalvision.ru/index.php/Преобразование_Хафа,_его_обобщения_и_модификации)
4. OpenCV [Электронный ресурс]. Режим доступа: - <https://opencv.org>

## **РОБОТ С РАСПОЗНАВАНИЕМ ЛИЦА ПОЛЬЗОВАТЕЛЯ**

*Ковбаса Г. А., Юревич В. С.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Стракович А. И. – ассистент*

Задачи, связанные с компьютерным зрением, являются весьма сложными для решения на основе архитектуры современных компьютеров. В данной работе рассмотрен механизм создания робота с возможностью распознавания лица пользователя, описание использованных алгоритмов, а также способы применения такого робота.

Множество задач, связанных с компьютерным зрением и распознаванием образов, являются нетривиальными, так как разработка алгоритмов решений зависит от конкретной задачи. Однако живые существа, имеющие зрение, например, тараканы, отлично справляются с данными задачами. Увидев человека, таракану не надо много времени, чтобы осознать угрозу и рассчитать положение ближайшей щели в трёхмерном пространстве. Данный набор действий происходит на уровне инстинктов. При этом для решения задач обнаружения человека и расчета траектории движения при помощи компьютерного зрения используются достаточно сложные алгоритмы. Задачи, которые встают перед разработчиками в данной области, возможно решать различными способами, в зависимости от вычислительных мощностей платформы и временных ресурсов.

В качестве примера реализации задач обнаружения человека на встроенных системах можно упомянуть разработки таких компаний, как Google и Microsoft, которые предложили оснастить компьютер Raspberry Pi системой распознавания лиц и речи [1,2]. Raspberry Pi с системой распознавания образов можно будет встраивать в кормушку для домашних животных, поливочную систему и систему электронных замков. Ранее разработчики в Microsoft ужали алгоритм нейронной сети до нескольких бит, причем точность распознавания изображений сжатой версией алгоритма не уступает точности полной [2]. На следующем шаге предполагается загрузить нейронную сеть на устройство и сократить ее энергопотребление в 10 тысяч раз [2]. А компании Raspberry Pi и Google предложили «конструктор» домашнего голосового помощника [3].

В данном проекте механизм распознавания лица человека реализован на базе робота на 4 колесах. Кроме непосредственно распознавания лица, он способен выполнять следующие действия:

1. Подъезжать к пользователю, при обнаружении лица пользователя в поле зрения робота;
2. Избегать препятствий при движении;
3. Реагировать на звуки и движения пользователя;
4. Запоминать определенного пользователя и в дальнейшем распознавать его как хозяина;
5. Издавать характерные звуки.

Разрабатываемый робот функционирует на базе платформы Raspberry Pi 3 Model B+, которая отвечает за алгоритмы распознавания лиц и расчет траектории движения. Непосредственно за движение робота отвечает платформа Arduino Uno. Был разработан набор команд для взаимодействия данных платформ при помощи эмуляции COM-порта через USB. Для этого робот оснащен датчиками и камерой в передней части корпуса. Модуль камеры Raspberry Pi можно использовать для съемки видео высокой четкости, а также фотографий неподвижных объектов.

Программное обеспечение робота, обеспечивающее распознавание человеческого лица, основано на OpenCV и язык Python. OpenCV (Open Source Computer Vision Library) [4] представляет собой библиотеку программного обеспечения для компьютерного зрения и машинного обучения с открытым исходным кодом. OpenCV была создана для обеспечения общей инфраструктуры приложений для компьютерного зрения и ускорения использования восприятия машины в коммерческих продуктах. Данная библиотека содержит более 2500 оптимизированных алгоритмов, которые включают в себя полный набор классических и современных алгоритмов машинного обучения и компьютерного зрения. Эти алгоритмы могут использоваться для идентификации объектов, обнаружения и распознавания лиц, отслеживания движения камеры, классификации действий человека в видео, извлечения 3D-моделей объектов и отслеживания движущихся объектов.

Процесс распознавания лица можно разделить на следующие этапы:

- Обнаружение лица на изображении;
- Создание набора данных с изображениями;
- Обучение распознавателя на созданном наборе;
- Сравнение обнаруженного лица с уже известными (непосредственно распознавание).

Для обнаружения лиц на изображении используется метод каскадов Хаара [5], также известный как алгоритм Виолы-Джонса. Этот подход основан на механизме обучения: каскадная функция Хаара используется для обнаружения объектов на других изображениях. Для того, чтобы определить, является ли объект человеческим лицом, необходимо выделить примитивы, такие как губы, нос, глаза, лоб. Если данные шаблоны соответствуют конкретным областям на изображении, делается вывод, что лицо на изображении есть. Чтобы определять на изображении лица разного размера, используется метод скользящего окна. Вышеназванные примитивы вычисляются внутри окна, которое перемещается по изображению. После этого размер окна увеличивается, и вычисления производятся еще раз. Повторяется это до тех пор, пока размер окна не совпадёт с размером исходного изображения. Для достаточно часто используемого алгоритма детектирования лиц, OpenCV имеет уже обученный классификатор, но также предоставляется возможность обучить свой собственный классификатор для обнаружения любого объекта.

Для обучения распознавателя необходимо создать набор данных, содержащий изображения лиц, подлежащих идентификации. Для распознавания лиц использован метод локальных бинарных шаблонов (LBP) [6]. Суть его заключается в том, что изображение разбивается на части и в каждой части каждый пиксель сравнивается с соседними пикселями. Если значение центрального пикселя больше соседнего, то на соответствующую позицию записывается 0, в противном случае 1. Функции, предоставляемые OpenCV, позволяют задавать количество соседних пикселей для сравнения, а также их удаленность от центрального пикселя. Чем больше пикселей в окрестности точки будет обработано, тем точнее будет распознавание. Для каждого пикселя получается определенное число. Далее на основе этих чисел для всех частей, на которые разбивалась фотография, считается гистограмма. Все гистограммы со всех частей объединяются в один вектор, характеризующий изображение в целом. Чтобы узнать, как сильно похожи два лица, для каждого из них необходимо вычислить такой вектор и сравнить их. Обучение распознавателя также проводится средствами OpenCV [7].

Заключительным этапом является распознавание лица. Каждое лицо, обнаруженное камерой робота, будет проанализировано распознавателем. Для изображения человека, уже известного распознавателю, будет возвращен идентификатор этого человека, а также индекс, показывающий степень уверенности распознавателя. Важно учитывать, что индекс доверия будет возвращать 0, если он будет считаться идеальным совпадением.

Таким образом, комбинация методов Хаара и LBP является оптимальным вариантом, так как данный подход не сложен в реализации и эффективен для встроенных и мобильных систем при правильном подходе.

В качестве преимуществ перед нейронными сетями следует отметить более низкое энергопотребление, меньшее время, необходимое для обучения, а также большее количество источников информации.

Данное устройство имеет большой потенциал к применению в реальной жизни. Робота можно использовать для удаленного мониторинга состояния помещения, при длительном отъезде из дома, в качестве интерактивной охранной системы или просто как развлечение для детей или всей семьи. Централизованно управляющую несколькими роботами систему на основе разработанной нами модели можно использовать для перевозки грузов в условиях производства или для прокладки кабеля в шахте, своими размерами не позволяющей работать там человеку.

Важно отметить, что подключая такого робота на постоянной основе к сети интернет необходимо тщательно продумать все аспекты в плане сетевой безопасности и шифрования данных, чтобы не допустить получения доступа к инструментам робота и информации на нем посторонним лицам.

Расширять и совершенствовать функционал устройства можно не только при помощи дополнительных датчиков и устройств, таких как датчик давления, влажности воздуха, температуры, дополнительный модуль камеры, но также совершенствуя алгоритм распознавания лиц или добавляя новые функции для взаимодействия с пользователем. Например, приветствие пользователя по имени, заданном при первом появлении в поле зрения робота. Также возможно усовершенствовать и саму конструкцию робота: использовать более надежную и мощную платформу с двигателями, добавить защитный корпус, чтобы избежать повреждения аппаратного обеспечения, добавить солнечные панели для подзарядки во время работы и многое другое. Дальнейшая модернизация устройства ограничивается лишь числом портов ввода-вывода как на контроллере Arduino, так и на одноплатном компьютере Raspberry.

**Список использованных источников:**

1. Raspberry.org. Google is bringing AI to your Raspberry Pi [Электронный ресурс]. Режим доступа: <https://www.raspberrypi.org/blog/google-tools-raspberry-pi/>. – Дата доступа: 4.03.2019.
2. Microsoft blog. AI's big leap to tiny devices opens world of possibilities [Электронный ресурс]. Режим доступа: <https://blogs.microsoft.com/ai/ais-big-leap-tiny-devices-opens-world-possibilities/>. – Дата доступа: 10.03.2019.
3. Информационный портал dev.by. Raspberry Pi и Google предложили «конструктор» домашнего голосового помощника за \$50 (видео) [Электронный ресурс]. Режим доступа: <https://dev.by/news/magpi-diy-google-home>. – Дата доступа: 15.03.2019.
4. Информационный ресурс opencv.org. About OpenCV [Электронный ресурс]. Режим доступа: <https://opencv.org/about.html>. – Дата доступа: 8.02.2019.
5. Информационный ресурс opencv.org. Face Detection using Haar Cascades [Электронный ресурс]. Режим доступа: [https://docs.opencv.org/3.3.0/d7/d8b/tutorial\\_py\\_face\\_detection.html](https://docs.opencv.org/3.3.0/d7/d8b/tutorial_py_face_detection.html). – Дата доступа: 10.02.2019.
6. Информационный ресурс opencv.org. Local Binary Patterns Histograms [Электронный ресурс]. Режим доступа: [https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec\\_tutorial.html#local-binary-patterns-histograms](https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html#local-binary-patterns-histograms). – Дата доступа: 10.02.2019.
7. Информационный портал habr. Распознаем лица на фото с помощью Python и OpenCV [Электронный ресурс]. Режим доступа: <https://habr.com/ru/post/301096/>. – Дата доступа: 8.02.2019

## ИДЕНТИФИКАЦИЯ ПОЛЬЗОВАТЕЛЯ ПО ЕГО РУКОПИСНОЙ ПОДПИСИ

*Коляго Н.Р.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Искра Н.А. – старший преподаватель*

В данной статье изучается возможность использовать устройство с сенсорным экраном в качестве приспособления для сбора образцов подписей, различные способы представления образца (офлайнный и онлайнный) и алгоритмы верификации подписей на основе динамической трансформации временной шкалы, выделения локальных экстремумов, разложения функций в ряды, скрытой марковской модели, нейронных сетей.

Распознавание людей - очень важный в современном мире вид деятельности. Во множестве приложений гарантия идентичности личности и ее авторизация являются необходимым условием. Одним из множества способов распознать человека является биометрия.

Биометрия - наука об идентификации или верификации личности по физиологическим или поведенческим отличительным характеристикам.

Биометрических параметров существует множество. Они могут быть физическими и поведенческими. Физические параметры, в целом, довольно статичны и редко подвергаются сильным изменениям. Даже с возрастом они практически не изменяются. Единственный случай, когда эти

параметры могут сильно измениться - какое-либо воздействие на организм, например, травма, хирургическая операция или болезнь. Поведенческие же параметры изменяются в течение определенного периода времени и могут быть намеренно изменены.

Остановимся поподробнее на поведенческих параметрах, а в частности, на рукописной подписи. Из биометрических параметров рукописная подпись применяется дольше и чаще всех. Подпись интересна тем, что подписывающийся может сам определить «отличительные признаки» и «уникальность» своей подписи. Подделка здесь может быть с нулевым усилием, после тренировки, из-за плеча и профессиональная подделка.

Технологии автоматической верификации подписей делятся на следующие виды по способам получения образцов:

1) офлайновые («статистические») - подпись получается в виде изображения. Единственная доступная информация - координаты точек подписи;

2) онлайнные - содержат динамические характеристики, такие как координата пера в конкретный момент времени, угол наклона пера, сила нажатия.

Недостатком офлайновых технологий является то, что их проще подделать, так как достаточно подделать только форму подписи. Преимущество онлайнных технологий - кроме формы подписи злоумышленнику необходимо повторить еще и динамику начертания подписи, что на порядок сложнее. Недостаток у них по сравнению с офлайновыми, пожалуй, всего один - необходимо более специализированное оборудование для получения образца.

Методы верификации подписи весьма разнообразны. Среди них были выбраны следующие:

1) алгоритм динамической трансформации временной шкалы. Алгоритм не отличается точностью. Он не самый надежный, но наиболее прост в реализации;

2) алгоритм выделения локальных экстремумов. Более точен, чем динамическая трансформация временной шкалы, так как здесь для сравнения используется не приведенная форма кривой подписи, а ее «примечательные точки» - локальные экстремумы траектории;

3) алгоритм на основе разложения функций в ряды – временные функции координат точек подписи раскладываются в ряды, на основании которых происходит сравнение;

4) алгоритм на основе скрытой марковской модели. Используется для верификации офлайновых образцов;

5) нейронные сети. Могут быть использованы как для распознавания офлайновых, так и онлайнных образцов. В первом случае, процесс сводится к распознаванию бинарного изображения, во втором - к распознаванию временной последовательности точек.

**Список использованных источников:**

1. Zhang Z. A Survey of On-line Signature Verification / Zhang Z., Wang K., Wang Y. // Biometric Recognition. 6th Chinese Conference, CCBR 2011, Beijing, China, December 3-4, 2011. Proceedings / ed.: Sun Z., Lai J., Chen X., Tan T. – Berlin, Heidelberg : Springer, 2011 – P. 141–149

2. L.G. Hafemann, Offline handwritten signature verification - literature review [Electronic resource] / L.G. Hafemann, R. Sabourin, L.S. Oliveira, // Cornell University. - Mode of access: <https://arxiv.org/pdf/1507.07909v4.pdf>. - Date of access: 23.03.2019.

## **КОРРЕКТИРОВКА ВЕКТОРА ДВИЖЕНИЯ В РЕАЛЬНОМ ВРЕМЕНИ**

*Коников А.Д.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Самаль Д.И. – к.т.н., доцент*

Автономные мобильные роботы применяются во многих областях жизнедеятельности человека. Для уменьшения габаритов и стоимости системы может применяться множество идентичных компактных самоорганизующихся роботов. В данной статье рассмотрена система перемещения компактного робота при помощи одного пьезоэлемента, а также проведён анализ проблем, возникших в ходе разработки, и приведены пути их решения.

Сфера автономных мобильных роботов набирает высокую популярность и получает всё большее распространение. Наибольшее развитие она получит в течение ближайших 5-10 лет [1]. Одним из направлений в данной области являются исследования коллективного интеллекта, иными словами роевого поведения. Таким образом, множество независимых компактных и не очень роботов могут объединяться для решения одной большой и сложной задачи, непосильной для решения одним роботом, например, перемещение крупного и тяжёлого объекта. уменьшение времени выполнения задачи, путём распределения частей задачи между группой роботов.

Наиболее подходящими для реализации компактных роботов являются электрические, в том числе и вибрационные двигатели, т.к. они обладают очень высоким отношением мощность/вес,

имеют компактные размеры, обладают низким тепловыделением, уровнем шума, а также высоким КПД по сравнению с двигателями внутреннего сгорания, что позволяет существенно увеличить время автономной работы робота.

В данном проекте применяется вибрационный двигатель на основе пьезоэлемента разработанный и запатентованный профессором Владимиром Тимофеевичем Минченей [2].

Платформа представляет собой равнобедренный треугольник с загнутыми краями, изготовленный из нержавеющей стали, к которому приклеен пьезоэлемент (рисунки 1а,б). При подаче переменного напряжения с определённой частотой, определяемой эмпирически, на пьезоэлемент, начинаются колебания. Т.к. пьезоэлемент жёстко соединён с платформой при помощи цианокрилата, то все колебания пьезоэлемента передаются на стальное основание.

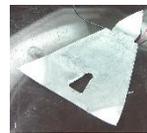


Рисунок 1а – Платформа с пьезоэлементом    Рисунок 1б – Платформа с пьезоэлементом

Волны механических колебаний распространяются по металлу и вызывают пружинящие колебания (изгибания) очень малой амплитуды ножек платформы. При попадании в частоту резонанса, из-за различных расстояний от пьезоэлемента до концов ножек платформы в определённый момент вектор суперпозиции сил оказывается ненулевым и направленным в определённую сторону, что вызывает движение платформы. Сужающиеся углы треугольника вызывают существенный рост амплитуды волн колебаний на концах ножек основания, что обеспечивает ещё большее смещение платформы. Отверстие в платформе изначально расположено в точке наибольших механических напряжений и вызывает переотражение и усиление колебаний.

При движении платформы, из-за шероховатости поверхности, по которой она движется, возникает отклонение движения робота от намеченного курса. Чтобы устранить отклонения в движении требуется вводить корректировки, рассчитываемые на основании системы обратной связи, построенной на базе оптического сенсора компьютерной мыши.

В идеале, пьезоэлемент должен быть приклеен ровно посередине платформы и над отверстием, однако из-за необходимости подключения оптического сенсора компьютерной мыши пьезоэлемент пришлось сместить, а отверстие расширить, чтобы оптический сенсор мог реагировать на поверхность, по которой движется робот. Причиной таких изменений стало то, что центр тяжести использованной мыши находится приблизительно в области оптического сенсора и требовалось, чтобы центр тяжести находился над платформой, а не за её пределами для обеспечения устойчивости платформы и равномерной нагрузки на её ножки. Равномерность распределения нагрузки является одним из необходимых условий для приведения данной системы в легко контролируемое движение.

Основные достоинства данной платформы - простота и дешевизна конструкции, позволяющая создать множество экземпляров для изучения роевого поведения роботов.

Главным недостатком является сложность при изготовлении стальной платформы, т.к. требуется точно соблюсти пропорции, и при подборке частот необходимых для движения.

В качестве системы управления роботом [3], используется платформа Arduino Pro Micro, т.к. она компактна, обладает достаточным количеством портов ввода-вывода, обладает низкой стоимостью и под эту платформу написано множество библиотек, что позволяет сократить время на разработку программной составляющей системы. Arduino используется в качестве генератора сигнала, форма полученного сигнала приведена на рисунке 2.

Амплитуда сигнала на выходе генератора недостаточна, для приведения системы в движение, поэтому требуется усилитель. Первоначально применялся усилитель на биполярных транзисторах (схема приведена на рисунке 3), однако из-за применения в цепи прохождения сигнала электролитического конденсатора в сочетании с трансформатором, образуется LC фильтр высоких частот, что приводит к значительному искажению сигнала на малых частотах и появлению нежелательных гармоник. Второй причиной являлось применение трансформатора от импульсного блока питания, рассчитанного на работу на частотах 80-120кГц, поэтому магнитопровод трансформатора обладал неподходящей передаточной характеристикой.

Форма сигнала на выходе усилителя на биполярных транзисторах приведена на рисунках 4 и 5.

В дальнейшем был собран другой усилитель на полевых транзисторах (рисунок 6) и изготовлен новый трансформатор. Выбор полевых транзисторов обусловлен малым сопротивлением открытого канала транзистора, что приводит к снижению потерь и увеличению КПД усилителя. Из схемы был убран конденсатор в цепи прохождения сигнала, что в купе с изготовлением трансформатора, на

более низкочастотном магнитопроводе, позволило улучшить передаточную характеристику и снизить искажения.

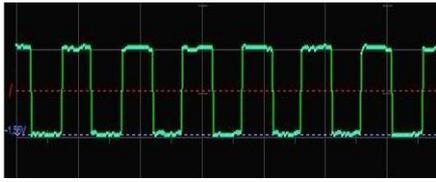


Рисунок 2 – Форма сигнала на выходе генератора

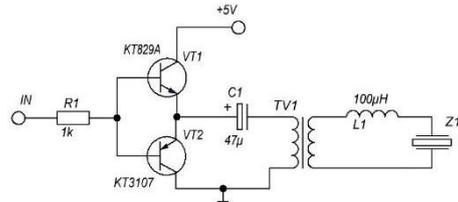


Рисунок 3 – Усилитель на биполярных транзисторах

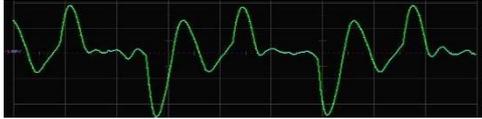


Рисунок 4 – 15кГц на выходе старого усилителя

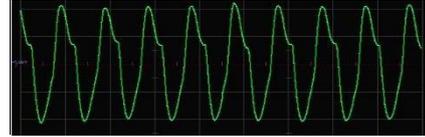


Рисунок 5 – 51кГц на выходе старого усилителя

В обоих усилителях на выходе установлен дроссель, он необходим для увеличения напряжения, поступающего на пьезоэлемент. Форма сигнала на выходе нового усилителя представлена на рисунках 7 и 8.

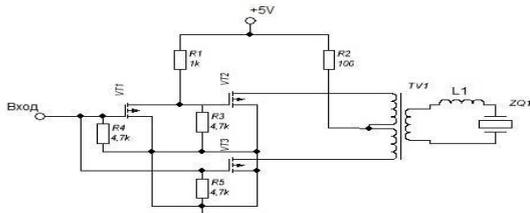


Рисунок 6 – Усилитель на полевых транзисторах

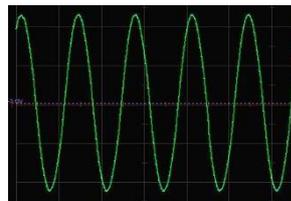


Рисунок 7 – 15кГц

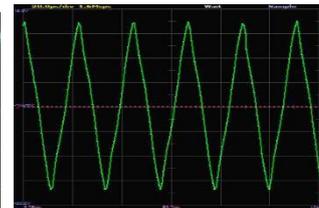


Рисунок 8 – 45кГц

Одной из перспектив развития данного проекта является использование роботов на основе пьезоэлементов в дефектоскопии: после того, как робот приведён в движение и продолжает двигаться по инерции, при помощи пьезоэлемента можно регистрировать затухающие колебания и анализируя их, можно судить о наличии микродефектов на поверхности, по которой двигался робот. Компактные размеры робота позволят анализировать поверхности в труднодоступных местах, например, внутреннюю поверхность труб.

**Список использованных источников:**

1. Gartner Hype Cycle [Электронный ресурс]. – Режим доступа : <https://www.gartner.com/smarterwithgartner/5-trends-emerge-in-gartner-hype-cycle-for-emerging-technologies-2018/>
2. Цифровые системы управления технологическим оборудованием : уч. пособие / В. Т. Минченя [и др.]. – Минск : Энциклопедикс, 2016. – 108 с. Информационные технологии и управление : материалы 49 науч. конф. аспирантов, магистрантов и студентов, Минск, 6–10 мая 2013 г. / Белорус. гос. ун-т информатики и радиоэлектроники ; редкол.: Л. Ю. Шилин [и др.]. – Минск : БГУИР, 2013. – 103 с.
3. Схемотехника цифровых преобразователей перемещений: Справочное пособие/ В.Г. Домрачев, В.Р. Матвеевский, Ю.С. Смирнов. – М.: Энергоатомиздат, 1992. - 392с.: ил.

## СИСТЕМА УПРАВЛЕНИЯ ДОКУМЕНТООБОРОТОМ КАФЕДРЫ

*Куприянова Д.В., Валентюкевич Д.А.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Искра Н.А. – старший преподаватель*

Документооборот - проблема каждой организации и кафедры университета не являются исключением. Необходимость создания систем управления документооборотом организации появилась давно и требует быстрого решения в современном мире.

На текущий момент каждая организация старается оптимизировать работу своих сотрудников и аккуратное ведение документации является ключевым требованием каждого руководителя. Всю документацию требуется структурировать, хранить в строгом порядке, а также беречь её сохранность.

Согласно закону Республики Беларусь «Об электронном документе и электронной цифровой подписи» [1], предприятия могут использовать электронную документацию, что позволяет разработать системы управления документооборотом в каждой организации.

Было предложено разработать систему на платформе 1С, для решения проблемы управления документацией на одной из кафедр университета. В процессе разработки системы управления документооборотом кафедры, были выбраны следующие пользователи:

- заведующий кафедрой;
- секретарь кафедры;
- заместитель заведующего кафедрой;
- учёный секретарь;
- ответственный по практике (студентов, магистрантов);
- ответственный за дипломное проектирование;
- секретари ГЭК (студентов, магистрантов).

Каждый из пользователей имеет свои права доступа к системе документооборота, а также свои права на редактирование и ведение документации. Ключевым пользователем является заведующий кафедрой, в его распоряжении просмотр всей документации пользователей системы управления. На данный момент реализованы следующие задачи:

Пользователи		Задачи
Заведующий кафедрой	Докладные записки	о изменении в графике отпуска
		о принятии сотрудников на работу
Учёный секретарь кафедры	Докладные записки	о закупках оборудования
		о премировании сотрудников
		о почасовой оплате
		о внесении изменений в расписание
Секретарь кафедры	Докладные записки	о передаче нагрузки на другую кафедру
		о пожеланиях совместителей к расписанию
		о тьюторах
		на изменение штатной численности
Ответственный по практике	Докладные записки	о истечении срока избрания преподавателя
		о оплате нагрузки преподавателя (замена по болезни)
		выписка заседания кафедры
		о отчислении студентов, магистрантов, аспирантов
Ответственный по дипломному проектированию	Распоряжения	о аттестации магистрантов, аспирантов
		о распределении выпускников
		о назначении ответственного
		о снятии с должности
Секретарь ГЭК	Приказы	о создании комиссии для приема зачёта экзамена
		о повторной практике
		на практику студентов I степени образования
		на практику студентов II степени образования
Секретарь ГЭК	Докладные записки	о начале производственной практики
		о начале преддипломной практики
		о начале практики (студентов II степени образования)
		о повторной практике
Секретарь ГЭК	Отчёт	о итогах практики производственной практики
		о итогах практики преддипломной практики
		о итогах практики (студентов II степени образования)
		о назначении тем студентов I и II степени образования
Секретарь ГЭК	Приказы	о изменении темы студентов I и II степени образования
		о изменении руководителя темы студентов I и II степени образования
		о создании рабочей комиссии
		о допуске к защите дипломных проектов, магистерских диссертаций
Секретарь ГЭК	График защиты	о защите дипломных проектов, магистерских диссертаций (по датам)
		о отчислении в связи с неявкой на защиту
		о рекомендации председателей ГЭКа
		о составе ГЭКа
Секретарь ГЭК	Докладные записки	о рецензентах дипломных проектов, магистерских диссертаций
		о назначении рецензентов дипломных проектов, магистерских диссертаций
		о итогах работы комиссии по защите дипломных проектов, магистерских диссертаций
		о итогах работы комиссии по защите дипломных проектов, магистерских диссертаций

Рисунок 1 – Задачи пользователей системы управления документооборотом кафедры

**Список использованных источников:**

1. Об электронном документе и электронной цифровой подписи: Закон Республики Беларусь от 28 декабря 2009г. №113-З (с изменениями и дополнениями) // Национальный Интернет-портал Республики Беларусь [Электронный ресурс] / Нац. центр правовой информ. Республики Беларусь. – Минск, 2013. – Режим доступа: <http://www.pravo.by>.

## MICROSOFT ORLEANS КАК РЕАЛИЗАЦИЯ МИКРОСЕРВИСНОЙ АРХИТЕКТУРЫ

Мазуркевич Е.А.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Насуро Е.В. – к.т.н.

Цель работы: исследовать структуру организации технологии microsoft orleans, рассмотреть применение данной технологии для реализации микросервисной архитектуры.

Microsoft Orleans - это инфраструктура, которая обеспечивает простой подход к созданию распределенных крупномасштабных вычислительных приложений без необходимости изучать и применять сложный параллелизм или другие шаблоны масштабирования [1]. Orleans предоставляет интуитивно понятный способ построения приложения, когда различные объекты бизнес-логики выглядят как множество изолированных глобально адресуемых объектов различных типов, именуемых Grain (зерно). Данные зерна распределены по кластеру серверов, которые в терминологии orleans называются Silo (хранилище). Данная структура отображена на рисунке 1.

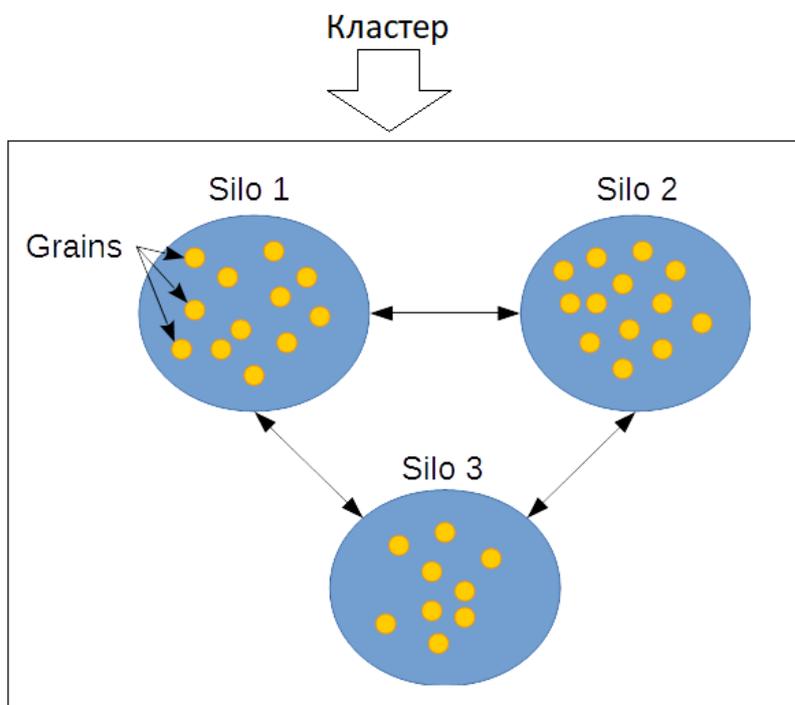


Рисунок 1 - Структура Microsoft Orleans

Реализация Orleans основана на модели актера, существующей с 1970-х годов. Однако, в отличие от актеров в более традиционных актерских системах, Orleans зерна - виртуальные актеры. Самым большим отличием является то, что физические экземпляры зерен полностью абстрагированы и автоматически управляются средой выполнения Orleans. Модель виртуального актера гораздо больше подходит для масштабных динамических рабочих нагрузок, таких как облачные сервисы, что является основным нововведением Orleans.

Преимущество виртуальных актеров в том, что в любой момент ваш код может получить у среды выполнения Orleans прокси для обращения к конкретному зерну по его интерфейсу и ключу [2]. При вызове методов прокси посылается сообщение кластеру из нескольких серверов, где оно будет доставлено одному зерну с заданным ключом. Среда выполнения гарантирует, что такое зерно будет создано в единственном экземпляре на одном из серверов и последующие вызовы будут доставлены ему. Среда выполнения также автоматически удаляет из памяти (деактивирует) зерна, которые не получали вызовов заданное время, таким образом постоянно собирая "мусор". Если сервер, на котором находились ранее зерна, стал недоступным — среда выполнения быстро поднимет экземпляры данных зерен на другом хранилище, что может сказаться только на небольшой задержке

на запуск грейнов. Если вызов должен прийти на тот же самый сервер — среда выполнения это оптимизирует и вызов будет локальным.

Пользой виртуальных актеров в том, что это структура очень легко масштабируется в облаках: хранилища проверяют доступность друг друга и определяют, когда кто-то недоступен, перераспределяют зерна недоступного хранилища между собой. Кластер будет доступен и выполнять свои функции, пока есть хотя бы одно активное хранилище, и при подключении новых участников кластера — они получают свой диапазон грейнов и начнут активно обрабатывать запросы.

Также среда выполнения дает гарантии что при выполнении метода в зерне, никакой другой вызов на то же самое зерно не придет, то есть среда выполнения гарантирует однопоточное выполнение кода, в рамках одного зерна.

Микросервисная архитектура — вариант сервис-ориентированной архитектуры программного обеспечения, ориентированный на взаимодействие небольших, насколько это возможно, слабо связанных и легко изменяемых модулей — микросервисов. Философия микросервисов гласит, что каждый сервис должен «делать что-то одно, и делать это хорошо» и взаимодействовать с другими сервисами простыми средствами [3]. Основные характеристики микросервисной архитектуры определяются принципами слабой связи между сервисами и высокой связанностью внутри сервиса. Приложение с микросервисной архитектурой отличается четкой структурой сервисов, которые представляют часть функционала приложения.

Философия микросервисной архитектуры ложится на модель структуры Orleans, где зерна выступают в роли сервисов, реализующих узкую часть бизнес логики приложения, которая изолирована в рамках зерна. Коммуникация между зернами-сервисам реализована на уровне среды выполнения Orleans, посредством интерфейсов реализованных зернами.

Однако Orleans не позволяет в полной степени реализовать все возможности микросервисной архитектуры. В первую очередь, разработку приложения не получится легко разделить между несколькими командами разработчиков, поскольку все сервисы описываются в одном проекте. Также одним из плюсов микросервисной архитектуры, который невозможно реализовать с Orleans, является использование различных языков программирования и технологий, подходящих под конкретные задачи. Еще к минусам реализации микросервисной архитектуры с помощью Orleans можно отнести отсутствие возможности развертывания и обновления только части сервисов, поскольку необходимо разворачивать хранилища целиком со всеми зернами.

Однако микросервисная архитектура с использованием Orleans будет очень полезна для начального этапа разработки микросервисного приложения, поскольку сразу позволяет разбивать логику приложения между зернами, которые в дальнейшем могут стать настоящими сервисами. Также нет необходимости в дополнительных трудозатратах на масштабирование приложения и настройки коммуникации между сервисами, поскольку среда выполнения реализует все эти функции самостоятельно. Данные трудозатраты являются основными причинами отказа от микросервисной архитектуры на начальном этапе разработки, однако Orleans позволяет уменьшить издержки.

**Список использованных источников:**

1. Microsoft Orleans | Microsoft Orleans Documentation [Электронный ресурс]. – Электронные данные. – Режим доступа <https://dotnet.github.io/orleans>
2. Getting Started with Microsoft Orleans [Электронный ресурс]. – Электронные данные. – Режим доступа <https://medium.com/@kritner/getting-started-with-microsoft-orleans-882cdac4307f>
3. Wagner, .NET Microservices: Architecture for Containerized .NET Applications / Bill Wagner, Cesar de la Torre, Mike Rousos – Redmond, Washington, 2018.

## **СТРУКТУРЫ ДАННЫХ ДЛЯ РАБОТЫ СО СДЕЛКАМИ НА БИРЖЕ**

*Митьковец А.А.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Поттосин Ю.В. – к.ф.-м.н., доцент*

В статье обсуждается применение классических структур данных, позволяющих работать с потоком данных с биржи. Дается структура данных для ведения книги сделок на основе связанного списка и хэш-таблицы. Обсуждается проведение сделок в контексте асинхронного программирования, даются рекомендации по использованию futures и promises.

Общение с биржей всегда строится по модели «клиент-сервер». Биржа представляет собой (готовый) сервер, который предоставляет API клиентам. В случае с конвенциональными биржами речь

может идти об интеграции с MetaTrader. Криптовбиржи чаще предоставляют REST API для выполнения активных действий и WebSocket API для наблюдения за состоянием биржи. Для общности, будем считать, что абстрактная биржа предоставляет абстрактный синхронный и асинхронный канал для коммуникаций.

Для успешной торговли необходимо моделировать на клиенте состояние биржи – по большей части на основе данных из асинхронного канала. Для того, чтобы успешно оставлять сделки, на клиенте необходимо моделировать существующие списки сделок (order book) – в сторону продажи и покупки для каждого используемого инструмента. Такие списки формируются на основе исходной копии списка и дальнейших асинхронных частичных обновлений. Каждая сделка состоит из цены, объема и направления (покупка или продажа).

С теоретической стороны к идеальному списку предъявляются следующие требования:

- 1) чтение первых двух-трех вершин за  $O(1)$ ;
- 2) список всегда отсортирован по ценам;
- 3) при адресации по цене – вставка, чтение и удаление за  $O(1)$ .

Простой структурой данных, которая похожа по требованиям, является order-statistic tree [1] – аугментированное красно-черное дерево, которое поддерживает выборку  $i$ -го крайнего значения за  $O(\log n)$  в придачу к остальным обыкновенным операциям над деревом тоже за  $O(\log n)$ .

У списков сделок есть своя специфика – во-первых, на ликвидном рынке разница в цене между соседними сделками минимальна, во-вторых, допустимые ценовые уровни зачастую определяются биржей в виде дискретной сетки (с шагом, например, 10 рублей, \$0,5).

При учете этой специфики, следующая структура данных позволяет добиться лучших показателей:

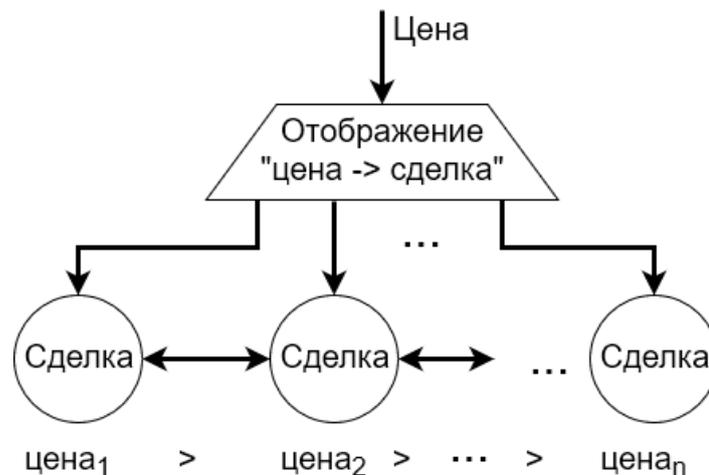


Рисунок 1 – Список сделок на основе двунаправленного списка и отображения

Сделки объединены в связный список, каждая сделка ссылается на две соседние по цене. Хранится только ссылка на первый элемент. Выборка первых  $N = \text{const}$  элементов происходит за  $O(1)$ . Отображение из цен на сделки означает выборку по цене и удаление за  $O(1)$ .

Вставка по цене с теоретической точки зрения занимает  $O(n)$  операций, где  $n$  – размер списка. На практике, для ликвидного рынка такие вставки, во-первых, редки, а во-вторых, ценовые дыры между сделками малы в силу ликвидности. Если цены на рынке дискретные, то перебор соседних цен через отображение приносит неплохие результаты.

Эмуляция состояния биржи на клиенте зависит от следующих главных асинхронных потоков данных:

- 1) обновления списка сделок;
- 2) обновления статуса собственных сделок (проведение торгов по ним, отмена, закрытие).

Поскольку обновления асинхронные, архитектура клиента должна это учитывать и предоставлять интерфейс, который бы скрывал хотя бы часть этой сложности. Например, типичная позиция на бирже включает в себя сделку на вход, экстренную сделку на выход (stop-loss) и take profit сделку на выход. Каждая сделка может быть в пяти состояниях: не выставлена, открыта, частично заполнена, выполнена, отменена. Для трех сделок это означает до 125 возможных состояний, причем почти все переходы случаются асинхронно. Вместе с этим необходимо получать обновления списка сделок.

Для контроля этой сложности хорошо зарекомендовал себя подход с созданием futures для ожидаемых переходов совместно с использованием их в языке, поддерживающем async/await конструкции.

Следующая схема помогает абстрагироваться от прямой работы с асинхронным каналом:

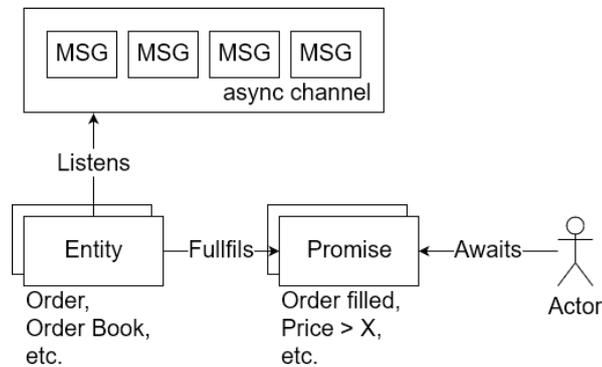


Рисунок 2 – сокрытие асинхронного канала при помощи promise/future без callbacks

Списку сделок достаточно иметь один вид следующие Promise: «цена покупки больше/меньше заданной». Не рекомендуется ждать строго указанной цены – в случае движения цены Promise сработает, однако он не сработает, если сделка окажется отмененной и этот ценовой уровень займет сделка в противоположную сторону.

Информация о собственных сделках также поступает из асинхронного источника, поэтому для них тоже нужны Promises:

1) Сделка отменена. Биржа может отменять сделки по своей инициативе (margin call, недостаточно средств и так далее).

2) Сделка частично выполнена. Может случиться при большом объеме, когда контрагентами по покупке/продаже выступают несколько других участников рынка.

3) Сделка полностью выполнена.

Имея следующие примитивы, можно реализовывать устойчивые и стабильные программы по работе с биржей. Программный код становится похож на набор последовательных инструкций – чего не происходит при прямом чтении из асинхронного канала или при использовании подхода с callbacks.

```

order_book = await OrderBook.create('USD/BYN')
price = await order_book.buy_price_lt(2.00)
buy_order = await Order.buy(price, 1000)
await buy_order.one_fill

take_profit = order_book.sell_price_gt(2.10)
stop_loss = order_book.sell_price_lt(1.90)
outcome = await first(take_profit, stop_loss)
cancel_buy = buy_order.cancel()

await (await Order.sell_all(outcome.value))
await cancel_buy
    
```

Рисунок 3 – псевдокод покупки и последующей продажи BYN

**Список использованных источников:**

1. Cormen, T. Introduction to Algorithms / T. Cormen, C. Leiserson, R. Rivest, C. Stein // MIT Ppess. – 1990. – P. 339–345.

## УСТРАНЕНИЕ ДИСТОРСИИ НА ШИРОКОУГОЛЬНЫХ ИЗОБРАЖЕНИЯХ

*Некревич С.А., Сапронова Ю.И.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Лукашевич М.М. – к.т.н., доцент*

Показан метод нахождения суммарной дисторсии, основанный на измерениях искажений тестового изображения (шахматной доски или квадратной сетки). Для устранения дисторсии (получения изображения, близкого к идеальному) необходимо к координатам каждой точки реального изображения добавить величину полученной с помощью тестового изображения дисторсии. Рассмотрены проблемы, которые могут возникнуть в результате произведенной коррекции.

Широкоугольное изображение представляет собой снимок (кадр), полученный с помощью объектива с коротким фокусным расстоянием. Преимуществом таких изображений, является большой угол зрения, однако они имеют и недостаток – дисторсию. Это означает что, линии объекта, которые являлись прямыми, искажаются, за исключением тех, которые лежат в плоскости оптической оси объектива, а величина искажения нелинейно зависит от параметров объекта. Изображения с дисторсией не пригодны для анализа ввиду того, что нарушается геометрическое подобие исходного объекта и его изображения, в связи с чем требуется предварительное исправление дисторсии.

При наличии только сферических искажений, исправить дисторсию можно зная положение точки на изображении и радиус кривизны линзы [1]. Если рассматривать изображение, в котором присутствует суперпозиция искажений необходимо предварительно рассчитать суммарную дисторсию для тестового изображения [2]. В данном случае, оно основано на определении искажений шахматной доски (либо квадратной сетки) по двум ортогональным осям (рис. 1).

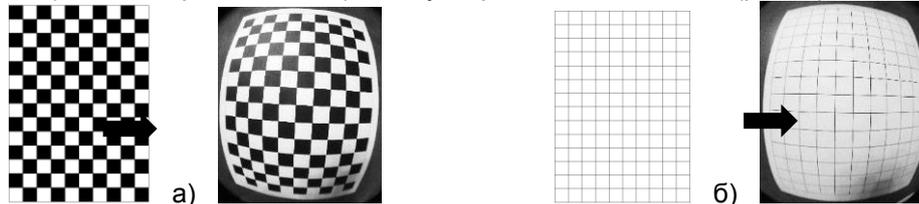


Рисунок 1 – Идеальное и реальное изображение шахматной доски (а) и сетки (б) для калибровки

Для вычисления дисторсии необходимо определить точку идеального объекта, в которой должно находиться изображение по законам оптики. В этом случае, некоторая точка реально объекта изображается в виде точки  $A'(x', y')$ , не совпадающей с точкой идеального изображения  $A'_0(x'_0, y'_0)$ . В таком случае дисторсия является отклонением точки идеального изображения  $A'_0$  от точки реального изображения  $A'$ . Величина дисторсии ( $\Delta x'$  и  $\Delta y'$ ), в общем случае, является различной для различных точек изображения и представляет собой некоторую функцию координат точек изображения. Массивы чисел ( $\Delta x', \Delta y'$ ) позволяют получить нелинейную зависимости дисторсии от реальных координат узловых точек изображения шахматной доски (сетки). Для этого необходимо:

- восстановить контур шахматной доски (квадратной сетки);
- определить отклонение узловых точек реального изображения от идеального;
- по величине отклонений определить вид функции изменения дисторсии;
- произвести интерполяцию значения дисторсии для точек, отличных от узловых.

Таким образом, с использованием полученной суммарной дисторсии для каждой точки тестового изображения, можно произвести коррекцию реального изображения другого объекта. Для устранения дисторсии (получения изображения, близкого к идеальному) необходимо к координатам  $(x', y')$  каждой точки  $A'$  реального изображения добавить величину дисторсии ( $\Delta x', \Delta y'$ ).

После коррекции искажений, края изображения могут стать изогнутыми, в результате чего может потребоваться кадрирование исправленного изображения. Кроме того, после устранения дисторсии может быть перераспределено разрешение изображения: при положительной дисторсии («подушка») резкость изображения по краям может увеличиться по отношению к центру, при отрицательной дисторсии («бочка») – наоборот, резкость по краям может уменьшиться.

**Список использованных источников:**

1. Капустин, В.В. Коррекция координатных искажений в телевизионно-вычислительных системах / В.В. Капустин // Доклады ТУСУРа, № 4 (38), декабрь 2015 – Томск, 2015. – С. 174–177.
2. Жимбуева, Л.Д. Метод определения суммарной дисторсии цифровых изображений / Л.Д. Жимбуева // Компьютерная Оптика, том 35, №3. – Самара, 2011. – С. 347-355.

## СИСТЕМА ПРОФИЛИРОВАНИЯ ДЛЯ ОПЕРАЦИОННОЙ СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ

Оверченко А.С.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Сапронова Ю.И. – ассистент

В статье рассмотрены основы систем инструментирования и операционных систем реального времени. Начата разработка собственной системы профилирования и сбора информации о событиях системы с использованием веб-интерфейса для операционной системы реального времени TI-RTOS.

Сегодня встраиваемые системы встречаются практически повсеместно. Встраиваемая система – это система, которая выполняет определенный производителем набор функций [1]. Изменение набора этих функций доступно лишь производителю устройства, что накладывает определенный отпечаток при разработке программного обеспечения для них. Также к таким системам обычно выдвигаются высокие временные требования и требования отказоустойчивости. Это приводит как к увеличению трудозатрат на разработку таких систем, так и к использованию специализированных операционных систем, которые разработаны для работы в режиме реального времени.

Операционная система реального времени – это операционная система (ОС), которая предназначена для обслуживания многопоточных приложений, которые обрабатывают данные по мере их поступления [2]. Обычно ядро такой ОС состоит из специально спроектированного планировщика задач и примитивов синхронизации. Также к базовым компонентам таких ОС относят драйверы используемых периферийных устройств, сетевой стек и файловые системы. Но они являются опциональными и добавляются только по мере необходимости.

Инструментирование программного обеспечения связано с набором возможностей, предоставляемых для измерения производительности программного продукта, поиска компонентов, ограничивающих скорость его работы, а также записи информации трассировки и нахождения некорректно работающих модулей [3]. В инструментировании выделяют следующие направления:

- трассировка кода;
- отладка и обработка исключений;
- профилирование;
- подсчет производительности;
- логирование данных.

Стоит отметить, что обычно вышеперечисленные направления взаимосвязаны и реализуются в виде набора инструментов для разработки.

Профилирование является способом анализа программного обеспечения во время его выполнения и позволяет измерять различные параметры работы программы. Под подсчетом производительности понимают процесс предоставления информации о качестве работы разрабатываемого программного продукта с помощью счетчиков производительности. Логирование – это процесс сбора и хранения произошедших в системе событий.

К программному обеспечению, предназначенному для инструментирования, выдвигается одно важное требование: минимальное влияние на анализируемое программное обеспечение. Это делается для того, чтобы уменьшить расхождение работы программы при использовании средств инструментирования и без них.

Встречаются случаи, когда конечному пользователю также необходимо использовать средства профилирования для встраиваемой системы. Использовать классические средства профилирования, которые используются при разработке таких систем, не является возможным. Но разработчики могут предусмотреть этот вариант и добавить в систему возможность отладки с использованием специально разработанного программного обеспечения.

Эта задача и побудила автора для разработки собственной системы профилирования и логирования. Разрабатываемая система подразумевает использование ОС реального времени RTOS и предназначена для использования во встраиваемых системах с возможностью подключения с помощью Ethernet или Wi-Fi. Информация для конечного пользователя выводится на страницу веб-браузера и отображает как общую загрузенность для каждого ядра используемого микропроцессора, так и процент процессорного времени, используемого каждой задачей.

**Список использованных источников:**

1. Heath, Steve. Embedded Systems Design / Steve Heath. — ELSEVIER, 2003. — P. 2-8.
2. RTOS (Real-Time Operating System) — Национальная библиотека им. Н. Э. Баумана [Электронный ресурс]. – Режим доступа: [https://ru.bmstu.wiki/RTOS\\_\(Real-Time\\_Operating\\_System\)](https://ru.bmstu.wiki/RTOS_(Real-Time_Operating_System)). – Дата доступа: 23.03.2019г.
3. Source code instrumentation overview [Электронный ресурс]. — Режим доступа : [https://www.ibm.com/support/knowledgecenter/SSSHUF\\_8.0.0/com.ibm.rational.testrt.doc](https://www.ibm.com/support/knowledgecenter/SSSHUF_8.0.0/com.ibm.rational.testrt.doc). – Дата доступа: 12.03.2019г.

# ОБНАРУЖЕНИЕ И РАСПОЗНАВАНИЕ ОБЪЕКТОВ В ВИДЕОПОТОКЕ НА ОСНОВЕ R-CNN ДЕТЕКТОРОВ

Пац Е.А.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Насуро Е.В. – к.т.н., доцент каф. ЭВМ

Средства интеллектуальной обработки кадров видеопоследовательности для автоматизации поиска и классификации объектов в видеопотоке. Сравнительный анализ.

Одной из актуальных проблем современного мира является обнаружение и распознавание объектов на видеопотоке. Для работы с видеопотоком необходимо разбить его на кадры, которые более удобны для обработки. Для раскадровки видео существует библиотека с открытым исходным кодом решающая задачи компьютерного зрения openCV[1]. Библиотека кроссплатформенна и поддерживается такими языками программирования, как Python, C++, Java. Также библиотека разработана для использования в системах реального времени, что подходит для решения поставленной задачи. Присутствует возможность использования многоядерных процессоров для обработки информации. После завершения обработки каждого из кадров, есть возможность восстановить последовательность видеопотока.

Зачастую основой интеллектуальной системы выступает нейронная сеть. На данный момент разработано большое количество моделей нейронных сетей. Модель нейронной сети выбирается в зависимости от поставленной задачи.

На практике, нейронные сети используются для решения задач:

1. Классификации;
2. Кластеризации(категоризации);
3. Аппроксимации;
4. Предсказания(прогноза);
5. Оптимизации.

Обычно для распознавания объекта на изображении достаточно прибегнуть к классификации. Для этого, зачастую, используются сверточные нейронные сети [2]. Данная модель зарекомендовала себя хорошим процентом распознавания и скоростью работы. Однако данный способ имеет ограничение: наличие только одного объекта на изображении.

Таким образом, для решения задач обнаружения и распознавания, придется прибегнуть к сегментации изображения с последующей классификацией сегментов. В итоге, по завершению работы нейронной сети, должна быть получена информация о регионах изображения, в которых были найдены признаки соответствующего объекта, а также процент вероятности достоверности объекта.

Однако, проблемой данного подхода, является то, что объекты на изображении могут иметь различное местоположение и пропорции. Таким образом, с большой вероятностью появятся необходимость в обработке большего количества регионов, следовательно, подход требует значительных вычислительных мощностей для решения поставленной задачи. В связи с этим, были разработаны следующие алгоритмы обнаружения объектов [3]: R-CNN, Fast R-CNN, Faster R-CNN, YOLO.

Алгоритмы R-CNN и Fast R-CNN используют выборочный поиск для выделения значимого региона. Использование выборочного поиска ресурсоемкий и времязатратный процесс. Использование алгоритма Faster R-CNN решает эту проблему. В алгоритме Faster R-CNN на вход сверточной сети, также подается изображение, которое представляет собой сверточную карту признаков. Однако, вместо использования выборочного поиска, для выделения предполагаемых регионов, используется отдельная сеть. Предсказанные области имеет возможность дальнейшего изменения для более точного выделения объекта на изображении.

Данный подход значительно сокращает время обработки изображения, что позволяет использовать алгоритм Faster R-CNN в системах реального времени (Рисунок 1).

При работе с нейронными сетями важным параметром является база данных используемая для обучения сети. Зачастую, используется специализированная выборка данных, которая содержит подготовленные, размеченные данные с информацией об объектах.

Для решения задачи обнаружения и распознавания разумно использовать базу данных, которая имеет большое количество классов, которые часто встречаются в повседневной жизни. Одной из подходящих выборок данных является COCO dataset (Общие объекты в контексте) [4]. Данная база данных, является крупномасштабным набором данных для обнаружения, сегментации и классификации объектов.

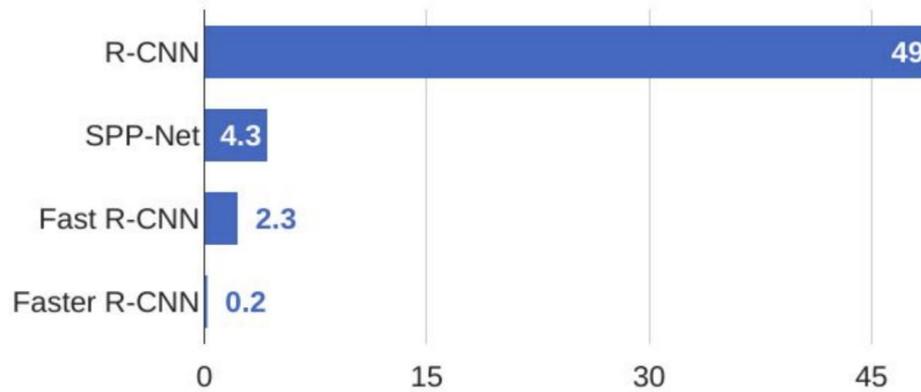


Рисунок 1 – Сравнение тестовой скорости алгоритмов обнаружения объектов

В разрабатываемой системе обнаружения и распознавания объектов в видеопотоке будут поэтапно применены описанные средства, что позволит идентифицировать объекты в режиме реального времени.

**Список использованных источников:**

1. Библиотека компьютерного зрения OpenCV [Электронный ресурс] – Режим доступа: <https://opencv.org/> – Дата доступа: 24.03.2019.
2. Онлайн курс “Программирование глубоких нейронных сетей на Python” [Электронный ресурс] – Режим доступа: <https://www.asozykin.ru/courses/nnp/python> – Дата доступа: 24.03.2019.
3. Блог платформа Medium [Электронный ресурс] – Режим доступа: <https://medium.com/> – Дата доступа: 24.03.2019.
4. COCO Dataset “Common objects in context” [Электронный ресурс] – Режим доступа: <http://cocodataset.org> – Дата доступа: 24.03.2019.

## СИСТЕМА АВТОМАТИЧЕСКОГО ПОЛИВА РАСТЕНИЙ

*Писчик А.С., Науменко Д.К.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Сапронова Ю.И. – ассистент*

На частоту и обильность полива влияют такие факторы как вид растения, температура и влажность окружающего воздуха, время суток и время года. В работе изучены вопросы, связанные с автоматизацией процесса полива растений в зависимости от указанных факторов.

Комнатные растения являются не только лишь украшением в квартире. Они выделяют во внешнюю среду кислород, устраняют запыленность помещений. Поэтому очень важно, чтобы растение всегда оставалось здоровым, а для этого нужно правильно за ним ухаживать. Данная работа посвящена изучению вопросов, связанных с автоматизацией процесса полива растений в зависимости от их вида и условий внешней среды.

Целью работы является определение факторов, влияющих на частоту и интенсивность полива растений, изучение особенностей построения систем на базе платформ семейства Arduino и их модулей, и разработка систему полива, позволяющую автоматизировать процесс ухода за растениями.

Было определено, что на частоту полива влияют следующие факторы: температура воздуха, влажность воздуха, тип растения, время суток и время года. В связи с этим в состав устройства были включены следующие модули:

- 1) управляющий микроконтроллер Arduino Uno R3 [1];
- 2) датчик влажности почвы [2], необходим для контроля недостатка или избытка влаги;
- 3) датчик температуры воздуха [3] т.к. при снижении окружающей температуры растение нуждается в меньшем количестве воды;
- 4) датчик влажности воздуха [4], т.к. при низкой влажности происходит более интенсивное испарение из почвы и листьев, следовательно, растение нуждается в большем поливе, а при высокой влажности – наоборот;
- 5) насос для подачи воды растению;

- б) датчик уровня воды [5] для слежения за количеством оставшейся воды, чтобы вовремя оповестить пользователя о том, что резервуар пуст;
- 7) экран [6] для отображения необходимой информации (например, дата, время, влажность и температура воздуха, влажность почвы);
- 8) кнопки [7] для управления (например, изменения частоты полива вручную, изменения типа растения, ввод режима полива и т.д.);
- 9) часы реального времени [8], необходимы для контроля за режимом полива (для некоторых растений необходим более редкий полив зимой, некоторые растения лучше поливать утром и т.д.).

Функционал разработанного устройства планируется расширить, путем возможности управления поливом нескольких растений с одного управляющего модуля. Для этого датчика влажности выделяется в отдельный измерительный узел. Каждый измерительный узел будет подключен к отдельному горшку. При этом, группу горшков планируется подсоединить к одному резервуару с водой, насос которого также будет подключаться к управляющему модулю и отвечать за подачу воды в каждый из горшков растений отдельно. Взаимодействие измерительного узла, насоса и управляющего модуля планируется организовать по беспроводному (например, радио) каналу.

Для обеспечения автономности работы системы необходимо также рассмотреть электропитание устройства. В связи с тем, что большую часть времени устройство будет находиться в режиме пониженного энергопотребления и основная энергия будет тратиться насосом, можно рассматривать альтернативные источники энергии для питания устройства.

**Список использованных источников:**

1. Микроконтроллер Arduino Uno [Электронный ресурс] – Режим доступа : <https://datasheet.octopart.com/A000066-Arduino-datasheet-38879526.pdf> – Дата доступа : 25.03.2019;
2. Датчик влажности почвы [Электронный ресурс] – Режим доступа : [http://www.fecegypt.com/uploads/dataSheet/1480854383\\_water%20and%20soil.pdf](http://www.fecegypt.com/uploads/dataSheet/1480854383_water%20and%20soil.pdf) – Дата доступа : 25.03.2019;
- 3,4. Датчик температуры и влажности воздуха KY-015 [Электронный ресурс] – Режим доступа : <https://arduino-modules.info/ky-015-temperature-humidity-sensor-module/> – Дата доступа: 25.03.2019;
5. Датчик уровня воды [Электронный ресурс] – Режим доступа : [http://www.fecegypt.com/uploads/dataSheet/1480850810\\_water.pdf](http://www.fecegypt.com/uploads/dataSheet/1480850810_water.pdf) – Дата доступа : 25.03.2019;
6. LCD-Экран I2C 1602 [Электронный ресурс] – Режим доступа : <https://opencircuit.nl/ProductInfo/1000061/I2C-LCD-interface.pdf> – Дата доступа : 25.03.2019;
7. Нормально разомкнутая Кнопка [Электронный ресурс] – Режим доступа : <https://www.arduino.cc/documents/datasheets/Button.pdf> – Дата доступа : 25.03.2019;
8. Часы реального времени DS1302 [Электронный ресурс] – Режим доступа: <https://datasheets.maximintegrated.com/en/ds/DS1302.pdf> – Дата доступа : 25.03.2019;

## АЛГОРИТМ ВЗАИМОДЕЙСТВИЯ МОБИЛЬНЫХ РОБОТОВ В СОСТАВЕ РОБОТОТЕХНИЧЕСКОГО КОМПЛЕКСА

*Пичкур Е.В.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Дадыкин А.К. – к.т.н., доцент*

В настоящее время, роботы заменяют человека в основном на конвейерных линиях заводов, однако всё большее внимание уделяется мобильным роботам и интеллектуальным комплексам, состоящим из нескольких мобильных роботов. Такие комплексы могут снять с человека бремя выполнения скучных, грязных и опасных работ — например, обнаружение и обезвреживание мин, высокоточное сельское хозяйство и обнаружение лесных пожаров. В работе детализирована структура такого комплекса при помощи диаграмм прецедентов, и подробно рассмотрен алгоритм взаимодействия элементов комплекса, позволяющий решать задачу обнаружения мин.

Исследуемый комплекс состоит из летающего робота (далее БПЛА), наземного робота (далее БПНА) и командного центра (далее КЦ) [1-3]. БПЛА в комплексе выступает в роли разведчика, БПНА — в роли разведчика или робота-сапёра( в зависимости от указаний оператора), КЦ выступает как единый центр управления и контроля. Общий алгоритм работы комплекса можно описать следующим образом:

- 1) Оператор задает маршрут полета для БПЛА, используя КЦ;
- 2) БПЛА совершает полет по заданному маршруту и выполняет подповерхностную радиолокацию почвы;
- 3) При обнаружении мины, БПЛА отправляет ее GPS координаты на КЦ;
- 4) КЦ наносит полученные координаты мины на свою карту;

- 5) Пункты 3-4 повторяются до тех пор, пока БПЛА не завершит полет по заданному маршруту;
- 6) КЦ строит маршрут для БПЛА на основании карты местности и полученных данных о местонахождении мин;
- 7) БПЛА перемещается по заданному маршруту, обезвреживая мины и обнаруживая новые, если они есть.

Рассмотрим структуру комплекса более подробно, для определения требований к каждому элементу. Диаграмма прецедентов командного центра приведена на рисунке 1.

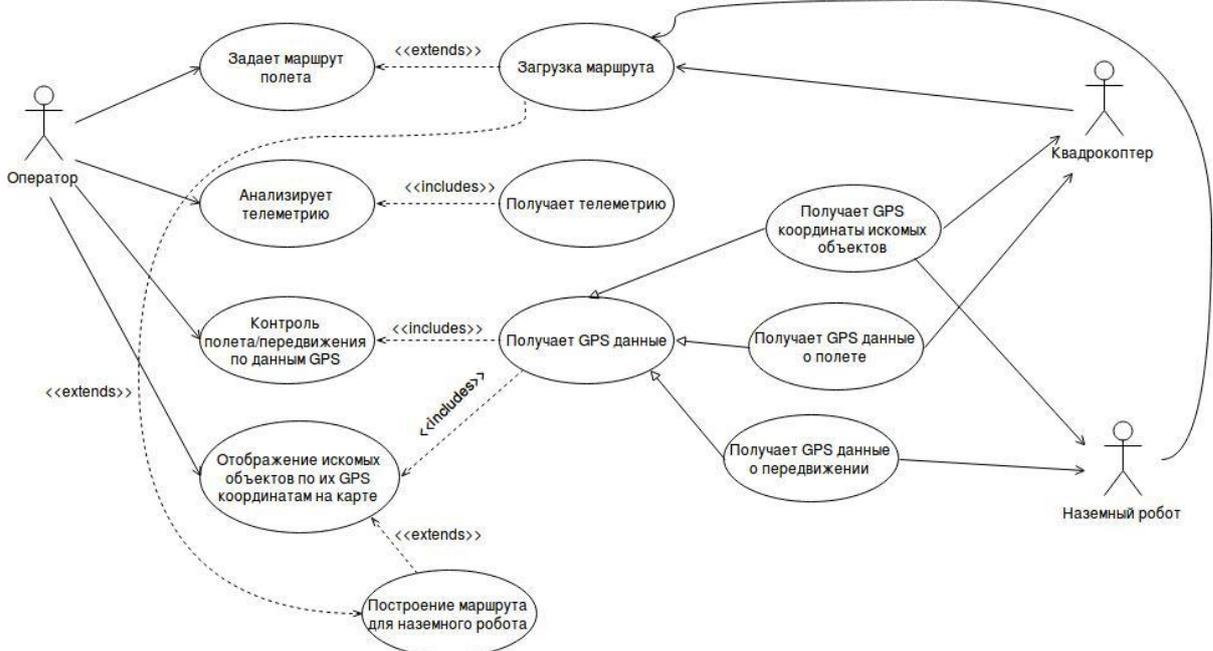


Рисунок 1 — Диаграмма прецедентов командного центра

Большую часть задач, описанных на рисунке 1 решает программное обеспечение Mission Planner[4]. Поскольку это программное обеспечение относится к типу open-source было принято решение использовать его в качестве КЦ. При этом стоит учесть, что Mission Planner дает возможность построить путь для робота автоматически, но не учитывает препятствия на его пути. Предлагаемый алгоритм построения пути для БПЛА таков:

- 1) Произвести семантическое сегментирование карты с использованием нейронной сети;
- 2) Представить карту в виде графа, где каждый пиксель является 8-связным узлом и разметить ее на основании данных семантического сегментирования;
- 3) Зная начальную точку и список точек, которые необходимо пройти, проложить путь, используя алгоритм A\* [5].

Теперь рассмотрим диаграмму прецедентов мобильного робота, приведенную на рисунке 2. Данная диаграмма является общей для БПЛА и БПНА.



Рисунок 2 — Диаграмма прецедентов мобильных роботов

- Из рисунка 2 видно, что мобильный робот должен уметь решать 5 задач:
- Определять и корректировать свои GPS координаты, а также координаты обнаруженных объектов;
  - Обнаруживать искомые объекты;

- Избегать препятствий на пути;
- В случае отсутствия GPS сигнала использовать визуально-инерциальную навигационную систему для оценки своего перемещения в пространстве;
- Уметь перемещаться на основе данных GPS.

Последняя задача решается при помощи контроллера перемещения с установленной на него прошивкой Ardupilot Rover или Ardupilot Copter соответственно [6], а также с помощью КЦ, работающего под управлением вышеупомянутого ПО Mission Planner. Задача обнаружения мин была рассмотрена в [3], в данной работе принимается что GPS координата найденной мины совпадает с GPS координатой самого робота (с поправкой на высоту).

Что касается определения и корректировки GPS координат, то было принято решение использовать технологию DGPS (Differential GPS), поскольку она позволяет сильно уменьшить ошибку позиционирования [7]. На основании этого формируется корректирующее сообщение [8].

Для избежания столкновений с препятствиями используется механизм построения границы вокруг робота. Пространство вокруг робота делится на сектора по 45 градусов, и камера, смотрящая вперед, оценивает расстояние до препятствия. Это расстояние используется, чтобы построить вокруг робота восьмиугольник границы. Наконец, если расстояние становится меньше заданного порога, робот пытается обогнуть препятствие или останавливается.

Разработка новой визуально-инерциальной системы навигации является довольно сложной задачей, поэтому автор выбрал использование подхода из [9], комбинирующего данные с камеры и ряда инерциальных сенсоров (гироскопы, акселерометры) с последующей обработкой расширенным фильтром Калмана.

#### **Список использованных источников:**

1. Tatur, M. M. Multifunction System of Mobile Robotics / M. M. Tatur, A. K. Dadykin, M. Kurdi // The Third International Conference on Electrical, Electronics, Computer Engineering and their Applications. - 2016. - P. 110 - 113.
2. Efficient navigation system of mobile robot with thermography image processing system and mapping of quadcopter / M. Kurdi and others // The 4th International Conference on Computer Science, Computer Engineering, and Education Technologies (CSCEET2017). - Institute of Electrical and Electronics Engineers (IEEE), 2017. - Pp. 23-28.
3. Cooperative Unmanned Air and Ground Vehicles for Landmine Detection / M. Kurdi and others // International Journal of Electrical and Electronic Engineering & Telecommunications, 2019.
4. Mission Planner Documentation [Electronic resource] // Mission Planner Home. - Mode of access: <http://ardupilot.org/planner/index.html>. - Date of access: 10.01.19.
5. Radiation Search Operations using Scene Understanding with Autonomous UAV and UGV / G. Christie and others // Journal of Field Robotics. - 2016. - P. 110 – 137.
6. Mission Planner Documentation [Electronic resource] // DOWNLOADS: Firmware. - Mode of access: [http://ardupilot.org/planner/docs/common-downloads\\_firmware.html](http://ardupilot.org/planner/docs/common-downloads_firmware.html). - Date of access: 10.01.19.
7. Википедия [Электронный ресурс] // DGPS. - Режим доступа: <https://ru.wikipedia.org/wiki/DGPS>. - Дата доступа: 22.02.19.
8. RACELOGIC support centre [Electronic resource] // How does DGPS (Differential GPS) work?. - Mode of access: [https://racelogic.support/01VBOX\\_Automotive/01General\\_Information/Knowledge\\_Base/How\\_does\\_DGPS\\_\(Differential\\_GPS\)\\_work%3F](https://racelogic.support/01VBOX_Automotive/01General_Information/Knowledge_Base/How_does_DGPS_(Differential_GPS)_work%3F). - Date of access: 22.02.19.
9. Weiss, Stephan Real-Time Metric State Estimation for Modular Vision-Inertial Systems / Weiss Stephan, Siegwar Roland Y. // 2011 IEEE International Conference on Robotics and Automation : (ICRA 2011) ; Shanghai, China, 9 - 13 May 2011. - 2011 . - P. 4531-4537

## **УМНОЖЕНИЕ КВАТЕРНИОНОВ С ИСПОЛЬЗОВАНИЕМ ЛОГАРИФМИЧЕСКОЙ СИСТЕМЫ СЧИСЛЕНИЯ**

*Сапронова Ю.И.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Петровский Н.А. – к.т.н., доцент*

Показано исследование умножения кватернионов, представленных в форме Кэли-Дисконна с использованием логарифмической системы счисления. На вход множителя поступают коэффициенты, представленные в логарифмической полярной форме. Процесс вычисления разбивается на две параллельные практически одинаковые части. За счет особенностей вычисления, в один момент времени каждая часть работает либо с функцией суммы, либо с функцией разности. При получении результата необходимо произвести корректировку квадранта получаемого комплексного числа. Для получения действительных коэффициентов кватерниона необходимо произвести обратное преобразование из логарифмической полярной формы.

Умножение кватернионов является затратной вычислительной операцией, требующей оптимизации, поскольку включает в себя 16 умножений и 12 сложений действительных чисел [1]. Кроме этого, немалую роль играет динамический диапазон значений, которые можно использовать при умножении. Представление с фиксированной точкой позволяет производить быстрые вычисления, однако имеет узкий динамический диапазон. Использование представления с

плавающей точкой позволяет расширить этот диапазон, но при этом увеличивает время вычислений. Применение логарифмической системы счисления может помочь при вычислении результата произведения двух кватернионов, коэффициенты которых представлены с использованием фиксированной точки, таким образом, что динамический диапазон значений будет больше (по сравнению с представлением с фиксированной точкой без применения логарифмической системы счисления), а время вычисления будет меньше (по сравнению с представлением с плавающей точкой без применения логарифмической системы счисления). Целью работы является исследование структуры устройства для умножения кватернионов с использованием логарифмической системы счисления.

Как показано в [2] применение логарифмической системы счисления для умножения кватернионов, представленных в виде  $Q = Q_1 + Q_2i + Q_3j + Q_4k$  и  $R = R_1 + R_2i + R_3j + R_4k$ , не приносит никакого выигрыша в количестве реализуемых операций ввиду не коммутативности произведения, т.е.  $\ln \ln (QR) \neq \ln \ln (Q) + \ln \ln (R)$ . Другим способом представления кватернионов  $Q$  и  $R$  является конструкция Кэли-Диксона [3]:

$$Q = Q_{12} + Q_{34}j \text{ и } R = R_{12} + R_{34}j, \quad (1)$$

где  $Q_{12} = Q_1 + Q_2i$ ,  $Q_{34} = Q_3 + Q_4i$ ,  $R_{12} = R_1 + R_2i$  и  $R_{34} = R_3 + R_4i$ , представляют собой комплексные числа. Использование логарифмической системы счисления заключается в представлении каждого из указанных комплексных чисел вида  $C = C_1 + C_2i$  в логарифмической полярной форме [3] по основанию  $b$ :

$$c = c_1 + c_2i = \log \log \left( \sqrt{C_1^2 + C_2^2} \right) + \arctg \left( \frac{C_2}{C_1} \right) \cdot i \quad (2)$$

Таким образом, сходные кватернионы  $Q$  и  $R$  будут представлены в виде пары логарифмических коэффициентов  $q_1, q_3$  и  $r_1, r_3$ , а также пары углов (аргументов)  $q_2, q_4$  и  $r_2, r_4$  соответственно. При этом при переводе значений нужно также учитывать квадрант, в котором находится исходное комплексное число. Эти значения поступают на вход умножителя, представленного на рисунке 1. Для начала входные значения формируют промежуточные значения (3), необходимые в дальнейших расчетах.

$$\begin{aligned} z_1 &= q_1 - q_3 + r_1 - r_3, \\ z_2 &= q_2 - q_4 + r_2 + r_4 + \frac{\pi}{2}, \\ z_3 &= q_1 - q_3 - r_1 + r_3, \\ z_4 &= q_2 - q_4 + r_2 + r_4. \end{aligned} \quad (3)$$

Как видно, значения  $z_2$  и  $z_4$  различаются на  $\frac{\pi}{2}$ , таким образом, нет необходимости иметь два блока для вычисления функций  $\log(|\cos(x)|)$  и  $\log(|\sin(x)|)$ . Дальнейшие вычисления разделяются на две параллельно работающие практически одинаковые части. Единственное различие заключается в том, какая функция суммы  $s(x)$  или разности  $d(x)$  [4] будет использоваться. Процесс вычисления построен таким образом, что за счет особенностей промежуточных значений (3) каждая половина умножителя использует только одну из функций в один момент времени, т.е. если одна половина использует функцию суммы, то другая использует функцию разности. За определение используемой функции отвечает блок «Выбор функции».

За счет функций  $\log(|\cos(x)|)$  и  $\log(|\sin(x)|)$  возникает необходимость коррекции квадрантов выходного результата (т.е. аргументов комплексных чисел). Коррекция осуществляется за счет добавления или вычитания константы ( $\frac{\pi}{2}$  или  $\pi$ ) и возможного изменения знака функции и  $\arctg(b^{-x})$  (а в некоторых случаях значение этой функции нет необходимости рассчитывать).

Для формирования результатов умножения  $m_i$  необходимы входные значения  $q_i$  и  $r_i$ , а также рассчитанные ранее значения с учетом коррекции. Полученные значения также представлены в логарифмической полярной форме, поэтому необходимо применить обратное преобразование [3]:

$$C = C_1 + C_2i = b^{c_1} \cdot \cos(c_2) + b^{c_1} \cdot \sin(c_2) \cdot i \quad (4)$$

Необходимость применения нескольких умножений подряд может сократить накладные расходы на перевод в логарифмическую систему счисления.

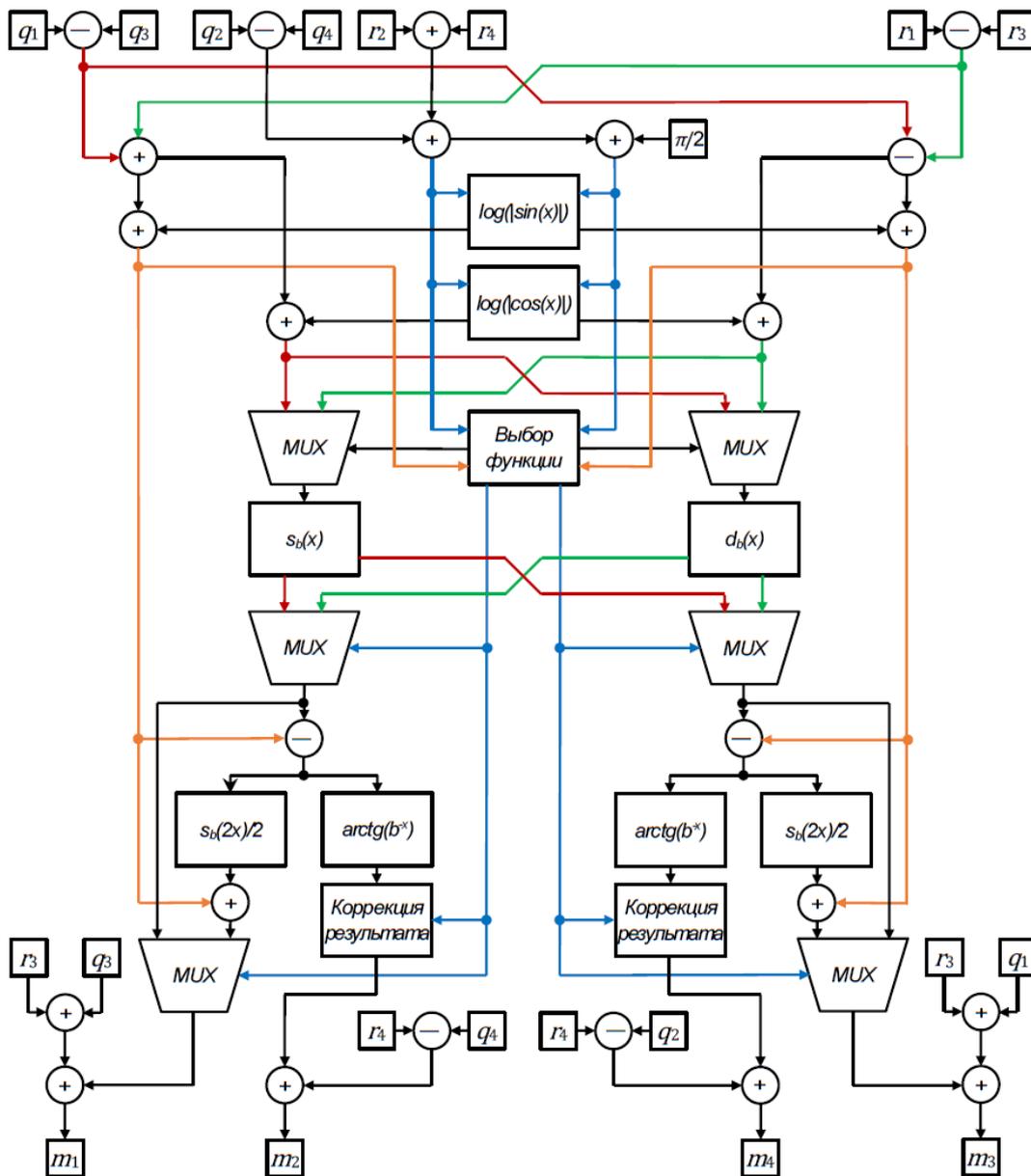


Рисунок 1 – Схема умножителя кватернионов с использованием логарифмической системы счисления

**Список использованных источников:**

1. Ward J.P. Quaternions and Cayley Numbers // Springer Science+Business Media Dordrecht, 1997. – 242 p.
2. Arnold, M.G. Towards a Quaternion Complex Logarithmic Number System // 20th IEEE Symposium on Computer Arithmetic, 2011. – pp. 33-42.
3. Sangwine, S.J. Quaternion Polar Representation with a Complex Modulus and Complex Argument Inspired by the Cayley-Dickson Form // Advances in Applied Clifford Algebras, 2010. – pp. 111-120.
4. Swartzlander E.E. The Sign/Logarithm Number System // IEEE Transactions on Computers, 1975. – pp. 1238-1242.

# МЕТОДИКА ПОИСКА ОПТИМАЛЬНОГО МАРШРУТА НА ОСНОВЕ АНАЛИЗА ДАННЫХ ОТ КАРТОГРАФИЧЕСКИХ СЕРВИСОВ

Соловцов В.В.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Одинец Д.Н. – к.т.н., доцент

В настоящее время программные средства для навигации имеют большую популярность. Существующие навигационные решения предлагают следующие оптимизации поиска маршрута: быстрый маршрут и кратчайший маршрут [1,2]. В данной работе будет предложен алгоритм оптимизации маршрутов по ресурсам, которые должны быть затрачены для преодоления маршрута.

Постановка задачи: требуется найти маршрут между двумя точками (маршрут также может включать некоторое количество промежуточных точек), на прохождение которого должно быть затрачено наименьшее количество ресурсов.

Будем исходить из того, что на подъем расходуется больше ресурсов, чем на спуск, тогда наименее затратный маршрут — маршрут с наименьшим количеством подъемов. Идея состоит в том, чтобы каждому маршруту между каждыми соседними двумя точками, присвоить вес. Для этого маршрут разбивается на множество участков, для каждого участка известна длина  $L$ , высота начальной точки  $H_{start}$  и высота конечной точки участка  $H_{end}$ .

Введем понятие *функционала энергии*:

$$E = f(\alpha) \quad (1)$$

где  $\alpha$  — это угол наклона участка маршрута. Угол может быть посчитан по формуле (2):

$$\alpha = \sin \frac{H_{end} - H_{start}}{L} \quad (2)$$

Введем понятие *функционала затрат*:

$$b = f(E, L) \quad (3)$$

где  $E$  — введенный ранее функционал затрат,  $L$  — длина участка.

Значение веса для всего маршрута будет высчитываться как сумма значений функционала затрат для каждого участка:

$$B = \sum_{i=1}^n b_i \quad (4)$$

Веса рассчитываются для каждого маршрута. Таким образом, каждому маршруту будет соответствовать свой суммарный вес, пропорциональный затратам на его прохождение. Искомый маршрут — маршрут, состоящий из соответствующих частей с наименьшими значениями рассчитанных весов.

В работе в качестве картографических сервисов использован сервис Google Maps, а именно Google Maps Direction API [3] для получения маршрутов и Google Maps Elevation API [4] для получения высот точек маршрута.

Работа алгоритма была проверена как для гористой и холмистой местности, так и для равнинной местности. По анализу проведенных испытаний можно сделать вывод, что результат работы алгоритма лучше, чем аналогичное решение от Google Maps для холмистой и гористой местности. Для равнинной местности итог работы алгоритма совпадает с результатами, которые получены от Google Maps.

Для холмистой и гористой местности описанный алгоритм выбирает маршруты, отличающиеся от Google Maps меньшим содержанием количества подъемов, что соответственно делает маршруты менее энергозатратными.

## Список использованных источников:

1. Тарасян В.С., Полушкин А.Я. ОПТИМИЗАЦИЯ ПУТИ В НЕОДНОРОДНОЙ СРЕДЕ // Фундаментальные исследования. [Электронный ресурс]. – 2017. – Режим доступа: <http://www.fundamental-research.ru/ru/article/view?id=41828> – Дата доступа: 27.02.2019.
2. Improving Operations with Route Optimization – Towards Data Science [Электронный ресурс]. – 2018. – Режим доступа: <https://towardsdatascience.com/improving-operations-with-route-optimization-4b8a3701ca39> – Дата доступа: 27.02.2019.
3. Developer Guide | Directions API | Google Developers [Электронный ресурс]. – 2019. – Режим доступа: <https://developers.google.com/maps/documentation/directions/intro> – Дата доступа: 27.02.2019.
4. Developer Guide | Elevation API | Google Developers [Электронный ресурс]. – 2019. – Режим доступа: <https://developers.google.com/maps/documentation/elevation/intro> – Дата доступа: 27.02.2019.

## СОВРЕМЕННЫЕ АЛГОРИТМЫ МОДЕЛИРОВАНИЯ АМОРФНЫХ ОБЪЕКТОВ

Стаховский А.В.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Кукин Д.П. – к.т.н., доцент

Реалистичная симуляция огня играет огромную роль в моделировании чрезвычайных ситуаций, моделировании боевых действий, создании спецэффектов для фильмов и компьютерных игр. Огонь является довольно сложным объектом для симуляции и рендеринга из-за его нечетких границ и постоянного изменения. Современные алгоритмы симуляции огня можно разделить на несколько групп, отличающихся по интерактивности, конфигурируемости, степени реалистичности, применимости в реальном времени и пространственно-временной сложности.

Аморфные объекты, такие как огонь, дым, вода, туман вызывают определенную сложность при симуляции и рендеринге из-за того, что эти объекты не имеют четко выраженных границ и крайне подвижны. Они являются субъектами постоянных исследований и разработок, т. к. представляют коммерческую ценность для кинематографа, индустрии видеоигр, силовых ведомств[1]. Особую сложность представляет компьютерная графика реального времени, где эффекты должны рассчитываться за время частоты кадра (60+ кадров/с).

Современные методы симуляции огня можно разделить на следующие основные группы: текстурный маппинг, системы частиц, физико-математические методы, клеточные автоматы, методы томографической реконструкции. Физико-математические методы позволяют достичь высокой реалистичности огня, однако имеют высокую пространственно-временную сложность. Поэтому данные методы невозможно было широко применять до недавнего времени для моделирования в реальном времени.

Стремительный рост производительности графических адаптеров в течение последнего десятилетия привел к росту исследований в области воксельного рендеринга. Воксели являются аналогами пикселей для трехмерного пространства. Воксельный рендеринг предоставляет широкие возможности для реалистичного моделирования объемных объектов, однако требует больших затрат аппаратных ресурсов, поэтому его применение в рендеринге реального времени крайне ограничено. Возросшая производительность графических адаптеров активизировала исследования в реалистичном моделировании различных аморфных объектов: дыма, огня, облаков.

Особый интерес представляет совместное использование воксельного рендеринга и методов физико-математической симуляции для создания крайне реалистичных спецэффектов. В данном направлении сейчас ведется множество исследований, направленных на создание алгоритмов, позволяющих создавать максимально реалистичное изображение за время смены кадра.



Рисунок 1 – Огонь на объемном столе. Металлический каркас открывается после сожжения верхнего деревянного слоя [2]

**Список использованных источников:**

1. ZhaoHui, W. Realistic fire simulation: A Survey / W. ZhaoHui, Z. Zhong, W. Wei // 12th International Conference on Computer-Aided Design and Computer Graphics – 2011 – P. 333–340.
2. Zhao, Y. Voxels on fire / Y. Zhao, X. Wei, Z. Fan, A. Kaufman, H. Quin – Center for Visual Computing and Department of Computer Science Stony Brook University Stony Brook, NY 11794-4400.
3. Vanzine, Y. Real-Time Volumetric Rendering of Fire in a Production System: Feasibility Study / Y. Vanzine. – Indiana University South Bend, 2007. – 88 P.

## СРЕДСТВА ЗАЩИТЫ ДАННЫХ ПРИ ОБЛАЧНЫХ ВЫЧИСЛЕНИЯХ

Стиняева В.М.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Ганжа В.А. – к.т.н., доцент

Управление, а также контроль над облаками — является проблемой безопасности и сохранности данных. Гарантий, что все ресурсы облака используются по назначению и в нем нет неуправляемых виртуальных машин, не запущено лишних процессов и не нарушена взаимная конфигурация элементов облака нет. Поэтому рассмотрим основные и актуальные способы защиты данных при облачных вычислениях.

Для контроля безопасности в облаках необходим комплексный подход в обеспечении как физической, так и сетевой безопасности. В основе обеспечения физической безопасности лежит строгий контроль физического доступа к серверам и сетевой инфраструктуре. А сетевая безопасность представляет собой построение надежной модели угроз, включающей в себя защиту от вторжений и межсетевой экран. Использование межсетевого экрана подразумевает работу фильтра, с целью разграничить внутренние сети центра обработки данных (далее – ЦОД) на подсети с разным уровнем доверия.

Наиболее эффективными способами обеспечения безопасности при облачных вычислениях являются:

- 1) шифрование – один из эффективнейших способов защиты данных. Провайдер, предоставляющий доступ к данным, шифрует данные клиента, хранящиеся в ЦОД, а также в случае отсутствия необходимости, безвозвратно удаляет их. Таким образом, облако всегда имеет дело только с зашифрованными данными.

Выделим плюсы шифрования данных:

- провайдер не сможет получить доступ к данным клиента;
- никакой элемент на пути следования трафика не получит доступ к данным клиента (например владелец wi-fi точки какого-либо заведения).

Главным условием с целью предоставления защищенности подобного решения считается отдельное использование облачного сервера и сервера управления ключами (рисунок 1);

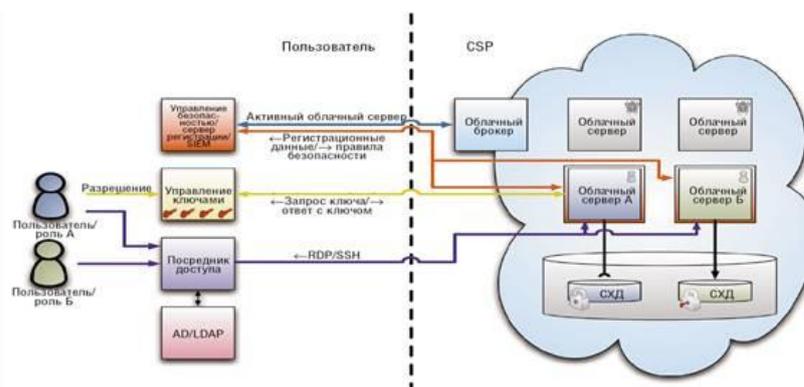


Рисунок 1 - Схема взаимодействия пользователя, сервера управления ключами и облачного сервера

- 2) защита данных при передаче. Зашифрованные данные при передаче в облако и при хранении в нем должны быть доступны только после аутентификации. Это позволит обеспечивать конфиденциальность данных даже если кто-то получит к ним доступ посредством ненадежных узлов в сети. Применяются такие алгоритмы и протоколы как AES, TLS, IPsec. Также для публичных облаков используется VPN-туннель, основанный на двухфакторной аутентификации;
- 3) аутентификация — процесс подтверждения подлинности субъекта. Для более высокой надежности прибегают к использованию сертификатов, токенов и двухфакторной аутентификации. Для прозрачного взаимодействия провайдера с системой идентификации при авторизации, также рекомендуется использовать стандарты LDAP (Lightweight Directory Access Protocol) и SAML (Security Assertion Markup Language);
- 4) изоляция пользователей. Самым актуальным и надежным способом изоляции пользователей является использование индивидуальной виртуальной машины и виртуальную сеть. Виртуальные сети должны быть развернуты с применением таких

технологий, как VPN (Virtual Private Network), VLAN (Virtual Local Area Network) и VPLS (Virtual Private LAN Service);

Таким образом можно подытожить, что обеспечение информационной безопасности при использовании облачных вычислениях зависит и от клиента, и от провайдера. Следует подойти серьезно к составлению договора с поставщиком облачных услуг и учесть все факторы, которые могут повлиять на обеспечение конфиденциальности данных. Со стороны клиента - это использование современных программных и аппаратных средств, работа профессионально подготовленного персонала, обеспечение безопасности сетевого оборудования, хранилища данных, сервера и гипервизора. Дополнительно возможно размещать в выделенном ядре антивирус для предотвращения заражения гипервизора через виртуальную машину, систему шифрования данных для хранения пользовательской информации в зашифрованном виде и средства для реализации зашифрованного тунелирования между виртуальным сервером и клиентской машиной. Также проводить своевременный аудит всех систем на выявление уязвимостей и контроль уровней доступа пользователей.

**Список использованных источников:**

1. Котяшичев И. А., Бырылова Е. А. Защита информации в «Облачных технологиях» как предмет национальной безопасности // Молодой ученый. — 2015. — №6.4. — С. 30-34. — URL <https://moluch.ru/archive/86/16357/> (дата обращения: 24.03.2019).
2. Удо Шнайдер Использование облачных сервисов [Текст] / У. Шнайдер //
3. Информационная безопасность в облачных вычислениях: уязвимости, методы и средства защиты, инструменты для проведения аудита и расследования инцидентов URL: <https://www.bibliofond.ru/view.aspx?id=783809> (дата обращения: 24.03.2019)
4. Арзамасов Е.В., Чурикова А.А. Угрозы облачных вычислений и методы их защиты // Технические и математические науки. Студенческий научный форум: электр. сб. ст. по мат. V междунар. студ. науч.-практ. конф. № 5(5). URL: [https://nauchforum.ru/archive/SNF\\_tech/5\(5\).pdf](https://nauchforum.ru/archive/SNF_tech/5(5).pdf)

## **ИДЕНТИФИКАЦИЯ ПОЛЬЗОВАТЕЛЯ МЕТОДОМ АНАЛИЗА ДИНАМИКИ ВВОДА ТЕКСТА**

*Супринович И.Ю.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Искра Н.А. – старший преподаватель*

В данной работе рассматриваются алгоритмы идентификации пользователя методом анализа динамики ввода текста. Помимо этого рассматривается возможность и сценарии применения этого метода идентификации, а также изучаются способы классификации, которые позволят сделать идентификацию пользователя более точной.

Потребность в распознавании людей по каким-либо уникальным, присущим только им признакам, возникла достаточно давно. Эту проблему призвана решить аутентификация — процесс идентификации и подтверждения личности пользователя. Обычно он сопровождается авторизацией, которая является механизмом определения того, какими правами наделяется пользователь после идентификации.

Аутентификация классифицируется следующим образом:

- 1) аутентификация на основе уникальной информации — для успешной идентификации пользователю необходимо ввести пароль;
- 2) аутентификация на основе уникального предмета — для успешной идентификации пользователю необходимо предъявить уникальный предмет, например пластиковую карту;
- 3) аутентификация на основе какой-либо уникальной характеристики пользователя, физиологической или поведенческой (биометрия).

Одним из представителей поведенческой биометрии является аутентификация методом анализа динамики ввода текста (АМАДВТ) — распознавание пользователя с помощью анализа динамики ввода текстовых последовательностей. Данный метод считается достаточно надёжным, так как очень сложно изучить темп и стиль ввода другого человека. Также он не требует какого-либо специального оборудования в отличие от методов физиологической биометрии.

В основе АМАДВТ лежат алгоритмы выявления аномалий — алгоритмы, позволяющие выделить необычные значения в выборке данных. От выбора алгоритма зависит точность распознавания пользователя. Наиболее часто используемыми метриками для выявления аномалий являются стандартное расстояние Махаланобиса, а также его разновидности — фильтрованное и масштабированное.

Большинство алгоритмов АМАДВТ состоят из следующих этапов:

- 1) сбор данных — собираются данные о времени нажатия и отпускания клавиш во время ввода пользователем пароля;
- 2) создание модели пользователя — преобразование данных, полученных на первом этапе, в модель с параметрами HoldTime (время удержания клавиши), DownDownTime (время между последовательными нажатиями двух клавиш) и UpDownTime (время между отпусканием клавиши N и нажатием клавиши N + 1). В течение некоторого времени эти данные накапливаются, после чего их можно использовать;
- 3) расчёт схожести полученной выборки и идеальной — расчёт расстояния между моделью пользователя и полученной на этапе ввода выборкой. Чем меньшее значение получено, тем более похожими являются выборки;
- 4) анализ полученных результатов — расчёт вероятности того, что данные были введены истинным пользователем.

Для улучшения распознавания пользователя можно использовать несколько моделей пользователя, которые отличаются временем сбора данных (темп набора пользователя может изменяться в зависимости от времени суток).

У АМАДВТ есть серьёзный недостаток: его точность напрямую зависит от удобства клавиатуры (эргономика, ход клавиш, размеры), то есть при смене пользователем устройства ввода необходимо обучить систему аутентификации заново. Также стоит отметить тот факт, что модель пользователя может быть применена только для того устройства, на котором она создавалась, поэтому АМАДВТ желательно использовать в качестве дополнительного уровня идентификации.

Таким образом, аутентификация методом анализа динамики ввода текста является достаточно хорошим способом аутентификации, хоть и не лишённым недостатков.

**Список использованных источников:**

1. А.Г. Дьяконов, А.М. Головина — Выявление аномалий в работе механизмов методами машинного обучения — 2017. — С.389-396.
2. Dr. Abdulameer Khalaf Hussain, Mustafa Nouman Al-Hassan — Multifactor Strong Authentication Method Using Keystroke Dynamics — 2013. — Vol.2, No.2, — С.31-34.

## ТМОФ УЯЗВИМОСТЬ

*Трафимук М.П.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Селезнев И.Л. – к.т.н., доцент*

В современном мире все большее количество услуг и сервисов предоставляется посредством сети Интернет, поскольку Интернет и, соответственно, все его ресурсы доступны повсеместно. Помимо описанных положительных сторон, есть и отрицательные: доступность, стабильность и надежность зависят не от пользователя, а от владельца ресурса. Обеспечить максимальную стабильность – это первостепенная задача администратора ресурса, для выполнения которой необходимо знать возможные уязвимости и способы их устранения. Не менее важно помнить и про известные проблемы, которые не считаются уязвимостями, вроде ТМОФ – too many open files.

С течением времени все большее количество услуг и сервисов предоставляется посредством сети Интернет. Повсеместное распространение доступа к сети и ее ресурсам является основной причиной этого явления. Помимо положительных сторон, у этого явления есть и отрицательные моменты: доступность, стабильность и надежность зависят не от пользователя, а от владельца ресурса. Обеспечить максимальную стабильность – это первостепенная задача администратора ресурса, для выполнения которой необходимо знать возможные уязвимости и способы их устранения.

На сегодняшний день ключевые ресурсы сети Интернет являются достаточно устойчивыми к различного вида атакам, использующим уязвимости протоколов UDP и TCP, а также DNS-серверов и других широко используемых сущностей. Ранее использование недоработок в этих компонентах для реализации атак приводило к неприятным последствиям для владельцев и администраторов ресурсов сети Интернет. Ключевой идеей этих, устаревших на данный момент, атак является генерация и отправка больших потоков данных атакуемому ресурсу. Некоторые современные атаки используют схожие механизмы, но избегают условий, в которых воздействие считается атакой и нейтрализуется существующими методами противодействия вроде «blackholing».

Современные атаки используют узкопрофильные уязвимости, которые, тем не менее, могут покрывать достаточно большой процент ресурсов сети Интернет. Например, уязвимостям, направленным на подсистемы серверов на базе Unix, подвержены свыше 40% всех серверов сети

Интернет (суммарное количество серверов на базе Apache и Nginx по данным декабря 2018) [1]. Одна из таких уязвимостей использует Pluggable Authentication Modules (PAM, подключаемые модули аутентификации), поставляемый практически во всех версиях операционных систем на базе Unix, и имеет название Too Many Open Files exploit (ТМОФ, «слишком много открытых файлов»).

Уязвимость заключается в следующем. Каждое подключение создает в PAM 2 файловых дескриптора: один – для передачи данных, второй – для чтения передаваемого файла. Максимальное количество существующих дескрипторов по умолчанию является 1024, как правило, при повышении нагрузки вручную увеличивается до 8000-16000, что достаточно для обслуживания крупного потока пользователей. При достижении этого лимита сервер перестает обслуживать новые подключения до освобождения ресурсов, добавляя в системный журнал сервера ошибку “Too Many Open Files (24)”.

Алгоритм атаки следующий:

1. создать сокет,
2. подключиться к серверу,
3. закрыть сокет на прием данных,
4. отправить пакет серверу,
5. уничтожить сокет.

Таким образом, на сервере создается как минимум 1 файловый дескриптор, который не уничтожится до истечения таймаута подключения. Средний современный компьютер способен сгенерировать и отправить 23000-45000 пакетов в секунду, в результате перегрузив целевой ресурс за несколько секунд. Как правило, сервер настроен таким образом, что при определенных критических ошибках или критическом количестве ошибок он останавливает свою работу и требует перезагрузки и обслуживания администратором, что и происходит в данном случае. Например, для внешнего наблюдателя сервер на любой HTTP запрос будет отправлять код ответа из 500 серии, как правило, код 503 или, в редких случаях, 500. На момент написания данной статьи ошибка Too Many Open Files является достаточно известной, однако использование ее в качестве уязвимости не имеет упоминаний на тематический форумах и новостных лентах, что делает ее достаточно опасной. Нейтрализацией данной уязвимости будет настройка роутера таким образом, чтобы более 2-3 подключений с одного IP-адреса блокировались, что приемлемо далеко не для всех.

**Список использованных источников:**

1. December 2018 Web Server Survey [Электронный ресурс]. – Режим доступа: <https://news.netcraft.com/archives/2018/12/17/>

## **ЗАЩИТА ПЕРЕДАВАЕМОЙ ИНФОРМАЦИИ В АВТОМОБИЛЬНОЙ СИГНАЛИЗАЦИИ**

*Турок М.А.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Сечко Г.В. – к.т.н., доцент*

Решается задача защиты информации в микроконтроллерах автосигнализации мобильных объектов, причём под защитой понимается противодействие нарушению целостности (несанкционированному изменению, искажению, уничтожению информации) и противодействие считыванию.

Современная автосигнализация мобильных объектов, и в первую очередь транспортных средств, реализуется на микроконтроллерах. В каждом микроконтроллере заложена своя программа, в соответствии с которой контроллер управляет каким-либо устройством, выдавая управляющие сигналы. На разработку данного программного обеспечения производители тратят большие средства и время. В этих условиях актуальной является задача защиты информации в микроконтроллерах автосигнализации мобильных объектов, причём под защитой понимается противодействие нарушению целостности (несанкционированному изменению, искажению, уничтожению информации) [1] и противодействие считыванию. Поскольку устройства автосигнализации используют радиоканал передачи данных для взаимодействия между центральным блоком и пультом дистанционного управления, то параллельно с защитой необходимо повысить помехоустойчивости канала связи между данными устройствами.

На современном рынке существует три вида автосигнализаций: статические, динамические и диалоговые. Так как основную долю рынка занимают динамические автосигнализации, то было принято решение защищать информацию именно в них. Для понимания сути работы алгоритма введем следующие понятия: протокол передачи данных, посылка, сообщение. Сигнал в радиоэфир

передается по определенному правилу. В общем случае протокол передачи данных состоит из преамбулы (специфический набор данных дающий понять принимаемому устройству, что принимаемая информация есть искомая), и информационной части (информация, которая представляет команду для устройства управления). Посылка представляет собой команду, передаваемую устройству управления автосигнализацией (рисунок 1).

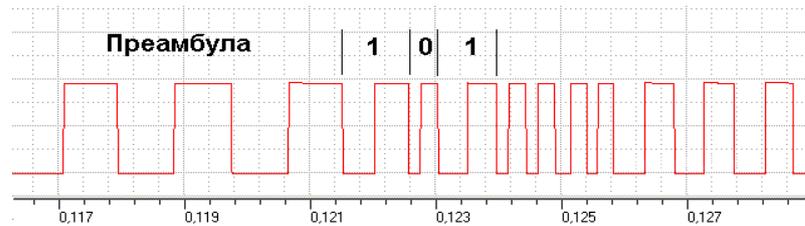


Рисунок 1 – Пример посылки сигнализации StarLine

Сообщение – это совокупность посылок (как минимум одна), которые идентичны друг другу. Суть алгоритма перехвата такова, что после подачи в радиозфир сигнала, предназначенного для управления автосигнализацией, устройство перехвата, анализируя передаваемый сигнал, выдает в эфир радиопомеху, которая ставится в определенном месте информационной части посылки, и искажает данную команду [2]. Само устройство перехвата в данный момент времени запоминает неискаженную часть посылки. Данное действие не позволяет устройству управления автосигнализацией воспринимать передаваемую команду адекватно. Так как команда (посылка) в радиозфир передается неоднократно, то это позволяет устройству перехвата одновременно принимать передающуюся посылку, выставлять помеху и запоминать искаженную команду, однако помеха ставится в месте, отличном от предыдущего, что дает возможность устройству перехвата восстановить истинную команду путем сложения неискаженных частей.

Предложенный алгоритм учитывает очистку передаваемых автосигнализацией сигналов от случайных помех в радиозфире, возникающих вблизи крупных промышленных объектов, которых в густонаселенных городах достаточно много.

**Список использованных источников:**

1. Закон Республики Беларусь от 10 ноября 2008 г. No 455-3 «Об информации, информатизации и защите информации» / Нац. реестр правовых актов Респ. Беларусь. – No 2/1552 (зарегистрировано 17 ноября 2008 г.).
2. Карпушкин, Э. М. Радиосистемы передачи информации / Уч. метод. пособие для студентов учреждений, обеспечивающих получение высшего образования по спец. "Радиоэлектронные системы". – Минск: БГУИР, 2008. – 62 с.

## СИСТЕМЫ УПРАВЛЕНИЯ ПРОЦЕССАМИ ПРОЕКТИРОВАНИЯ ПРОГРАММ

Ульянко В.Г.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Сечко Г.В. – к.т.н., доцент

В настоящее время системы управления процессами проектирования программ актуальны, востребованы и должны решать широкий диапазон задач. Уже существует ряд систем такого рода, однако у всех есть свои недостатки: от недостаточной глубины функциональности до чрезмерно высокой цены. Для того, чтобы превзойти аналоги, необходимо реализовать функциональность автоматического анализа проекта, с возможным автоматическим решением проблем.

В первую очередь стоит определить, что из себя представляет управление проектами. Основой успеха проектного управления есть наличие конкретного, заранее подготовленного набора действий для уменьшения рисков и ответвлений от начального плана [1].

Исходя из этого, можно выделить определение для систем управления процессами проектирования программ – это набор программных продуктов, состоящий из инструментов планирования задач, подготовки расписания, управления ценой и денежными средствами, распределения ресурсов, организации работы, а также из инструментов управления администрированием системы и документированием разрабатываемого продукта [2].

Для того, чтобы лучше понять значение и функции систем данного типа, стоит обратить внимание на решаемые задачи, которые можно разделить на 3 типа.

Одним из наиболее востребованных типов является создание плана событий и управление задачами. От того, как используется инструмент на практике требования могут быть различны. К задачам этого типа можно отнести:

- организация связанных друг от друга различных событий;
- определение больших составных проектных частей и их декомпозиция, посредством которой создается структура декомпозиции работ;
- формирование расписания для работы коллектива и определение людей на определенные задачи;
- подсчет временных затрат, необходимых для завершения каждой из задач;
- распределение задач по срокам их завершения;
- визуализация графика работ в виде диаграммы Ганта;
- одновременное контролирование несколькими проектами [3, 4].

В первую очередь, стоит классифицировать СУППП по месту их расположения. ПО может быть установлено на компьютере конечного пользователя (настольное приложение). Такой вариант создает наиболее податливый интерфейс. Эти проекты могут создавать файл с информацией, которая в будущем может быть выложена для других пользователей в общий доступ или же могут быть сохранены в централизованной базе данных (БД) [5].

В последнее время наиболее популярным являются приложения, ориентированные на веб-направление (рисунок 1), доступ к которым предоставляется с помощью браузера. К положительным моментам такого подхода можно отнести:

- доступ может быть получен с различных компьютеров, которые не требуют наличия дополнительного программного обеспечения;
- не сложный контроль за доступом;
- многопользовательский протокол общения;
- серверная программа, которая удаленно работает в единственном экземпляре;
- большинство таких приложений доступны с мобильных устройств.

Однако имеются и несколько недостатков:

- скорость общения с приложением значительно меньше чем с обычными приложениями;
- проблемы с соединением, а соответственно и к доступу серверу или его отказу ведут к полной невозможности получения или отправки информации [6].

Веб-приложения действительно более удобны для разработчиков, так как они могут полностью контролировать все процессы, происходящие на серверах, что позволяет не только избегать дополнительных ошибок, но и быстрее исправлять существующие [7].

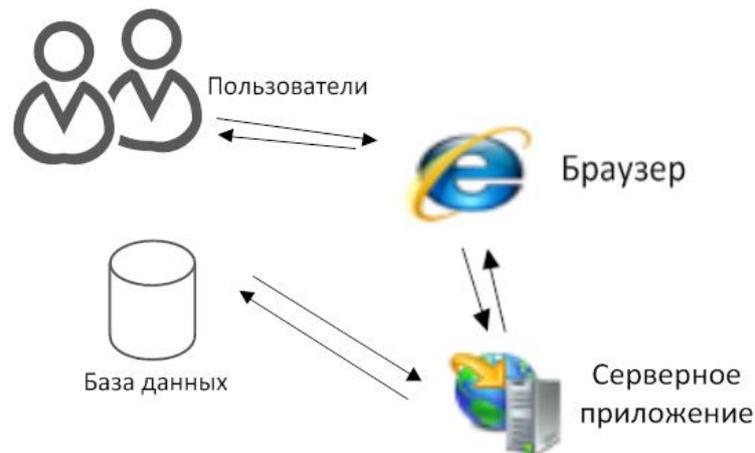


Рисунок 1 – Пример работы веб-приложения

Классифицировать СУППП можно также по целевой аудитории:

- личные СУППП, которые предназначены для организации мелких и локальных проектов с небольшим количеством людей в команде;
- однопользовательские СУППП для управления компаниями с небольшим штатом сотрудников;
- многопользовательские СУППП, которые служат для организации действий большого количества пользователей (зачастую СУППП такого типа имеют клиент-серверную архитектуру).

Естественно, что наиболее востребованными являются многопользовательские системы, так как они решают наибольшее количество задач и имеют самую большую сферу применения. Такие

системы и более выгодны разработчикам, ведь угодить каждому пользователю персонально сложнее, чем потребностям руководства или сотрудников крупной компании [8].

Одним из способов анализа и графического представления является диаграмма Ганта — класс столбчатых диаграмм, которыми пользуются для визуализации плана, а также графика работ по проекту.

В общем, диаграмма Ганта состоит из линий, направленных вдоль оси времени. Каждая линия на диаграмме визуализирует определенную задачу в составе проекта (вид ее работы), её начало и конец — моменты старта и завершения задачи, её длина — время работы. Вертикальной осью диаграммы служит набор задач.

Как заключение, на рынке уже существуют такого рода системы, имеющие широкий набор возможностей, но у них есть недостатки. Чтобы конкурировать с ними при создании новой системы, необходимо создать систему с учётом достоинств и недостатков предшественников. Необходимо взять всё лучшее и добавить некоторую уникальную функциональность и особенности.

**Список использованных источников:**

1. Швоулб, К. Информационные технологии в управлении проектами / К. Швоулб. — Cengage Learning, 2013. — 672 с.
2. Ависон, Д. Информационные системы управления проектами / Д. Ависон. — Sage Publications, 2009. — 496 с.
3. Киризоглу, Дж. Как улучшить производительность компании / Дж. Киризоглу. — Sage Publications, 2009. — 55 с.
4. Керзнер, Г. Управление проектами: системный подход к планированию, формированию графика и управлению / Г. Керзнер. — Wiley, 2009. — 1120 с.
5. Матвеев, А.А. Модели и методы управления портфелями проектов / А.А. Матвеев, Д.А. Новиков, А.В. Цветов — М.: ПМСОФТ, 2005. — 206 с.
6. Баркалов, П.С. Задачи распределения ресурсов в управлении проектами / П.С. Баркалов, И.В. Буркова, А.В. Глаголев — Москва: ИПУ РАН, 2002. - 65 с.
7. Васильев, Д.К. Типовые решения в управлении проектами / Д.К. Васильев, А.Ю. Заложнев, Д.А. Новиков — М.: ИПУ РАН (научное издание), 2003. 75 с.
8. Хэллоус, Дж. Информационные системы управления проектами / Дж. Хэллоус. — Amacom Div American Mgmt Assn, 2005. — 304 с.

## ОСНОВНЫЕ ПОДХОДЫ ОПРЕДЕЛЕНИЯ НЕВНИМАТЕЛЬНОСТИ ВОДИТЕЛЯ ЗА РУЛЕМ

*Фоменок В.В.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Лукашевич М.М. — к.т.н., доцент*

Статистика дорожно-транспортных происшествий, связанных с невнимательностью водителя, неутешительна, так как все больше и больше инцидентов происходит по этой причине. Эта статистика может стать еще более тревожной, так как количество возможных отвлекающих факторов внутри автомобиля продолжает расти. Большое количество дисплеев и новых информационно-развлекательных устройств в автомобилях проблема более критичная, чем кажется на первый взгляд. Именно поэтому очень важно, чтобы автомобили были оснащены системами, которые смогут определять степень внимательности водителя и предупреждать об опасности заранее.

В настоящее время существует несколько подходов для определения степени внимательности водителя на дороге. Первый из них - физиологические датчики. Такой подход обнаруживает физиологические особенности такие как активность мозга, частота сердцебиения или влажность рук. В частности, электроэнцефалограф был признан действительной, объективной и точной мерой измерения невнимательности водителя с использованием  $\delta$ ,  $\theta$ ,  $\alpha$ ,  $\beta$  и  $\sigma$  мозговой волновой активности. Однако, физиологические датчики не удобны и не могут быть использованы внутри автомобиля для коммерческих приложений. Их можно использовать как основу для исследований, но они не представляют собой реалистичное решение для контроля невнимательности водителя.

Второй подход - производительность водителя. Данный подход использует внешнюю информацию (положение автомобиля, движения рулевого колеса или педали) и показатели производительности водителя, чтобы сделать вывод об уровне невнимательности. В большинстве случаев, отвлечение внимания водителя связано с отсутствием управления транспортным средством, таким как дрейф со стороны дороги, или неожиданное изменение скорости. В 2007 году компания Volvo представила систему управления оповещениями водителя, постоянно отслеживающую дорогу с помощью камеры и предупреждающую водителя в случае опасного поведения [1]. Эти методы взаимосвязаны с невнимательностью водителя, но они также связаны с внешними факторами, такие как опыт водителя, тип дороги, погода и уличный свет. Кроме того, эти меры основаны на

долгосрочной статистике, и система не может предсказать непосредственные опасности, например, такие как краткосрочный сон.

Третий подход - компьютерное зрение. Он, вероятно, самый популярный и опирается на визуальное состояние водителя. Когда водитель невнимателен, лицо и тело демонстрируют характерное поведение. Установка камеры перед водителем и анализ выражения лица и движений рассматривается как эффективный способ оценки невнимательности водителя. В частности, продолжительность взгляда, направленного не на дорогу или продолжительности поворота головы в сторону от дороги являются действительными показателями невнимательности водителя и могут быть легко оценены с помощью камеры. Для определения усталости водителя, наилучшими показателями являются продолжительность времени закрытых глаз водителя, кивков головой или зевоты. Существующие системы обычно используют простые цветные камеры или инфракрасные камеры для захвата изображения. После этого инструменты обработки изображений и компьютерного зрения определяют основные компоненты на изображении, такие как лицо, глаза и рот. Это отправная точка для извлечения признаков невнимательности, таких как зевота, кивок, направление взгляда, поворот головы, закрытые глаза. После извлечения признаков полученная информация объединяется и определяется уровень невнимательности. Наиболее популярными методами объединения являются нечеткая логика, байесовские сети, динамические байесовские сети, нейронные сети или простые правила принятия решений. Если уровень внимательности недостаточный, подается предупреждение для предотвращения опасной ситуации.

Гибридные системы используют комбинацию трех вышеупомянутых методов. Существуют системы, которые объединяют камеру, установленную перед водителем, и дорожную информацию, фиксируется взгляд водителя чтобы обнаружить, пропустил ли он какой-либо дорожный знак [2]. Еще один пример, объединение компьютерного зрения, сигнала рулевого колеса и пульсоксиметр для определения усталости водителя.

**Список использованных источников:**

1. Driver alert control [Электронный ресурс]. - Режим доступа: <https://www.media.volvocars.com/global/en-gb>
2. L. Fletcher, G. Loy, N. Barnes, and A. Zelinsky, Correlating driver 11 gaze with the road scene for driver assistance systems, Robotics and Autonomous Systems: [http://www.nada.kth.se/~gareth/homepage/local\\_site/papers/Fletcher\\_RAS05.pdf](http://www.nada.kth.se/~gareth/homepage/local_site/papers/Fletcher_RAS05.pdf): 24 мая 2005 г.

## ПЕРЕНОС СТИЛЯ МЕЖДУ ИЗОБРАЖЕНИЯМИ

*Царикович А.А.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Фролов И.И. – к.т.н., доцент*

В работе рассматривается способ переноса стиля с одного изображения на другое при помощи сверточных нейронных сетей. Описана архитектура первой нейронной сети, которая была использована для решения данной задачи. Также рассматриваются вспомогательные средства и решения, позволяющие существенно упростить разработку решений данной задачи.

Задача переноса стиля сводится к работе над двумя изображениями:

- входное изображение, на которое необходимо перенести стиль и сохранить контент изображения (фотография);
- изображение с ярко выраженной стилистикой, которую необходимо для переноса на входное изображение.

Чтобы перенести стиль одного изображения на другое, необходимо синтезировать новое изображение, которое одновременно будет соответствовать представлению содержимого входного изображения и представлению стиля изображения-картины.

Поставленная задача решается при помощи сверточных нейронных сетей. Данный тип нейронных сетей используется по причине своей главной особенности – сеть «ищет» на изображении заданные ей признаки. Признаком является та или иная особенность исследуемой картинки, например острые углы, определенные размытости и т.д. Также сеть является универсальной и обладает высокой точностью работы.

Типичная архитектура сверточной нейронной сети представляет собой последовательно соединенные блоки субдискретизирующего и сверточных слоев. Основными функциями активации являются:

$$- \text{softmax } y_i = \frac{e^{z_i}}{\sum_{j=1}^n e^{z_j}};$$

- $\text{ReLU}(x) = \max(0, x)$ ;
- сигмоидная функция  $\sigma(x) = \frac{1}{1+e^{-x}}$ .

Для обучения нейронной сети используется метод обратного распространения ошибки.

В работе [1] предложено использование сети архитектуры VGG для получения контента изображения. Было использовано 16 сверточных, 5 слоев субдискретизации. В данной работе было описано, что выходы сверточных слоев VGG содержат в себе достаточно детальную информацию о контенте изображения. Функция потерь, отвечающая за сохранение контента изображения, выглядит (1) следующим образом:

$$L_l^{\text{content}}(x_{\text{content}}, y) = \frac{1}{c_l H_l W_l} \|F_l(x_{\text{content}}) - F_l(y)\|_F^2 \quad (1)$$

Функция потерь, которая отвечает за перенос стиля  $x_{\text{style}}$  определяется (2):

$$L_l^{\text{style}}(x_{\text{style}}, y) = \frac{1}{c_l^2} \|G_l(x_{\text{style}}) - G_l(y)\|_F^2 \quad (2)$$

Для подсчета  $L_l^{\text{content}}(x_{\text{content}}, y)$  используется лишь один слой VGG, для  $L_l^{\text{style}}(x_{\text{style}}, y)$  подсчета – разные слои с подобранными весами [2].

Обобщенная функция потерь будет выглядеть следующим образом (3):

$$L_l^{\text{total}}(y, x_{\text{content}}, x_{\text{style}}) = \sum_l \beta_l^{\text{content}} L_l^{\text{content}}(x_{\text{content}}, y) + \alpha \sum_l \beta_l^{\text{style}} L_l^{\text{style}}(x_{\text{style}}, y) \quad (3)$$

В любом случае, невозможно синтезировать изображение, которое на 100% удовлетворяло бы и стилю, и содержанию двух разных изображений. Однако минимизируя одну из составляющих функции потерь, которая представляет собой комбинацию функций потерь контента и стиля можно плавно регулировать акцент на переносе стиля либо на сохранении контента изображения.

На данный момент существует определенное количество программных решений, которые позволяют существенно упростить задачу разработки решений переноса стиля изображения:

- 1) OpenCV;
- 2) Tensorflow;
- 3) Pytorch;
- 4) Caffe.

**Список использованных источников:**

1. Leon A. Gatys, Alexander S. Ecker, Matthias Bethge. Image Style Transfer Using Convolutional Neural Networks, 2016.
2. Николаев С.В. Совершенствование метода автоматического переноса стиля изображений, Москва 2018.

## ПРИМЕНЕНИЕ НЕЙРОННЫХ СЕТЕЙ ДЛЯ РЕДАКТИРОВАНИЯ ИЗОБРАЖЕНИЙ

*Черноокий Р.С.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Насуро Е.В. – к.т.н., доцент*

Редактирование изображений является сложным процессом, однако существуют методы для упрощения автоматизации этого действия. Одним из них являются нейронные сети. Их особенностью является возможность редактировать изображения согласно их содержанию, опираясь на образы и объекты, увиденные на изображении. Это позволяет проводить редактирование не слепо, а осознанно.

Из всех способов редактирование изображений нейронные сети являются наиболее актуальным, они позволяют проводить его без затрат времени и сил художников, но при этом результат их работы в частном случае может быть непредсказуем. У нейронных сетей есть два основных применения в редактировании изображений:

- увеличение разрешение изображения;
- улучшение качества изображения.

Из них увеличение разрешения изображений наиболее актуален, он не стремится изменить изображение и обладает наивысшими показателями качества из всех распространенных алгоритмов увеличения разрешения изображений, что нивелирует такие из его недостатков, как склонность к артефактам и химерам, ведь другие методы увеличения разрешения изображений тоже ведут к появлению артефактов.

Увеличение разрешения изображений необходимо очень часто. Современные мониторы и видеокарты способны отображать только одно разрешение одновременно, они не могут отобразить часть экрана в одном разрешении, а другую часть в другом разрешении, поэтому чтобы отобразить

изображения и видео, сделанные в более низком разрешении, чем разрешение монитора, необходимо увеличить их разрешение. Сейчас это по традиции делается с помощью малозатратных и эффективных алгоритмов, ведь большинство компьютеров, доступных рядовому пользователю было маломощными, однако рост производительности видеокарт в последнее время привел к тому, что все больше и больше компьютеров обладает необходимой мощностью, но не использует ее, потому что большинство решений для воспроизведения видео и отображения изображений не применяет высококачественные алгоритмы. Существующие решения, как madVR, сложны в использовании, нуждаются в дополнительной трудоемкой настройке и недоступны для рядового пользователя, поэтому существует необходимость продвижения этих методов в массы.

Увеличения разрешения также можно использовать для улучшения качества изображения путем применения суперсэмплинга, увеличение разрешения выше разрешения монитора и последующее его уменьшение. Это позволяет увеличить четкость и плавность изображения, а также избавиться от некоторых видов артефактов.

Еще одним преимуществом увеличения разрешения изображения является уменьшение необходимого объема хранилища, используемого для хранения одного файла. При уменьшении разрешения в два раза, размер файла может уменьшиться в 4 раза.

Другим способом улучшения качества изображения с помощью нейронных сетей является изменение изображения без изменения его разрешения. Это применяется для достижения следующих эффектов:

- убирание прыщей;
- улучшение цвета кожи;
- убирание морщин.

У этих методов более узкая зона действия и они, в отличие от суперсэмплинга, стремятся именно изменить изображение. Более точечная область действия приводит к уменьшению вероятности возникновения артефактов, но может привести.

**Список использованных источников:**

1. Li Deng, Dong Yu deep learning: methods and applications // Now Publishers Inc.. – 2014. – Vol. 7, № 4. – P. 197-387.
2. Christopher M. Bishop Pattern Recognition and Machine Learning// Springer-Verlag - New York, 2006. – 738 с.

## **ПРИМЕНЕНИЕ АЛГОРИТМОВ КОНТУРНОГО АНАЛИЗА ПРИ СОЗДАНИИ ПАНОРАМНЫХ ИЗОБРАЖЕНИЙ**

*Чирук Д.В.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Сапронова Ю.И. – ассистент*

В данной статье рассмотрена возможность использования алгоритмов контурного анализа изображений применительно к созданию панорамных изображений. Кратко указана актуальность темы. Обоснован выбор данного типа алгоритмов для использования при склейке исходных изображений при создании панорамы, рассмотрены основные факторы, влияющие на выбор. Приведено краткое описание алгоритма, а кроме него необходимых этапов предварительной и последующей обработки изображений для создания панорам.

Совмещение изображений является актуальным вопросом во многих направлениях: медицина (получение снимков внутренних органов, костей и др.), авиация, космические исследования и оборонная промышленность (комбинированное видение, совмещение снимков с приборов ночного видения, камер, зондов и др). Также широко распространенными в наше время являются панорамные фотографии, перекрывающие поле зрения человека до 180, а то и 360 градусов (так называемые сферические панорамы). Во многих сферах жизни человека так или иначе встречается необходимость совмещения изображений.

Стоит отметить, что под панорамными фотографиями стоит понимать те, соотношение сторон в которых больше, чем 2:1, и также стоит учитывать, что данные фотографии, как правило, получаются именно при наложении нескольких снимков друг на друга, а не использованием специальных технических средств (широкоугольных объективов и пр.).

Стоит обозначить некоторые ограничения для данного алгоритма:

- 1) минимальная скорость работы с сохранением достаточной точности;
- 2) простота реализации алгоритма;
- 3) относительная невосприимчивость к качеству исходных изображений;
- 4) легкость предварительной обработки изображения перед применением алгоритма.

Решение задачи сводится к поиску в двух и более полученных изображениях одних и тех же объектов, их сопоставлению, геометрическому преобразованию одного изображения к плоскости другого изображения и конечном слиянии данных изображений для отображения в виде панорамы. Путем исследований для ускорения процесса сопоставления было решено сопоставлять “ближайшие” друг к другу 35% исходных изображений. Контурные основных объектов на изображениях остаются неизменными. Отличия в перекрывающихся частях изображений возникают из-за геометрических искажений при съемке.

После теоретических изысканий было решено воспользоваться алгоритмом выделения границ объектов на основе вычисления порога Отсу для необходимых частей изображений и бинаризацию по данному порогу. После подготовки изображений необходимо произвести их совмещение. Для этого были рассмотрены два различных алгоритма совмещения контуров[1]:

Первый основан на идеях комплексного контурного анализа, а второй на использовании матрицы гомографии. Первый алгоритм позволяет учитывать преобразования изображений: сдвиг вдоль вектора, поворот относительно начала общей системы координат, изменение масштаба. Но он не учитывает проекционные искажения, которые будут возникать при съемке, поэтому он не подходит для совмещения изображений при съемке панорам.

Второй алгоритм - алгоритм совмещения контуров с помощью методов комплексного контурного анализа, позволяет учитывать сдвиг вдоль вектора, изменение масштаба, поворот относительно начала общей для двух изображений декартовой системы координат, что вполне достаточно.

Учесть влияние проекционных искажений можно в алгоритме совмещения контуров с помощью матрицы гомографии. Идея заключается в умножении этой матрицы на вектор, компоненты которого являются координатами  $i$ -й точки совмещаемого контура (изображения).

Одним из самых важных достоинств обнаружения объектов в последовательности снимков при съемке панорам и их совмещению с использованием контурного анализа является слабая зависимость точности от искажений яркости при различных условиях съемки, также стоит указать вычислительную простоту алгоритмов контурного анализа, и, следовательно высокую скорость выполнения алгоритма, что свидетельствует о возможности использования в системах реального времени.

**Список использованных источников:**

1. Логинов А.А., Новиков А.И., Саблина В.А., Щербакова О.В. Исследование возможности применения комплексного контурного анализа в задачах классификации и совмещения контуров // Вестник РГРТУ, №1 (выпуск 43). Рязань, 2013. с.20-25.

## **АУДИТ БЕЗОПАСНОСТИ ТРАФИКА В СИСТЕМЕ IP-ТЕЛЕФОНИИ**

*Шабан А.И.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Одинец Д.Н. – к.т.н., доцент*

В данной работе были рассмотрены возможные виды атак при использовании протокола установления связи в системе IP-телефонии. Также рассмотрены методы для повышения безопасности в системе IP-телефонии и предотвращения рассмотренных видов атак.

В IP-телефонии для установления связи между клиентами используется протокол SIP. Структура и синтаксис сообщений похож на те что используется в HTTP, что представляет собой набор текстовых строк, заголовки и тело сообщения разделены пустой строкой.

Протоколы SIP и RTP, используемые для передачи медиа данных, были разработаны без учета необходимости защищать передаваемую информацию, в следствии чего возможны следующие виды атак:

- 1) Фрод звонков – атакующий проникает в систему IP-телефонии и имеет возможность совершать звонки, украсть пароли и имена пользователей [1];
- 2) Вирусы – вирус, попавший в сеть IP-телефонии может начать рассылать спам ее абонентам. Т.к. клиенты для IP-телефонии могут быть реализованы в виде программ, то данному виду атаки подвержен не только клиент, но и операционная система, в следствии чего могут быть украдены данные;
- 3) Нарушение звонков – атакующий рассылает пакеты клиентам звонка, что приводит к ухудшению качества, задержкам и даже прекращению звонка [2];

- 4) DoS – данная атака нарушает доступность сервиса, также не позволяет авторизованным пользователям использовать сервис. Данная атака может осуществляться за счет отправки большого количества сообщений “Invite” и “Registrar”, что нарушает работу компонентов SIP;
- 5) Подслушивание – атакующий прослушивает весь трафик в IP-телефонии;
- 6) Подбор паролей;
- 7) Man-In-The-Middle – атакующий проникает в звонок между пользователями, и может не только прослушивать, но и изменять сообщения между клиентами.

Для защиты передаваемых данных, может быть использован протокол TLS – протокол защиты транспортного уровня. Данный протокол использует асимметричное шифрование для аутентификации, симметричное шифрование для конфиденциальности и коды аутентичности сообщений для сохранения целостности сообщений.

Также для защиты данных может быть использован IPsec – набор протоколов для обеспечения защиты данных, передаваемых по протоколу IP, обеспечивающих аутентификацию, проверку целостности и/или шифрование IP-пакетов [3].

При сравнении протоколов TLS и IPsec можно отметить что TLS имеет следующие преимущества:

- 1) Отсутствие постоянного соединения между сервером и клиентом;
- 2) Реализуется на уровне представления.

Недостатки:

- 1) Не может быть использован с протоколом UDP;
- 2) Требуется отслеживать состояние соединения.

Однако протокол IPsec реализуется на сетевом уровне и позволяет шифровать данные, но при этом сложен в реализации, предъявляет дополнительные требования к оборудованию сети.

Для улучшения защиты может быть использован VPN. В случае невозможности использования VPN, необходимо использовать VLAN. Данные решения создают виртуальную сеть, что позволяет передавать данные в ненадежных сетях [4].

Шифрование это один из ключевых механизмов, не позволяющего атакующему получить ценную информацию из перехваченных сообщений.

На данный момент аутентификация происходит в два этапа, что не является надежным методом. CHAP – протокол аутентификации с косвенным согласованием. Является алгоритмом проверки подлинности и предусматривает передачу не самого пароля пользователя, а косвенных сведений о нём. Аутентификация узла выполняется путём трехэтапной процедуры согласования.

Преимущества CHAP:

- 1) Значение идентификатора и переменного значения открытого ключа возрастают, что предотвращает атаки повторного воспроизведения;
- 2) Метод проверки подлинности основан на том, что аутентификатору и узлу известен секрет, который никогда не посылается по каналу.

Анализ трафика может быть осуществлен при помощи DDP, что позволяет анализировать содержимое передаваемых пакетов и блокировать вирусы.

Для передачи медиа данных необходимо использовать SRTP, который предназначен для шифрования, установления подлинности сообщения, целостности данных.

Для шифрования данных, SRTP использует алгоритм AES, который может использоваться в двух режимах:

- 1) Сегментированный целочисленный счётчик — режим, который осуществляет произвольный доступ к любым блокам, что является существенным для трафика RTP, передающегося в публичных сетях с непредсказуемым уровнем надежности и возможной потерей пакетов. AES, работающий в этом режиме, является алгоритмом шифрования по умолчанию, с длиной шифровального ключа по умолчанию в 128 бит и ключом сессии длиной в 112 бит;
- 2) f8-режим — вариант режима способа обратной связи, расширенного, чтобы быть доступным с изменённой функцией инициализации. Значения по умолчанию для шифровального ключа и ключа сессии — то же, что и в AES в режиме.

Рассмотрев приведенные методы защиты, нужно отметить, что для обеспечения безопасного соединения, необходимо использовать TLS, т.к. данный протокол реализуется на транспортном уровне и не требует поддержки данного протокола на промежуточных устройствах, в отличие IP-Sec.

Так как TLS не обеспечивает защиту сообщений в случае их перехвата, то необходимо воспользоваться шифрованием данных. В данном случае лучшим выбором будет использование VPN который позволяет шифровать данные, так как в этом случае нет необходимости реализовывать шифрование на клиентах и повышает совместимость, отсутствует риск ошибок в реализации шифрования, VPN может быть обновлен для использования более современных методов шифрования без необходимости обновления кода клиента.

Так же необходимо использовать аутентификацию CNAP. Для передачи медиа информации должен использоваться протокол SRTP.

**Список использованных источников:**

1. <http://www.centurylinkbrightideas.com/how-to-address-voip-security-challenges>
2. <https://habr.com/en/company/pt/blog/212839/>
3. [https://www.researchgate.net/publication/235601569\\_SIP\\_Server\\_Security\\_with\\_TLS\\_Relative\\_Performance\\_Evaluation](https://www.researchgate.net/publication/235601569_SIP_Server_Security_with_TLS_Relative_Performance_Evaluation)
4. <https://searchunifiedcommunications.techtarget.com/feature/SIP-network-security-measures>

## ОСНОВНЫЕ ЭТАПЫ 3D РАСПОЗНАВАНИЯ ЛИЦ

*Шакун Р.А.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Лукашевич М.М. – к.т.н., доцент*

Трехмерная информация о поверхности лица человека является мощной биометрической модальностью, которая может повысить точность идентификации и проверки точности систем распознавания лиц в сложных ситуациях. При наличии вариантов освещения, выражения и позы традиционные алгоритмы распознавания лиц на основе двумерных изображений обычно сталкиваются с проблемами. С доступностью трехмерной (3D) информации о форме лица, которая по своей природе не чувствительна к освещению и создает изменения, эти сложности могут быть эффективно устранены

Система 3D распознавания лиц обычно состоит из следующих этапов: получение 3D образцов лица, преобразование образцов, извлечение признаков, запись признаков в базу, поиск соответствий между признаками.

Рабочий процесс может быть разбит на две фазы (обучения и тестирования) и пять этапов. Первым этапом обучения является сбор данных. Получение трехмерных образцов лица включает в себя специальное аппаратное оборудование, которое можно отнести к категории активных систем сбора данных и пассивных систем сбора данных в соответствии с используемыми технологиями. Активные системы сбора данных излучают невидимый свет, например, инфракрасные лазерные лучи, чтобы осветить лицо человека. Затем системы измеряют отражение, чтобы определить особенности формы целевой грани. Пассивные системы содержат несколько камер, которые расположены отдельно друг от друга. Система сопоставляет точки, наблюдаемые с другой камеры, и рассчитывает точное трехмерное местоположение совпадающей точки. Набор совпавших точек формирует трехмерное лицо.

Полученные трехмерные данные лица не могут быть непосредственно использованы в качестве входных данных алгоритмов извлечения признаков, поскольку данные содержат отвлекающие элементы, такие как волосы, уши, шея, очки и ювелирные изделия. Такие особенности, как волосы, очки и украшения, могут время от времени меняться, а особенности ушей и шеи не могут быть достоверно идентифицированы в разных позах головы, поэтому, дабы не вводить в заблуждение алгоритмы распознавания, они должны быть удалены до извлечения признаков.

Следующим этапом предварительной обработки является определение положения и ориентации человеческого лица. Геометрические преобразования используются для «поворота» человеческого лица непосредственно к оси камеры. Затем предварительная обработка использует помощь от четко определенных частей лица, таких как нос, чтобы изолировать область человеческого лица от областей отвлекающих элементов. Эта операция называется сегментацией.

Предварительно обработанные образцы лицевых данных часто интерпретируются в трех форматах модели: глубинное изображение, облако точек и сетка.

Затем для обработки данных применяются алгоритмы извлечения признаков, чтобы найти те, которые можно использовать для распознавания лиц. Самый простой способ выделения признаков состоит в том, чтобы рассматривать все лицо как единый вектор признаков, который называется глобальным подходом [1]. При таком подходе все лицо сохраняется в базе данных. На этапе сопоставления признаков лицо цели сравнивается с лицами в базе данных с использованием статистических классификационных функций. В противоположность глобальному подходу, компонентный подход фокусируется на локальных характеристиках лица, таких как нос и глаза. Он использует операторы графа для извлечения части носа и глаз и сохранения этих локальных особенностей в базе данных. Когда целевая грань вводится для распознавания, компонентный подход сначала извлекает соответствующие детали из целевых граней, а затем ищет соответствующий набор деталей из базы данных объектов. Существуют гибридные подходы, которые объединяют функции, используемые глобальными и локальными подходами. При более высоких

вычислительных затратах гибридный подход может обеспечить более высокую точность распознавания

После извлечения, признаки каждого лица сохраняются в базе данных.

На этапе тестирования целевое лицо проходит этапы сбора, предварительной обработки и выделения признаков, которые идентичны этапам обучения. На этапе сопоставления признаков признаки целевого лица сравниваются с хранящимися в базе данных признаками, и рассчитывают оценки соответствия. Когда результат совпадения достаточно высок, тогда можно считать, что целевое лицо распознается.

**Список использованных источников:**

1. Feature extraction for robust physical activity recognition [Электронный ресурс]. - Режим доступа: <https://hcis-journal.springeropen.com/articles/10.1186/s13673-017-0097-2>

**СЕКЦИЯ**  
**«ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ**  
**ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»**

## ПРОГРАММНОЕ СРЕДСТВО УПРАВЛЕНИЯ ЗАДАЧАМИ ПО МЕТОДОЛОГИИ КАНБАН

Акульчик В.С.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Сурков Д.А. – ст. преп. каф. ПОИТ

В современном мире планирование является залогом успеха в любой сфере жизнедеятельности. При составлении плана можно точно распределить свое время и повысить свою продуктивность. План помогает расставить приоритет в предстоящих задачах и сконцентрировать свое внимание на самые важные из них. Составление плана позволяет сократить время на выполнение поставленных задач, что приводит к общей экономии времени. Для планирования и управления задачами могут применяться самые различные методологии. Рассмотрим одну из самых популярных в настоящее время.

Канбан – система организации производства и снабжения, позволяющая реализовать принцип «точно в срок». Слово «канбан» в переводе с японского языка означает «рекламный щит, вывеска». Появление данного термина можно связать с перечислением стандартных операций конвейерного процесса производства. Мастера участков указывали на листке бумаги работы, выполняемые на данном участке, и вывешивали их на доске, размещенной на видном месте. Рядом с ними вывешивали такие же списки мастера других участков. Это позволяет легко ориентироваться в ситуации, складывающейся на производстве, и быстро вносить корректировки в план, чтобы в конечном итоге успеть выполнить всё запланированное точно в срок. Система канбан была разработана и впервые в мире реализована компанией «Toyota». Это позволило компании разгрузить склады своей продукции и стать ближе к потребителю, благодаря тому, что они начали производить только то, что действительно было необходимо покупателям, а не пытаться прогнозировать спрос на свою продукцию. Значительно позже похожий принцип был использован и в разработке программного обеспечения.

Канбан – методология управления разработкой программного обеспечения, которая позволяет реализовать принцип «точно в срок» и способствует более равномерному распределению нагрузки между всеми членами команды. При данном подходе весь процесс разработки становится прозрачным для всех участников. Для иллюстрации работы применяется доска, состоящая из нескольких колонок. На доске всегда отражается актуальное состояние всех задач. Доска, как правило, разделена на три логические секции: «сделать», «в процессе» и «готово». Сотрудники перемещают заметки в ту секцию доски, которая соответствует статусу задачи. Задачи по мере поступления заносятся в отдельный список, откуда каждый участник команды разработки может взять необходимую задачу, перенести в другую колонку и тем самым приступить к её выполнению. Таким образом, канбан можно считать наглядной системой разработки, которая всегда отражает реальное положение дел на проекте. Такие доски можно рассматривать как вариацию на тему традиционных канбан карточек. Обычно карточки обозначали потребность определенного участка конвейера, а теперь обозначают отдельно взятые задачи. Также вместо бумажных карточек можно использовать фишки или магниты. Передвижение одного из этих объектов по доске по мере выполнения задач соответствует движению процесса производства.

В методологии канбан порядок задач на доске имеет значение. Чем выше в колонке располагается задача, тем более важной она является. Благодаря такому легкому визуальному приему решается очень важная проблема с приоритетностью задач. В любой момент времени понятно, какую из задач нужно выполнить первой, а какие можно отложить на более позднее время. Методология канбан позволяет оперативно изменять приоритет задач. Для этого достаточно всего лишь переместить задачу в колонку вверх или вниз, тем самым повысив или понизив её приоритет. Это позволяет сконцентрировать внимание на самых главных задачах в конкретный момент времени.

С течением времени технологии развивались, и изменения коснулись и канбан доски. На смену физической доске с карточками пришла её электронная версия. Карточки стали еще более красочными и постепенно получали все большую известность. Однако их главная функция осталась прежней – иллюстрировать процесс управления задачами по методологии канбан. В последнее время электронные канбан-доски получили широкое распространение как среди разработчиков программного обеспечения, так и среди обычных пользователей персональных компьютеров.

Цель данной работы заключается в разработке программного средства для управления задач по методологии канбан. Разработанное программное средство предоставляет возможность планировать и управлять задачами не только в рамках процесса разработки программного обеспечения, но и при планировании повседневных задач в любой сфере жизнедеятельности.

### Список использованных источников:

1. Канбан [Электронный ресурс]. – Электронные данные. – Режим доступа <https://ru.wikipedia.org/wiki/Канбан>. – Дата доступа: 01.05.2018

## WEB APPLICATION “ELECTRONIC CALENDAR”

*Dzmitry Aliksandrau*

*Belarusian State University of Informatics and Radioelectronics  
Minsk, Belarus*

*Svetlana Boltak – department of Software for Information Technologies assistant*

Electronic calendars are not useful only for business people, but are also a very helpful tool for schoolchildren and students. To help them with management of life activities like planning work, meetings, make work notes, manage workers, create reports.

The time when people were using pen and paper for making notes and planning work activities has passed. With the new technologies and an increase in Internet accessibility, electronic calendars came widely into use. Big corporations are massively using them for easier communication and control over the work of employees, as well as for the organization of work. Not only companies, but also individuals with lots of activities, are using an electronic calendar in order to plan their work and family activities. No matter the industry and organization, electronic calendars have proven to be a very effective tool.



Picture 1 – Electronic calendar web assistance

### *Benefits of using electronic calendars:*

**Never forget the event.** Have you ever written down an important event in your calendar and then completely forgot about it? Are you questioning why? It's because you forgot to regularly check the calendar or/and nobody warned you on upcoming events. With electronic calendar, this is the past. You can make reminders (also: text message or email reminders) for special events, important meetings and work that has to be done in your daily activities or life [1].

**Access everywhere and anywhere.** Another positive side of electronic calendar is that you can access to it everywhere and anywhere. Even when you are on the go, out of your office or home. Basically, you can update your work log of the situation and schedule your future maintenance activities directly from your smartphone or laptop. Access to the calendar 24/7 allows you to check and update your meetings, events and daily activities anytime, day-and-night [1].

**Never lose data.** If you keep track of your maintenance activities on electronic calendar you will never again need to worry about losing your paper records of management tasks or important activities. All the records are available to you at any given moment, and directly analyzable. Having everything in one place also allows you to compare your yearly activities with previous time and figure out some important things to be improved [1].

**Create reports.** Every work activity is stored in a data base and can help create future events and appointments. While some require special forms to be filled others don't, either way, electronic calendars comes in handy in order to gather the data and create printed reports of yearly activities. You don't really need to print the reports, simply save it to your computer and open when needed [1].

**Track due dates.** Keeping an accurate and up-to-date calendar of due dates of important assignments, tests, projects and class discussions reminds you of what you need to complete and when.

Rather than being overwhelmed at seeing coursework as a long checklist of things you need to do, a calendar provides the space to plan for each due date at the appropriate time.

**See availability, quickly.** It's an unfortunate feeling when you receive and accept an invitation to a fun event, only to realize later that you already have a commitment during that time.

A calendar allows you to see quickly when you are free and when you have prior commitments. Planning events or other special occasions also becomes easier when you know your availability.

**Calendar tips.** Calendars are easy to personalize, depending on your work habits, access to technology and other preferences. Regardless of which route you choose, here are some insightful tips to get the most out of keeping your calendar:

- color code your various schedules – work, home, school, etc. – to easily distinguish where you spend your time;

- keep your calendar in a location that's easily accessible and visible;

- plan study times at consistent times to encourage the habit of studying;

- want to use electronic and paper calendars? Keep both for different uses or in separate locations.

For example, use a paper month-view calendar for special events and an electronic week or day-view calendar for more detailed assignments, meetings, etc.

Maintaining an accurate calendar can help you both in your academics and in your personal life – and can ease the overwhelming feeling of balancing a busy schedule.

**Technology stack.** From development prospective make sense to organize headless architecture.

*React* was chosen as a library for building user interfaces. Why *React*? From my expertise it is the most popular library created by Facebook. *React* makes it painless to create interactive UIs. Design simple views for each state in your application, and *React* will efficiently update and render just the right components when your data changes. Declarative views make your code more predictable and easier to debug. Build encapsulated components that manage their own state, then compose them to make complex UIs. Since component logic is written in JavaScript instead of templates, you can easily pass rich data through your app and keep state out of the DOM [2]. Most *React* applications use one state manager called *Redux*.

*Redux* is a predictable state container for JavaScript apps. It helps you write applications that behave consistently, run in different environments (client, server, and native), and are easy to test. On top of that, it provides a great developer experience, such as live code editing combined with a time traveling debugger.

Moreover, *React* code base can be shared for building native mobile applications. *React Native* lets build mobile apps using only JavaScript. It uses the same design as *React*, letting compose a rich mobile UI using declarative components. The apps which are building with *React Native* aren't mobile web apps because *React Native* uses the same fundamental UI building blocks as regular iOS and Android apps.

*Node.js* is a good choice for back-end. *Node.js* is a JavaScript runtime built on Chrome's V8 JavaScript engine. Many people wonder how a single-threaded *Node.js* can compete with multithreaded back ends. As such, it may seem counterintuitive that so many huge companies pick *Node* as their back end, given its supposed single-threaded nature. To know why, we have to understand what we really mean when we say that *Node* is single-threaded. JavaScript was created to be just good enough to do simple things on the web, like validate a form or, say, create a rainbow-colored mouse trail. It was only in 2009 that Ryan Dahl, creator of *Node.js*, made it possible for developers to use the language to write back-end code. Back-end languages, which generally support multithreading, have all kinds of mechanisms for syncing values between threads and other thread-oriented features [3].

*MongoDB* is an open source database management system (DBMS) that uses a document-oriented database model which supports various forms of data. It is one of numerous no relational database technologies which arose in the mid-2000s under the NoSQL banner for use in big data applications and other processing jobs involving data that doesn't fit well in a rigid relational model. Instead of using tables and rows as in relational databases, the *MongoDB* architecture is made up of collections and documents [4].

*Docker* unlocks the potential of organizations by giving developers and IT the freedom to build, manage and secure business-critical applications without the fear of technology or infrastructure lock-in. By combining its industry-leading container engine technology, an enterprise-grade container platform and world-class services, *Docker* enables to bring traditional and cloud native applications built on Windows Server, Linux and mainframe into an automated and secure supply chain, advancing dev to ops collaboration and reducing time to value. Because *Docker* increases productivity and reduces the time it takes to bring applications to market, organizations have the resources needed to invest in key digitization projects that cut across the entire value chain, such as application modernization, cloud migration and server consolidation. [5] Also *Docker* protects data base. It is not possible to connect to data base directly and steal all data while port will not be exposed from *Docker* container to local machine. Only other containers have access to data base.

**List of used sources:**

1. eVineyardBlog. [Digital resource]. – Digital data. – Access mode: <https://www.evineyardapp.com/blog/2015/07/20/how-electronic-calendar-can-help-with-vineyard-management> Access date: 24.03.2019.
2. React. [Digital resource]. – Digital data. – Access mode: <https://reactjs.org/> Access date: 24.03.2019.
3. Node.js [Digital resource]. – Digital data. – Access mode: <https://nodejs.org/en/> Access date: 24.03.2019.
4. Mongo DB. [Digital resource]. – Digital data. – Access mode: <https://www.mongodb.com/> Access date: 24.03.2019.
5. Docker. [Digital resource]. – Digital data. – Access mode: <https://www.docker.com/> Access date: 24.03.2019.

## ПРИЛОЖЕНИЕ ДЛЯ АВТОМАТИЗАЦИИ ОРГАНИЗАЦИИ ЗАДАЧ И РАБОЧЕГО ГРАФИКА ПРЕПОДАВАТЕЛЯ

Арзуманян А.С.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Трус В.В. – ст. преподаватель

В рамках интеграции информационных технологий в жизнь человека, преподаватели всё чаще сталкиваются с необходимостью организации рабочего графика с помощью различных сервисов. Они значительно облегчают процесс ведения расписаний и задач, так как позволяют сделать управление рабочим графиком более удобным, а также позволяют в любой момент получить необходимую информацию о расписании. Существует множество мобильных приложений для организации рабочего графика, но большинство из таких сервисов не учитывают преподавательскую специфику, не имеют синхронизаций с веб-сервисами, а также не дают статистическую информацию о затраченном времени.

Ключевым преимуществом использования мобильного приложения для организации расписания является то, что вся необходимая информация легкодоступна и имеет формализованный вид. Пользователь может в режиме реального времени использовать только актуальные данные.

Планирование расписания и задач является важным пунктом в работе преподавателя, которому необходимо держать постоянный контроль над большим объёмом информации, так как профессия преподавателя предполагает работу с широким спектром задач и коммуникацию с людьми, которых обучает преподаватель, и с коллегами по работе[1].

Управление временем – это действие или процесс тренировки сознательного контроля над временем, потраченным на конкретные виды деятельности, при котором специально увеличиваются эффективность и продуктивность. Управление временем может помочь получить ряд навыков, инструментов и методов, используемых при выполнении конкретных задач, проектов и целей. Управление временем включает в себя широкий набор решаемых задач, среди которых:

- постановка целей;
- планирование времени;
- анализ затрат времени;
- определение приоритетов;
- создание списков [2].

Современные мобильные приложения хорошо справляются с вышеупомянутыми задачами. Благодаря интерактивному и интуитивно понятному интерфейсу пользователь в пару кликов может добавлять, редактировать или просматривать необходимую информацию о расписании. Большинство таких приложений организованы согласно клиент-серверной архитектуре. Нередко случается, что приложения используют в качестве серверной части какой-нибудь сторонний сервис. Так, например, многие приложения для управления расписанием используют сервер Google-календаря, имеют синхронизацию с ним и используют его базу данных для хранения. Данный подход является удобным, так как пользователю предоставляется большой функционал, а программисту – простота разработки.

Нередко, так называемые, приложения-планировщики позволяют выставить степень важности задаче. Однако, помимо важности существует другой не менее важный фактор, который не учитывают данные приложения – срочность исполнения. Так, один из президентов Америки изобрёл собственную методику для управления временем, называемую “таблицей Эйзенхауэра”. Суть метода состоит в том, что все задачи делятся на следующие категории:

- срочные (важные) задачи, которые должны выполняться в первую очередь;
- срочные (менее важные) задачи, которые при необходимости можно делегировать;
- менее срочные (важные) задачи, которые не критичны и могут быть выполнены в ближайшее время;
- менее срочные (менее важные) задачи, которые лучше всего сделать при наличии свободного времени или делегировать другому человеку.

Для более эффективного планирования расписания и задач преподавателя, а также для получения статистических данных было разработано мобильное приложение на базе операционной системы Андроид. Таким образом преподаватель сможет удобным ему образом создавать элемент

расписания, просматривать и редактировать его. Ключевыми особенностями являются возможности просмотра статистических данных о затраченном времени преподавателя и выставления степени важности и срочности задачи.

**Список использованных источников:**

1. [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://kinderklub.ru/professiya-uchitel/upravlenie-vremenem-kak-instrument-prepodavatelya/>.
2. Википедия [Электронный ресурс]. – Электронные данные. – Режим доступа: [https://ru.wikipedia.org/wiki/Управление\\_временем](https://ru.wikipedia.org/wiki/Управление_временем).

## ПРОГРАММНОЕ СРЕДСТВО «KASISKI-BABBAGE FREQUENCY ANALYSIS»

*Биткин Н.С., Болтак С.В.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Болтак С.В. – ассистент*

Современные криптосистемы отличаются своим многообразием, уровнем криптостойкости и сложностью понимания. Поэтому начинающему криптографу необходим более простой и понятный пример для ознакомления с огромным миром шифров и методов взлома. Разработанный программный продукт позволяет провести криптоанализ шифротекста, зашифрованного шифром Виженера с прогрессивным способом генерации ключевой последовательности.

Шифр Виженера может использовать так называемый прогрессивный ключ (дублирование ключевого слова осуществляется с одновременным сдвигом букв ключа на одну позицию при каждом новом повторении) [1]. В этом случае задача взлома становится не совсем тривиальной.

*В алгоритм взлома необходимо внести следующие изменения:* повторяющиеся последовательности необходимо учитывать только в позициях, чьи номера кратны длине алфавита исходного текста (это очевидно, так как минимально необходимое количество сдвигов ключа для получения исходного равняется мощности алфавита).

Данная работа представляет собой анализ возможностей метода Касиски-Бэббиджа при атаке шифротекста, полученного с использованием шифра Виженера с прогрессивным ключом. Средняя скорость обработки – порядка миллиона символов в минуту. Исходя из полученных результатов можно сделать следующие выводы:

- 1) метод абсолютно неэффективен на шифротексте, длина которого не превышает произведение мощности алфавита на длину ключа;
- 2) при достижении определённых размеров шифротекста метод взлома позволяет получить результат с вероятностью более 90% для ключа любой длины;
- 3) при повторном шифровании ранее зашифрованного текста вероятность успеха стремится к нулю, тоже самое касается шифрования текста, не имеющего смысла;
- 4) научно-популярная литература более устойчива к рассматриваемой атаке. Из-за насыщенности таких текстов терминами, которые в большинстве своём являются словами с большим количеством букв, вероятность совпадения последовательностей меньше;

Разработанное программное средство позволяет наглядно продемонстрировать работу метода Касиски-Бэббиджа на шифротексте, полученном шифром Виженера с прогрессивным ключом. Как видно из рисунка 1, успешность взлома зависит от длины шифротекста.

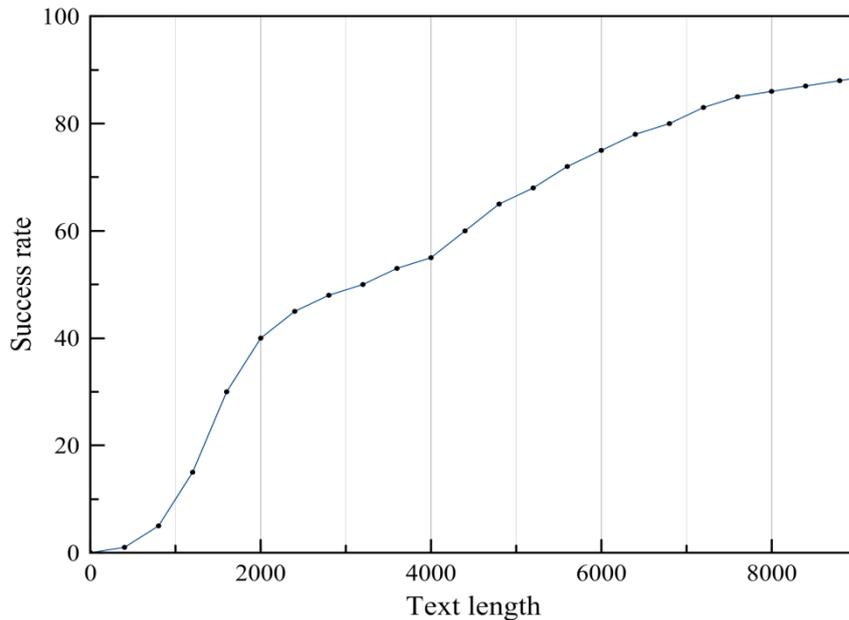


Рисунок 1 – Процент успешности атаки при фиксированной длине ключа

**Список использованных источников:**

1. Tilborg H.C.A. Fundamentals of Cryptography. A Professional Reference and Interactive Tutorial. Kluwer, 1999. с. 9-16.

## **ПРОГРАММНОЕ СРЕДСТВО МОНИТОРИНГА РАБОТЫ СЕРВИСОВ В РАСПРЕДЕЛЕННОЙ СИСТЕМЕ**

*Бобков А.А.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Смолякова О.Г. – к.т.н., доцент*

Информационные технологии полностью изменили жизнь современного человека. Мгновенный и беспрепятственный доступ к любой информации упростил и ускорил большинство процессов, связанных с различными сферами деятельности. С ростом серверных приложений появляется проблема, связанная с тем, что приложения не помещаются на одном сервере. Данная проблема решается созданием распределенных систем. В докладе сформулированы цели и особенности реализации программного средства, осуществляющего контроль работоспособности серверов в данной системе.

С методологической точки зрения мониторинг программ можно рассматривать как процедуру по оценке, целью которой является выявление и (или) измерение эффектов продолжающихся действий без выяснения причин. Мониторинг выступает в качестве внутренней процедуры, основанной на индикаторах и результатах, а также как инструмент сбора информации и отчетности.

Важно различать понятия «мониторинг» и «оценка программ». Сущность этих терминов одинакова — отслеживание результатов работы программ и выдача данных лицам, принимающим решение. Однако между ними есть и различия, заключающиеся в вопросах, на которые должно ответить проведение оценки или мониторинга. Мониторинг программ подразумевает ответ на вопрос: как идут дела? Он основывается на отслеживании текущей ситуации и сравнении текущего положения дел с ранее разработанным планом.

Программное средство мониторинга работы сервиса в распределенной системе является веб-инструментом отслеживания, который в режиме реального времени отображает активность различных сервисов в распределенной системе [1].

Основные возможности, предоставляемые программным средством:

- возможность просмотра всех сервисов, а также их статусов;
- возможность добавления и редактирования сервисов;
- возможность автоматического подъема сервиса;

– возможность SMS-оповещения о неисправном поведении сервиса.

Принцип реализации структуры приложения на рисунке 1:

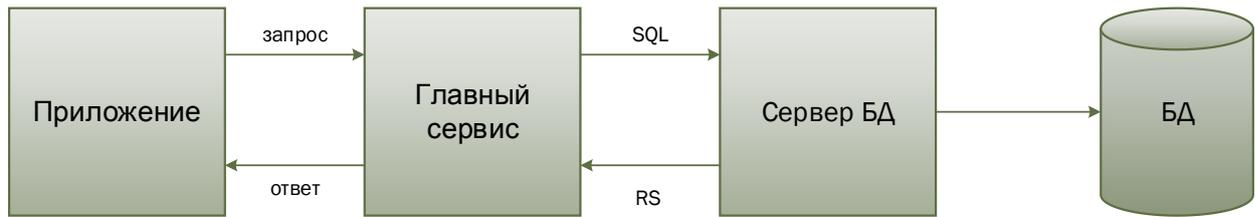


Рисунок 1 – Структура приложения

Основной принцип работы заключается в том, что приложение постоянно отправляет запросы сервисам, которые включены в мониторинг, и анализируя код ответа, выставляет статус для определенного сервиса [2]. В случае, если сервис не отвечает после первого запроса, то будет произведена повторная попытка обращения. Обращение будет идти до той степени, пока количество попыток не превысит установленное ограничение и таким образом главный сервис понимает, что данный сервис находится в нерабочем состоянии.

Приложение собирает всю информацию о состоянии сервисов в реальном времени, а это дает главное преимущество — возможность посредством SMS-оповещения мгновенно получить информацию об этом и, в случае экстренной ситуации, при которой не сработает автоматический подъем сервиса, поднять сервис вручную, используя приложение.

**Список использованных источников:**

1. Seth Gilbert and Nancy Lynch, "Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services", *ACM SIGACT News*, Volume 33 Issue 2 (2002), pg. 51–59.
2. Албахари, Джозеф. С# 6.0. Справочник. Полное описание / Джозеф Албахари, Бен Албахари. – 6-е изд. – ООО "И.Д. Вильямс", 2016. – 1040 стр.

## ЗАДАЧИ АВТОМАТИЗАЦИИ БАНКОВСКИХ ПРОЦЕССОВ В КРЕДИТОВАНИИ

*Богдан Г.О.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Парамонов А.И. – к.т.н., доцент*

Современная экономическая модель в мире напрямую зависит от уровня технического прогресса. Технический прогресс формирует не только новые экономические отношения, но и позволяет оптимизировать их. В связи с этим банки все чаще проявляют повышенный интерес к автоматизированным системам оценки риска и кредитоспособности, которые позволили бы минимизировать участие экспертов и влияние человеческого фактора на принятие решений.

В отечественных и зарубежных банках применяются различные подходы к определению кредитного риска физических лиц, начиная с субъективных оценок кредитными экспертами коммерческих банков и заканчивая автоматизированными системами оценки риска. Большинство банков в своей практике используют два основных метода оценки кредитоспособности физических лиц:

- экспертные системы оценки кредитоспособности заемщиков;
- балльные системы оценки кредитоспособности клиентов.

Использование балльных систем оценки кредитоспособности клиентов – более объективный и экономически обоснованный метод принятия решений при выдаче кредитов, чем экспертные оценки.

В современных бизнес-процессах активно начался развиваться тренд на автоматизацию. Среди процессов, автоматизация которых повлияет на эффективность банковской системы, можно выделить:

- предварительный расчет платежей и размера кредита осуществляется на стадии изучения материалов для предоставления кредита. Данная процедура необходима для информирования клиента для принятия окончательного управленческого решения о заключении сделки;

– автоматизированное составление документов обеспечивает формирование текста кредитного договора, договора залога, срочного обязательства по кредиту, служебных записок и тому подобное. Реализация данной функции обеспечивается наличием шаблонов типовых форм документов по кредитованию. При оформлении документа необходимо внести только реквизиты в указанные места в соответствии с шаблоном [1];

– исполнение сопровождения операций по кредитным соглашениям;  
– контроль исполнения мероприятий по кредитным соглашениям. Обеспечивает контроль выполнения мероприятий по соглашениям согласно календарю [1];  
– составление отчетов обеспечивает формирование обязательных отчетов согласно должностных обязанностей эксперта по кредитованию.

В зависимости от конкретных функциональных обязанностей работников и подразделений банка при предоставлении кредита, данный список может дополняться или изменяться.

Предлагается решение, при котором автоматически будет проверяться кредитная история потенциального заемщика, запрашивая данные из Бюро кредитных историй. Таким образом, сразу будут отброшены «гиблые» заявки, на дальнейшее рассмотрение которых нет смысла тратить время. Следует также изучить вопрос внедрения скорингового принятия решения по кредитным заявкам [2]. Кредитный скоринг представляет собой дискретную математическую или статистическую модель, с помощью которой банк пытается определить, насколько велика вероятность того, что потенциальный заемщик вернет кредит в срок.

Автоматизация процессов кредитования позволит значительно упростить и ускорить все составляющие его процессы, начиная от формирования заявки от потенциального клиента до момента выдачи кредита, что приведет к увеличению количества выданных кредитов и, соответственно, к росту банковской прибыли [2].

Зачастую для автоматизации работы банки используют веб-приложение, к которому будет доступ как клиентов, так и сотрудников банка. Логика веб-приложения распределена между сервером и клиентом, хранение данных осуществляется на сервере, а обмен информацией происходит по сети. Одним из преимуществ такой системы в отличие от офлайн-приложений, что клиенты не зависят от конкретной операционной системы пользователя, поэтому именно веб-приложения являются межплатформенными универсальными сервисами.

Использование специального программного обеспечения позволит полностью контролировать процесс создания кредитной заявкой, что обеспечит полноту и корректность данных.

**Список использованных источников:**

1. Оптимизация кредитного процесса: процесс принятия решений / Материалы конференции решения SAS для кредитного конвейера / 24 августа 2017г. / Алматы
2. Автоматизация кредитного процесса: старые проблемы и новые решения [Электронный ресурс] / Аналитический правовой портал. – Москва, 2015. – Режим доступа: <http://отрасли-права.рф/article/8534>. – Дата доступа: 07.06.2015

## ИГРОВОЕ ПРИЛОЖЕНИЕ «МОМО»

*Бондарева Т. О.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Данилова Г.В. – м.т.н., ассистент*

Многие люди мечтают о домашних любимцах, но не все могут себе это позволить. Внедрение данной компьютерной программы станет уникальным шансом завести себе домашнего любимца.

«Тамагочи»— игрушка, виртуальный домашний питомец.

Изобретателем «Тамагочи» считается Аки Маита, сотрудница отдела маркетинга японского производителя игрушек Bandai. По легенде, Маита очень любила домашних животных и хотела иметь зверька, которого можно всюду брать с собой. Она год вынашивала идею подобной игрушки, а окончательно мысль сформировалась после того, как ей попалась телереклама, в которой мальчик хотел взять с собой в школу черепашку.

Маита разработала концепт игрушки, а само устройство помог создать Акихиро Йокои из компании WiZ, также занимавшейся производством товаров для детей. Уже в октябре Маита начала показывать тестовую версию устройства школьницам, и, по её воспоминаниям, «их глаза тут же загорались». Она несколько недель следила за реакцией примерно 200 девочек на игрушку, чтобы довести её до совершенства.

23 ноября 1996 года компания Bandai выпустила в продажу первую версию «Тамагочи» – небольшой пластиковой игрушки в форме яйца с чёрно-белым экранчиком и тремя кнопками. По официальной версии, слово «Tamagotch» образовано от слияния японского слова «tamago» – «яйцо» и английского «watch» – «смотреть».

В настоящее время существует большое количество аналогов «Тамагочи» и сделать выбор питомца становится все тяжелее. Одни из них содержат лишние функции, которые используются весьма редко, другие имеют недостаточно удобный интерфейс, а некоторые и вовсе очень требовательны к системным ресурсам.

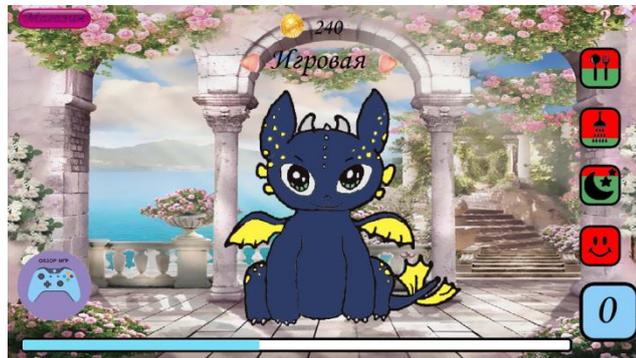


Рисунок 1– Игровое поле

Идея программы заключается в уходе за дракончиком по имени Момо (рисунок 1). Его надо кормить, мыть, укладывать спать и необходимо с ним играть. В приложение доступны три мини-игры: «Найди пару», «Пятнашки», «Розочки». Играя в мини-игры, пользователь получает монетки, которые можно расходовать на покупку еды для дракончика. В игре также присутствует анимация и звуковое сопровождение. А именно, в главном меню и в мини-играх присутствует музыка, при кормлении слышен звук чавканья дракончика и еда постепенно исчезает с тарелки, при купании слышен звук льющейся воды и видна анимация капель воды, а при сне слышен храп дракончика.

Программа имеет развлекательный характер. Задачей, поставленной перед данной курсовой работой, является разработка игрового приложения, которое поможет пользователю провести время с удовольствием.

Приложение имеет возможность наращивания функционала и расширения возможностей. Со временем данное приложение будет совершенствоваться: будет добавлены новые мини-игры, реализации приложения на экранах с разным разрешением, возможность выбора персонажа и одежды для него.

#### **Список использованных источников:**

1. Научные исследования компьютерных игр (электронный ресурс). – Электронные данные. – Режим доступа: <http://gamestudies.ru>.
2. Серебряная, Л.В. Марина, И.М. Структуры и алгоритмы обработки данных: учебно-метод. пособие для студ. спец. «Программное обеспечение информационных технологий» всех форм обуч. / Л. В. Серебряная, И. М. Марина. – Минск: БГУИР, 2012. – 49 с. : ил.

## **СПОСОБЫ АВТОМАТИЧЕСКОГО РАСПОЗНАВАНИЯ НОМЕРОВ ВАГОНОВ ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА**

*Борисевич А.Н., Деменковец Д.В.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Серебряная Л.В. – к.т.н. доцент*

В данной работе представлены подходы распознавания номеров вагонов железнодорожного транспорта. Для решения этой задачи использовались сверточные искусственные нейронные сети, обученные на наборах данных Modified National Institute of Standards and Technology и The Street View House Numbers. Наилучший результат показал алгоритм You Only Look Once (YOLO), но результат оказался недостаточным для применения в коммерческой среде.

Одной из необходимых задач, решаемых в автоматизированных системах фото и видеофиксации процесса взвешивания вагонов железнодорожного транспорта, является распознавание их номеров. Это связано с тем, что основным идентифицирующим атрибутом

железнодорожного вагона является его номер. Поэтому актуальным является автоматическое распознавание номера. Решению названной задачи посвящена данная работа.

В результате анализа моделей, подходящих для решения поставленной задачи, были выбраны искусственные нейронные сети (ИНС). ИНС обучается на размеченном наборе исходных данных, представляющем собой фотографии цифр. Под разметкой подразумевается выделение каждой цифры в рамку с указанием порядкового номера цифры.

Первоначально было решено использовать популярный набор данных Modified National Institute of Standards and Technology (MNIST) (рисунок 1). Этот набор содержит 60000 черно-белых изображений рукописных цифр с разрешением 20x20 пикселей в тренировочной выборке. На этих данных была обучена ИНС, состоящая из двух сверточных слоев с размерами фильтра 5x5 и функцией активации ReLU, двух сверточных слоев с размером фильтра 3x3 и функцией активации ReLU, двух полносвязных слоев с функциями активации ReLU и softmax. Точность такой сети составляет 0.997 на тестовой выборке, состоящей из 10000 изображений. Чтобы применить такую сеть к данным о номерах вагонов, было выполнено морфологическое трансформирование изображений с помощью библиотеки `opencv`. Преобразования позволили привести фотографии номеров вагонов к виду, сравнимому с фотографиями из набора MNIST. Однако, кроме этого, требовалось определять яркость текста или фона номера. На фотографиях вагонов были определены два случая. При ручном подборе наиболее подходящих параметров предварительной обработки изображений модель показывала неудовлетворительные результаты. Точность ее работы составляла менее 22%. Поэтому было решено использовать другой размеченный набор данных.

The Street View House Numbers (SVHN) – набор данных, состоящий из 73257 цифр на цветных фотографиях с разрешением 32x32 пикселя (рисунок 2). Был использован классификатор, предварительно обученный на выборке из 26032 цифр. Точность его распознавания составляет 94.2%. Применение этой модели для решения задачи распознавания номеров вагонов требует предварительного разделения номеров на цифры. Сначала был использован линейный оператор, устраняющий перспективы на рисунке номера, а затем делалась попытка разделить цифры по гистограмме яркости. Но и этот способ не дал ожидаемых результатов по причине того, что на многих фотографиях цифры находились на близком расстоянии друг от друга, а также во многих линиях цифр имелись разрывы (рисунок 3). Следующим был применен метод “скользящего” окна, но и он не дал удовлетворительных результатов (рисунок 3).

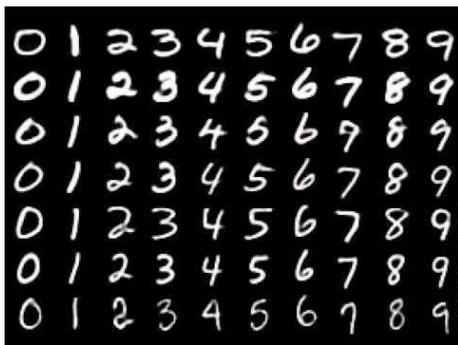


Рисунок 1 – Пример изображений из MNIST



Рисунок 2 – Пример фотографий из SVHN

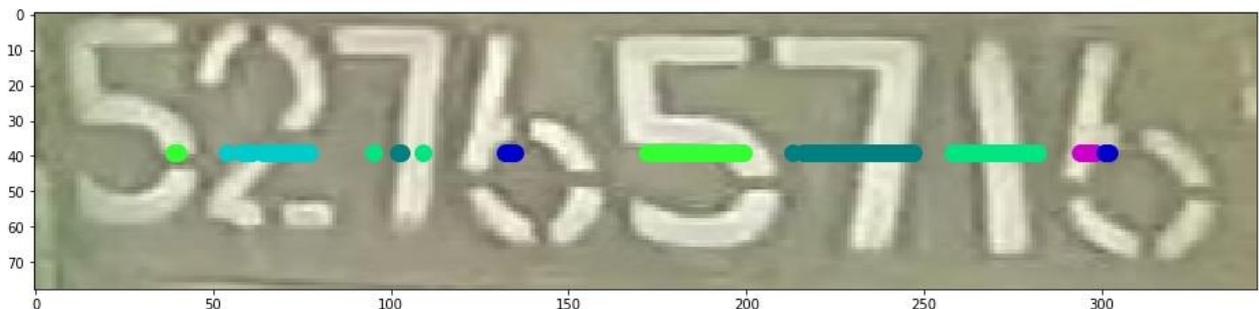


Рисунок 3 – Номер с разрывами в цифрах. Цветные точки – координаты центра “скользящего” окна, которые ставились в случае, если вероятность наличия цифры в окне превышает 0.8

Последним был опробован алгоритм детекции YOLO (рисунок 4), обученный на наборе данных SVHN. С целью повышения точности работы модели в фотографиях увеличивалась контрастность и устранялся шум с помощью алгоритма кластеризации k-means. Необходимая

точность работы модели не была достигнута из-за переобучения. Кроме того, невысокая точность YOLO обусловлена еще и тем, что этот алгоритм относится к одноэтапным алгоритмам распознавания.

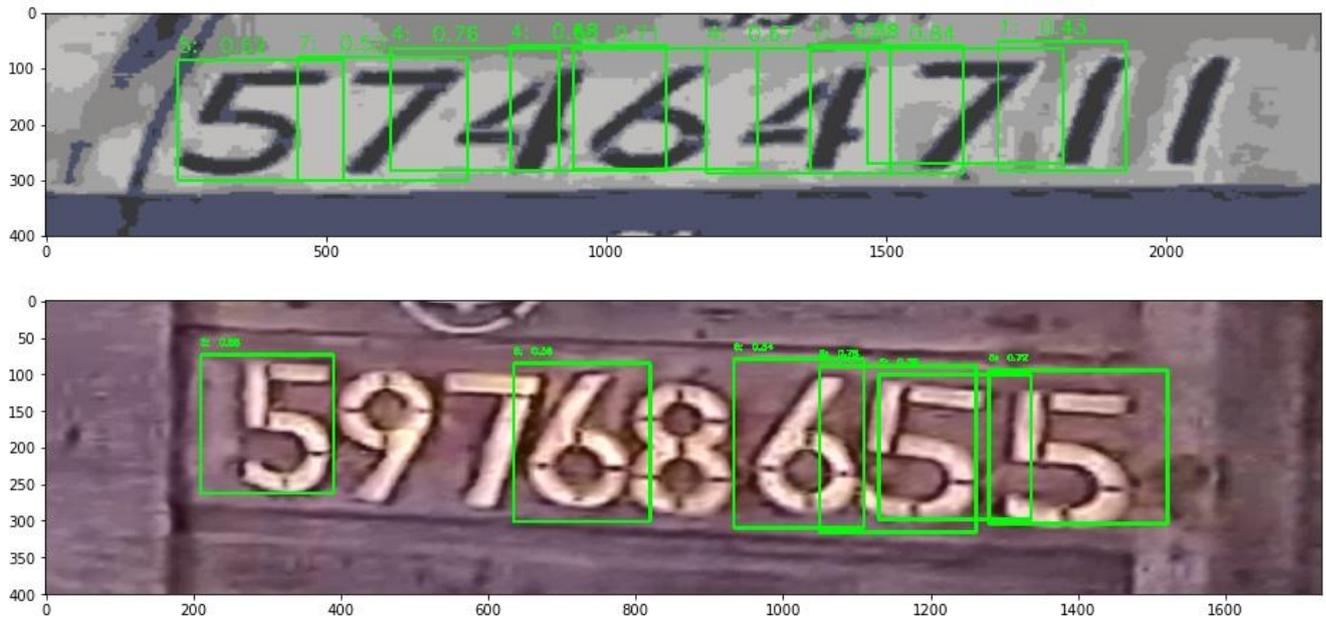


Рисунок 4 – Результат работы алгоритма YOLO

В работе было опробовано несколько моделей и алгоритмов для решения задачи распознавания номеров железнодорожных вагонов. Лучшие результаты были получены с помощью алгоритма YOLO, но и его качества распознавания оказались недостаточно для применения в реальной системе автоматизации взвешивания. Однако рассмотренные алгоритм одноступенчатой сверточной ИНС (YOLO) и алгоритм быстрой двухступенчатой сверточной ИНС (F-RCNN) определили направление для решения поставленной задачи. Поэтому планируется дальнейшее изучение и применение различных моделей, которые позволят решить задачу автоматического распознавания номеров вагонов.

**Список использованных источников:**

1. Деменковец, Д.В., Буйко, А. О. Библиотека БГУИР [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://libeldoc.bsuir.by/handle/123456789/32062>
2. Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi. Библиотека Cornell University [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://arxiv.org/abs/1506.02640>
3. Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, Piotr Dollár. Библиотека Cornell University [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://arxiv.org/abs/1708.02002>
4. Серебряная, Л. В., Третьяков, Ф. И. Библиотека БГУИР [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://libeldoc.bsuir.by/handle/123456789/10713>

## ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ ПЕРСОНАЛИЗИРОВАННОГО ПОДБОРА ПОДАРКОВ

Булойчик А.С.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Куликов С.С. – к.т.н., доцент

Во многих культурах принято дарить друг другу подарки. И иногда на выбор подарка уходит существенное количество времени, так как необходимо учитывать много фактов. Поэтому и была поставлена цель разработки приложения для персонализированного подбора подарков, которое сможет помочь пользователям быстрее и качественнее подбирать подарки конкретному человеку.

Основной задачей приложения является помочь пользователю подобрать подходящий подарок для конкретного человека, затратив при этом как можно меньше времени.

В настоящее время множество интернет-магазинов и сайтов подарков используют такие методы, как:

- 1) разбивание подарков на категории и подкатегории;
- 2) поиск по названиям;
- 3) поиск по отдельным характеристикам товаров;
- 4) создание категорий подарков определенной тематики: к событию, по полу и т.д.

Однако это не сильно упрощает задачу, так как необходимо просмотреть огромное количество подарков, чтобы найти подходящий. Кроме того, такие сайты предоставляют только свою продукцию, поэтому нужно просмотреть и не один интернет-магазин.

После анализа выше описанного было принято, что приложение должно:

- 1) предоставлять пользователю большую базу данных подарков из разных интернет-магазинов;
- 2) осуществлять подбор по характеристикам конкретного человека;
- 3) иметь возможность сохранять данные о подаренных человеку подарках, для того чтобы уменьшить диапазон поиска, а так же сохранять уже введенные данные о человеке для упрощения подбора подарков в следующие разы.

В реализации первого и третьего пунктов нет каких-то особенностей, а вот второй необходимо рассмотреть подробнее.

Для начала необходимо определить информацию, которая будет использоваться для подбора: пол, возраст, тип отношений (брат, друг, бабушка), событие, на которое подбирается подарок, увлечения. Для формирования рекомендаций по первым четырем критериям возможно просто увеличения приоритета для подарка по соответствующему критерию каждый раз, когда пользователь выбирает подарок для человека. Таким образом, если подарок подошел конкретному человеку, значит данный подарок подходит всем его характеристикам. И при подборе для человека со схожими характеристиками будет предложен этот подарок.

Однако в случае увлечений такой способ не получится использовать, т.к. если подарок связан с каким-то одним увлечением, совсем не значит, что он подойдет и для других увлечений этого человека. Одним из подходов для решения данной задачи является построение графа интересов в социальных сетях, где на основе информации о человеке формируются рекомендации друзей[1]. Вершинами такого графа будут увлечения и категории подарков. В графе могут быть различные типы связей: категория подарка – категория подарка, увлечение – категория подарка, увлечение – увлечение. Граф так же может быть взвешенным, где вес ребра будет означать силу взаимосвязи между вершинами. И изначально взаимосвязь между вершинами устанавливается бесконечно большим числом, а далее если будет обнаружена связь между ними, то значения веса ребра между вершинами будет уменьшено. И уже на основе этого графа будут учитываться увлечения.

При предоставлении пользователю результатов необходимо исключить подарки, уже подаренные данному человеку, либо даже исключить подкатегории этих подарков.

Пользователь так же должен иметь возможность исключать подарки из рекомендаций для конкретного человека. Это значит, что если пользователь точно знает, что данный подарок не подойдет человеку, он может добавить его в черный список, и этот подарок никогда больше не будет отображаться в рекомендациях для данного человека.

Таким образом алгоритм персонализированного подбора подарков для состоит в следующем:

- 1) формирование списка подарков согласно характеристикам человека в порядке убывания значения этих характеристик у подарков;
  - 2) исключение уже подаренных подарков и подарков из черного списка данного человека.
- Таковы основные особенности приложения для персонализированного подбора подарков.

**Список использованных источников:**

1. Medium [Электронный ресурс]. Дата обновления: 28.05.2014. URL: <https://medium.com/eggheadoscience/778c92d20cea>

## ПРИМЕНЕНИЕ МОБИЛЬНЫХ УСТРОЙСТВ ДЛЯ КРИПТОГРАФИЧЕСКИХ ВЫЧИСЛЕНИЙ

Грачев Я.Ю.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Петровский Н.А. – к.т.н., доцент

Постоянно увеличивающаяся производительность мобильных процессоров и устройств позволяет применять их для более широкого круга задач, в том числе криптографических вычислений. В перспективе неоднородные распределенные вычислительные системы на основе мобильных устройств позволят достичь близкой к x86 архитектуре производительности вычислений.

Вычисление хэш-функции занимает достаточно большое время относительно передачи и обработки необходимых для осуществления этого вычисления данных. Таким образом возможно применение мобильных устройств для осуществления криптографических вычислений. При работе с распределенной неоднородной вычислительной системой, которая основана на множестве мобильных ARM устройств, возможно достижение близкой к x86 производительности [1].

В качестве примера для подтверждения возможности и практического смысла осуществления криптографических вычислений с применением мобильных устройств рассматривается обратная криптографическая задача для хэш-функции MD5. Наилучшим способом является атака по нахождению прообраза – поиск сообщения с заданным значением хэша. Но в действительности такая атака не является практически пригодной, так как на практике она не может быть проведена за разумное время при разумных затратах ресурсов. Поэтому, исключая радужные таблицы, единственным вариантом является коллизионная атака [2]. В качестве мобильных устройств используются Xiaomi Redmi Note 5A (ARM Qualcomm Snapdragon 425), Samsung Galaxy Note II (ARM Samsung Exynos 4412), Nexus 5X (эмулятор, x86 Intel i7-6800k).

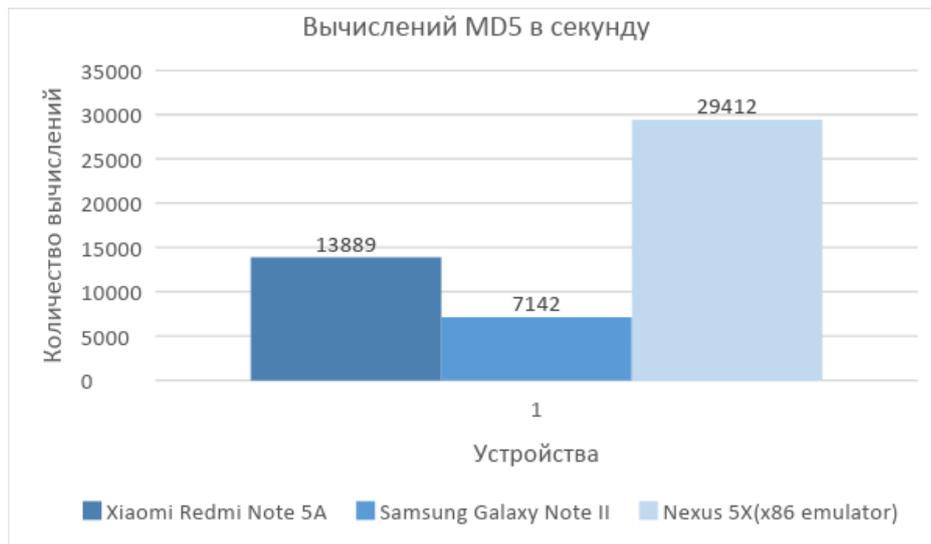


Рисунок 1 – Сравнение вычислений MD5 в секунду для разных устройств

В результате, при анализе количества вычислений MD5 в секунду (рисунок 1) на одинаковых для всех устройств, но разных на каждой итерации, данных наиболее эффективным является эмулируемое устройство на основе x86 процессора, однако современный, но не самый производительный в своем классе, ARM процессор от устройства Xiaomi Redmi Note 5A позволяет считать MD5 хэши всего в 2.1 раза дольше, в то время как более старый Samsung Galaxy Note II еще более значительно уступает x86, но тем не менее показывая положительную и значительную динамику роста производительности ARM устройств и, как следствие, подтверждение возможности и практического смысла осуществления криптографических вычислений на мобильных ARM устройствах, особенно при их объединение в неоднородные распределенные вычислительные системы.

### Список использованных источников:

1. Грачев, Я. Ю. Неоднородная распределенная вычислительная система / Я. Ю. Грачев // Компьютерные системы и сети: материалы 54-й научной конференции аспирантов, магистрантов и студентов, Минск, 23 – 27 апреля 2018 г. / Белорусский государственный университет информатики и радиоэлектроники. – Минск, 2018. – С. 58 - 59.

2. Rogaway, P. Cryptographic Hash-Function Basics: Definitions, Implications, and Separations for Preimage Resistance, Second-Preimage Resistance, and Collision Resistance. Fast Software Encryption / P. Rogaway, T. Shrimpton. – Berlin; Springer-Verlag. 2004, – 371-388 p.

## PLACE – ЛЕГКИЙ ПОИСК ПАРКОВОЧНЫХ МЕСТ

*Деменковец Д.В., Прудников В.М., Ивончик К.С., Бондарь Е.А.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Леванцевич В.А. – ст. преподаватель*

Цель данной работы заключается в решении проблемы поиска парковочных мест. В ходе работы было разработано мобильное приложение, которое обладает достаточным функционалом для поиска свободного парковочного места. Кроме этого, программное средство позволяет всегда быть в курсе того, что происходит с автомобилем в текущий момент времени.

В наши дни парковка автомобилей во дворе, у дома или офиса становится проблемой не меньшей, чем городские пробки. Многие, подъезжая домой, вынуждены искать свободное место в своем или соседних дворах. Это отнимает немало времени.

На данный момент в Минске приблизительно 940 тысяч автомобилей. Кроме этого, еще примерно 5-6 тысяч приезжают в город. Однако, по статистике, парковочных мест гораздо меньше. Пользователи автомобилей не очень любят платить за место на платной парковке, и, к сожалению, они не всегда находятся рядом. По статистике каждый водитель тратит примерно 5-10 минут своего времени для поиска парковочного места. Если учесть факт того, что мы это делаем примерно 2 раза в день, то за год в сумме может составить около 70-80 часов.

Для решения данной проблемы было разработано приложение Place, которое обеспечит легкий поиск парковочных мест, надежную систему безопасности, а также навигацию до свободного места на парковке.

Система состоит из следующих компонентов: одна или несколько камер, установленных так, чтобы просматривать всю территорию парковки; маршрутизатор, который отвечает за связь с интернетом; сервер, который принимает данные о местоположении и уровне доступа пользователей; и, непосредственно, само мобильное устройство.

После того, как пользователь выберет нужное ему место парковки, данные о его местоположении и правах доступа будут отправлены на сервер, где из всего множества видеопотоков, ему будут отправлены те, которые он запросил. Далее, на экране своего смартфона пользователь сможет увидеть трансляцию с парковки в реальном времени, убедиться в наличии свободного места, проложить маршрут и двигаться в заданном направлении.

После добавления технологии искусственного интеллекта в приложение, появится возможность проложить маршрут к свободному парковочному месту. Также появится возможность подсчета и выделения парковочного места, а также подсчитать их количество.

Установив мобильное приложение на свой смартфон, пользователь будет иметь возможность оформить платную подписку для пользования данной системой. При входе в приложение пользователь сталкивается с простым и интуитивно понятным интерфейсом, который включает в себя 3 секции:

- 1) карта с доступными ему парковками;
- 2) обзор ближайшей от текущего местоположения доступной парковки;
- 3) навигация до выбранной парковки.

Таким образом, разработанная система позволит заблаговременно оценить ситуацию на парковке и целенаправленно двигаться к свободному месту, используя встроенную систему навигации в приложении. При этом, пользователь сэкономит время и нервы, которые он мог потратить на поиск незанятого места. С точки зрения безопасности, система Place позволит всегда быть в курсе того, что происходит с вашей машиной в режиме реального времени

Дополнительно приложение может способствовать разгрузке дворов. Place поможет заранее определить, есть ли свободные места во дворах и стоит ли туда ехать. Увидев на экране своего смартфона, что все парковочные места уже заняты, водитель может попытаться найти иное свободное парковочное место. Администрация города также будет заинтересована в разгрузке дворов. Коммунальные и экстренные службы смогут более эффективно работать.

### Список использованных источников:

1. Documentation for app developers [Электронный ресурс] // developer.android.com : Сайт разработчика URL: <https://developer.android.com/docs>(дата обращения: 19.12.2018).
2. Apple Developer Documentation [Электронный ресурс] // developer.apple.com : Сайт разработчика URL: <https://developer.apple.com/documentation/>(дата обращения: 21.12.2018).
3. Escam [Электронный ресурс] // escam.cn : Сайт разработчика URL: <http://www.escam.cn/>(дата обращения: 05.02.2019).
4. В. Олифер, Н. Олифер. Компьютерные сети. Принципы, технологии, протоколы. 4-е издание. [Электронный ресурс] // rulit.me : Сайт разработчика URL: <https://www.rulit.me/books/kompyuternye-seti-principy-tehnologii-protokoly-read-467113-1.html>(дата обращения: 14.12.2018).

# ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИИ БЛОКЧЕЙН ДЛЯ АВТОМАТИЗАЦИИ УПРАВЛЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫМИ АКТИВАМИ

*Довыдёнко М.А.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Петюкевич Н.С. – м. т. н., ассистент*

Интеллектуальные активы – экономические ресурсы организации, обладающие искусственно созданными свойствами редкости и исключительности и способные приносить экономические выгоды [1]. Эффективное использование этих ресурсов является важной задачей для любой организации. Разработанная система, призвана автоматизировать основные процессы управления патентами: учет, продажу и лицензирование. Это достигается с использованием блокчейна – выстроенной по определенным правилам, распределенной, непрерывной цепочки блоков, содержащих информацию. Использование блокчейна позволяет создать децентрализованное, доверенное хранилище, в котором каждый участник проверяет блоки, созданные остальными.

Цель каждой организации – получение прибыли, поэтому управление активами является одной из важных задач. Управление материальными и интеллектуальными активами имеет общую цель, но сами активы кардинально различаются:

- по механизмам ценообразования;
- по праву собственности;
- по правовой защите.

Наиболее важным различием является механизм ценообразования: рынки материальных активов хорошо известны и организованы, а в случае интеллектуальных активов – уникальны для каждого изобретения и в большинстве случаев неликвидны. Среди других уникальных критериев следует выделить условную собственность, то есть возможность использования актива неограниченным числом лиц, и ограниченную правовую защиту (по сроку и области действия).

Далее, говоря об интеллектуальных активах, будет использоваться термин «патент» – один из наиболее формализованных видов активов, предоставляющих исключительные права интеллектуальной собственности. Существуют следующие способы получения прибыли от патента:

- производство изобретения;
- патентная защита, то есть получение компенсации за нарушение патентных прав;
- продажа лицензий;
- продажа прав на патент.

Сейчас становится популярным бизнес, основной деятельностью которого является патентная защита. Компании, которые занимаются подобной деятельностью, также называют «непрактикующими организациями». Их единственными активами являются патенты, а способом монетизации – судебные иски [2].

В связи со сложившимися особенностями рынка продажа и лицензирование патентов является редкостью. Используя современные технологии, можно повысить количество операций с патентами. Блокчейн предлагает общедоступное хранилище информации о патентах и их истории. Имея доступ к этим данным, внешние системы могут предлагать различные виды услуг: аналитику, автоматизацию документооборота, упрощение процессов оплаты, поиск покупателей.

Существует два вида блокчейн-сетей: публичные и частные. Для подключения к частной сети требуется подтвердить свою личность, что позволяет верифицировать права пользователей при управлении патентами. Наиболее популярной технологией для создания частных блокчейн-сетей является Hyperledger Fabric. Основной единицей такой сети является «организация», которая в большинстве случаев совпадает с компанией из реального мира. Организация может по правилам, описанным заранее, изменять данные в сети, а также читать данные, к которым имеет доступ [3]. Хотя не существует четких различий между организациями, по целям и способам получения прибыли можно условно выделить следующие их типы:

- организации, предоставляющие аналитику;
- организации, владеющие патентами;
- патентные бюро – организации, предоставляющих списки и информацию о патентах;
- организации, автоматизирующие документооборот и получающие прибыль от сделок.

Таким образом, при создании доверенной платформы по управлению патентами можно достичь повышения ликвидности рынка и эффективности использования интеллектуальных активов.

## **Список использованных источников:**

1. Интеллектуальные активы [Электронный ресурс]. – Режим доступа: [https://ru.wikipedia.org/wiki/Интеллектуальные\\_активы](https://ru.wikipedia.org/wiki/Интеллектуальные_активы)
2. 4 способа заработать на патентах [Электронный ресурс]. – Режим доступа: <https://rb.ru/opinion/patent-trolls/>
3. A Blockchain Platform [Электронный ресурс]. – Режим доступа: <https://hyperledger-fabric.readthedocs.io/en/release-1.4/>

## АЛГОРИТМ АВТОМАТИЧЕСКОГО ВЗВЕШИВАНИЯ ЖЕЛЕЗНОДОРОЖНЫХ ВАГОНОВ

Деменковец Д.В., Дубицкая К.Ю.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Бранцевич П.Ю. – к.т.н., доцент

В докладе рассматривается структура программно-аппаратного комплекса, обеспечивающего взвешивание железнодорожных вагонов. Также описывается алгоритм автоматического взвешивания железнодорожных вагонов, перевозящих нефтепродукты.

Для правильной эксплуатации подвижного состава, организации движения и коммерческих расчетов большое значение имеет учет перевозимых по железной дороге грузов. Данный процесс играет значимую роль в транспорте углеводородов, сырой нефти и автомобильного топлива. Таким образом, проблема организации процесса взвешивания и учета железнодорожных вагонов является актуальной и важной.

Процесс взвешивания железнодорожных вагонов, перевозящих автомобильное топливо или сырую нефть, может быть длительным процессом, особенно при неполном заполнении вагона (около 70%). Этот процесс требует непосредственного участия оператора-весовщика, составителя поездов и машиниста локомотива. При статическом способе взвешивания тратится в среднем от 2 до 20 минут на установку вагона на весовые платформы вагона и ожидание стабилизации колебаний жидкого груза в цистернах. Кроме того, на некоторых предприятиях прием и выгрузка вагонов осуществляется в темное время суток, что также затрудняет процесс взвешивания. В докладе рассматривается пример организации учета взвешивания вагонов, перевозящих нефтепродукты ИООО «ЛУКОЙЛ Белоруссия».

Целью автоматизации взвешивания является разработка и внедрение программно-аппаратного комплекса (ПАК), позволяющего осуществлять данный процесс без непосредственного участия оператора-весовщика в автоматическом режиме. Взвешивание происходит при установке вагона на измерительную платформу маневровым локомотивом с помощью составителя поездов. Параметры взвешивания поступают от весоизмерительного индикатора на персональный компьютер оператора посредством протоколов TCP/IP поверх интерфейса Ethernet. Проблема уведомления составителя поездов о статусе процесса взвешивания решается отображением команд и управляющей информации на информационном табло. Кроме того, на нем отображаются результаты взвешивания. Дополнительно в базу данных сохраняются фотоизображения вагона с внешних IP-камер видеонаблюдения (фотофиксация процесса взвешивания).

Структурная схема ПАК автоматизации процесса взвешивания железнодорожных вагонов представлена на рисунке 1:

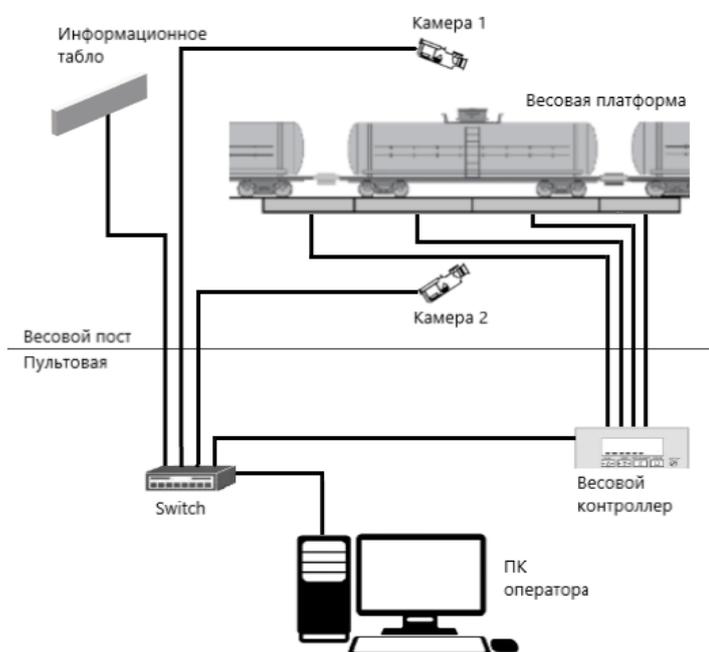


Рисунок 1 – Структурная схема ПАК

В автоматическом режиме данные циклически запрашиваются через определенный промежуток времени по срабатыванию таймера. Основной проблемой при разработке алгоритма автоматического взвешивания железнодорожных вагонов является определение момента добавления в базу данных новой карточки взвешивания (нового вагона). Дублирование данных или пропуск записи, соответствующей некоторому вагону, недопустимы, так как могут стать причиной серьезных нестыковок в финансовой отчетности.

В связи с этим запись параметров измерения производится единожды для каждого вагона при получении стабильного значения веса. При этом подача следующего вагона определяется отсутствием стабильности в течение некоторого заданного интервала времени. Это исключает дублирование значений при помехах связи, сбоях в работе весоизмерительного индикатора и при рывках, связанных с работой машиниста локомотива.

2: Диаграмма состояний режима автоматического взвешивания вагонов представлена на рисунке

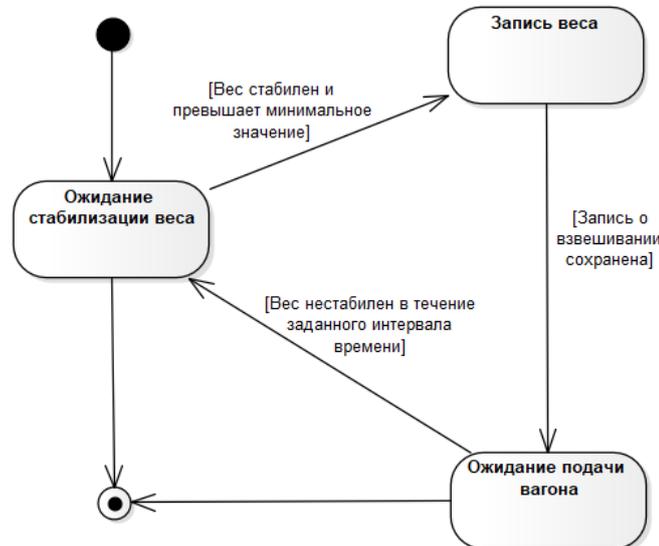


Рисунок 2 – Диаграмма состояний режима автоматического взвешивания

При срабатывании таймера автоматического режима взвешивания через определенные промежутки времени у весоизмерительного индикатора запрашиваются данные. Если значение веса превышает установленный минимум, анализируется полученный состояние стабилизации веса. Каждый раз при получении нестабильного значения от весоизмерительного индикатора запускается таймер на три секунды, если он не был запущен ранее. Значение временного интервала получено опытным путем и может быть изменено в зависимости от конкретной модели весоизмерительного индикатора. По срабатыванию таймера через указанный промежуток времени устанавливается флаг нестабильности. В случае, когда получен бит стабильности веса, таймер останавливается (если он был запущен). После этого анализируется значение флага нестабильности. Он показывает, предшествовал ли состоянию стабильности в данный момент период нестабильности длительностью от трех секунд и более. Этот интервал времени гарантирует, что промежуток нестабильности не обусловлен помехами связи и перебоями в работе индикатора, а вызван установкой следующего вагона на весовую платформу. На основе этого можно сделать вывод, что текущее стабильное значение относится к уже новому вагону, и выполняется сохранение новую записи о взвешивании.

Таким образом, вышеописанный алгоритм позволяет автоматизировать процесс взвешивания вагонов с нефтепродуктами, представляющих собой жидкие грузы. Это обеспечивает требуемый уровень надежности и точности сохраняемых данных и позволяет более эффективно организовать работу предприятия.

**Список использованных источников:**

1. Коваленко П.В., Клебанова М.Н. Проектирование и эксплуатация нефтебаз и нефтехранилищ // Уч. метод. комплекс для студ. спец. "Проектирование, сооружение и эксплуатация газонефтепроводов и газонефтехранилищ" – Новополоцк : ПГУ, 2006. – 340с.
2. Деменковец, Д. В. Система автоматической фото, видеофиксации и записи процесса взвешивания железнодорожных вагонов / Д. В. Деменковец // Новые горизонты – 2017 : сборник материалов Белорусско-Китайского молодежного инновационного форума, 2-3 ноября 2017 г. : в 2 т. – Минск : БНТУ, 2017. – Т. 1. – С. 22-24
3. Частное производственно-торговое унитарное предприятие «Саха-пром» - Автоматизация взвешивания железнодорожных вагонов [Электронный ресурс]~/ --- Режим доступа: <http://sakha-prom.by/smart/avtomatizatsiya-vzveshivaniya.html> --- Дата доступа: 02.03.2019.

# МЕТОД РАСПОЗНАВАНИЯ ЭМОЦИЙ ПО ИЗОБРАЖЕНИЮ НА ОСНОВЕ МАШИННОГО ОБУЧЕНИЯ

*Евтушенко Т.С.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Серебряная Л.В. – к.т.н., доцент*

Нейронные сети — это одно из направлений в разработке систем искусственного интеллекта. В основном они применяются для прогнозирования, распознавания образов, принятия решений, оптимизации и анализа данных. Распознавание эмоций по фотографии лица человека – одна из тех задач, для решения которых используются нейронные сети.

Актуальность выбранной темы обусловлена тем, что именно сейчас, с быстрым темпом интеграции различных информационных систем в повседневную жизнь, мы можем найти все больше и больше способов применения систем распознавания эмоций. Сейчас такие системы могут быть использованы в различных сферах жизни, в которых идет взаимодействие с людьми. Например, распознавание эмоций покупателей можно использовать для целевого маркетинга и для эффективного размещения товаров в магазинах, пользователей какого-либо интернет ресурса – для определения реакции человека на размещенный на данном ресурсе контент.

Задачей данной работы является изучение существующих алгоритмов распознавания лиц, выявление наиболее точных и надежных из них, реализация программного средства, использующего веб-камеру пользователя для получения изображения, алгоритм классификатора для выделения на нем лица и нейронную сеть для определения с ее помощью эмоции на изображении лица. В качестве исходных данных выступает набор данных для обучения модели, который содержит в себе фотографии со всеми, нужными нам, человеческими эмоциями: злость, удивление, счастье, грусть, отвращение, страх, нейтральность. На основе данного набора мы сможем обучить нашу модель и интегрировать её в систему. Результатом работы программного средства будет являться вывод об эмоции лица на изображении, считанного через веб-камеру. Результирующая эмоция будет получаться как эмоция с самой большой вероятностью присутствия на изображении из всех рассматриваемых эмоций.

Задачу по распознаванию эмоций можно разделить на 2 этапа: обучение модели нейронной сети и распознавание эмоций с помощью этой обученной модели.

В качестве нейронной сети для идентификации эмоции на изображении лица была выбрана сверточная нейронная сеть. Сверточная нейронная сеть — это специальная архитектура искусственных нейронных сетей, цель которой - эффективное распознавание образов. Данная сеть относится к технологиям глубокого обучения, имеет однонаправленную, многослойную структуру. Сверточная нейронная сеть берёт исходное изображение и пропускает его через ряд сверточных, нелинейных слоев, слоев объединения, полносвязных слоёв и генерирует вывод в виде класса или вероятности классов, которые лучше всего описывают изображение [1]. Для обучения данной сети было использовано обучение с учителем и метод обратного распространения ошибки. На вход сети передавалось множество изображений лиц с определенной эмоцией и информация о данной эмоции. Так нейронная сеть обучалась на множестве изображений и типах эмоции, пока не была достигнута необходимая вероятность выдачи нейронной сетью правильных результатов.

Для распознавания эмоций после обучения нейронной сети используются изображения, считанные через веб-камеру пользователя. Перед тем, как передать изображение обученной модели нейронной сети для идентификации вероятности эмоций, стоит убедиться, содержит ли исходное изображение лицо человека и если да, то передать на вход модели фрагмент исходного изображения, содержащий только лицо. Для данной задачи хорошо подходит метод Виолы-Джонса — алгоритм, позволяющий обнаруживать объекты на изображениях в реальном времени, он имеет низкое количество ложных результатов и высокую точность обнаружения искомых объектов [2]. С помощью данного алгоритма в исходном изображении выделяется лицо человека, передается на вход нейронной сети, а нейронная сеть возвращает вероятность каждой эмоции на переданном изображении лица.

Эксперименты показали, что программное средство гарантирует неплохую достоверность (около 80%) распознавания эмоций на изображениях с хорошей освещенностью и поворотом лица не более, чем на 30 градусов, и очень недостаточную (около 50%) на остальных изображениях. Полученное программное средство может быть доработано путем дальнейшего обучения модели нейронной сети, что позволит с большей точностью определять вероятность эмоций.

#### **Список использованных источников:**

1. Convolutional neural network [Электронный ресурс]. – 2019. – Режим доступа: [https://en.wikipedia.org/wiki/Convolutional\\_neural\\_network](https://en.wikipedia.org/wiki/Convolutional_neural_network)

## ГРУППОВАЯ РОБОТОТЕХНИКА И ИДЕИ ЕЕ ПРИМЕНЕНИЯ

*Ермолович В.О.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Хмелева А.В. – к.т.н., доцент*

В докладе рассматриваются понятие и предмет групповой робототехники, выделены ее достоинства и недостатки, приводятся идеи ее практического применения.

Групповая робототехника – направление робототехники, в основе которого лежит изучение поведения группы роботов, взаимодействующих между собой и с окружающей средой. Причем роботы, используемые в групповой робототехнике, имеют достаточно простое строение и направлены на решение конкретной, специфичной задачи. Коллективное поведение появляется благодаря коммуникации роботов друг с другом за счет чего достигается эффект параллелизма при решении задач, на которые заточены роботы. Особенности рассматриваемого вида робототехники является децентрализованный подход в управлении роботами, практически полное отсутствие синхронности идентичность и сравнительная простота производства используемых роботов [1]. Все это указывает на отсутствие некоего особенного, главного робота и на взаимозаменяемость роботов, т.к. каждый робот способен выполнять один и тот же набор функций. Благодаря отсутствию главного робота и некоторой избыточности достигается высокая отказоустойчивость. Каждый робот взаимодействует только со своими непосредственными соседями. Кроме того, он локально хранит только ту информацию, которую сам получил со своих датчиков. Вторым преимуществом такой децентрализованной системы является гибкость: при отказе одного робота, любой из оставшихся может завершить задачу. Система роботов может компенсировать некоторые аппаратные ограничения за счет их способность к кооперации. Такими ограничениями могут являться слабый электропривод отдельно взятого робота при решении задачи передвижения предмета, но скооперировавшись, роботы суммарным усилием могут передвинуть предмет. Третьим преимуществом является масштабируемость. Используемые методы для управления группой роботов продолжают быть эффективными независимо от количества добавляемых в группу роботов. Это возможно благодаря тому, что роботы взаимодействуют только со своими соседями. Соответственно такие методы, как широковежательная рассылка сообщений на всю группу роботов, неприменимы, т.к. не являются масштабируемыми. Неотъемлемой частью групповой робототехники являются методы коммуникации между роботами. Выделяют явную и неявную коммуникацию. Явная коммуникация подразумевает наличие отправителя, получателя, явного канала между ними, по которому передается сообщение. В качестве неявной коммуникации можно привести в пример восприятие одним роботом присутствие другого робота и изменение своего состояния. Задачи, которые могут решаться с помощью групповой робототехники включают:

- агрегация и кластеризация роботов (является полезным с точки зрения исследования окружающей среды);
- рассредоточение роботов (является полезным для мониторинга на больших площадях);
- самоорганизация роботов (например, для рассмотренной выше задачи перемещения предмета).

Далее будут рассмотрены несколько идей применения групповой робототехники. Первым применением является оценка качества почвы на полях. Роботы будут представлять собой небольшие передвижные машины с модулями для оценки качества почвы. С помощью группы роботов будут решаться задачи рассредоточения, сбора информации о качестве почвы, выделение наиболее плодородных участков или, наоборот, участки почв с низкими показателями плодородия, которые могут быть заражены и потребовать вмешательство человека. Еще одним применением является мониторинг чистоты акваторий на больших площадях. В таком случае робот может представлять собой буй, в нижней части которого будут находиться датчики качества воды, а в верхней части, расположенной над уровнем воды, передатчик для связи с соседними роботами. Главной проблемой при решении данной задачи мониторинга чистоты акватории является большая площадь самой акватории, что означает либо частичную, либо полную потерю соединения с интернетом. Рассматриваемая система не требует подключения к интернету, достаточно

подключится к наиболее близкому роботу с учетом реализованного протокола общения между устройствами.

Подводя итог, отметим достоинство групповой робототехники как децентрализованной системы, хорошо масштабируемой и работающей на больших площадях, но при этом не являющейся достаточно универсальной и требующей отдельной реализации для решения частных задач.

**Список использованных источников:**

1. Hamann, H.: Swarm Robotics: A Formal Approach. Springer International Publishing AG 2018

## ОДНОПЛАТНЫЕ КОМПЬЮТЕРЫ В СИСТЕМАХ ПОЛУЧЕНИЯ ПОТОВОКОВЫХ АУДИОДАННЫХ

*Ермолович И.А., Пилинко Н.А., Фадеева Е.Е*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Леванцевич В.А. – м.т.н., старший преподаватель*

Описана реализация автономного приемника потоковых аудиоданных для воспроизведения передач с интернет радиостанций на базе одноплатного компьютера Raspberry Pi и операционной системы RaspbianOS.

В настоящее время широко распространено вещание радиостанций в УКВ диапазоне с весьма хорошим качеством. Однако у эфирных радиостанций есть несколько недостатков: во-первых, это неуверенный прием сигнала в некоторых районах, во-вторых ограниченный выбор радиостанций.

Целью проекта стало создание автономного приемника потоковых аудиоданных, способного воспроизводить вещание с интернет радиостанций on-line [1]. Для реализации проекта был выбран полярный микрокомпьютер Raspberry Pi, причем можно использовать любую модель: B, B+, 2 или Zero [2]. Подключение к сети Интернет может быть как проводным, так и по WiFi. В качестве операционной системы выбрана RaspbianOS как наиболее оптимизированная система для данного семейства микрокомпьютеров, позволяющая достичь максимальной производительности.

Система имеет клиент-серверную архитектуру. Сервер построен на связке LAMP (Linux, Apache, MySQL, PHP), клиент использует Python и Bash для управления и MPD (Music Player Daemon) для получения мультивещательного потока данных, а также имеет MPC (Media Player Classic) для упрощения управления MPD системы (Рис 1).

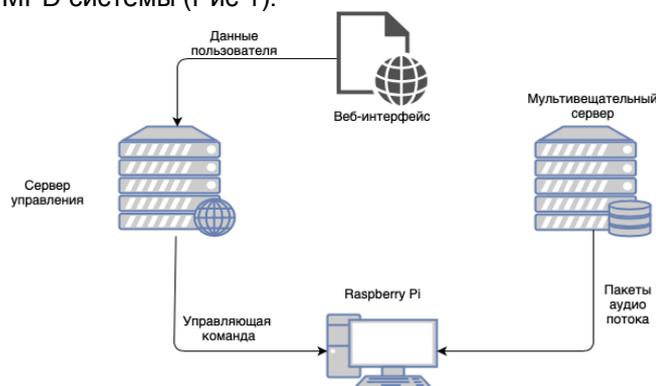


Рисунок 1 - Структура клиент-серверной системы

Программа на устройстве использует системные функции и не требует дополнительных библиотек для своей работы. Установка сделана на основе Bash-скрипта и самих файлов программы. По запуску Bash-скрипт помещает все необходимые файлы в загрузочные сектора системы и производит настройку программы для автозапуска при включении системы. Для работы графической части используется веб-страница, запускаемая в веб-браузере Chromium на устройстве. Данная связка позволяет производить улучшения системы без необходимости обновления ПО на самом устройстве. Также HTML+CSS предоставляет лучшие инструменты для проектирования пользовательского интерфейса, чем библиотеки поставляемые для Python. Программа на устройстве делает запрос на сервер управления каждые 300мс. Сервер сообщает устройству необходимые команды, которые оно впоследствии выполняет. Используется безопасный протокол HTTPS, поэтому дополнительного шифрования трафика не производится.

Список использованных источников

1. Как работает Интернет-радио <https://radio-tochka.com>
2. Пети, В. Микрокомпьютеры Raspberry Pi. Практическое руководство/ В.А. Петин – БХВ-Петербург, 2017.-240с

## ВЕБ-ПРИЛОЖЕНИЕ СОЗДАНИЯ И ПРОВЕДЕНИЯ ОПРОСОВ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИЙ REACT, REDUX, ASP.NET WEBAPI 2

Ермошин М.А.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Болтак С.В. – ассистент

В современном мире многим компаниям для того, чтобы оставаться конкурентоспособными необходимо собирать отзывы своих клиентов и быстро реагировать на их изменения. Для этого есть много способов. К ним относятся такие как эксперимент, наблюдение, изучение различных документов. Но один из лучших способов сбора этой информации – проведение опроса.

Опрос – это чаще всего небольшой перечень вопросов, способных дать необходимую информацию. Для улучшения качества результатов вопросы могут разбиваться по темам и вводиться вопросы-фильтры для отсеивания анкет, заполненных ненадлежащим образом.

Вот некоторые преимущества онлайн-опросов:

- экономия ресурсов (денег, времени и трудозатрат);
- широта охвата (преодоление границ и расстояний, доступ к различным социальным группам и сообществам);
- оставляют возможность выбора удобного времени и места участия, могут быть завершены в любое удобное для респондента время;
- релевантность (самостоятельность) коммуникации, т. е. более низкий уровень влияния интервьюера на респондента, возможность давать более развёрнутые ответы;
- широта охвата предметных полей (возможность изучать деликатные и закрытые для публичного обсуждения темы);
- возможность автоматической проверки анкет.

Однако пока что в онлайн-опросах так же присутствуют некоторые недостатки, которые являются критическими для некоторых методологий проведения опроса. Главный недостаток онлайн-опросов связан с проблемой обеспечения репрезентативности выборки. Во-первых, это отсутствие основы выборки. Эта проблема может быть успешно решена в исследованиях организаций с широкими сетевыми базами, а также при построении выборки по результатам оффлайн опроса. Во-вторых, проблема охвата, то есть неспособность используемой выборочной процедуры охватить реальную генеральную совокупность (то есть задать известную ненулевую вероятность попадания в выборку для каждой единицы совокупности). И, в-третьих, это не-ответы либо отказы от участия. Обычно успешно удаётся решить первые две [1].

В разрабатываемом приложении учтено большинство из преимуществ онлайн-опросов. Процедура создания опроса проста и интуитивно понятна. Создать опрос можно на любую интересующую создателя опроса тему. Сразу же после создания опрос публикуется, и респонденты-пользователи сайта смогут проходить опрос. Для приглашения человека поучаствовать в опросе необходимо любым способом (*email*, почта, звонок, факс и т. д.) переслать ему ссылку на опрос.

Стек технологий для приложения был выбран неслучайно. Для клиентской стороны приложения были выбраны такие библиотеки как *React* и *Redux*. Они легковесные, легко дополняемые другими модулями и просты в использовании. Так же их использование уменьшит количество работы, исполняемой на серверной части, что позволит снизить требования к аппаратному обеспечению.

*React* используется для генерации пользовательского интерфейса. В основе данной библиотеки лежит компонентный подход, т. е. вся разметка разбивается на много компонентов различного уровня. Каждый компонент отвечает только за одну определённую цель.

Клиентское приложение требует хранения его состояния (идентификатор пользователя, вошедшего в систему, промежуточные данные при создании опроса и т. д.). Так как в *React* нету необходимых компонент для хранения состояния, выбор пал на *Redux*. Эта библиотека предоставляет единственное хранилище состояния приложения, которое и является «единственным источником правды» для всего приложения.

Для серверной части приложения была выбрана технология .Net, а именно её фреймворк *WebApi 2*, который позволяет быстро создавать простой *restful* сервис.

В качестве базы данных выбрана *MSSQL* как входящая в комплект поставки *Microsoft Visual Studio Enterprise*. Для работы с базой данных используется *EntityFramework* с подходом *code first*.

**Список использованных источников:**

1. Девятко И.Ф., Онлайн исследования и методология социальных наук: новые горизонты, новые (и не столь новые) трудности/ Онлайн-исследования в России 2.0. М.: РИЦ «Северо-Восток», 2010. — С.17-31. — ISBN 978-5-9901939-1-8 [https://www.hse.ru/data/2013/01/23/1306499729/Online\\_Research\\_in\\_Russia\\_2\\_complete.pdf](https://www.hse.ru/data/2013/01/23/1306499729/Online_Research_in_Russia_2_complete.pdf).

## ИСПОЛЬЗОВАНИЕ АЛГОРИТМА PATCHWORK ДЛЯ РЕАЛИЗАЦИИ СИСТЕМЫ ВОДЯНЫХ ЗНАКОВ

Ждан В.А.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Ярмолик В.Н. – д.т.н., профессор

В работе рассматриваются угрозы интеллектуальной собственности, а также способы защиты от них. Особое внимание уделяется системам цифровых водяных знаков, а также использованию стеганографического алгоритма patchwork в рамках его применения для реализации системы водяных знаков.

Для программного обеспечения и других объектов интеллектуальной собственности актуальными являются следующие угрозы: несанкционированное использование, обратное проектирование и несанкционированная модификация.

Одно из направлений для ликвидации данных угроз – системы цифровых водяных знаков. Данное направление использует различные методы: графическая стеганография, текстовая стеганография, обфускация и другие.

Стеганография — это наука, которая занимается вопросами скрытой передачи информации путём сохранения в тайне самого факта передачи. Открытый текст, где скрыта зашифрованная стеганографическим алгоритмом информация, называется контейнером. Графическая стеганография использует в качестве контейнера графические файлы, например растровые изображения [1].

Алгоритм Patchwork является одним из базовых алгоритмов графической стеганографии. В его основе лежит статистический подход. Суть этого алгоритма состоит в следующем. При помощи криптостойкого генератора псевдослучайных чисел, используя заранее известный секретный ключ, выбираются два пикселя изображения. Затем значение яркости одного из них увеличивается или уменьшается на некоторое значение в зависимости от типа встраиваемой информации. Значение яркости другого – уменьшается или увеличивается на ту же величину соответственно. Процесс повторяется около 10000 раз. Значение приращения варьируется от 1 до 5 [2].

Пусть значения яркостей выбираемых на каждом шаге пикселей  $a_i$  и  $b_i$ , а величина приращения  $\delta$ . Тогда сумма разностей значений пикселей после  $n$  преобразований:

$$S_n = \sum_{i=1}^n [(a_i \pm \delta) - (b_i \mp \delta)] = \pm 2\delta n + \sum_{i=1}^n (a_i - b_i)$$

Математическое ожидание величины  $\sum_{i=1}^n (a_i - b_i)$ , то есть суммы разности значений пикселей в незаполненном контейнере равно нулю, и его значение в неизменном изображении будет много меньше чем  $n$ . Математическое ожидание величины  $S_n$  в заполненном контейнере будет равно  $\pm 2\delta n$ , а вычисленное значение  $S_n$  будет иметь большой соответствующий  $n$  порядок. В стегодекодере используя ключ проверяется значение  $S_n$ . Решение о значении встроенного бита выносится в зависимости от знака  $S_n$ . При положительной разности 1, а при отрицательной – 0 [3].

Таким образом, владелец может доказать свои интеллектуальные права, предъявив секретный ключ, который использовался для выборки изменяемых пикселей изображения.

Существует способы улучшения алгоритма для повышения его робастности. Например, можно использовать квадратные или прямоугольные блоки вместо отдельных пикселей изображения. Алгоритм Patchwork является достаточно стойким к операциям усечения, сжатия, изменения гистограммы изображения. Основным недостатком алгоритма – это неустойчивость к геометрическим преобразованиям: сдвигу, повороту, масштабированию. Другим недостатком является малая пропускная способность, что не дает возможности встраивания значительных по размеру цифровых водяных знаков. К примеру, в базовой версии алгоритма для передачи 1 бита скрытой информации необходимо порядка 20 000 пикселей. Таким образом в 20-мегапиксельном изображении можно передать только 1000 бит, то есть около 120 байт скрытого сообщения.

**Список использованных источников:**

1. Грибунин, В. Г. Цифровая стеганография / В. Г. Грибунин, И. Н. Оков, И. В. Туринцев. – М.: СОЛОН-Пресс, 2002. – 261 с.
2. Семёнов К. П. Алгоритмы встраивания цифровых водяных знаков в растровые изображения / К. П. Семёнов, П. В. Зайцев // Информационная безопасность регионов : научно-практический журнал. – 2011. – №1. – С. 46–50.
3. Bender W. Techniques for Data Hiding / W. Bender, D. Gruhl, N. Morimoto, A. Lu // IBM Systems Journal. – 1996. – Vol. 35.

## **МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ ПРОСМОТРА РАСПИСАНИЯ ОБЩЕСТВЕННОГО ТРАНСПОРТА НА ПЛАТФОРМЕ XAMARIN С ИСПОЛЬЗОВАНИЕМ УСТРОЙСТВА ТИПА «УМНЫЕ ЧАСЫ»**

*Жизневский В.С.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Трус В.В. – ст. преподаватель каф. ПОИТ*

Все большую популярность набирают устройства типа «умные часы». Некоторые современные модели являются многофункциональными устройствами под управлением полноценных операционных систем. Например, линейка Gear от компании Samsung под управлением операционной системы Tizen. Однако полностью свой потенциал «умные часы» раскрывают при работе в связке со смартфоном. В связи с этим большой перспективой обладают комплексные решения, в состав которых входят как приложение для смартфона, так и приложение для часов.

Не редки ситуации, когда необходимо на ходу составлять маршруты внутри черты города, или же иметь быстрый доступ к актуальной информации движения общественного транспорта, например знать ближайшее время прибытия на остановку автобуса по определенному маршруту. И на данный момент есть приложения для смартфонов, которые предоставляют функции для составления маршрутов, просмотра расписаний движения общественного транспорта. Но использование в некоторых случаях часов дает преимущества, т. к. часы являются более мобильным устройством по сравнению с телефоном. Одним из эффективных сценариев использования часов является отображение наиболее актуальной для пользователя информации. Однако редактирование и ввод информации лучше производить на смартфоне, т. к. он обладает более удобными средствами для ввода информации по сравнению со смарт-часами.

В рамках проекта было принято решение о создании мобильного приложения для просмотра расписания общественного транспорта с использованием устройства типа «умные часы». Помимо основного функционала, в которых входит отображение остановок общественного транспорта на карте, просмотр расписания общественного транспорта, добавление расписаний движения общественного транспорта через остановку в избранное, данное приложение будет подразумевать сценарий использования смарт-часов, а именно отображение расписаний движения общественного транспорта через остановку из списка избранного.

В качестве платформы для части приложения, которая будет работать на стороне смартфона, была выбрана платформа Xamarin, т.к. данная платформа позволяет создавать кросс-платформенные приложения для Android и iOS [1]. В качестве целевых устройств типа «умные часы» были выбраны устройства линейки Gear от компании Samsung под управлением операционной системы Tizen. Согласно отчету компании Strategy Analytics за 4-й квартал 2018 устройства компании Samsung входят в тройку по количеству проданных, что говорит о популярности моделей данных часов [2]. Для разработки части приложения, работающего на часах будут использованы инструменты Tizen SDK [3]. Тип приложения будет Tizen Web, т. к. приложения такого типа обладают наибольшей совместимостью с устройствами линейки Gear [4].

В качестве компонента отображения ближайшего времени отправления на смарт-часах будет использован виджет.

Предполагаемый сценарий создания и инициализации виджета:

Пользователь синхронизирует список избранного на телефоне и смарт-часах при помощи интерфейса Bluetooth, далее пользователь добавляет виджет в список активных. После, при нажатии на виджет, происходит инициализация, которая подразумевает выбор пользователем расписания движения общественного транспорта через остановку из списка избранного. В результате данных действий на экране виджета будет отображаться ближайшее время отправление транспорта с остановки по маршруту.

Таким образом, данный сценарий использования позволит ещё быстрее получать доступ к наиболее необходимой информации касательно расписаний движения общественного транспорта, что выгодно отличает данное приложение от аналогов.

**Список использованных источников:**

1. Документация по Xamarin [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://docs.microsoft.com/ru-ru/xamarin>.

2. Strategy Analytics Wearable Device Ecosystems (WDE) service report [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://www.strategyanalytics.com/strategy-analytics/blogs/devices/wearables/wearables/2019/02/28/apple-watch-captures-half-of-18-million-global-smartwatch-shipments-in-q4-2018>.

## ОБЗОР РЕШЕНИЙ СТАТИСТИЧЕСКОГО АНАЛИЗА КОДА

*Жух А.В.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Парамонов А.И. – к.т.н., доцент*

В данной статье рассмотрены проблемы, решаемые с помощью статического анализа кода, а также проанализированы характерные особенности статических анализаторов кода на примере компилятора Roslyn. Проблема статического анализа кода является актуальной, так как языки программирования, как и относящиеся к ним технологии, продолжают развиваться, масштабы проектов увеличиваются, вместе с ними и объем самого кода, а, следовательно, и кодревью усложняется. Статический анализ кода является неотъемлемой частью процесса разработки.

Статический анализ кода — анализ программного кода, который производится без его выполнения. Статический анализ необходимо рассматривать в качестве автоматизированного процесса обзора кода [1]. Кодревью является одним из самых полезных методов для выявления дефектов программного кода. В процессе обзора кода выявляются ошибки, которые могут привести к некорректному поведению. Основным недостатком кодревью является то, что данный процесс трудоёмкий и времязатратным, так как полностью производится человеком, а, следовательно, и дорогостоящий. Говоря о статическом анализе, следует подчеркнуть, что он позволяет найти большинство ошибок по мере их появления в коде, а, следовательно, и своевременно их исправить. Данная особенность приводит к улучшению качества самого кода. Следует учитывать и то, что чем раньше ошибка была выявлена, тем меньше будет стоить ее исправление. Статический анализ кода может быть использован, как в своём стандартном виде, с целью поиска дефектов кода, так и с целью обучения либо же более усложненный вариант - проверка кода с точки зрения архитектурного решения. Преимущества статического анализа кода в данных направлениях являются очевидными. Возможность поиска дефектов для них полностью сохраняется только будут усложнены правила распознавания того, что является дефектом.

На данный момент находится в использовании широкий спектр анализаторов кода на базе статического анализа, основная специфика которых заключается в том, что они заточены под конкретный язык программирования. Рассмотрим основные особенности реализации статических анализаторов для C# на примере компилятора Roslyn.

Roslyn — это не просто компилятор, а целая платформа, которая разрабатывается корпорацией Microsoft. [2] Она предоставляет возможности для разбора кода и его последовательного анализа. Roslyn также является основой для других анализаторов кода. С помощью данного компилятора можно производить детальный разбор кода для дальнейшего анализа всех поддерживаемых языковых синтаксических конструкций. Коданализ проходит в несколько этапов. Вначале идет сбор необходимых данных, что включает в себя получение проектного решения, затем компиляции и разбор полученных файлов, на основе чего строится неизменяемое дерево синтаксиса и семантическая модель. Roslyn строит дерево синтаксиса на основе программного кода, полученного из проектного файла, состоящее из синтаксических узлов, лексем и дополнительной информации. [3] Полученное дерево используется для анализа конструкций языка. По дереву синтаксиса и происходит перемещение во время анализа, которое осуществляется с помощью рекурсии. Синтаксические узлы содержат основные синтаксические языковые конструкции. На синтаксических узлах основываются

диагностические правила. Основной задачей анализатора кода становится обработка синтаксических узлов. У каждого узла есть свой тип, что упрощает навигацию по самому дереву. Лексемы — узлы дерева, неподлежащие дальнейшему разложению. К ним относят такие элементы, как идентификаторы, специальные символы, а также ключевые слова. Дополнительная информация — это такие узлы дерева, которые после компиляции не будут прорасти в IL-код, например, комментарии. Семантическая модель создается в результате компиляции с использованием синтаксического дерева. Семантическая модель содержит в себе информацию о всех объектах. Более того, оно хранит в себе и информацию о типах этих объектов, что даёт возможность проводить более комплексный анализ.

Процесс статического анализа кода всё еще не способен справляться с более сложными задачами, к примеру, прогнозирование результатов проектной архитектуры и предложения путей коррекции.

**Список использованных источников:**

1. "Статический анализ кода"[Электронный ресурс] – Режим доступа: <https://www.viva64.com/ru/t/0046/>– Дата доступа: 14.03.2019.

## **ОСОБЕННОСТИ РАЗРАБОТКИ WEB-ПРИЛОЖЕНИЙ ПРИ НИЗКОМ УРОВНЕ КОМПЬЮТЕРНОЙ ГРАМОТНОСТИ ПОЛЬЗОВАТЕЛЕЙ**

*Жлобич А.В.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Куликов С.С. – к.т.н., доцент*

Современные тенденции ведут к тому, что большую часть рынка завоевывает web-сфера из-за своей простоты в использовании. Рядовому пользователю привычно понятие сайта и он с лёгкостью использует web-технологии в повседневной жизни. Тем не менее, существуют проекты (ситуации), когда web-приложение используется людьми, далёкими даже от понятия «компьютер», в связи с чем существуют особенности разработки сайтов для такой аудитории.

Чаще всего такие ситуации встречаются на стыке двух сфер: когда программные продукты используются для автоматизации процессов в сфере образования, медицины и т.п., где часто встречаются люди, непривыкшие к работе на компьютерной технике. Их знания и умения часто ограничиваются настольными программами семейства Microsoft Office и браузером.

Таким образом, в случае разработки программного обеспечения для данной аудитории существуют проблемы и особенности, которые будут рассмотрены на примере web-сферы:

- 1) Устаревшие браузеры. Несмотря на то что большинство браузеров обновляется автоматически, в данном сегменте гораздо чаще можно встретить случаи использования устаревших версий браузеров и редко встречающихся браузеров в принципе. Сложность проекта возрастает, потому что нельзя доверять усреднённой статистике по использованию браузеров, приходится отказываться от новых технологий и возможностей только для поддержания совместимости. Дополнительным минусом является тот факт, что вынужденное использование во frontend-части методов, помеченных как *deprecated*, плохо сказывается на продвижении сайтов в поисковых системах;
- 2) Вопросы безопасности. Компьютеры пользователей с низким уровнем компьютерной грамотности чаще содержат в себе вредоносные программы, что вынуждает принимать дополнительные меры по обеспечению безопасности;
- 3) Понятность интерфейса.
  - Стоит избегать большого кол-ва изображений. Сайт должен выглядеть максимально просто;
  - Меню сайта должно быть унифицированным и обеспечивать доступ к большей части сайта;
  - Некоторая часть пользователей данного сегмента не имеет представления о разделении на вкладки и такой стандартной возможности браузеров, как «вернуться назад». В связи с этим не рекомендуется открывать ссылки в новом окне;
  - Если предполагается управление некими данными или ресурсами, во-первых, для любой операции должно запрашиваться подтверждение, во-вторых, нельзя доверять решению пользователя: любые даже подтверждённые изменения, включая удаления, должны быть отменяемыми. Приветствуется сохранение резервных копий без участия пользователя.
- 4) Шрифт. Не приветствуется использование нестандартных шрифтов. Также стоит учитывать вероятность того, что сайтом будут пользоваться люди с плохим зрением. Существует два

решения этой проблемы: совместимость шрифтов сайта с параметром браузеров «режим для слабовидящих» или же разработка отдельной версии интерфейса для людей с плохим зрением, как это сделано на таких сайтах как [www.vtb.ru](http://www.vtb.ru), [www.rosbank.ru](http://www.rosbank.ru);

- 5) Навигация с помощью клавиатуры. Привычными для многих пользователей являются стрелки на клавиатуре, клавиша Enter, клавиша Tab. Сайт должен адекватно реагировать на их использование;
- 6) Обновления. Интерфейс web-приложений, предназначенных для постоянного использования, не должен претерпевать слишком сильные изменения от версии к версии, так как пользователям данного сегмента сложно перестраиваться на непривычные им способы взаимодействия с сайтом
- 7) Руководство. Для web-приложений, предназначенных для регулярного использования, стоит сделать подробное руководство. Вполне к месту наличие видеоруководства и удобной обратной связи.

Как можно заметить, сложность проекта, потенциальная аудитория которого - пользователи с низким уровнем компьютерной грамотности, в разы выше, чем у обычного проекта. Тем не менее, несмотря на общие особенности таких проектов, в каждой смежной сфере, для которой могут разрабатываться приложения, могут быть свои дополнительные требования.

**Список использованных источников:**

1. Якоб Нильсен, Хоа Лоранжер. Web-дизайн: удобство использования Web-сайтов. — М.: «Вильямс», 2007. — С. 368.
2. Фисун А. П., Гращенко Л. А. и др. Теоретические и практические основы человеко-компьютерного взаимодействия / А. П. Фисун. — Деп. в ВИНТИ 15.10. 2004 г. № 1624 – В 2004. — Орел: Орловский государственный университет, 2004. — 169 с.

## **ХРАНИЛИЩЕ ДАННЫХ КАК ЭЛЕМЕНТ ПРОЦЕССА ПРИНЯТИЯ РЕШЕНИЙ**

*Ивченко А.В.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Куликов С.С. – к.т.н., доцент*

В данной статье рассматриваются причины и преимущества использования хранилищ данных, их отличия от транзакционных систем, описываются методы поддержания историчности данных, исследуются и анализируются существующие подходы к построению хранилищ данных.

В эпоху информационных технологий одним из важнейших активов любой организации является информация. Каждый день базы данных пополняются новыми записями, которые в дальнейшем могут быть использованы для анализа. Задача Business Intelligence (BI) состоит в том, чтобы упорядочить данные организации, представить их в человекочитаемой форме и предоставить бизнес-пользователям для принятия решений [1].

Направление Business Intelligence тесно связано с понятием «хранилище данных». Хранилище данных (англ. Data Warehouse) – предметно-ориентированная, интегрированная, поддерживающая хронологию и неизменяемая база данных, используемая для процесса принятия решений [2].

В хранилище объединяются данные из различных источников. Данные преобразуются, унифицируются, обрабатываются на наличие ошибок и приводятся к определенной структуре, что облегчает дальнейшую работу с ними.

Основным отличием хранилища данных в сравнении с транзакционной системой является сохранение историчности данных, что имеет большое значение при анализе данных.

Выделяют несколько типов поддержания хронологии данных. Первый заключается в том, что значения перезаписываются без сохранения устаревшего варианта, когда история изменения сущности не важна. Во втором данные заносятся в хранилище как новая запись, без изменения имеющегося значения, в этом случае специальный атрибут указывает на актуальность данных. Кроме того, хранилище может содержать сущности такого типа, что записи в них вовсе не могут быть изменены. Таким образом обеспечивается необходимая степень историчности данных.

В отличие от транзакционных систем, в хранилище данные поступают с определенной периодичностью, так как анализ производится относительно какого-либо периода и мгновенная синхронизация с транзакционной системой не требуется.

Существует два основных подхода к построению хранилищ данных: Инмона и Кимбалла.

Подход Инмона подразумевает наличие общего Enterprise-хранилища, находящегося в третьей нормальной форме и хранящего данные в детализированном виде. Для представления предметно-ориентированных данных на основе Enterprise-хранилища создаются витрины данных, где данные

агрегируются. Такое хранилище легче расширяется и за счет атомарности данных может быть использовано при возникновении новых вопросов для анализа, так как хранит данные в неагрегированном виде.

Подход Кимбалла заключается в построении хранилища на основе схемы «звезда» или «снежинка» с денормализованными данными и слабой детализацией. Денормализация данных позволяет упростить и ускорить процесс построения отчетности, являющийся завершающим этапом построения BI-решения.

В настоящее время популярен подход объединения концепций Инмона и Кимбалла для использования преимуществ обоих.

Так, например, для построения отчетности в большинстве случаев предпочтительным является представление данных в денормализованном и агрегированном виде, так как выполнение данных операций во многих инструментах построения отчетности требует большого количества времени и вынуждает пользователя отчета ждать окончания расчета для отображения визуализаций. Кроме того, зачастую инструменты построения отчетности имеют достаточно ограниченный набор калькуляций и функционал объединения данных. Однако в условиях быстрого изменения окружающего мира у организаций постоянно возникают новые вопросы, требующие анализа, поэтому расширяемость хранилища также имеет большое значение.

Таким образом, хранилище данных позволяет сохранить историчность данных, необходимую для анализа, представить данные в наиболее удобном виде, снять нагрузку, связанную с аналитикой, с транзакционной базы данных, упростить и ускорить процесс построения отчетности.

**Список использованных источников:**

1. Kimball R., Ross M. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling, Third Edition. – Indianapolis, 2013. – P. 27.
2. "What is a Data Warehouse?" W.H. Inmon, Prism, Volume 1, Number 1.

## **ВЛИЯНИЕ ФИЗИЧЕСКОГО СОСТОЯНИЯ СПОРТСМЕНА НА РЕЗУЛЬТАТ ТЕННИСНОГО МАТЧА**

*Канунников И.С.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Парамонов А.И. – к. т. н., доцент*

В работе приводится исследование по влиянию на результат теннисного матча таких характеристик игрока как рост, вес и возраст. Выполнен анализ динамики изменения указанных характеристик для игроков топ-10 мирового рейтинга. Сформулированы выводы о влиянии рассмотренных показателей на итоговый результат матча. Введен ряд новых величин, которые характеризуют интенсивность игровой нагрузки и времени на восстановление между матчами.

В научно-методической литературе по теннису не выявлены характерные особенности телосложения современных лучших теннисистов мира, хотя антропометрические данные спортсмена существенно влияют на спортивный результат. Цель данной работы сводится к установлению тенденций изменения антропометрических показателей профессиональных теннисистов мира за прошедшие годы 21 века и их влияние на спортивный результат.

В работе ставилась задача провести сравнение средних антропометрических показателей (возраста, роста, массы, индекса массы тела и других показателей) лучших теннисистов-профессионалов среди мужчин 2018 года и найти статистическую связь антропометрических показателей с рейтингом и скоростью подачи.

Рост и вес – одни их важнейших антропометрических показателей человека. Профессиональным спортсменам, в том числе и теннисистам, критически важно следить за данными параметрами своего тела и держать их в соответствии друг другу. Практика показывает, что рассматривать данные показатели отдельно друг от друга малоэффективно. Для оценки соответствия веса и роста человека существует показатель «Индекса массы тела» – ИМТ (body mass index - BMI). Данный показатель рассчитывается по формуле (1).

$$BMI = \frac{m}{h^2}, \quad (1)$$

где  $m$  – масса тела в килограммах,  $h$  – рост в метрах. [1,2]

В 2014 году Francis KH Wong, Jackie HK Keung, Newman ML Lau и другие [3] опубликовали исследование, в котором изучалось в том числе и влияние ИМТ на скорость подачи в теннисе.

Ученые пришли к следующим выводам: возраст, продолжительность тренировок и рост игрока не имеют значительной корреляции со скоростью подачи; ИМТ значительно ( $r = 0.577$ ;  $p < 0.05$ ) коррелирует со скоростью подачи. Теннисисты с ИМТ  $20,45 \pm 2.57 \text{ кг/м}^2$  имеют скорость подачи до  $150 \text{ км/ч}$ . Теннисисты с ИМТ  $23,87 \pm 1.89 \text{ кг/м}^2$  имеют скорость подачи выше  $150 \text{ км/ч}$ .

В работе собрана и проанализирована информация по росту и весу игроков топ-10 мирового рейтинга АТР по итогам сезонов 2000-2018 гг. (на дату 13.10.2018). Информация бралась с официального сайта АТР. [4] Анализировались следующие 3 фактора: индекс массы тела, вес, рост. Исследовалось для 4 состояний: 1-я ракетка мира, Среднее среди топ-10, Максимальное среди топ-10, Минимальное среди топ-10. В результате был найден «идеальный» теннисный ИМТ  $23.89 \frac{\text{кг}}{\text{м}^2}$ . Влияние показателя ИМТ было оценено по формуле (2).

$$P = 1 - \left| \frac{BMI_i - BMI_{ideal}}{BMI_{ideal}} \right| = 1 - \left| \frac{BMI_i - 23.89}{23.89} \right|, \quad (2)$$

где  $BMI_i$  – индекс массы тела  $i$ -го игрока.

На сегодняшний день теннисные аналитики и журналисты отмечают, что современные профессиональные теннисисты в среднем старше, чем когда бы то ни было ранее. [6-10] В 2016 г средний возраст в мужском теннисном туре был 27.7 лет – наибольший за всю историю тура. [5] Средний возраст окончания карьеры в теннисе – 30-33 года. [14] В работе введено понятие «пиковый теннисный возраст» - возраст, когда теннисист показывает максимальные игровые результаты. Данное значение получилось равным 26.75 лет.

Влияние показателя возраста игрока было оценено по формуле (3).

$$P_{\text{вз}} = 1 - \left| \frac{age_i - PeakAge_{30+}}{PeakAge_{30+}} \right| = 1 - \left| \frac{age_i - 26.75}{26.75} \right|, \quad (3)$$

где  $age_i$  – возраст  $i$ -го игрока.

Введено понятие игровой нагрузки – суммарное время на корте за турнир. Теннисные матчи у мужчин на турнирах большого шлема играют до побед в трех сетах, то есть максимально возможное количество сетов в матче равно пяти. В настоящее время продолжительность теннисного матча на турнирах большого шлема у мужчин достигает шести и более часов. Учитывая, что в основной сетке необходимо *выиграть семь матчей*, чтобы победить в Турнире Большого Шлема, то суммарное время пребывания спортсменов на корте может намного превысить двузначную отметку в часах.

Введено понятие времени на восстановление между турнирами - фактическое время отдыха (в неделях) игроков разных возрастных групп. Восстановление сил между турнирами очень важно для продуктивного выступления теннисистов на последующих соревнованиях. В зависимости от возраста, человеческому организму требуется различное время для восстановления. Для того, чтобы установить, сколько времени требуется на восстановление теннисистам топ-10 различных возрастных категорий условно разделили всех игроков на три возрастные категории: до 25 лет, 25-30 лет, 30+ лет. Для каждой категории взяты наиболее успешные представители в рейтинге в данный момент и проверены их результаты за сезон 2018г (на дату 08.10.2018). Для анализа пользовались данные из источников [11-13].

Введено понятие непрерывной игровой нагрузки – средняя нагрузка между периодами отдыха. В 2013 г Tuomo Ojala and Keijo Häkkinen исследовали [15] влияние 3-дневного теннисного турнира на физическую активность, гормональные реакции и тонус мышц. Помимо прочего ученые сделали вывод, что даже через 48 часов после окончания турнира организм спортсменов не восстановился полностью. Для того, чтобы установить среднюю нагрузку между периодами отдыха у теннисистов топ-10 различных возрастов, все игроки были условно разделены на 3 возрастные категории: до 25 лет, 25-30 лет, 30+ лет. Для каждой категории был взят наиболее успешный представитель в рейтинге в данный момент и проверены его результаты за сезон 2018г (на дату 08.10.2018). Моделирование показало рост показателя непрерывной игровой нагрузки с возрастом от 524,6 минут (для возрастной группы до 25 лет) до 823.9 минут (для возрастной группы 30+ лет.)

Введено понятие предельной игровой нагрузки - такой суммарной продолжительности времени на корте без учета периодов отдыха, после которой игрок снимется с матча ввиду травмы или иного недомогания. Отсчет суммы ведется с момента последнего снятия с матча. Как правило, после снятий с игр спортсмены делают длительные перерывы для восстановления здоровья. Проанализировав, как сказывается суммарное время, проведенное на корте, на снятиях с матчей спортсменов в сезоне 2018г на дату 08.10.2018, получены цифры 2541,3 минуты для возрастной группы 25-30 лет и 3295,5 минут для возрастной группы 30+ лет.

В работе показаны закономерности изменения роста, веса и ИМТ у теннисистов топ-10 мирового рейтинга, проанализирован возраст игроков Тура и введен ряд новых величин, характеризующих интенсивность игровой нагрузки и времени на восстановление между матчами.

Исследование будет представлять интерес для игроков и тренеров для составления годового календарного плана спортсмена и планирования игровых нагрузок и восстановительных мер.

Данное исследование будет продолжено для моделирования результата матча на основе предыгровых показателей.

**Список использованных источников:**

1. Body Mass Index: Obesity, BMI, and Health: A Critical Review, Frank Q. Nuttall, Nutr Today. 2015 May; 50(3):, 117–128. Published online 2015 Apr 7. doi: 10.1097/NT.0000000000000092
2. Калькулятор подсчета индекса массы тела, 01.03.19, Режим доступа: <http://ironzen.org/calc-4/>
3. Effects of Body Mass Index and Full Body Kinematics on Tennis Serve Speed, Francis KH Wong, Jackie HK, Keung, Newnan ML Lau, Douglas KS Ng, Joanne WY Chung, Daniel HK Chow, J Hum Kinet. 2014 Mar 27; 40: 21–28., Published online 2014 Apr 9. doi: 10.2478/hukin-2014-0003
4. Официальный сайт ATP, 01.03.19, Режим доступа: <https://www.atpworldtour.com/en/rankings/singles>
5. Professional tennis is older than it's ever been, Nick Wells, Eric Chemi, 01.03.19, Режим доступа: <https://www.cnbc.com/2017/01/28/professional-tennis-is-older-than-its-ever-been.html>
6. Bjorn Borg at 27: Retiring Early and Sleeping Well, Jane Leavy, 01.03.19, Режим доступа: [https://www.washingtonpost.com/archive/sports/1983/10/29/bjorn-borg-at-27-retiring-early-and-sleeping-well/a635ec9b-1cf0-47a8-ad5c-cfa34b2f1c60/?noredirect=on&utm\\_term=.15abc0dbe10b](https://www.washingtonpost.com/archive/sports/1983/10/29/bjorn-borg-at-27-retiring-early-and-sleeping-well/a635ec9b-1cf0-47a8-ad5c-cfa34b2f1c60/?noredirect=on&utm_term=.15abc0dbe10b)
7. Edberg calls time on Sweden's golden era, John Roberts, 01.03.19, Режим доступа: <https://www.independent.co.uk/sport/edberg-calls-time-on-swedens-golden-era-1526578.html>
8. Courier ends career after 12 classy years, John Roberts, 01.03.19, Режим доступа: <https://www.independent.co.uk/sport/tennis/courier-ends-career-after-12-classy-years-277852.html>
9. Clay-court legend Bruguera retires, 01.03.19, Режим доступа: <http://news.bbc.co.uk/sport2/hi/tennis/1956741.stm>
10. Sampras to retire officially at US Open, Howard Fendrich, 01.03.19, Режим доступа: <https://www.independent.co.uk/sport/tennis/sampras-to-retire-officially-at-us-open-101371.html>
11. Электронное хранилище теннисной статистики, 01.03.19, Режим доступа: <http://www.tennisabstract.com/cgi-bin/player.cgi?p=AlexanderZverev>
12. Электронное хранилище теннисной статистики, 01.03.19, Режим доступа: <http://www.tennisabstract.com/cgi-bin/player.cgi?p=DominicThiem>
13. Электронное хранилище теннисной статистики, 01.03.19, доступа: <http://www.tennisabstract.com/cgi-bin/player.cgi?p=RafaelNadal>
14. Федерер: с пониманием отношусь к людям, которые называют меня старым, 01.03.19, Режим доступа: <https://www.championat.com/tennis/news-3570725-federer-s-ponimaniem-otnoshus-k-ljudjam-kotorye-nazyvajut-menja-starym.html>
15. Effects of the Tennis Tournament on Players' Physical Performance, Hormonal Responses, Muscle Damage and Recovery, Tuomo Ojala, Keijo Häkkinen, J Sports Sci Med. 2013 Jun; 12(2): 240–248. Published online 2013 Jun 1.

## ИСПОЛЬЗОВАНИЕ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ ДЛЯ ПРОГНОЗИРОВАНИЯ ПОГОДНЫХ УСЛОВИЙ

*Клещиков А.С., Медведев С.А.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Медведев С.А. – к.т.н., доцент*

Прогнозирование погоды является жизненно важной областью в метеорологии и была одним из самых научно и технологически сложных проблем в мире в прошлом веке. В данной статье рассматривается использование методов интеллектуального анализа данных с использованием искусственных нейронных сетей для прогнозирования погодных условий, таких как максимальная температура воздуха, скорость ветра, количества осадков и др.

Основная цель данной статьи – рассмотреть использование искусственных нейронных сетей для задач прогнозирования численных характеристик, таких как температура воздуха, скорость ветра, количество осадков, испарение и другие.

Интеллектуальный анализ данных – это методика поиска новой и потенциально полезной информации из большого количества исходных данных. В отличие от стандартных статистических методов, методы искусственного анализа данных позволяют найти интересную информацию, не требуя априорных гипотез, а вид шаблонов, обнаруженных во время анализа данных, зависит от задач, поставленных в ходе их анализа.

Искусственной нейронной сетью называется динамическая система, состоящая из совокупности связанных между собой по типу узлов направленного графа элементарных процессоров, называемых искусственными или формальными нейронами, и способная генерировать выходную информацию в ответ на входное воздействие. Модель искусственного нейрона состоит из трех элементов: синапсов (входов), каждый из которых характеризуется величиной синаптической связи  $w_i$ , телом нейрона (сумматор) и аксоном (выходом). Данная модель представлена на рисунке 1.

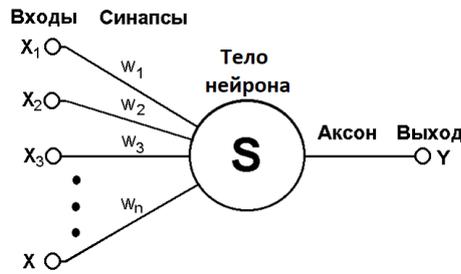


Рисунок 1 – Модель искусственного нейрона.

Математически выход  $Y$  нейрона можно записать в виде следующей формулы:

$$Y = f(S) = f\left(\sum_{i=1}^n w_i x_i + w_0\right) \quad (1)$$

Для целей прогнозирования погодных условий целесообразно применить схему многослойного персептрона с обратным распространением ошибки. Многослойный персептрон позволяет аппроксимировать любую непрерывную функцию или границу между классами со сколь угодно высокой точностью. В качестве функции активации  $f$  используется сигмоидная функция (2).

$$f(x) = \frac{1}{1+e^{-x}} \quad (2)$$

Для обучения сети используется алгоритм обратного распространения ошибки:

- 1) инициализировать синаптические веса маленькими случайными значениями;
- 2) выбрать очередную пару из обучающего множества; подать входной вектор на вход сети;
- 3) вычислить выход сети;
- 4) вычислить разность между выходом сети и требуемым выходом;
- 5) подкорректировать веса сети для минимизации ошибки;
- 6) повторять шаги с 2 по 5 для каждого вектора обучающего множества до тех пор, пока ошибка на всем множестве не достигнет приемлемого уровня.

Данная модель позволяет достаточно точно предсказывать численные характеристики погоды с высокой точностью. Однако, такой метод не подходит, если входные данные характеризуются высоким шумом или большим пропуском значений во временном ряде.

**Список использованных источников:**

1. Bregman, J.I. Environmental Impact Statements / Bregman, J.I. – Boca Raton : MI Lewis Publication, 2006. – 264 p.
2. Folorunsho O. Application of Data Mining Techniques in Weather Prediction and Climate Change Studies / Folorunsho O. // I.J. Information Engineering and Electronic Business. – 2012. – p. 50-51.

## РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА ГЕНЕРАЦИИ UML- ДИАГРАММ НА ОСНОВЕ ОПИСАНИЯ НА МЕТАЯЗЫКЕ

*Кобец И. А.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Смолякова О.Г. – к.т.н., доцент*

В статье рассматриваются вопросы разработки программного средства для визуализации UML-диаграмм на основе описания на проблемно-ориентированном метаязыке. Приводится общее описание языка и составляющих его компонентов: отвечающих за описание генерируемых диаграмм и их характеристик, а также используемых для расширения языка

Разработанное программное средство представляет собой транслятор кода на метаязыке в векторное либо растровое изображение UML-диаграммы. Необходимость описания диаграмм на метаязыке может возникать при использовании систем компьютерной вёрстки наподобие Latex, либо разнообразных облегчённых языков разметки, таких как Markdown. Одним из преимуществ автоматической генерации диаграмм на основе описания на метаязыке, является то, что увеличивается скорость их получения, так как нет необходимости в ручном размещении элементов диаграммы. Такая необходимость может появляться только в редких случаях, когда необходимо определённое расположение элементов, для большей выразительности получаемого изображения. Как показала практика проекта PlantUML, текстовое описание UML-диаграмм позволяет слабовидящим студентам и инженерам как разрабатывать, так и читать UML-диаграммы [1,2].

Существует несколько подобных программных средств, самые популярные среди которых PlantUML и Graphviz. Среди основных недостатков у всех аналогов можно выделить слишком общий синтаксис для любых видов диаграмм, описываемых при помощи предметно-ориентированных языков данных программ. Из-за того, что текст на общем для всех видов диаграмм языке сложен для синтаксического анализа, то это не позволяет создать удобные средства для произведения статического анализа кода, автодополнения, навигации, поиска, подсветки синтаксиса, форматирования, а также генерации кода.

При работе над языком разрабатываемого программного средства были учтены недостатки аналогов. И было принято решение, что в метаязыке необходимо обеспечить строгую типизацию, которая бы разграничивала как типы диаграмм и их элементы между собой, так и разницу между используемыми графическими нотациями: UML, IDEF0, IDEF1X и так далее. Также, для упрощения разработки необходимых инструментов для написания кода на данном метаязыке, было решено описать язык с использованием грамматики разбирающей выражение, т.к. они являются однозначными, что упрощает написание синтаксического анализатора [3].

В результате разработки метаязыка было решено выделить два его подмножества: язык для описания диаграмм и элементов их составляющих, а также язык для описания и расширения типов диаграмм и графических нотаций. Язык для описания диаграмм представляет собой предметно-ориентированный язык, при помощи которого описываются конкретные диаграммы, изображения которых необходимо получить на выходе работы программы. Основные элементы этого подмножества языка – диаграммы. Описание каждой диаграммы представляет собой блок, содержащий в себе перечисление вершин, составляющих диаграмму, а также связей между вершинами. Каждый блок диаграммы содержит информацию о том, к какому типу диаграмм он относится, что позволяет использовать внутри блока элементы необходимого типа. Например, Actor и Use Case для диаграммы вариантов использования.

Расположение диаграммы на плоскости происходит в соответствии с выбором способа размещения, каждый тип диаграмм имеет список определённых вариантов визуализации диаграммы, один из которых является способом по умолчанию. Типы размещений для большинства диаграмм – это иерархическое, радиальное, прямолинейное. В архитектуре приложения предусмотрено дальнейшее расширение вариантов расположения. Для расширения существующих нотаций и относящихся к ним типов диаграмм используется второе подмножество языка. В данном подмножестве описывается, как будут выглядеть определенные элементы диаграммы и из каких элементов может состоять диаграмма определённого типа.

В результате разработано расширяемое программное средство, автоматизирующее размещение UML-диаграмм на плоскости и генерацию изображений.

**Список использованных источников:**

1. Langdon, Patrick, et al. Inclusive designing: joining usability, accessibility, and inclusion. Cham New York: Springer, 2014.
2. Miesenberger, Klaus. Computers helping people with special needs: 13th International Conference, ICCHP 2012, Linz, Austria, July 11-13, 2012, Proceedings. Berlin New York: Springer, 2012.
3. Proceedings of the 31st ACM SIGPLAN-SIGACT symposium on Principles of programming languages. New York, NY: ACM, 2004.

## **ВЕБ-СЕРВИСЫ ВОПРОСОВ И ОТВЕТОВ**

*Коваленко И.А.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Парамонов А.И. – к.т.н., доцент*

Целью проекта является разработка веб-сервиса вопросов и ответов. Приложение позволит пользователю получить ответ на свой вопрос в автоматическом режиме, используя базу уже имеющихся вопросов и ответов.

Все чаще и чаще люди обращаются за ответами к ресурсам сети Интернет. И если одни пытаются счастья в поисковых системах, то другая половина уже является постоянным пользователем одной из информационных систем вопросов и ответов. Эти системы предоставляют возможность сообществу обмениваться знаниями, зачастую в определенной предметной области.

Ежедневно на подобных сервисах пользователи задают десятки тысяч повторяющихся вопросов. Существующие аналогичные сервисы не анализируют вопрос пользователя перед его публикацией. Соответственно, в таких сервисах, как toster.ru [1], Ответы @Mail.ru [2] довольно много повторяющихся пар ответов/вопросов.

Существующий сервис Ответы@Mail.ru используют сложную систему ранжирования пользователей, учитывающую размер и качество их вклада. Ключевая метрика: КПД — это отношение лучших ответов к общему числу ответов (лучший ответ выбирается автором вопроса или по результатам голосования). От него зависит число полученных за ответ баллов: чем выше КПД участника, тем больше баллов начисляется. За вопросы и нарушение правил система вычитает баллы. С достижением определённого числа баллов пользователю присваивается новый статус (от «новичка» до «высшего разума»), расширяющий лимит на число вопросов, ответов и голосований за один день.

Сервис Quora [3] использует алгоритм ранжирования ценности ответа каждого пользователя основываясь на оценках его предыдущих ответов. Также сервис использует алгоритм машинного обучения для классификации тематики вопроса на основе истории вопросов и ответов пользователя.

Похожий сервис ASKfm [4] предлагает пользователю после регистрации заполнить свою анкету и начать, как от своего имени, так и анонимно, задавать и отвечать на вопросы других пользователей. ASKfm интегрирован с другими социальными сетями, такими как Facebook, Twitter, ВКонтакте и Tumblr, которые повлияли на широкое распространение этой сети. В основу сервиса с самого начала была заложена соревновательная модель, когда наиболее активные и «полезные» пользователи оказывались выше в рейтинге участников.

Существует также популярный сервис StackOverflow для программистов [5]. На нем, как и в других системах подобного рода, предоставляется возможность оценивать вопросы и ответы, что поднимает или понижает репутацию зарегистрированных пользователей.

Сервис Toster [1] позволяет искать по ответам, чтобы пользователь мог найти интересующую его информацию не только в вопросах. К поиску по ответам тоже можно применить фильтр по времени, по наличию ответов, решений и по времени создания контента. Полезной функцией является возможность вставлять блоки с результатами работ в тела вопросов, ответов и комментариев. Причем, демонстрировать не только код, но и результат его воспроизведения.

На основе обзора предлагается альтернативный проект. В сервисе предлагается использовать систему голосования за лучший ответ, что позволит системе выбирать наиболее релевантный ответ. Автор вопроса имеет возможность выбрать ответ-решение своей проблемы, что облегчит поиск ответа для другого человека, задающего аналогичный вопрос. Если пользователи сервиса смогут получить ответ еще до создания вопроса, количество однотипных вопросов будет практически нулевым. Так как авторы ответов будут тратить меньше времени на поиск уникальных ответов, то они смогут отвечать на большее количество вопросов. Таким образом система экономит время на поиск готового ответа. В основе сервиса лежит распределенное хранилище ключ-значение, которое позволяет обрабатывать большое количество одновременных запросов от пользователей. Для реализации веб-части используется платформа Ruby on Rails, в качестве хранилища данных — база данных MySQL.

**Список использованных источников:**

1. Сервис вопросов и ответов для IT специалистов Обновление на Тостере [Электронный ресурс] / TechMedia: «Хабр». – Режим доступа: <https://habr.com/ru/company/toster/blog/419535/>. – Дата доступа: 22.03.2019
2. Ответы@Mail.Ru [Электронный ресурс] / Wikipedia® – Режим доступа: <https://ru.wikipedia.org/wiki/Ответы@Mail.Ru> – Дата доступа: 22.03.2019
3. Quora — социальный сервис обмена знаниями [Электронный ресурс] / Режим доступа: <https://www.quora.com/> – Дата доступа: 22.03.2019

## **АВТОМАТИЗАЦИЯ ДЕЯТЕЛЬНОСТИ МЕДИЦИНСКИХ УЧРЕЖДЕНИЙ**

*Корховая А.Б.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Колотыгин К.Е. – ассистент каф. ПОИТ*

В докладе рассмотрены возникающие трудности в работе медицинских учреждений и предполагаемые пути решения путем использования приложения по автоматизации деятельности медицинских учреждений.

Одна из самых главных составляющей жизни — здоровье человека. Из года в год появляется все больше специалистов, которые работают в сфере медицинского оздоровления. И тем самым увеличивается и количество самих поликлиник, больниц, кабинетов и других учреждений.

Издавна вся документация велась при помощи бумаг, что в наше время утратило актуальность. Из-за огромного количества информации, хранящейся не в электронном виде, в медицинских

учреждениях возникают задержки в очереди, утери документов и необходимость их восстановления, что очень трудоёмкий процесс, если он вообще возможен.

Сейчас активно внедряются программные средства по автоматизации работы всех видов учреждений [2]. В сфере медицины уже известны ПО для автоматизации очереди при сдаче анализов и ПО для ведения приемов врачей. Но проблема с огромным количеством медицинских карт и прочих документов до сих пор не решена.

Данные проблемы могут быть решены путем использования унифицированной системы для автоматизации деятельности медицинских учреждений. Предполагается, что пользоваться системой смогут люди, занимающие руководящие должности, врачи, лаборанты, медсестры, регистраторы и сами пациенты. Данная система может распространяться на несколько стран и в любых регионах. Ответственному лицу необходимо будет внести медицинское учреждение в систему и добавить работающих в нем специалистов (после их регистрации).

Предлагаемая система будет иметь довольно широкий функционал. Для каждой роли будут предусмотрены свои функции [1].

Возможности руководящих должностей:

- 1) просмотр данных о медучреждении, работниках мед. учреждения, пациентах и больничных листах;
- 2) изменение данных о медучреждении;
- 3) подтверждение изменений должностных данных;
- 4) запрос на изменение должностных данных;
- 5) подтверждение запроса на отпуск;
- 6) подтверждение добавления нового работника или пациента;
- 7) запрос на добавление нового работника.

Возможности заведующих врачей:

- 1) просмотр данных о медучреждении, работниках мед. учреждения, пациентах и больничных листах;
- 2) запрос на изменение должностных данных;
- 3) подтверждение открытия, закрытия или продления больничного листа;
- 4) запрос на подтверждение отпуска главным врачом;
- 5) подтверждение отпуска подчиненного персонала;
- 6) запрос на добавление нового работника или пациента.

Возможности врачей-специалистов:

- 1) просмотр данных о медучреждении, работниках мед. учреждения, пациентах карт и больничных листах;
- 2) запрос на подтверждение отпуска заведующим врачом;
- 3) изменение данных пациента;
- 4) запрос на открытие, закрытие или продление больничного листа;
- 5) добавление поставленного диагноза к медицинской карте пациента;
- 6) просмотр очереди пациентов по выданным талонам;
- 7) запрос на добавление нового пациента в систему.

Возможности лаборанта:

- 1) просмотр данных о медучреждении, работниках мед. учреждения, пациентах и больничных листах;
- 2) запрос на подтверждение отпуска заведующим врачом;
- 3) добавление данных о проведенных анализах пациента.

Возможности медсестры:

- 1) просмотр данных о медучреждении, работниках мед. учреждения, пациентах и больничных листах;
- 2) запрос на подтверждение отпуска заведующим врачом;
- 3) изменение данных пациента;
- 4) просмотр очереди пациентов по выданным талонам.

Возможности регистратора:

- 1) просмотр данных о медучреждении, работниках мед. учреждения, пациентах и больничных листах;
- 2) запрос на подтверждение отпуска заведующим врачом;
- 3) просмотр и изменение очереди пациентов к врачу;
- 4) добавление выданного талона к врачу-специалисту в систему.

Возможности специалиста по работе с кадрами:

- 1) просмотр данных о медучреждении, работниках мед. учреждения, пациентах и больничных листах;
- 2) изменение должностных данных;
- 3) добавление новых сотрудников медицинского учреждения.

Возможности пациента:

- 1) просмотр данных о медучреждении и работниках мед. учреждения;
- 2) просмотр данных по своей медицинской карте и анализах;
- 3) изменение данных о себе;
- 4) онлайн-заказ талона к врачу-специалисту;
- 5) открытие диалогов с конкретным врачом мед. учреждения.

Тем самым можно отметить, что предлагаемая система предназначена для учета рабочих кадров, пациентов, медицинских карт, интеграции больничных листов. Разные пользователи будут иметь различные возможности.

Пользователи с доступом к редактированию рабочих кадров смогут изменять должностные данные (должность, оклад, рабочий кабинет и др.). Пользователи врачи будут иметь доступ к изменению данных пациентов.

Данная система должна будет предусматривать работу с больничными листами для каждого из пациентов. Пациенты смогут просмотреть свои медицинские карты и результаты анализов, так же они смогут заказать талон через приложение к выбранному врачу-специалисту в любом медицинском учреждении.

Данная система должна использовать развернутую базу данных, агрегируя следующими сущностями:

- медицинское учреждение;
- пользователи;
- документы пациентов;
- документы работников медицинских учреждений;
- запрос пациентов;
- запрос работников медицинских учреждений.

База данных данного приложения должна быть оптимизирована для работы с большим объемом данных [3], т.к. система рассчитана на использования не только в разных городах, но и в разных странах.

Предлагаемое приложение избавит пациентов от необходимости звонить в медицинское учреждение, чтобы узнать результаты проведенных анализов (при появлении результата оповещение придет на почту). Так же уйдет необходимость ведения нескольких медицинских карт на одного пациента в разных учреждениях. Вся история будет храниться в карте пациента, и любой врач будет иметь к ней доступ.

Со временем ПО наберет большое количество информации, что даст возможность внедрить в систему анализ данных, возможность прогнозирования вспышек заболеваний и подбора оптимального лекарства по диагнозу, полу и возрасту пациента. Таким образом предложенное программное средство будет востребовано во многих учреждениях, так как поможет избавиться от бумажного хранения информации и позволит оптимизировать работу медицинских учреждений.

Можно отметить ещё одну из особенностей системы: одинаковое удобство использования как работниками, так и пациентами. В следствии чего и врач может являться пациентом, не переходя в другое приложение, а только лишь сменив свою роль.

**Список использованных источников:**

1. Приказ министерства здравоохранения республики Беларусь 1 октября 2007 г. N 792 [Электронный ресурс] – Электронные данные – Режим доступа: <http://nerbol.bsmu.by/wp-content/uploads/2015/10/Приказ-МЗ-РБ-Об-утверждении-форм-первичной-медицинской-документации.pdf>.
2. Информационные технологии и управление: материалы 54 науч. конф. аспирантов, магистрантов и студентов, Минск, 23-27 апреля 2018 г. / Белорус. гос. ун-т информатики и радиоэлектроники. – Минск : БГУИР, 2018. – 191 с.
3. Работа с MySQL, MS SQL Server и Oracle в примерах: практ. пособие. / С. С. Куликов. — Минск: БОФФ, 2016. — 556 с.

## **ПРОГРАММНОЕ СРЕДСТВО СБОРА И АНАЛИЗА ТЕХНОЛОГИЧЕСКИХ ПАРАМЕТРОВ СТРЕЛЫ ФРОНТАЛЬНОГО ПОГРУЗЧИКА**

*Кошелёв Д.М.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Деменковец Д.В. – ассистент*

Эксплуатация спецтехники связана с постоянными высокими нагрузками. Подобные условия приводят к значительному увеличению расхода горючего. Чрезмерная эксплуатация транспортного средства и его нецелевое использование приводит к

быстрому износу дорогостоящей техники. Программное средство сбора и анализа параметров стрелы фронтального погрузчика помогает решать подобного рода проблемы.

Программное средство сбора и анализа технологических параметров стрелы фронтального погрузчика реализует основные функции, необходимые для сбора и анализа данных о работе техники, к примеру это расчет наработки техники и количества перемещенного груза за смену, неделю, месяц. Это значение очень важно при расчете себестоимости одного часа работы машины [1].

Данное программное обеспечение разработано под платформу Android. Это позволяет его использовать на мобильных устройствах, телефонах и планшетах без необходимости использовать персональный компьютер.

Совместно с программным средством используется специальное аппаратное устройство с датчиками угла наклона и давления в гидравлических цилиндрах стрелы фронтального погрузчика. Один датчик наклона крепится на стрелу, а второй на корпус фронтального погрузчика. Два датчика давлений крепятся в гидравлическую систему подъема. Именно с помощью этих датчиков будет рассчитываться вес перевозимого груза [2].

Разработанное программное средство сбора и анализа параметров работы фронтального погрузчика выполняет следующие функции:

- получение значений с датчиков угла наклона и давлений в гидроцилиндрах;
- вывод значений датчиков в режиме реального времени;
- вывод значений датчиков наклона в виде графиков в режиме реального времени;
- архивирование значений с датчиков в базу данных;
- отображение почасовой суммарной нагрузки на технику за текущие сутки.

Графический интерфейс приложения представлен на рисунке 1:

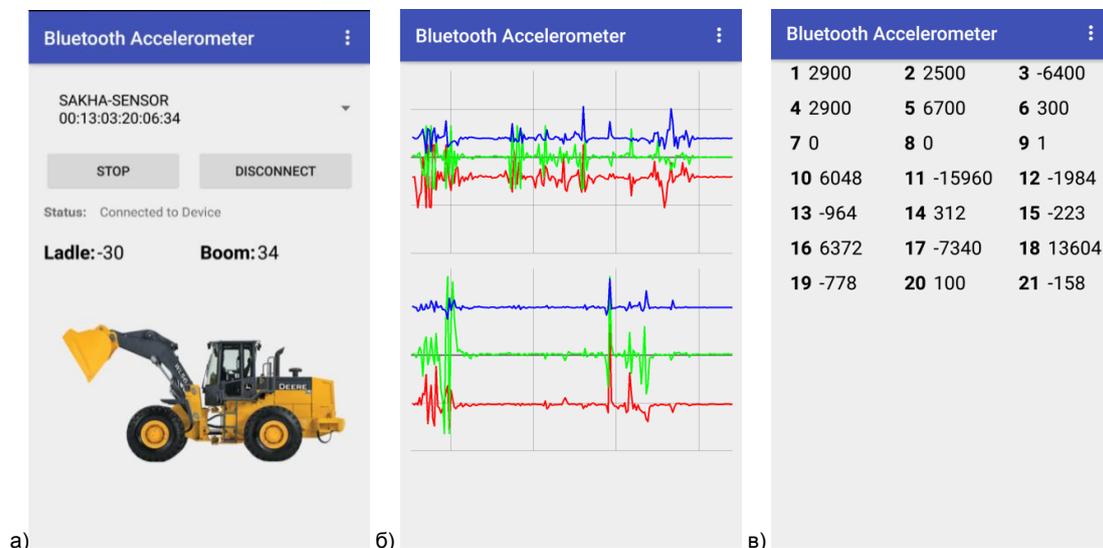


Рисунок 1 – Графический интерфейс ПС: а) главный экран ПС; б) экран графиков реального времени; в) экран текущих значений с датчиков

В дальнейшем планируется дополнительно добавить функции сохранения статистики с помощью облачной технологии (напр. Google Drive). Проблему вибрации планируется решить использованием более совершенных фильтров и выполнением предобработки данных с датчиков.

**Список использованных источников:**

1. Максименко А.Н. ТЕХНИЧЕСКАЯ ЭКСПЛУАТАЦИЯ СТРОИТЕЛЬНЫХ И ДОРОЖНЫХ МАШИН / А. Н. Максименко, В. В. Кутузов; под ред. И.В. Тургель. – Минск : Победа, 2015. – 255 с.
2. АНАЛИЗ НАГРУЖЕННОСТИ ГИДРОЦИЛИНДРА МЕХАНИЗМА ПОДЪЕМА ТЕЛЕСКОПИЧЕСКОЙ СТРЕЛЫ ПОГРУЗЧИКА / М.И. Жилевич, П.Н. Кишкевич, А.А. Зубрицкий // Вестник Костромского государственного университета. Серия тех. науки – 2017. – №12 – С. 31-34.

# ПРОГРАММНОЕ СРЕДСТВО УПРАВЛЕНИЯ ПРОЦЕССОМ ПОДАЧИ ДОКУМЕНТОВ В ВЫСШЕЕ УЧЕБНОЕ ЗАВЕДЕНИЕ

Кудрявцева О.В.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Данилова Г.В. – м. т. н., ассистент каф.: ПОИТ

Цель разработки – создание программного средства, автоматизирующего процессы подготовки и планирования проведения приёмной комиссии, подачи документов, зачисления абитуриентов, сбора статистики и формирования отчётов. Простой и удобный интерфейс должен обеспечить удобство и комфорт при работе с ПС.

Программное средство представлено в виде веб-приложения. Веб-приложение реализовано в виде трехуровневой системы, включает серверную и клиентскую части, базу данных. Данные от сервера клиенту передаются с использованием REST подхода в виде JSON формата. Для клиентской части выбран язык программирования JavaScript и фреймворк Vue.js, для серверной части язык программирования Java и фреймворк Spring. В качестве системы управления базой данных используется MySQL [1-3].

ПС включает две роли пользователей: администратор и пользователь. Администратор имеет доступ ко всем данным (управление документами пользователей, создание и редактирование: факультетов, специальностей, предметов, преподавателей и т.д). Для абитуриента предусмотрено: просмотр информации о факультетах и специальностях; регистрация и авторизация в системе; заполнение и редактирование информации о себе, своих родителях, набранных баллах на централизованном тестировании, информации о среднем балле аттестата; указание льгот, влияющих при поступлении; просмотр информации о поданных заявлениях в режиме реального времени и о текущем рейтинге. Абитуриент имеет возможность подать документы сразу на несколько специальностей разных форм обучения с указанием приоритета. Предусмотрено оповещение абитуриентов о результатах зачисления и проведения приёмной комиссии. Пример страницы ПС показан на рисунке 1.

ID	School subject name	Group number	Active	Created at	Updated at	Operations
16	Math	1	✓	11:54:21 06.03.2019	20:30:16 18.03.2019	[edit] [delete]
17	English	2	✓	11:54:39 06.03.2019	16:38:26 24.03.2019	[edit] [delete]
18	Belarusian	3	✓	19:53:19 18.03.2019	16:38:34 24.03.2019	[edit] [delete]
19	Physics	4	✓	19:53:39 18.03.2019	16:38:45 24.03.2019	[edit] [delete]
20	Biology	5	✓	15:34:47 23.03.2019	16:38:52 24.03.2019	[edit] [delete]
21	Russian	3	✓	16:38:05 24.03.2019	16:39:07 24.03.2019	[edit] [delete]
22	History	6	✓	16:38:13 24.03.2019	16:39:14 24.03.2019	[edit] [delete]

Рисунок 1 – Страница просмотра и редактирования предметов

ПС по сравнению с аналогами обладает следующими достоинствами:

- онлайн подача документов, с возможностью в дальнейшем привезти оригиналы документов;
- возможность в режиме реального времени видеть информацию о поданных заявлениях, текущем проходном балле, рейтинге абитуриентов;
- подача документов сразу на разные специальности и разные формы обучения с указанием приоритета;
- приятный интерфейс, веб-приложение работает без перезагрузки страницы.

#### Список использованных источников:

- Куликов С. С. Работа с MySQL, MS SQL Server и Oracle в примерах / Куликов С. С. – Минск: БОФФ, 2016. – 556 с.
- Дэвид Макфарланд "Большая книга CSS" 2014, Питер. – 608 с.
- Обучение JavaScript. [Электронный ресурс]. – Режим доступа: <http://learn.javascript.ru/>

# СРАВНИТЕЛЬНЫЙ АНАЛИЗ МОДЕЛЕЙ ПРОГНОЗИРОВАНИЯ ПОГОДЫ

Лабоха А.К.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Шамына А.Ю. – ассистент

Численное прогнозирование погоды использует модель атмосферы, построенную на основе системы уравнений физических параметров, описывающих аэро- и термодинамические процессы в атмосфере. В работе рассматриваются ключевые характеристики атмосферных моделей, их влияние на точность прогноза, а также некоторые компьютерные модели предсказания погоды.

Модели прогнозирования погоды можно систематизировать на основании различных критериев. На основании охватываемой прогнозом территории выделяют глобальные и региональные модели. Существует разделение моделей на термотропные, баротропные, гидростатические и негидростатические в зависимости от концептуального взгляда на атмосферу и выделения атмосферных процессов, оказывающих преобладающее влияние на формирование прогнозной модели [1]. Модели подразделяются и по временному диапазону, на которой выполняется прогноз; распространено деление на краткосрочные и долгосрочные прогнозы, однако иногда также выделяются прогнозы средней и повышенной дальности. Таким образом, каждая модель имеет свою специфику, которую необходимо учитывать для выбора наиболее точного прогноза с учетом различных параметров и предъявляемых требований.

В гидростатических моделях гидростатическое приближение позволяет исключить уравнение вертикальной составляющей, что предполагает малость вертикальных ускорений по сравнению с ускорением свободного падения [2]. Уровни давления или давления сигмы в моделях данного типа используются в качестве вертикальных координат. В то же время, с повышением точности модели вероятность нарушения данного допущения возрастает, что может привести к значительным искажениям прогноза, поскольку особенности рельефа конкретной местности исключены из рассмотрения. Негидростатические модели отказываются от идеи гидростатического приближения и ищут другие способы решения системы уравнений. Атмосферные модели данного типа большее внимание уделяют рельефу местности и вместо уровней давления используют уровни высоты или высоты сигмы.

Глобальные прогнозные модели охватывают прогнозом всю площадь Земли, наиболее распространены модели GFS, ECMWF, UKMET, NOGAPS, GEM. Существуют специальные модели для предсказания ураганов, такие как HWRF, GFDL.

Модель **GFS** (Global Forecast System) разработана американским центром NCEP (National Centers for Environmental Prediction) и предоставляет прогнозы на срок до 16 дней [3]. Модель GFS относится к гидростатическим моделям и использует 64 уровня давления сигмы в качестве вертикальных координат. В качестве модели прогнозирования атмосферы используется модель GSM (Global Spectral Model), для представления результатов используется Гауссовская сетка. Базовые условия для составления прогноза предоставляются системой GDAS (Global Data Assimilation System), осуществляющей сбор всех доступных спутниковых, традиционных (радиозондирование, авиационные, поверхностные) и радиолокационных наблюдений. Обновление прогноза GFS происходит 4 раза в день с интервалом 6 часов. Данные могут быть представлены с точностью до 1, 0.5 и 0.25 градуса ( $0.25^\circ \approx 25$  км). Прогноз доступен с шагом 1 час на ближайшие 5 суток, с прогнозируемым интервалом 3 часа на 6-10 сутки и с шагом 12 часов на 11-16 сутки. Немаловажным аспектом является общедоступность результатов работы модели GFS: файлы прогнозов доступны бесплатно, для доступа могут быть использованы протоколы ftp или https.

Европейский центр **ECMWF** (European Centre for Medium-Range Weather Forecasts) предоставляет модель прогнозирования погоды IFS (Integrated Forecasting System). Текущее ядро модели IFS гидростатично (IFS-ST), однако разработано новое ядро IFS-FVM, которое является негидростатическим и использует конечно-объемную дискретизацию [4]. Ядро IFS-FVM позволяет учитывать негидростатические динамические эффекты, влияние которых значительно при горизонтальном масштабе, меньшем 10 км (рисунок 1). В рамках модели IFS доступно несколько моделей, вычисляющих прогноз для указанного временного диапазона с указанным пространственным разрешением [5]. Модель HRES вычисляет прогноз с шагом 9 км ( $0.1^\circ$ ) на ближайшие 10 суток с дискретностью 1 час. Результаты прогнозов представлены с помощью октаэдрических уменьшенных Гауссовских сеток. Значимой особенностью моделей ECMWF является платный доступ к прогнозным данным, что существенно ограничивает возможность их использования в некоммерческих инициативах и в небольших проектах без существенных денежных вложений.

Модель **UKMET** (Unified Model of the Met Office) доступна благодаря деятельности британского центра Met Office. Данное учреждение придерживается концепции комплексного моделирования (seamless modelling), при котором одно и то же динамическое ядро системы вместе со схемами параметризации многократно используются. Таким образом, разработанная центром модель может использоваться одновременно для численного прогнозирования погоды, сезонного прогноза и климатического моделирования, а срок прогноза может варьироваться от нескольких дней до десятилетий [6]. Модель UKMET негидростатична и широко использует параметризацию таких значений, как конвекция, турбулентность пограничного слоя и др. Разработанная модель MOGREPS рассчитывает данные для всего мира с пространственной точностью 20 км (на 7 суток), а для Великобритании – с точностью 2.2 км (на 5 суток); обновление прогнозных данных выполняется 4 раза в сутки с интервалом 6 часов. Для предоставления данных прогноза используются разностная сетка «С» Аракавы и вертикально-разностная сетка Чарни-Филлипса. Решение о предоставлении данных прогноза в каждом случае принимается индивидуально, реализуются программы по поддержке исследовательских проектов.

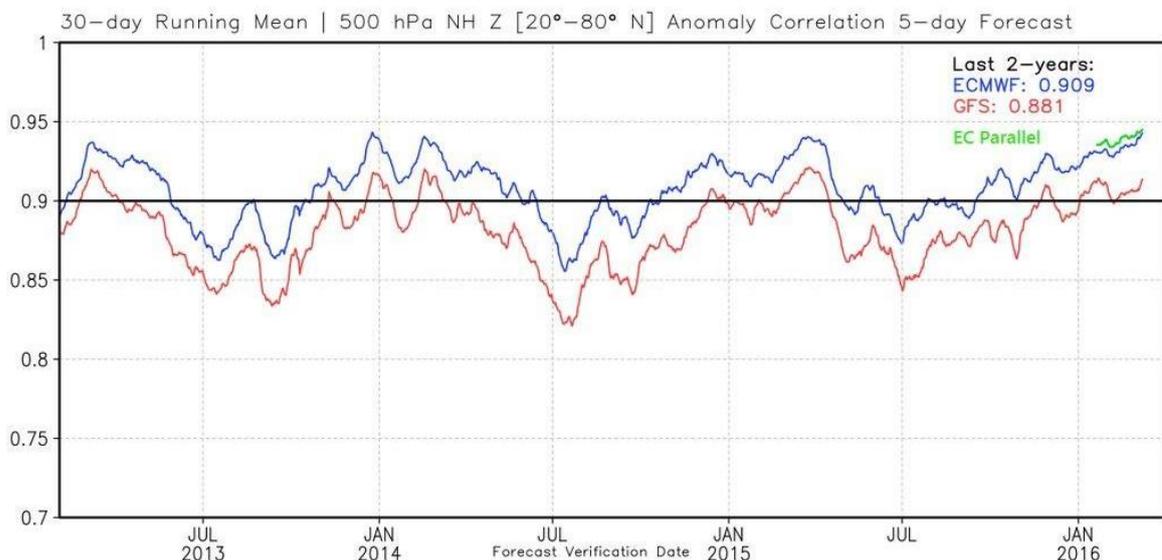


Рисунок 1 – Сравнение точности прогноза европейской модели ECMWF (синяя линия) и американской модели GFS (красная линия) в 2013-2015 гг.

Локальные модели прогноза погоды охватывают только определённую часть территории; ярким примером является модель **AROME** (Applications de la Recherche à l'Opérationnel à Méso-Echelle) французского центра Météo-France, которая предоставляет прогнозные данные только для территории Франции. Однако такое сужение рассматриваемого пространства позволило выполнять прогнозы с пространственной точностью 1.3 км [7], а также учитывать такие параметры, как рельеф, тип поверхности, экосистема и др. Ещё одним значимым аспектом является наличие интеграции с базами данных, предоставляющими фактические значения регистрирующих приборов.

Таким образом, локальные модели прогноза погоды могут предоставлять более полную и детальную информацию по сравнению с глобальными моделями благодаря концентрации внимания на характеристиках конкретной пространственной области. Однако на территориях, не охватываемых локальными моделями, частным случаем которых является Республика Беларусь, необходимо использовать глобальные модели как единственный источник прогнозных данных. Как показывает вышеприведенный анализ, при выборе конкретной модели необходимо учитывать заданные пространственную и временную точности, конкретный перечень искомых параметров погоды, характеристики прогнозной модели, а также соотносить стоимость данных прогнозных модели с финансовыми возможностями конкретного реализуемого проекта.

**Список использованных источников:**

1. Jacobson, Mark Zachary. Fundamentals of atmospheric modeling. - Cambridge University Press. - 2005. - P. 138–143.
2. Гутман Л. Н. Применение численного метода длинных волн в задаче обтекания гор // ДАН. СССР, 1957, N. 3 – С. 115-124.
3. National Centers for Environmental Prediction [Электронный ресурс]. – Режим доступа: <https://www.emc.ncep.noaa.gov/GFS/doc.php>. Дата доступа: 15.03.2019.
4. Atmospheric dynamics | ECMWF [Электронный ресурс]. – Режим доступа: <https://www.ecmwf.int/en/research/modelling-and-prediction/atmospheric-dynamics>. Дата доступа: 19.03.2019.
5. Documentation and support | ECMWF [Электронный ресурс]. – Режим доступа: <https://www.ecmwf.int/en/forecasts/documentation-and-support>. Дата доступа: 19.03.2019.
6. Unified Model - Met Office [Электронный ресурс]. – Режим доступа: <https://www.metoffice.gov.uk/research/modelling-systems/unified-model>. Дата доступа: 20.03.2019.

## ИСПОЛЬЗОВАНИЕ МНОГОКРАТНЫХ ТЕСТОВ С ИЗМЕНЯЕМЫМ НАЧАЛЬНЫМ СОСТОЯНИЕМ ДЛЯ ПСЕВДОИЩЕРПЫВАЮЩЕГО ТЕСТИРОВАНИЯ ОЗУ

*Леванцевич В.А.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Ярмолик В.Н. – д.т.н., профессор*

Исследуется возможность использования многократных вероятностных тестов с изменяемым начальным состоянием ячеек памяти для псевдо-исчерпывающего тестирования ОЗУ. Определено среднее число итераций многократного теста для исчерпывающего тестирования заданного количества ячеек памяти

Одним из эффективных способов определения неисправных ячеек ОЗУ является исчерпывающее тестирование [1]. Однако его применение ограничено сложностью и как следствие значительными временными затратами на реализацию подобного тестирования. Поэтому используют одну из аппроксимаций исчерпывающего тестирования в виде псевдо-исчерпывающих тестов [2].

Псевдо-исчерпывающим тестом является тест  $Ts(M, k)$ , который для любого заданного количества ячеек  $k$  из общего количества ячеек памяти  $M$ , при  $k < M$ , обеспечивает генерирование всех  $2^k$  двоичных разрядов. То есть таким тестом мы можем организовать исчерпывающее тестирование в любых  $k$  ячейках памяти.

В качестве примера псевдо-исчерпывающего теста можно привести тест для шести ячеек памяти  $Ts(6, 2) = \{000000, 000011, 011100, 101101, 110110, 111011\}$ . Можно заметить, что в данном тесте для любых двух ячеек памяти присутствуют возможные  $2^2=4$  двоичные комбинации. При этом некоторые комбинации присутствуют более, чем по одному разу. На рисунке 1 приведены примеры псевдо-исчерпывающих тестов для различного количества ячеек памяти.

$T(3, 2)$	$T(4, 2)$	$T(4, 3)$	$T(5, 2)$	$T(5, 3)$
000	0000	0000	11111	10000
011	0111	0011	10000	01000
101	1011	0110	01000	00100
110	1101	0101	00100	00010
	1110	1100	00010	00001
		1111	00001	01111
		1010		10111
		1001		11011
				11101
				11110

Рисунок 1 – Примеры псевдо-исчерпывающих тестов

Одним из недостатков псевдо-исчерпывающих тестов, является сложность вычисления тестовых наборов, входящих в псевдо-исчерпывающий тест. Поэтому на практике для формирования подобных тестов используют многократные вероятностные тесты.

Классический однократный маршевый для тестирования  $k$  произвольных ячеек можно представить как совокупность двоичных векторов, разрядностью  $k$ , которая называется орбитой. [3]. Конкретный набор векторов, входящих в орбиту, зависит от трех основных факторов: правила формирования орбиты, очередности формирования адресов ячеек памяти и исходного содержимого ячеек памяти (рисунок 1).

Орбиты	$O_0$	$O_1$	$O_2$	$O_3$
Начальное состояние $P_0$	000...00	000...00	111...11	111...11
Последовательность адресов	Прямая (↑)	Обратная (↓)	Прямая (↑)	Обратная (↓)
$P_0$	000...00	000...00	111...11	111...11
$P_1$	000...01	100...00	111...10	011...11
$P_2$	000...11	110...00	111...00	001...11
...	...	...	...	...
$P_{k-2}$	001...11	111...00	110...00	000...11
$P_{k-1}$	011...11	111...10	100...00	000...01
$P_k$	111...11	111...11	000...00	000...00

Рисунок 2 – Четыре орбиты классического маршевого теста

Можно выделить четыре орбиты для классического маршевого теста, который имеет прямую и обратную очередность формирования адресов и два исходных содержимых ячеек памяти.

Анализ таблицы показывает, что каждая орбита, при тестировании произвольных  $k$  ячеек памяти, содержит  $k+1$  двоичные вектора  $P_0, P_1, \dots, P_k$ , разрядностью  $k$ .

Для повышения эффективности многократных тестов используется методы их формирования в которых на каждом шаге многократного теста изменяется начальное содержимое ячеек памяти, что позволяет существенно повысить покрывающую способность тестов [3].

Рассмотрим ОЗУ, которое содержит  $M = 2^n$  однобитных ячеек и каждую ячейку можно адресовать с помощью уникального  $n$ -разрядного адреса. Для любых  $k$  ячеек ОЗУ, где  $k$  больше единицы и меньше  $2^n$  существует  $2^k$  исходных состояний  $P_0$  для которых классический маршевый тест формирует  $2^k$  оригинальных орбит для фиксированной очередности следования адресов ячеек памяти [3].

Как показано в [3] для определенного  $k$  и фиксированной очередности следования адресов существуют  $2^k - (k^2 + k) / 2 - 1$  орбит  $O_n$  с разным исходным состоянием ОЗУ, в состав которых входят вектора, не входящие в орбиту  $O_0$ , а также количество орбит, равное  $(k^2 + k) / 2$  в состав которых входят два вектора, из орбиты  $O_0$ .

Для определения среднего количества итераций  $O_s$  многократного теста с изменяемым исходным состоянием памяти и фиксированным порядком следования адресов ячеек памяти предположим, что они формируются случайным образом и их значения равномерно распределены с вероятностью  $2^{-k}$ . Тогда для определения среднего количества итераций при реализации псевдо-исчерпывающего тестирования  $k$  ячеек ОЗУ, можно применить классическую задачу «Coupon Collector's Problem», где в качестве очередного купона служит орбита с новым исходным состоянием памяти [4]:

$$O_s = 1 + \frac{2^k}{2^k - 1} + \frac{2^k}{2^k - 2} + \dots + \frac{2^k}{2} + 2^k = 2^k \sum_{n=1}^{2^k} \frac{1}{n} . \quad (1)$$

Используя выражение 1 среднее значение количества итераций  $O_s$  для формирования всех  $2^k$  двоичных векторов многократных тестов типа  $MATS++$  равно  $O_s(MATS++) = O_s / k + 1$ . Для тестов  $MarchC-$  это значение определяется выражением  $O_s(MarchC-) = O_s / 2k$  [3].

На рисунке 3 изображены данные сравнительного анализа среднего количества итераций многократных тестов  $MATS++$  и  $MarchC-$  – полученные экспериментально и теоретически.

Экспериментальные и теоретические значения  $O_s$  для теста  $MATS++$  и  $MarchC-$

$k$	2	3	4	5	6	7	8	9	10
$2^k$	4	8	16	32	64	128	256	512	1024
Теор. $O_s$	2,77	5,44	10,82	21,64	43,37	86,93	174,20	349,00	699,03
Эксп. $O_s$	–	4,44	9,42	19,54	40,90	83,57	170,36	341,10	680,86

$k$	2	3	4	5	6	7	8	9	10
$2^k$	4	8	16	32	64	128	256	512	1024
Теор. $O_s$	2,77	3,62	6,76	12,98	25,30	49,67	97,98	193,89	384,47
Эксп. $O_s$	–	2,33	4,42	9,34	19,55	40,55	82,62	169,32	339,48

Рисунок 3 – Сравнительный анализ средней кратности тестов (MATS++) и March C–

Анализ эксперимента показывает, что значения среднего количества итераций  $O_s$  являются реализуемыми, поэтому можно сделать вывод о применимости многократных вероятностных тестов с изменяемыми исходными состояниями для псевдо исчерпывающего тестирования ОЗУ.

**Список использованных источников:**

1. Barzilai, Z. Exhaustive Generation of Bit Pattern with Application to VLSI Self-Testing / Z. Barzilai, D. Coppersmith, A. Rozenberg // IEEE Transactions on Computers. – 1983. – Vol. C-31, № 2. – P.190–194.
2. Nicolaidis, M. Theory of transparent BIST for RAMs // IEEE Transactions on Computers. – 1996. – Vol. 45, № 10. – P. 114.
3. Ярмолик, В.Н. Псевдоисчерпывающее тестирование ОЗУ / В.Н. Ярмолик, И. Мрозек, В.А. Леванцевич// Информатика. – 2017. – №2(54). – С.58–69.
4. Д. Кнут Устойчивые паросочетания и другие комбинаторные задачи / Д.Э.Кнут / МЦМНО. 2014, с 26-28

## ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА ЗАЩИТЫ ПРИВАТНОСТИ И ФИЛЬТРАЦИИ НЕЖЕЛАТЕЛЬНЫХ ПИСЕМ В ЭЛЕКТРОННОЙ ПОЧТЕ

*Литвинко П.М.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Лапицкая Н.В. – к.т.н., доцент*

В настоящей работе предлагается новый подход к решению проблемы фильтрации спама за счёт распознавания массовых рассылок и дополнительной проверки отправителя при обнаружении сообщений личного характера. Проверка осуществляется при помощи автоматизированного теста Тьюринга. Разработанная модель позволяет добиться высокой точности распознавания (98.81%) и эффективно осуществлять фильтрацию писем, а также защиту от утечек личной информации.

Электронная почта остаётся неотъемлемой частью социального взаимодействия и повсеместным средством коммуникации. По данным исследования Radicati Group, число активных пользователей электронной почты составляет 3,9 млрд на 2019 год [1]. По прогнозам, общее количество отправляемых и получаемых электронных писем в день к концу 2023 года вырастет до 347 миллиардов, а число пользователей - до 4,3 миллиардов.

По мере того, как число пользователей и объём пересылаемых сообщений неуклонно растёт с годами, несанкционированное использование электронной почты в маркетинговых и мошеннических целях, распространения вредоносного ПО и кражи личной информации становится одной из самых серьёзных проблем в почтовых службах. По оценкам, в 2010 году более 97% писем были классифицированы как спам [2]. При этом становится всё более распространённым феномен «перегрузки электронной почтой», впервые описанный ещё в 1996 году [3]. Именно поэтому многие исследования, изучающие или анализирующие электронные письма, сосредоточены на аспекте категоризации электронных писем, а также поиске эффективных инструментов управления почтой и снижения информационной перегрузки. Однако борьба между спамерами и инструментами защиты от спама продолжается, поскольку каждая сторона пытается создать новые способы преодоления методов, разработанных другой.

Традиционные методы определения спама включают применение байесовского классификатора [4], системы оценки на основе правил [5], проверку DNS MX-записей и обратный поиск по адресу [6], ведение чёрных списков IP и DNS (DNSBL) [7]. Однако каждый из приведённых объективных методов классификации при достижении заданного предела точности приводит к ложным положительным или неверным отрицательным срабатываниям. Идеальное решение не может быть найдено, поскольку каждая сторона использует уязвимости прежних алгоритмов для разработки новых способов доставки писем.

Принципиальная сложность разработки эффективных систем фильтрации спама заключается в том, само определение спама является субъективным и получателем электронных писем также является субъект. В то же время общей характеристикой спама является массовый характер рассылок, поскольку отправитель рассчитывает лишь на заданный процент доставки и открытия электронных писем, что является уже объективным критерием.

В настоящей работе для решения задачи фильтрации нежелательных писем, предлагается сместить область применения объективных методов классификации для решения задачи распознавания рассылки массового характера и использовать принципиально новый подход для проверки легитимности доставки электронного письма. Использование комбинации двух методов позволит более эффективно решить общую задачу определения и фильтрации спама.

Схема работы системы представлена на рисунке 1. При получении входящего сообщения происходит извлечение текста и разделение его на n-граммы. Затем, с помощью многослойного перцептрона (MLP) осуществляется предсказание класса письма: рассылка (Newsletter) – информационные сообщения, не требующие ответа или личное сообщение (Personal) – письма, предполагающие ответ или взаимодействие. В ходе исследования, данная модель показала наивысшую точность распознавания на тестовой выборке (98.81%), для обучения использовалось 37800 размеченных писем. Далее, для личных сообщений, которые и представляют угрозу оказаться спамом, происходит проверка адреса отправителя по списку контактов и в случае, если письмо отправлено с неизвестного адреса, запускается процедура проверки легитимности доставки электронного письма.

Данная процедура включает следующие шаги:

- 1) подозрительное письмо помечается как небезопасное и удаляется из списка непрочитанных сообщений на почтовом сервере пользователя;
- 2) генерируется ответное письмо, содержащее инструкции и уникальную ссылку для прохождения автоматизированного теста Тьюринга отправителем;
- 3) в случае успешного прохождения теста, письмо помечается как проверенное, переносится в список входящих сообщений, а отправитель получает подтверждение доставки.

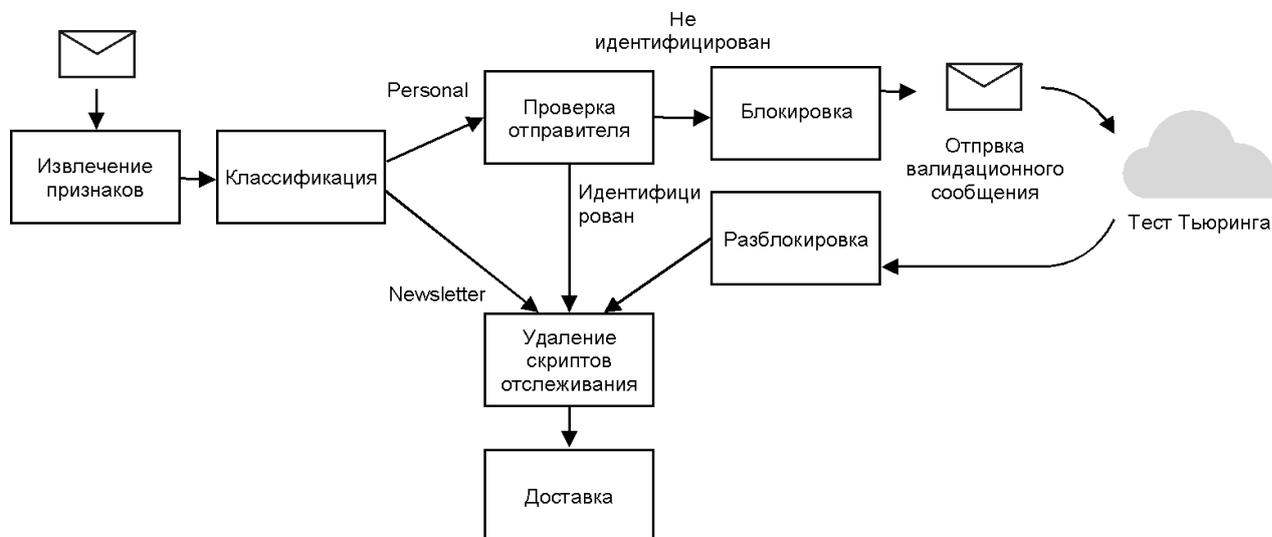


Рисунок 1 – Схема работы интеллектуальной системы фильтрации спама

В качестве автоматизированного теста Тьюринга в настоящей работе используется сервис геCAPTCHA от Google, обеспечивающий необходимый уровень сложности и в то же время позволяющий пропустить или упростить проверку для авторизованных пользователей. Для массовых спам-рассылок прохождение теста становится нецелесообразным и слишком трудоёмким для автоматизации, обеспечивая необходимую уверенность в том, что письмо было предназначено конкретно этому пользователю. Кроме того, на данном этапе может быть дополнительно осуществлена проверка личности отправителя для минимизации риска целенаправленного мошенничества (фишинга, вымогательства, попыток шантажа).

Помимо фильтрации спама, система выполняет также и проверку содержимого для предотвращения утечек персональной информации (IP адрес, местоположение пользователя, используемый браузер, почтовый клиент, операционная система и др.) и отслеживания таких действий пользователя как открытие/повторное открытие письма, переход по ссылке, открытие/просмотр вложений, пересылка письма и другие, широко применяющиеся в маркетинговых и мошеннических целях. В ходе работы было обнаружено более 40 различных способов внедрения скрытого кода отслеживания, каждый из которых был протестирован в 8 наиболее популярных клиентских приложениях электронной почты: Gmail (веб-версия), Gmail (Android), Gmail (iOS), Яндекс.Почта (веб-версия), Яндекс.Почта (iOS), Mail (Mac OS), Outlook (веб-версия), Mail.ru (веб-версия). В 44% случаев была зафиксирована утечка персональной информации при получении электронного письма. После внедрения разработанной системы этот показатель снизился до 3%.

Таким образом, разработанная модель позволяет эффективно распознавать массовые рассылки и персональные электронные письма, фильтруя нежелательные письма за счёт дополнительной проверки отправителя. Распознавание и изоляция скриптов отслеживания позволяет обеспечивать защиту личных данных получателя и блокировать триггерные маркетинговые рассылки. Разработанное программное средство может быть внедрено в существующие почтовые сервисы по протоколам SMTP и IMAP.

**Список использованных источников:**

1. The Radicati Group, Inc [Electronic resource] : Email Statistics Report, 2019-2023 – Executive Summary. – Mode of access: <https://www.radicati.com/wp/wp-content/uploads/2018/12/Email-Statistics-Report-2019-2023-Executive-Summary.pdf>. – Date of access: 24.03.2019.
2. Salomon, D. Elements of Computer Security / D. Salomon – Springer Verlag, 2010. – 374 P.
3. Email overload: Exploring personal information management of email : in proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 1996, Vancouver, British Columbia, Canada, April 13-18, 1996 / SIGCHI; ed.: S.Whittaker [et al.]. – Vancouver, SIGCHI 1996. P. 276-283.
4. Zdziarski, J. Ending Spam — Bayesian Content Filtering and the Art of Statistical Language Classification / J. A. Zdziarski – 5th Edition. – No Starch Press, San Francisco, 2005. P – 451.
5. Spam Classification Based on Supervised Learning Using Machine Learning Techniques : papers from the International Conference on Process Automation, Control and Computing, Coimbatore, 20-22 July, 2011 / Coimbatore Institute of Technology; ed.: D. K. Renuka [et al.]. – Coimbatore Institute of Technology, Coimbatore, 2011. P. 1-4.
6. DNS Resource Record Analysis of URLs in E-Mail Messages for Improving Spam Filtering : IEEE/IPSJ 11th International Symposium on Applications and the Internet (SAINT), Munich, 18-21 July, 2011 / IEEE Computer Society; ed.: S. Suwa, N. Yamai, K. Okayama, M. Nakamura – IEEE Computer Society, Munich, 2011. P. 439-444.
7. Improving the Efficiency of Spam Filtering through Cache Architecture : 15th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, Istanbul, 24-26 October, 2007. P. 303-309.

## ОБЗОР РЕШЕНИЙ ПО МОДЕЛИРОВАНИЮ ДВИЖЕНИЯ ПОТОКОВ ЛЮДЕЙ ПРИ ЭВАКУАЦИИ ИЗ ПОМЕЩЕНИЙ

*Лухута Е.И.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Парамонов А.И. – к.т.н., доцент*

В данной работе представлен сравнительный обзор современных решений, которые используются для разработки планов эвакуации и моделирования эвакуации потоков людей из зданий в случае чрезвычайной ситуации. Рассмотрены преимущества и особенности отдельных программных комплексов.

С точки зрения психологии паника представляет собой одно из самых опасных психологических состояний для жизни человека. Группа людей, которые подверглись панике, способны многократно увеличить общее число жертв в результате чрезвычайной ситуации (ЧС). Это подтверждают проведённые в данной области многочисленные исследования.

На сегодняшний день существует достаточно много математических моделей, на основе которых созданы различные действующие программные комплексы, способные моделировать движения потоков людей при эвакуации из различных зданий и сооружений в случае ЧС.

Использование современных специализированных программных комплексов является показательным способом при расчёте эвакуационных способностей зданий и сооружений. Основное преимущество данного подхода заключается в возможности моделирования чрезвычайной ситуации с учётом весомого числа внешних переменных.

На данный момент существует достаточное количество моделей, которые позволяют настроить детали организации внутренней среды, ключевые особенности движения потоков людей, а также особенности их поведения в условиях ЧС.

Различают следующие модели движения потоков людей:

- упрощенная аналитическая модель,
- имитационно-стохастическая модель,
- индивидуально-поточная модель.

Далее рассмотрим современные программные решения, которые способны эмулировать перечисленные выше модели при моделировании эвакуации из зданий и сооружений.

Программный комплекс «PedGo» способен моделировать перемещение толпы людей при эвакуации людей из зданий, сооружений, а также из самолетов, кораблей и прочих видов общественного транспорта [1]. Модель, используемая в данном комплексе, имитирует решение и движение каждого отдельно взятого человека, поэтому план помещения исследуемого сооружения делится на квадратные ячейки размером 0,4 x 0,4 м. Место, занимаемое человеком, а также стены,

мебель и прочие препятствия представлены данными ячейками. Данный продукт перед началом расчетов предлагает задать ряд следующих характеристик поведения людей: задержка перед началом эвакуации, бездействие, терпение, реакция.

Программный комплекс «Simulex» позволяет эмулировать движения крупных потоков людей в случае чрезвычайной ситуации из зданий с достаточно сложной геометрической архитектурой [2]. Основной особенностью данного продукта является работа с группами людей. Комплекс позволяет разделять людей на типы и объединять их в различные группы с учётом следующих параметров: форма и размер тела, скорость ходьбы, подъема и спуска и время реакции на тревогу. Изменение всех этих параметров позволяет протестировать широкий спектр типов людей.

Программный комплекс «Pathfinder» представляет собой необходимый инструмент для проектирования надежных решений относительно плана здания и систем противопожарной защиты [3]. Отличительной особенностью данного продукта от предыдущих является то, что при моделировании чрезвычайной ситуации возможно разбиение людей на две ключевые группы: клиенты и помощники. Главным отличием помощников от клиентов является возможность оказания помощи, в результате чего помощники могут выступать в роли спасателей. Одна группа помощников способна передать малоподвижного клиента другой, тем самым моделируя работу спасательной службы. А трёхмерное представление ситуации в реальном времени с возможностью настройки эскалаторов, лифтов и турникетов позволяет эмулировать ситуацию, максимально схожую с реальной.

**Список использованных источников:**

1. PedGo [Electronic resource]. – Mode of access: <https://www.traffgo-ht.com/en/pedestrians/products/pedgo/pedgo/index.html>. – Date of access: 05.03.2019.
2. Simulex [Electronic resource]. – Mode of access: <https://www.iesve.com/software/ve-for-engineers/module/Simulex/480#technical-detail>. – Date of access: 07.03.2019.
3. Pathfinder [Electronic resource]. – Mode of access: <https://www.thunderheadeng.com/pathfinder/pathfinder-features/>. – Date of access: 10.03.2019.

## **УНИФИЦИРОВАННЫЙ ИНТЕРФЕЙС ДЛЯ СОЗДАНИЯ КРОССПЛАТФОРМЕННЫХ ВЕБ-ОРИЕНТИРОВАННЫХ SMART TV ПРИЛОЖЕНИЙ**

*Максимченко А.В.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Таборовец В.В. – к.т.н., доцент*

Рассматриваются вопросы разработки и создания унифицированного интерфейса, который инкапсулирует логику взаимодействия приложения с конкретной версией телевизора Smart TV, распознаёт текущую платформу Smart TV, и предоставляет разработчикам открытый интерфейс, позволяющий реализовать различные функции для работы со Smart TV телевизорами.

В современном мире при появлении Smart TV телевизоров вопрос о создании для них веб-приложений становится все более и более острым и актуальным. За последнее 10 лет появилось большое количество Smart TV платформ, и каждая из которых, как и разработка приложений под них, в большинстве случаев отличается друг от друга. Для обеспечения возможности тв-приложения работать одинаковым образом на всех существующих и используемых платформах и операционных системах Smart TV телевизоров предлагается разработать унифицированный интерфейс для создания платформо-независимых веб-ориентированных Smart TV приложений.

Smart TV, или Connected TV – это технология интеграции интернета и цифровых интерактивных сервисов в современные телевизоры и ресиверы цифрового телевидения, а также в техническом симбиозе между компьютерами и телевизорами / ресиверами цифрового телевидения [1].

После анализа платформ и операционных систем современных телевизоров с функцией Smart TV было выявлено, что каждая система предоставляет свой набор характерных настроек.

Каждая операционная система Smart TV такая, как WebOS, Orsey, Tizen, Netcast, требует детального анализа и понимания своих характерных особенностей для успешной реализации различного рода функций в видео-плеере и с видео-материалом: воспроизведение, пауза, режим промотки в прямом и обратном направлении, на определенную позицию, регулирование громкости и многое другое.

Схожая проблема возникает и при реализации взаимодействия Smart TV телевизора и разрабатываемого приложения с клавишами на пульте телевизора. Каждая операционная система Smart TV предоставляет свой собственный набор кодов кнопок, которые приложение должно правильно обрабатывать: определять, к какой операционной системе они относятся, и реагировать должным образом. Примеры разнообразия кодов клавиш можно увидеть на ресурсах [2] и [3].

Часто при реализации веб-приложений для Smart TV приходится делать проверки на наличие интернет-соединения, определение готовности телевизора к взаимодействию с приложением или готовности осуществить выход из приложения. Поскольку многие телевизоры по-разному это осуществляют, разработчикам приходится писать немало программного кода, учитывающего все особенности каждой поддерживаемой телеплатформы для успешной реализации этих функций.

Решение этих проблем достигается путем разработки и реализации унифицированного интерфейса, который инкапсулирует логику взаимодействия приложения с конкретной версией телевизора, распознаёт текущую платформу Smart TV, и предоставляет разработчикам открытый интерфейс, позволяющий реализовать функции для работы со Smart TV телевизорами без углубления в работу самого интерфейса.

Унифицированный интерфейс реализует сервис, который позволяет создать видео-плеер, управление приложением при помощи контроллеров мыши и клавиатуры, осуществлять различные проверки телевизора на работоспособность, доступ к сети и многое другое. При его использовании, сервис определяет внутри себя текущую платформу и операционную систему, на которой выполняется приложение, и на основании этих определений производит, характерные для данного телевизора инструкции.

Разработанный унифицированный Smart TV интерфейс позволяет абстрагироваться от знаний о конкретных интерфейсах различных платформ и операционных систем Smart TV телевизоров и позволяет полностью сконцентрироваться на решении бизнес-проблем, что является важной его особенностью.

**Список использованных источников:**

1. Википедия [Электронный ресурс]. - Электронные данные. - Режим доступа: [https://ru.wikipedia.org/wiki/Smart\\_TV](https://ru.wikipedia.org/wiki/Smart_TV).
2. Handling Control Key Events | Samsung Developers [Электронный ресурс]. - Электронные данные. - Режим доступа: <http://developer.samsung.com/tv/develop/legacy-platform-library/art00046/index>.
3. LG | webOS TV Developer | Remote Control [Электронный ресурс]. - Электронные данные. - Режим доступа: <http://webostv.developer.lge.com/design/webos-tv-system-ui/remote-control>.

## **ПРОГРАММНОЕ СРЕДСТВО «ОБЩЕРЕСПУБЛИКАНСКИЙ БАНК ДАННЫХ УЧАСТНИКОВ РЕПЕТИЦИОННОГО ТЕСТИРОВАНИЯ» С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ ASP .NET CORE**

*Малец И.В., Тимофеев Н.И., Зайкина И.С.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Парамонов А.И. – к.т.н., доцент*

На первых этапах проведения репетиционного тестирования каждый пункт тестирования реализовывал свои возможности автоматизации данного бизнес-процесса. Это усложняло возможность централизованного управления репетиционным тестированием. Перечень функциональных возможностей каждой системы так же отличался. Поэтому возникла потребность в создании единого программного средства, которое могли бы использовать все пункты тестирования для организации и проведения репетиционного тестирования.

Эпоха новых информационных и коммуникационных технологий принесла значительные перемены в сферу производства, образования и деловой активности людей. Сферу образования можно назвать флангом, который принимает на себя основной удар. Одним из таких направлений является организация проведения репетиционного тестирования (РТ), как составляющей всего этапа проведения проверки знаний будущих абитуриентов. Обязанности организации и проведения РТ возложены на «Республиканский институт контроля знаний».

Предлагается решение в виде информационной системы, которая позволит организовать работу в рамках «одного окна», и осуществлять управление пунктами регистрации и проведения РТ.

В ходе проведения анализа предметной области, были выделены основные функциональные требования, предъявляемые к программному средству (ПС):

- филиальная возможность работы;
- регистрация новых пунктов;

- организация доступа к регистрации;
- формирования списков зарегистрированных пользователей;
- контроль явки на РТ;
- личный кабинет участника РТ;
- отслеживание статуса регистрации;
- формирование и подтверждение актов приема-передачи бланков ответов;
- осуществление платежей в режиме реального времени с помощью системы «Расчет»;
- оформления пропусков;
- экспорт и импорт данных из Excel-файла;
- формирование сводной ведомости;
- генерация статистики по репетиционному тестированию;
- загрузка результатов репетиционного тестирования.

На основании списка требований построена функциональная модель программного средства в нотациях IDEF0 [1], а также диаграмма вариантов использования (рисунок 1).

Были выделены следующие действующие лица: главный администратор, администратор, пользователь. Для разрабатываемой системы, исходя из потребностей действующих лиц, выделены следующие варианты использования: вход в систему, просмотр списка пунктов тестирования (ПТ), изменение ПТ, добавление ПТ, загрузка результатов, выбор файла результатов, подтверждение актов приёма-передачи бланков ответов, просмотр сеансов тестирований, изменение сеанса тестирования, просмотр списка зарегистрированных, печать ведомости, отметка явившихся, добавление сеанса тестирования, регистрация в системе, ввод своих данных, просмотр списка регистраций, отмена регистрации, просмотр подробной информации, регистрация на РТ.

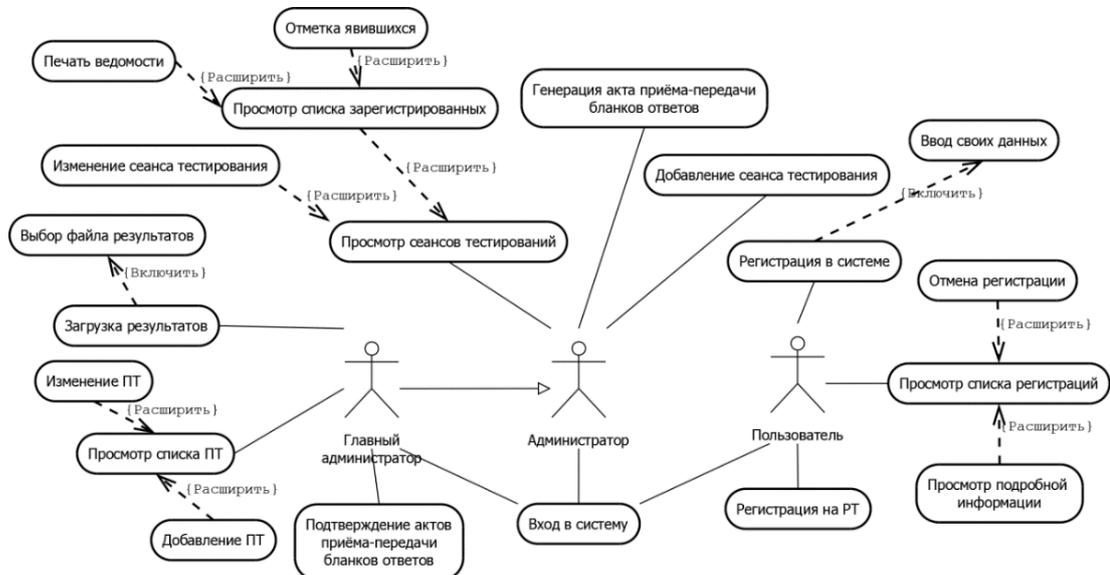


Рисунок 1 – Диаграмма вариантов использования

В качестве языка программирования для серверной части приложения был выбран C#, а именно фреймворк ASP.NET Core [2]. В качестве языка программирования для клиентской части приложения был выбран Typescript, фреймворк Angular [3]. Для хранения информации будет использовано СУБД Microsoft SQL Server 2017. Для разработки программного средства «Общереспубликанский банк данных участников репетиционного тестирования» будет использоваться среда разработки Visual Studio 2017.

Интерфейс программного средства представлен на рисунке 2.

Реализована возможность многопользовательского режима работы. Разработан понятный и простой интерфейс для работы с программным средством с учетом особенностей предметной области.

Выходными данными программного средства являются: обобщенная ведомость участников репетиционного тестирования, отображаемая в виде файла pdf; файл Microsoft Excel обобщенных данных участников тестирования, зарегистрированных на конкретную дату; пропуск для прохождения тестирования в выбранном пункте в виде файла pdf. Обобщенные сведения доступны только администраторам пунктов тестирования. Печать пропуска доступна всем участникам тестирования, которые произвели оплату созданной регистрации на выбранный предмет и выделенную дату.

 МС 2151450 Выйти



Тимофеев Никита Игоревич

Лицевой счёт для оплаты: **0000002**

Количество свободных оплат: **1**

✎ Редактировать

Номер регистрации	Дата	Название ПТ	Место	Предмет	Статус	
0000013	12.12.2018 14:00	ГУО «Средняя школа №4 г. Червеня»	ул. Барыкина, 91а, ФОК	Русский язык Язык теста: русский	Результат 65	<a href="#">👁 Подробнее</a>
0000049	21.12.2018 14:00	ГУО «Средняя школа №4 г. Червеня»	ул. Барыкина, 91а, ком. 15	Физика Язык теста: русский	Не явился	<a href="#">👁 Подробнее</a>
0000059	31.12.2018 14:00	УО «Республиканский институт контроля знаний»	Оранжевое здание, 2-ой этаж, к. 27	Русский язык Язык теста: русский	Оплачено	<span style="background-color: red; color: white; padding: 2px 5px;">✖ Отменить</span> <a href="#">👁 Подробнее</a> <span style="background-color: #0070C0; color: white; padding: 2px 5px;">🔄 Пропуск</span>
0000060	29.12.2018 14:00	ГУО «Средняя школа №4 г. Червеня»	ул. Барыкина, 91а, акт. зал	Физика Язык теста: белорусский	Ожидается оплата	<span style="background-color: red; color: white; padding: 2px 5px;">✖ Отменить</span> <span style="background-color: #0070C0; color: white; padding: 2px 5px;">💰 Оплатить</span> <a href="#">👁 Подробнее</a>

Рисунок 2 – Домашняя страница пользователя

Внедрение разработанного программного средства автоматизировало процесс регистрации на РТ во всех пунктах тестирования, а также облегчило работу организаторов репетиционного тестирования в пунктах его проведения. Личный кабинет участника РТ позволил легче производить регистрацию на репетиционное тестирование, а также отслеживать статус своих регистраций.

Отпала необходимость в проведении различного рода совещаний, командировок, письменных коммуникаций.

**Список использованных источников:**

1. Маклаков, С. В. Инструментальные средства создания корпоративных информационных систем / С. В. Маклаков. – Компьютер Пресс. 1998. – №7.
2. Чамберс Д., Пэккетт Д., Тиммс С., ASP.NET Core. Разработка приложений/ Д. Чамберс – 2018. – 464 с.
3. Файн Я., Моисеев А., Angular и TypeScript. Сайтостроение для профессионалов/ Я. Файн 2018 – 368 с.

## ИНФОРМАЦИОННАЯ СИСТЕМА СБОРА И ОБРАБОТКИ ТЕЛЕМЕТРИЧЕСКИХ ДАННЫХ

*Малец И.В., Тимофеев Н.И., Мохонкова Е.В.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Пармонов А.И. – к.т.н., доцент*

В современном мире, наполненном большим количеством информационных систем и технологий, важное место занимает их всесторонний мониторинг. Суть его заключается в сборе необходимых данных и их анализе. Анализ собранной информации об устройстве или программном средстве позволяет предотвратить ошибки или, на основании полученной информации, оптимизировать их производительность. Практически все современные устройства и программные средства имеют собственные системы мониторинга. Но в случае, когда количество систем для мониторинга слишком велико, или к ним нет непосредственного доступа, – стандартных средств становится недостаточно.

Сбор, хранение и обработка телеметрических данных становятся все более актуальными задачами. На сегодня известно уже большое количество систем и программных средств для её решения. Часть из них нацелена на мониторинг конкретных типов систем – программные средства для мониторинга сетей: Observium [1], SolarWinds Network Performance Monitor [2]. Другие, более гибкие, могут работать с большим количеством устройств и протоколов, по средствам добавления в них модулей или установки специальных приложений-агентов. Примером таких систем являются Zabbix [3] или CA UIM [4]. Практически все современные системы сбора и обработки телеметрических данных имеют веб-интерфейс для доступа к данным из любой точки мира. В последнее время многие системы для сбора и обработки телеметрических данных становятся доступны в формате облачных

сервисов, что упрощает для пользователей настройку, увеличивает надежность и отказоустойчивость системы и не требует от пользователя мощного оборудования и больших объемов памяти.

Существует два основных способа сбора телеметрических данных. Первый способ – система для сбора телеметрических данных сама опрашивает устройства или программное обеспечение по средствам протокола или программного интерфейса. Второй способ – приложения-агенты, которые опрашивают устройство или программное обеспечение и передают данные на сервер системы мониторинга. Приложения агенты так же, как и в первом случае, опрашивают устройства или программные средства с помощью протокола или программного интерфейса, либо же являются частью опрашиваемых систем. В одной системе одновременно могут использоваться оба способа. Системы использующие первый способ обычно более просты в конфигурации, но сложны в доработке. Второй способ может потребовать установки приложения-агента непосредственно на целевое устройство, а также настройки самой системы и приложений агентов по отдельности для каждого источника телеметрических данных.

Преимущества второго способа – это гибкость, возможность разработки новых приложений агентов и модификации существующих, независимо от основных частей системы и других приложений агентов.

Пример систем с приложениями агентами и без них представлены на рисунке 1.

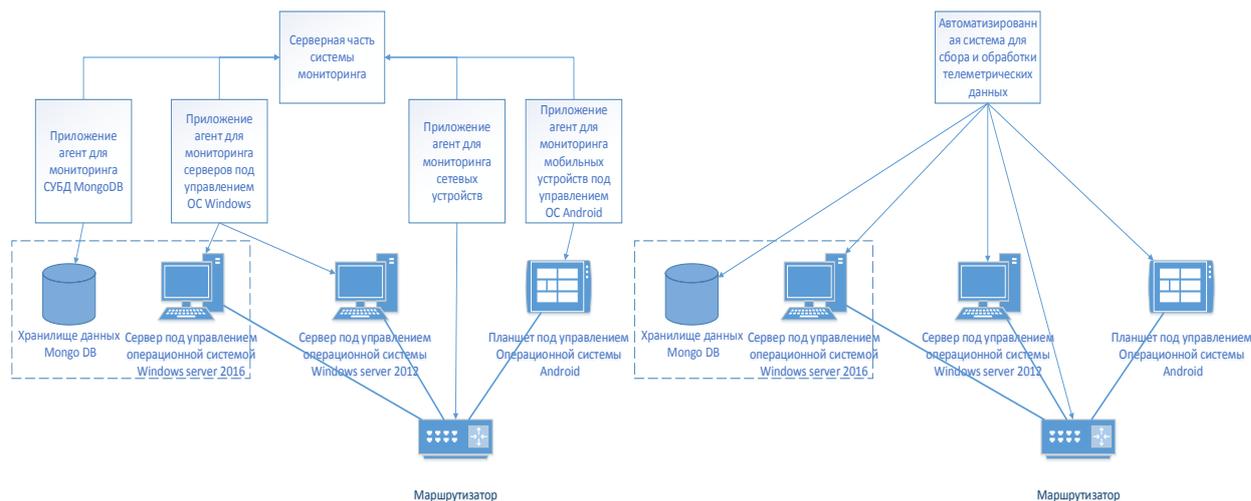


Рисунок 1 – Пример систем с приложениями агентами и без агентов

Анализ систем сбора и обработки телеметрических данных показал, что большинство существующих систем разрабатываются для конкретного сценария использования и типа пользователя (например, для администраторов сетей) и сложны в настройке. Поэтому была предпринята попытка разработать простую в настройке и использовании, гибкую систему для сбора и обработки телеметрических данных.

На рисунке 2 представлена страница веб интерфейса разработанного приложения с информацией об источнике данных (iPhone SE). На странице изображен блок с названием устройства, типом и статусом. Можно просмотреть данные устройства в реальном времени или за выбранный период, установить пороговые значения для данных.

Система имеет серверную часть, которая принимает, хранит и обрабатывает данные, простой веб-интерфейс с возможностью просмотра и анализа данных разного типа, получения данных по средствам приложений агентов. Система имеет открытый исходный код и открытый API. Открытый исходный код позволит конечным пользователям, при необходимости, доработать систему для своих нужд. Открытый API позволит пользователям или третьим лицам разрабатывать новых агентов для системы. На начальных этапах работы системы разработано несколько приложений-агентов для мониторинга наиболее популярных систем и протоколов (для персональных компьютеров, работающих на базе операционных систем Windows, Linux. Для SNMP протокола). Одним из первых разработано приложение-агент для мониторинга мобильных устройств. Несмотря на популярность такого типа устройств, приложения для их мониторинга практически отсутствуют.

Использование системы мониторинга позволит упростить получение детальной информации о различных системах за счет снижения количества рутинных операций.

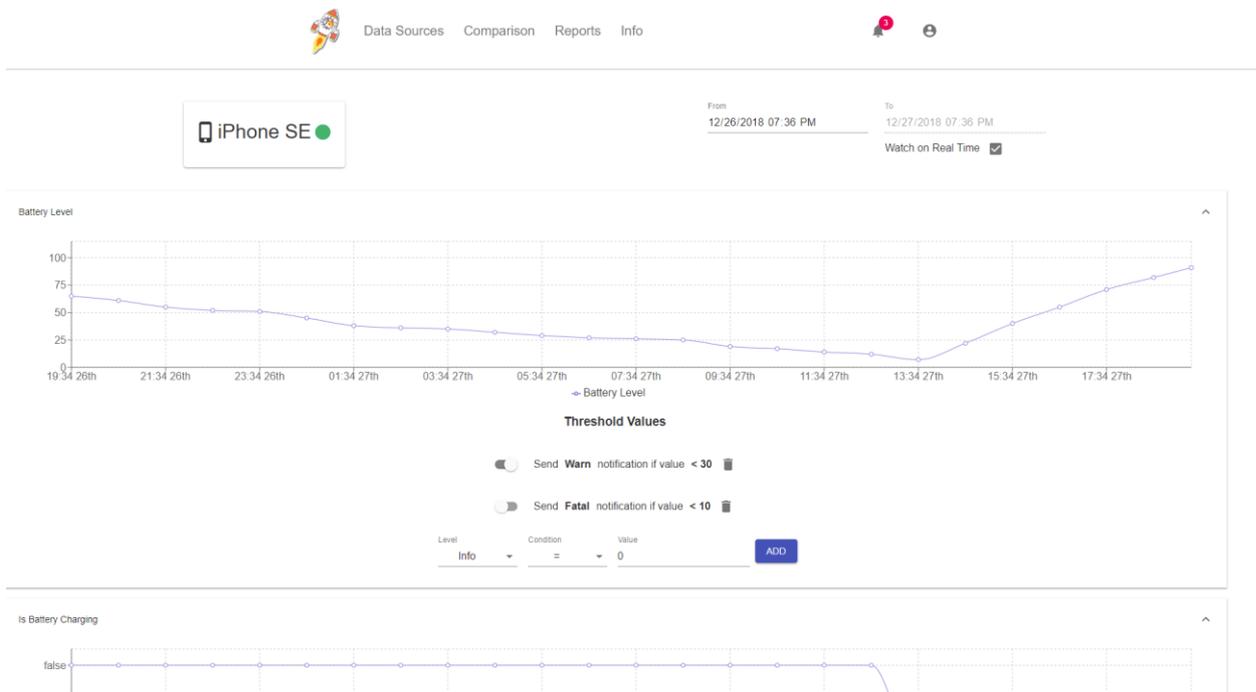


Рисунок 2 – Страница с данными устройства

Планируется распространять продукт в открытом доступе с открытым исходным кодом, а также при необходимости дорабатывать систему под требования заказчика. Такая система может быть использована системными администраторами для мониторинга сетей, пользователями с большим количеством устройств. Для мониторинга умных домов и многого другого, за счет разработки новых приложений агентов.

#### Список использованных источников

1. Observium [Электронный ресурс] – Электронные данные. – Режим доступа: <http://www.observium.org>. Дата доступа: 12.02.2019 г.
2. Solarwinds Network Performance Monitor [Электронный ресурс] – Электронные данные. – Режим доступа: <https://www.solarwinds.com/network-performance-monitor>. Дата доступа: 12.02.2019 г.
3. Zabbix [Электронный ресурс] – Электронные данные. – Режим доступа: <https://www.zabbix.com/ru/>. Дата доступа: 12.02.2019 г.
4. CA Unified Infrastructure Management (UIM) [Электронный ресурс] – Электронные данные. – Режим доступа: <https://www.ca.com/us/products/ca-unified-infrastructure-management.html>. Дата доступа: 12.02.2019 г.

## ПРОБЛЕМА БАЛАНСИРОВКИ НАГРУЗКИ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ

*Марков А.Н.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Парамонов А.И. – к.т.н., доцент*

Отражена проблема распределения нагрузки на сетевое и серверное оборудование, а также балансировка нагрузки на кластерных мощностях.

В современном мире облачные вычисления не только приобретают популярность, но и становятся основной частью инфраструктуры интегрированных систем, интернета вещей, организацией работы крупных компаний и сервисов. Повсеместная цифровизация приводит к значительному увеличению нагрузки на сетевое и серверное оборудование. Увеличение нагрузки же неизбежно влечет за собой вопрос распределения мощностей оборудования на выполняемые задачи.

Цель балансировки нагрузки – оптимизация использования ресурсов, максимизация пропускной способности, уменьшение времени отклика и предотвращение перегрузки какого-либо ресурса. Использование нескольких компонентов балансировки нагрузки вместо одного может повлиять

надежность и доступность за счет резервирования. Балансировка нагрузки предполагает обычно наличие специального программного обеспечения или аппаратных средств, таких как многоуровневый коммутатор или система доменных имен, как серверный процесс [1].

В данный момент существует несколько глобальных подходов к распределению нагрузки в облачных вычислениях. Условно их можно разделить по логике построения: сетевая балансировка и балансировка по загруженности серверного оборудования.

Сетевая балансировка нагрузки заключена в распределении информационных потоков от пользователей (клиентов) между кластерными системами. Когда речь идет о балансировках сетевой нагрузки, термины балансировки нагрузки и прокси используются примерно одинаково. Используя модель OSI логически балансировку нагрузки можно представить на транспортном и прикладном уровнях. Транспортный уровень (L4) управляет балансировкой с помощью протоколов управления передачи. Прикладной уровень (L7) управляет балансировкой за счет протоколов прикладного уровня.

Балансировка непосредственно на кластерных мощностях осуществляется с помощью анализа загруженности конкретного оборудования в данный момент времени, а также прогнозирование загрузки. Для долгосрочных прогнозов используется машинное обучение с учителем, основанное на гауссовских процессах.

Динамическое распределение ресурсов для виртуальных машин (VM) в облачной среде объединяет локальное и глобальное распределение ресурсов VM. Локальное распределение ресурсов означает выделение общих ресурсов хоста для VM в соответствии с текущей нагрузкой. Глобальное распределение ресурсов равносильно живой миграции, в случае перегрузки или недогрузки хоста, с целью обеспечения достаточной производительности VM или уменьшение количества хостов для экономии электроэнергии. Для учёта неопределенности долгосрочных прогнозов при обнаружении перегрузок в режиме онлайн строится модель распределения ошибки предсказания с использованием метода ядерной оценки плотности вероятности.

Однако совокупность сетевой балансировки и балансировки кластеров (виртуальных машин, оборудования и т.д.) зависит от SLA (Service Level Agreement) – термин методологии ITIL, обозначающий формальный договор между заказчиком услуги и её поставщиком, содержащий описание услуги, права и обязанности сторон и, самое главное, согласованный уровень качества предоставления данной услуги. Как правило термин SLA используется применительно к ИТ и телекоммуникационным услугам. В таком соглашении может содержаться детальное описание предоставляемого сервиса, включая перечень параметров качества, методов и средств их контроля, времени отклика поставщика на запрос от потребителя, а также штрафные санкции за нарушение этого соглашения. Для того, чтобы соблюсти SLA, поставщик услуг заключает операционное соглашение об уровне услуг (OLA, operational-level agreement) с другими внутренними подразделениями, от которых зависит качество предоставления услуг [2]. Таким образом непосредственно балансировка нагрузки в облачных вычислениях, учитывая факторы загрузки сетевого и серверного оборудования решена в распределенных системах.

**Список использованных источников:**

1. Д.А. Варенов, А.А. Больных. Исследование подходов к разработке распределенной вычислительной системы на базе кластерной архитектуры для функционирования веб-приложений и сервисов обработки научно-образовательных ресурсов. //Труды XVII Всероссийской научно-методической конференции «Телематика'2010». Том 2. – СПб.: "Университетские телекоммуникации", 2010. – С. 378-379. – ISBN 978-5-7577-0354-1.

2. Proceedings of the twenty-second annual symposium on Principles of distributed computing / Annual ACM Symposium on Principles of Distributed Computing. – Boston, Massachusetts, USA. – 2003. – 380 с. ISBN:1-58113-708-7

## **SAFE МЕТОДОЛОГИЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

*Марценюк Р.В.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Смолякова О.Г. – к.т.н., доцент*

В настоящее время среди предприятий, занимающихся разработкой программного обеспечения, все больше начинают распространяться Agile-методологии. Одним из подвидов гибкой методологии является Scaled Agile Framework или, как его называют, масштабированный Agile.

В большинстве случаев причиной перехода на управление процессами с использованием методологии SAFe является неудовлетворенность другими подвидами гибкой методологии

(например, Scrum и Kanban) или неспособность существующих методик удовлетворить требованиям, выдвинутым к процессам.

В настоящее время все больше крупных компаний сталкиваются с ситуацией, когда над одним проектом или продуктом работает одновременно несколько команд. В таких случаях применение Scrum методологии не приносит положительных результатов, так как данная методология направлена на работу небольшого количества команд над одним проектом. Первым возможным решением может быть увеличение количества работающего персонала в пределах одной команды, однако таким способом будет ухудшено взаимодействие между членами команды. Другим решением является увеличение числа команд, но стоит заметить, что методология Scrum не предоставляет механизмов для обеспечения синхронизации между большим количеством команд.

Для решения вышеописанной проблемы была разработана методология SAFe (иногда называют фреймворком). По своему определению данная методология предназначена для разработки программного обеспечения с использованием гибких методик в командах размером 50 и более человек. SAFe методология является объединением лучших качеств и инструментов других гибких методологий.

Одним из важнейших факторов при формировании рассматриваемой методологии являлись принципы Lean. Данное понятие возникло в 1990-ых годах в автомобильной промышленности и представляет собой подход к управлению организацией, направленный на повышение качества совершаемой работы за счет сокращения потерь. Ниже представлены 5 основных принципов Lean [1]:

- определение ценности продукта с точки зрения конечного пользователя;
- определение и детальное описание всех действий в цепочке производства;
- формирование действий таким образом, чтобы между ними не было простоев и ожиданий;
- ограничение количества выполняемых задач, за счет выполнения только тех, которые важны для конечного пользователя;

– стремление к постоянному совершенствованию процессов и продукта.

Другим влияющим фактором при формировании SAFe является Agile манифест – документ, содержащий основные принципы и ценности гибкой методологии разработки. Данный документ включает описание 4 ценностей и 12 принципов. Ниже представлены некоторые основные принципы [2]:

- наивысшим приоритетом является удовлетворение потребностей заказчика;
- изменение требований приветствуется на всех стадиях разработки;
- работающий продукт следует выпускать как можно чаще, с периодичностью от пары недель до пары месяцев;
- на протяжении всего проекта разработчики и представители бизнеса должны ежедневно работать вместе;
- постоянное внимание к техническому совершенству и качеству проектирования повышает гибкость проекта.

Еще одной причиной успешного распространения SAFe является разделение фреймворка на 4 уровня: командный, программный, системный и уровень портфолио. Так как команда является основной сущностью в любом подвиде Agile-методологии, то можно сделать вывод, что командный уровень – это самый главный и неотъемлемый уровень SAFe. Команда, как и в Scrum, состоит из 10 человек, включая специалистов из различных областей, владельца продукта (Product Owner) и скрам-мастера (Scrum-master). Задачи последних двух являются строго зафиксированными. Например, владелец продукта является голосом бизнеса внутри команды, определяет и принимает требования к выполняемым задачам. Скрам-мастер занимается ликвидацией препятствий для работы команды, защищает ее от внешних отвлекающих факторов, проводит и назначает митинги. Как и в Scrum, разработка происходит итеративно. Основной единицей является программный инкремент – период времени, включающий в себя 5 итераций. Каждая итерация длится 2 недели.

Присутствие следующих уровней может зависеть от потребностей и размеров предприятий.

На программном уровне обычно происходит приоритизация задач, построение планов по развитию архитектуры и осуществляется синхронизация между всеми командами. Именно на данном уровне должно происходить поддержание потока задач в таком виде, в котором это необходимо для достижения целей, поставленных заказчиком. Неотъемлемой частью SAFe процесса на программном уровне является планирование задач перед началом каждого программного инкремента.

Системный уровень служит для координации между различными отделами, их руководителями и заказчиками. На данном этапе производится анализ экономической целесообразности проведения изменений.

Уровень портфолио является самым верхним уровнем фреймворка SAFe. Здесь происходит распределение средств на различные направления в бизнесе, корректировка бюджета компании и управление человеческими ресурсами, необходимых для предоставления готовых решений.

Как утверждают создатели методологии, основными задачами SAFe являются [3]:

- улучшение производительности на 20–50%;

- улучшение качества продукта на 25–75%;
- ускорение выхода продукта на рынок на 30–75%;
- увеличение вовлеченности сотрудников и удовлетворенности работой на 10–50%.

Не стоит ожидать вышеописанных результатов сразу после внедрения SAFe в процесс производства. В ходе практического участия в SAFe процессах и проведения анализа было выявлено, что положительный эффект начинает проявляться через 1–2 программных инкремента. Особенно затруднителен переход к процессам разработки при помощи SAFe после использования таких классических методологий, как каскадная и спиральная. Это обусловлено тем, что разработчикам необходимо много времени, чтобы сформировать Lean-Agile мышление, которое позволяет понимать и применять принципы и методы SAFe.

Само название Lean-Agile характеризует два основных аспекта данного мышления. Первый аспект – это бережливое мышление. Такое мышление должно полностью отражать все принципы бережливого производства. Второй аспект – готовность к изменениям. На практике же именно данный аспект вызывает больше всего проблем. При переходе на SAFe процессы разработчикам требуется отвыкать работать согласно строго установленному набору требований и описанным спецификациям. Команды должны быть готовы в любой момент времени пересмотреть свои рабочие планы и подстроиться под текущие требования, убрать те задачи, которые не требуются для достижения целей разработки, и включить в работу новые задачи.

Еще одним фактором, замедляющим успешное внедрение методологии SAFe, является необходимость в децентрализованном принятии решений. В большинстве случаев в каждой крупной компании имеется человек или группа лиц, мнение которых является определяющим при решении задач. Любое решение, которое должно быть передано на более высокий уровень, приводит к задержке. При этом может быть ухудшено качество этих решений из-за отсутствия контекста. Децентрализация решений позволяет командам самостоятельно принимать решения в тех случаях, когда появившаяся проблема не является глобальной и времязатратной. При частом возникновении такие проблемы будут решаться локально членами команды, что уменьшит время задержки при принятии решений, улучшит процесс разработки и пропускную способность команды. Одним из главных инструментов, который может помочь в принятии решений децентрализованным способом, является доверие.

Для того, чтобы ускорить процесс внедрения SAFe происходит большое количество обучающих тренингов для членов команд, на которых происходит не только обучение принципам SAFe, но и закладываются основы Agile-Lean мышления, которое со временем необходимо развивать самостоятельно.

Безусловно, рассматриваемая методология не является идеальным подвидом Agile-методологии. Помимо длительного времени ожидания положительных результатов, к недостаткам можно отнести большое количество используемых временных и финансовых ресурсов на организацию коммуникаций между командами и их членами. Однако, при определенном уровне терпения и стремления крупные предприятия могут в полной мере прочувствовать положительный эффект от использования гибких методик SAFe методологии при разработке программного обеспечения – продукт станет качественнее, а работники почувствуют уверенность в собственных силах и удовлетворенность от работы благодаря пониманию собственной значимости в разработке и увеличению прозрачности процессов, направленных на развитие продукта.

Список использованных источников:

1. The Lean Way [Электронный ресурс]. – Режим доступа: <https://theleanway.net/The-Five-Principles-of-Lean>. – Дата доступа: 20.03.2019.
2. Agile Manifesto [Электронный ресурс]. – Режим доступа: <https://agilemanifesto.org/iso/ru/principles.html>. – Дата доступа: 20.03.2019.
3. Scaled Agile Framework [Электронный ресурс]. – Режим доступа: <https://www.scaledagileframework.com/about/>. – Дата доступа: 20.03.2019.

# СТЕГАНОГРАФИЧЕСКАЯ СИСТЕМА, ОСНОВАННАЯ НА ИСПОЛЬЗОВАНИИ МЕРЫ СЛОЖНОСТИ ГРАФИЧЕСКОГО ОБРАЗА

Мельник М.А.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Ярмолик В.Н. – д.т.н., профессор

Информация – это один из самых ценных товаров в современном мире, а потому очевидно, что ее защита является весьма актуальной для настоящего времени проблемой. С развитием современных информационных технологий число компьютерных преступлений, в том числе хищений конфиденциальной информации, только растет. В таких условиях применение стеганографии является отличным способом защиты от несанкционированного доступа.

Стеганография, как и криптография, представляет собой набор методов, посвященных проблеме скрытой передачи информации. Но в отличие от криптографии, она также скрывает сам факт секретной передачи информации [1].

Общей особенностью всех стеганографических методов является то, что скрываемое сообщение внедряется в некоторый нейтральный объект, который не привлекает внимания. При применении криптографии наличие зашифрованного сообщения само по себе привлекает внимание, а вот в случае стеганографии наличие скрытой информации остается незаметным.

Основной принцип работы стеганографии приведен на рисунке 1. Сообщение – это любая информация, подлежащая скрытой передаче. Контейнер – любой непримечательный объект. Стего-ключ – криптографический ключ, который используют при встраивании сообщения в контейнер.

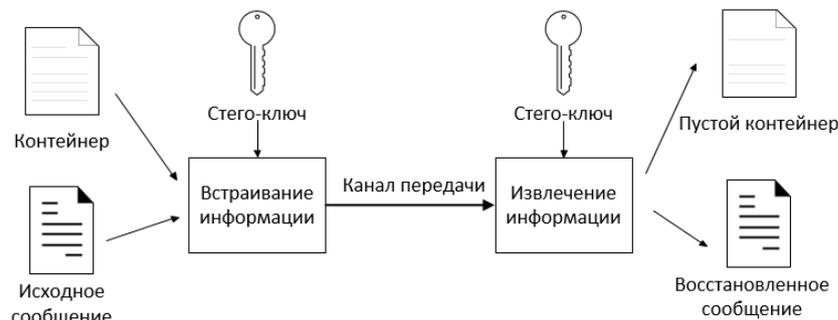


Рисунок 1 – Основной принцип сокрытия данных в стеганографии

Среди всего многообразия стеганографических методов особый интерес представляют алгоритмы встраивания информации в графические контейнеры с целью их последующей передачи. Такой интерес можно объяснить тем, что внесение искажений, которые трудно различимы для среднестатистического человека, не приводит к существенным изменениям этих контейнеров [2].

Среди методов сокрытия информации в графических изображениях наиболее простым и доступным для понимания является **метод LSB**. Суть данного метода заключается в изменении наименьшего из значащих бит в каждом блоке изображения, что не влияет на картину в целом. У данного метода есть один существенный недостаток: он довольно прост в реализации и также прост в обнаружении.

Большим отличием нового **алгоритма BPCS** от уже устоявшегося LSB являлось то, что пока LSB использовал младшие биты картинок и мог поместить данных на 10% от исходного размера контейнера, BPCS пытался вставить данные и в более старшие биты картинки там, где это по-прежнему не будет ухудшать сам контейнер. А объем данных, который можно было внедрять в картинку с использованием BPCS, вырос до 44%. Для определения зашумленных областей данный алгоритм использует *меры сложности*  $\alpha$  и  $\beta$ . Но представленные меры не всегда позволяют точно определить зашумленную область, а потому часто при применении данного метода вставка информации может сильно исказить контейнер.

Развитием BPCS стеганографии стал **метод ABCDE**, который использует сразу три меры сложности графического образа для определения шумоподобных блоков. Он отличается большей сложностью, а также и большим количеством внедренной информации: объем данных достигает 50% от объема исходного графического образа.

#### Список использованных источников:

1. Ярмолик, В.Н. Криптография, стеганография и охрана авторского права/ В.Н. Ярмолик, С.В. Ярмолик, С.С. Портянко – 2007. – № 10. – С. 149–167.
2. Википедия [Электронный ресурс]. - Электронные данные. - Режим доступа: <https://en.wikipedia.org/wiki/Steganography>.

## СХЕМА ГЕНЕРАЦИИ И ВОССТАНОВЛЕНИЯ ДАННЫХ ПРИ ЛОКАЛИЗАЦИИ ДЕФЕКТА

Моженкова Е.В., Пармонов А.И.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Пармонов А.И. – к.т.н., доцент

Для локализации дефектов работы корпоративных информационных систем на стороне разработчика предлагается реализовать механизм импорта/экспорта данных клиента, разработав автоматизированную информационную систему «Агрегатор данных». Механизм передачи данных на сторону разработчика будет осуществляться через конфигурационный файл. Предложена структура такого конфигурационного файла для генерации и восстановления данных при локализации дефектов на стороне клиента корпоративной сети.

Для обеспечения взаимодействия банка данных (БД) клиента и разработчика в системе обработки принято использовать конфигурационный файл. Схема выгружаемых данных должна представлять собой перечень полей выгружаемой структуры и их типы. Поля выгрузки могут быть сложно структурированными. В схеме выгружаемых данных могут появляться зависимости, которые усложняют логический разбор данных бизнес-объектов системы.

Состояние БД сохраняется в формате метаданных, поэтому структура конфигурационного файла оформляется в формате XML-документа.

Общее описание процесса обмена данными можно описать следующим образом:

- 1) под обработкой данных БД понимается экспорт/импорт объектов, где каждый объект относится к заранее определенному типу данных и характеризуется набором свойств. В общем случае значения свойств конкретного объекта могут храниться в нескольких таблицах;
- 2) при экспорте данные из базы данных (БД) выгружаются в XML-документ определенной структуры. В одном XML-документе могут содержаться объекты разных типов, при этом возможно существование взаимосвязей между объектами;
- 3) Описание типов данных будет храниться в специальной таблице, чтобы можно было настраивать для каждого типа данных, значения каких свойств объектов экспортировать или импортировать для каждой операции экспорта/импорта создается соответствующий протокол, содержащий краткую информацию об экспортированных/импортированных объектах;
- 4) при выполнении экспорта обеспечивается возможность подхвата для экспортируемых объектов для каждого типа данных могут экспортироваться как все объекты из БД, так и только определены при выполнении импорта данных обеспечивается просмотр содержимого XML-файла, фильтрация импортируемых объектов, возможность замещения существующих в БД объектов, либо создание новых объектов, а также сохранение ссылочной целостности между объектами;
- 5) при необходимости экспортированные/импортированные объекты можно удалить через протоколы экспорта/импорта данных;
- 6) файл, содержащий экспортированные данные, является правильным XML-документом.

Для хранения объектов БД в XML-документе будут использоваться следующие XML-элементы:

- <Data\_Root> – корневой узел XML-документа;
- <Descript> – описание содержимого XML-файла. В свою очередь имеет атрибуты: ExpSet\_Code (код настройки экспорта) и ExpSet\_Name (наименование настройки экспорта, по которой производился экспорт данных). Содержит описание данных, хранящихся в XML-документе;
- <MetaData> – метаописание данных. Используется для хранения описания данных. Содержит элементы Class;
- <Time> – родительский узел для хранения временных интервалов;
- <Filtr> – родительский узел для хранения данных фильтров менеджера сущностей;
- <Data> – родительский узел для хранения объектов. Используется для хранения данных. Без атрибутов (Sobj). Содержит элементы Collection;
- <Class> – класс объектов. Содержит описание класса объектов. В свою очередь имеет атрибуты: name (имя класса, должно быть уникальным для всех элементов Class), caption (наименование/заголовок класса, должно быть уникальным для всех элементов Class) и manager (признак менеджера сущностей). Включает в себя элементы Collection;
- <Collection> – коллекция однородных элементов. Используется для группировки однородных элементов. Имеет такие атрибуты: name (имя коллекции), caption (наименование/заголовок коллекции) и child\_class\_id (тип элементов, которые

могут входить в состав коллекции, должно быть одно из значений: "property", "prop\_value" или наименование класса объектов);

- <Object> – объект БД. Используется для хранения объекта экспорта/импорта. Имеет атрибуты: name (наименование объекта, используется при просмотре содержимого XML-документа), class\_id (имя класса, к которому объект относится) и id (уникальный в пределах класса, к которому относится объект, идентификатор объекта, используется для построения взаимосвязей между объектами). Содержит элементы Collection;

- <property> – свойство. Используется при описании класса объектов. Имеет атрибуты: name (имя свойства), caption (наименование/заголовок свойства), datatype (тип данных) и rit\_classes (список связанных классов объектов);

- <prop\_value> – значение свойства. Используется для хранения значения свойства объекта. Имеет атрибуты: prop\_name (имя свойства), rit\_class (имя класса объекта, на который ссылается свойство).

Структура конфигурационного файла проектировалась с учетом формальной модели информационных потоков и описания модели этих потоков в корпоративной сети [1, 2]. Фрагмент структуры файла представлен на рисунке 1.

```
<?xml version="1.0" encoding="windows-1251" ?>
<Data_Root>
  <Descript ExpSet_Code="1" ExpSet_Name="Настройка для экспорта алгоритмов">
    <![CDATA[описание XML-документа]]>
    </Descript>
    <MetaData>
      <Class name="AlgView" caption="Алгоритм представления">
        <Collection name="Properties" caption="Свойства" child_class_id="property">
          <property name="Name" caption="Наименование" datatype="0" rit_classes="" />
          <property .../>...<property .../>
        </Collection>
        ...
        <Collection .../>...</Collection>
      </Class>
      ...
      <Class .../>...</Class>
    </MetaData>
    <Time>...</Time>
  </Descript>
  <Filter>...</Filter>
  <Data>
    <Collection name="Data.AlgView" caption="Алгоритмы представления">
      <Object name="Период (1)" class_id="AlgView" id="2">
        <Collection name="Prop_Values" caption="Свойства" child_class_id="prop_value">
          <prop_value prop_name="name">
            <![CDATA[Период]]>
          </prop_value>
          ...
          <prop_value>...</prop_value>
        </Collection>...</Collection ...>
        ...
      </Object>
      ...
      <Object .../>...</Object>
    </Collection>
    ...
    <Collection .../>...</Collection>
  </Data>
</Data_Root>
```

Рисунок 1 – Структура конфигурационного файла

Состав элементов конфигурационного файла спроектирован на основе абстракции предметной области и имеет обобщенную структуру. В ходе интеграции с другими информационными системами в корпоративной сети и в результате опытной эксплуатации АИС «Агрегатор данных» возможно изменение конфигурации - введение дополнительных тегов и их атрибутов.

**Список использованных источников:**

1. Моженкова, Е. В. Модель построения общих требований по обработке банков данных в задачах сопровождения систем / Е. В. Моженкова, А.И. Парамонов // Компьютерные системы и сети: материалы 54-й науч. конф. аспирантов, магистрантов и студентов, Минск, 23 – 27 апреля 2018 г. / Белорус. гос. ун-т информатики и радиоэлектроники. – Минск, 2018. – С. 92 – 93.

2. Моженкова, Е. В. Модель представления информации в корпоративных информационных системах / Е. В. Моженкова, А. И. Парамонов // Информационные технологии и системы 2018 (ИТС 2018) = Information Technologies and Systems 2018 (ITS 2018) : материалы межд. науч. конф., Минск, 25 октября 2018 г. / Белорус. гос. ун-т информатики и радиоэлектроники ; редкол. : Л. Ю. Шилин [и др.]. – Минск, 2018. – С. 228 - 229.

# ПОСТРОЕНИЕ ИСКУССТВЕННОЙ НЕЙРОННОЙ СЕТИ ДЛЯ РАСПОЗНАВАНИЯ ГРАФИЧЕСКИХ ФИГУР ТЕХНИЧЕСКОГО АНАЛИЗА

Моисеев И.В.

Белорусский государственный университет информатики и радиоэлектроники  
Г. Минск, Республика Беларусь

Серебряная Л.В. – к.т.н., доцент

В данной работе рассмотрена задача построения искусственной нейронной сети типа многослойный перцептрон для распознавания графических фигур технического анализа; предложены способы повышения точности распознавания: применение функции SoftMax (вместо логистической функции Sigmoid) в качестве функции активации нейронов выходного слоя и аугментация обучающей выборки.

Задача распознавания образов играет важнейшую роль в области машинного зрения, прогнозирования, управления и принятия решений в различных сферах человеческой деятельности. Одним из направлений применения алгоритмов распознавания образов является анализ графических фигур технического анализа (рис. 1), цель которого - прогнозирование дальнейшего развития временного ряда. Фигура технического анализа - визуально различимая формация на ценовом графике, которая периодически повторяется вследствие шаблонного поведения участников рынка в сходных условиях [1].

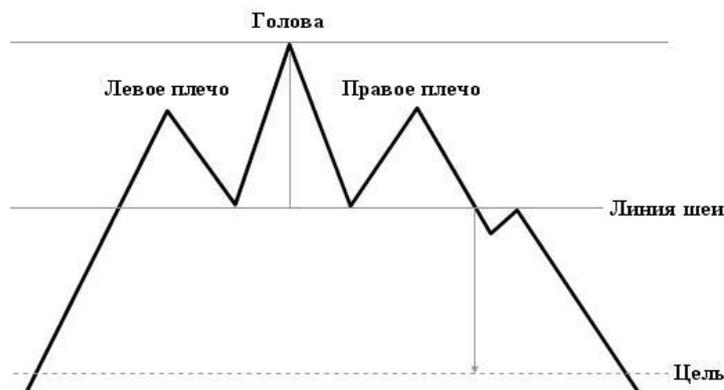


Рисунок 1 - Пример фигуры технического анализа: Голова и Плечи

Успешно зарекомендовавшим себя инструментом распознавания образов является искусственная нейронная сеть (ИНС) [2].

В рамках данной работы была выбрана сеть типа многослойный перцептрон, а также предложены способы повышения точности распознавания сетью графических фигур технического анализа. Сеть типа многослойный перцептрон является одной из наиболее подходящих архитектур ИНС для решения задачи распознавания поведенческих шаблонов на графиках временных рядов [3]. Преимуществом ИНС такого типа является возможность формировать на выходе произвольную многомерную функцию при соответствующем выборе количества слоев, диапазона изменения сигналов и параметров нейронов [2].

Архитектура данной сети представлена на рисунке ниже:

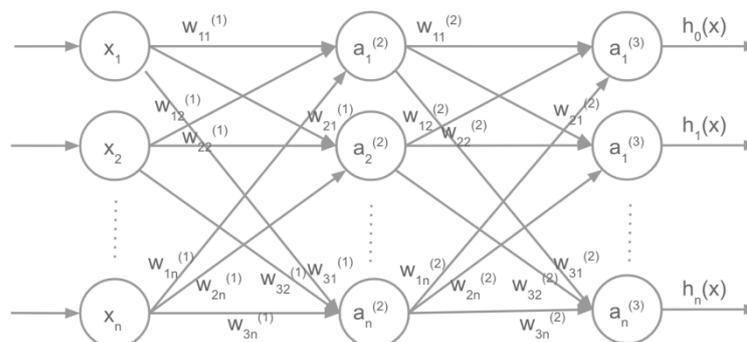


Рисунок 2 - Архитектура ИНС

Входной слой данной сети содержит 400 нейронов (распознавание изображения размером 20 на 20 пикселей требует обработки 400 признаков входного образа).

Скрытый слой содержит 25 нейронов.

Выходной слой содержит 11 нейронов (число классов, 11 различных фигур технического анализа).

В качестве активационной функции нейронов входного и скрытого слоя используется логистическая функция [4]:

$$F(x) = \frac{1}{1 + e^{-x}} \quad (1)$$

В качестве активационной функции нейронов выходного слоя была выбрана т.н. SoftMax функция [5]:

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (2)$$

Функция SoftMax является обобщением логистической функции [6]. Ее идея состоит в том, чтобы несколько преувеличить разницу между полученными значениями: softmax будет выдавать значения, очень близкие к нулю для всех  $x_j$ , существенно меньше максимального.

В качестве функции потерь используется перекрестная энтропия (cross-entropy) [6]:

$$\xi = - \sum_{i=1}^c t_i \log y_i \quad (3)$$

Для дальнейшего улучшения точности распознавания была проведена аугментация обучающей выборки. Методом искусственной генерации был создан дополнительный набор фигур технического анализа, каждому элементу которого была присвоена метка, означающая принадлежность к одному из рассматриваемых классов фигур технического анализа.

Результаты измерений точности распознавания ИНС представлены в таблице 1.

Таблица 1 - Результаты измерений точности распознавания ИНС

	Функция активации нейронов выходного слоя	
	Sigmoid	SoftMax
Без аугментации	74.54%	76.36%
С аугментацией	81.81%	85.45%

Благодаря изложенным выше операциям (использованию функции SoftMax в качестве активационной функции нейронов выходного слоя и аугментации обучающей выборки) удалось повысить точность распознавания фигур технического анализа на 11%.

**Список использованных источников:**

1. Графические фигуры [Электронный ресурс] / Forex Markets. – Режим доступа : <http://forex-markets.ru/pattern.html>. – Дата доступа : 20.04.2019.
2. Серебряная, Л. В. Методы и алгоритмы принятия решений : учебно-методическое пособие / Л. В. Серебряная, Ф. И. Третьяков. – Минск : БГУИР, 2016. – 64 с.
3. Найман, Э. Малая энциклопедия трейдера / Э. Найман. – М.: Альпина Паблишер, 2003. – 378 с.
4. Сигмоида [Электронный ресурс] / Wikipedia. – Режим доступа : <https://ru.wikipedia.org/wiki/Сигмоида>. – Дата доступа : 21.04.2019.
5. SoftMax [Электронный ресурс] / Wikipedia. – Режим доступа : <https://ru.wikipedia.org/wiki/Softmax>. – Дата доступа : 21.04.2019.
6. Николенко С., Кадури А., Архангельская Е. Глубокое обучение / С. Николенко, А. Кадури, Е. Архангельская. – СПб.: Питер, 2018. – 480с.

## ПОДХОДЫ К РЕАЛИЗАЦИИ КОНТРОЛЯ ДОСТУПА К ФАЙЛОВОЙ СИСТЕМЕ В ОС СЕМЕЙСТВА WINDOWS NT

Немкович А.В.

Белорусский государственный университет информатики и радиоэлектроники  
г.Минск, Республика Беларусь

Ярмолик В.Н. – д.т.н., профессор

Целью данной работы является анализ возможных подходов к реализации контроля доступа к файловой системе в операционных системах семейства Windows NT.

На данный момент одной из наиболее важных проблем в области информационных технологий является проблема защиты и контроля доступа к конфиденциальной информации, к различного вида цифровым данным, хранящимся на компьютерах в виде файлов. В тоже время самыми популярными операционными системами для компьютера среди пользователей являются операционные системы семейства Windows NT, что обуславливает важность изучения подходов к реализации контроля доступа к файловой системе в операционных системах семейства Windows NT.

Прежде чем рассматривать подходы к защите файловой системы с учётом особенностей и возможностей операционных систем Windows NT, необходимо ознакомиться с общепринятыми способами защиты секретной информации, среди которых можно выделить следующие методы:

- 1) стеганография;
- 2) криптография.

В случае стеганографии скрывается сам факт сокрытия секретной информации, то есть секретная информация передаётся под видом общедоступной. Наиболее распространенный стеганографический материал — это графические, звуковые и другие мультимедийные данные, формат которых позволяет замещать наименее значимую часть исходной информации на любую произвольную (собственно, на ту самую секретную информацию, которую требуется скрыть) [1].

Метод криптографии основан на прямо противоположном принципе: ни от кого не скрывается факт передачи или хранения секретной информации и не предпринимаются никакие действия для предотвращения доступа к ней посторонних лиц [1]. В большинстве случаев секретные данные шифруются с помощью определенного алгоритма, и, чтобы прочитать их, сначала потребуется их расшифровать, для чего также, скорее всего, необходимо будет знать ключ/ключи, которые использовались для шифрования данных [2].

В данной работе для защиты и контроля доступа к файлам конкретно в операционных системах семейства Windows NT предлагается метод «прозрачного доступа к файлам».

Суть метода «прозрачного доступа к файлу» заключается в том, что файл динамически шифруется при перезаписи/записи его на носитель информации и динамически расшифровывается при его чтении. В операционных системах семейства Windows NT для обеспечения возможности динамического шифрования файла при его записи на диск необходимо разработать свой фильтр-драйвер файловой системы. Фильтр-драйвер – это драйвер, который перехватывает запросы на ввод-вывод (IRPs – Input/Output Request Packets [3]), предназначенные для некоторых существующих программных модулей (например, файловая система или драйвер диска). В зависимости от места вставки в иерархии драйверов фильтр-драйвер будет получать определенные запросы от диспетчера ввода-вывода (I/O Manager [4]) и иметь определенные возможности расширить имеющуюся функциональность, поэтому фильтр-драйвер должен быть вставлен в иерархию драйверов над драйвером каждой файловой системы, чтобы шифровать данные перед тем как послать их файловой системе и дешифровать данные перед тем, как их посылать обратно пользователю. Таким образом, с учётом возможностей, предоставляемых фильтр-драйвером, становится возможным выполнение операций аутентификации и авторизации пользователя системы, а также шифрования данных файлов «на лету», то есть в момент обращения пользователя к файловой системе. При этом сам факт того, что данные файла зашифрованы также может скрываться, поскольку, если пользователь имеет права на доступ к файлу, то файл будет автоматически расшифровываться фильтр-драйвером, которому будет известен ключ шифрования, и пользователь сможет с ним работать так, словно он и не был зашифрован, в ином случае пользователю будет отказано в доступе к файлу.

### Список использованных источников:

1. Второе дыхание криптографии [Электронный ресурс] – Режим доступа: <https://compress.ru/article.aspx?id=10116> – Дата доступа: 17.03.2019.
2. С.Г. криптографии / С.Г. В.В. , П.Е. Серов. – М: линия – , – 52 с.
3. Rajeev, N. Windows NT File System Internals: A Developer's Guide / N. Rajeev. – O'Reilly. – 1997. – 774 S.
4. Schrieber, S.B. 2000 : a programmer's cookbook / S.B. Schrieber. – . – 2001, – 592 S.

## АЛГОРИТМЫ ПОСТРОЕНИЯ ПЕРСОНАЛИЗИРОВАННЫХ МУЗЫКАЛЬНЫХ РЕКОМЕНДАТЕЛЬНЫХ СИСТЕМ

Орлова А.С.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Петюкевич Н.С. – м.т.н., ассистент

На сегодняшний день существует большое количество сервисов для прослушивания музыки. Отличительной чертой подобных сервисов является наличие персонализированного подхода к каждому пользователю. Задача персонализированного подхода заключается в том, чтобы, основываясь на предпочтениях пользователя, предложить музыкальную композицию, которая будет ему интересна. Решением данной задачи занимаются специальные программы, называемые рекомендательными системами.

Рекомендательные системы – это программные средства, которые способны предсказать, какие объекты будут интересны пользователю при наличии определённой информации о его предпочтениях. Под объектом понимается то, что именно система рекомендует пользователю [1]. В данном случае в качестве объекта выступает музыкальная композиция.

Существует две основные стратегии построения рекомендательных систем - фильтрация на основе содержания и коллаборативная фильтрация. При фильтрации на основе содержания создаются профили пользователей и объектов. Профили пользователей могут включать различную информацию о самом пользователе, а также его ответы на заранее составленные вопросы, профили объектов могут включать названия жанров, имена исполнителей, оценки пользователей. Метод коллаборативной фильтрации не требует определения множества характеристик, а основывается только на информации о действиях и поведении пользователей в прошлом. Далее будут рассматриваться алгоритмы, применяемые для построения рекомендательных систем методом коллаборативной фильтрации.

В некоторых коллаборативных рекомендательных системах используется алгоритм Байеса. В его основе лежит теорема Байеса, которая заключается в вычислении вероятности наступления события (выставления пользователем той или иной оценки) в условиях, когда на основе наблюдений известна лишь некоторая информация о событиях. Основным недостатком алгоритма Байеса является требование большой выборки для обучения, а также тот факт, что значения спрогнозированных вероятностей не всегда являются достаточно точными. Поэтому в последнее время используются другие алгоритмы.

В центре большинства рекомендательных систем находится так называемая матрица предпочтений. Это матрица, строки которой соответствуют пользователям, а столбцы композициям (рисунок 1). На пересечении некоторых строк с некоторыми столбцами матрица заполнена оценками. Оценка – это показатель того, насколько пользователю понравилась прослушанная композиция по заданной шкале (например, от 1 до 5). Незаполненные поля соответствуют непрослушанным композициям. Основное допущение данного метода состоит в том, что те, кто одинаково оценивали какие-либо объекты в прошлом, склонны давать похожие оценки другим объектам и в будущем.

	Track 1	Track 2	Track 3	Track 4	Track 5
User 1	5	1	3	5	2
User 2	?	?	?	2	5
User 3	4	?	3	?	1
User 4	1	5	5	3	3

Рисунок 1 – Матрица оценок пользователей

Задача заключается в том, чтобы предположить, как оценил бы пользователь ту или иную композицию. Например, какую оценку поставил бы третий пользователь песне номер 2 или 4.

Самым популярным алгоритмом для нахождения данной оценки является алгоритм  $k$ -ближайших соседей [2]. Алгоритм включает следующие шаги:

- вычисление степени схожести пользователей;
- нахождение  $k$  самых похожих пользователей;
- рекомендация пользователю тех композиций, которые он еще не прослушивал.

Оценки каждого пользователя можно представить в виде вектора. Чем ближе два вектора находятся друг к другу, тем более похожими являются соответствующие оценки пользователей, а

значит, и более похожими являются их интересы. Расстояние между векторами в свою очередь можно найти из векторного произведения. Запишем векторное произведение двух векторов:

$$A \cdot B = \|A\| \|B\| \cos \theta,$$

где  $\theta$  - угол между векторами.

Чем меньше значение угла  $\theta$ , тем ближе друг к другу находятся векторы. Степень схожести пользователей может быть выражена как значение косинуса угла между векторами (чем ближе значение косинуса к 1, тем более похожие интересы у пользователей):

$$\cos \theta = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

После того, как найдены пользователи со схожими интересами, целевому пользователю могут быть порекомендованы композиции.

Зачастую такие матрицы разрастаются в объемах, становятся слишком большими и их неудобно, сложно и долго обрабатывать. Существует ещё один алгоритм, который используется для создания рекомендательных систем - факторизация матриц. С помощью этого метода изначальную матрицу предпочтений можно представить в виде произведения двух прямоугольных матриц меньшей размерности (рисунок 2). Одна матрица может быть представлена как матрица пользователей, где строки - это пользователи, а столбцы - это скрытые признаки. Вторая матрица - матрица песен, где строки - это скрытые признаки, а столбцы представляют собой сами песни.

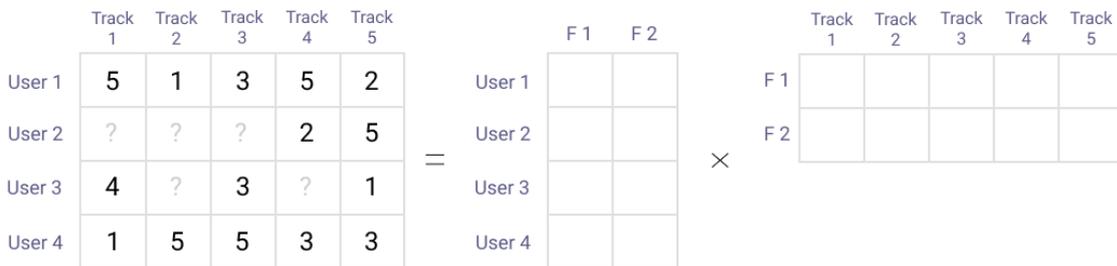


Рисунок 2 – Разложение матрицы оценок пользователей

Чем больше количество признаков, тем точнее будет приближение оценки, но в то же время алгоритм будет сложнее, работать будет дольше и размер матриц будет больше. Количество признаков следует выбирать разумно, проанализировав конкретную задачу.

Одним из методов разложения матриц является мультипликативное правило обновления [3].

где  $A$  – исходная матрица оценок размерностью  $n \cdot m$ ;

$$H_{ij} = H_{ij} \frac{(W^T A)_{ij}}{(W^T W H)_{ij} + \varepsilon}, \quad W_{ij} = W_{ij} \frac{(A H^T)_{ij}}{(W H H^T)_{ij} + \varepsilon},$$

$H$  – матрица песен, заполненная начальными случайными значениями, размерностью  $k \cdot m$ ;

$W$  – матрица пользователей, заполненная начальными случайными значениями, размерностью  $n \cdot k$ .

Данный метод является итеративным, в конце каждой итерации вычисляется ошибка. Алгоритм останавливается, когда значение ошибки приближается к 0:

$$\frac{\|A - WH\|}{\|A\|} \approx 0.$$

Данный метод решает проблему обработки большого количества данных, а также даёт хорошее приближение оценок пользователей.

**Список использованных источников:**

1. Francesco Ricci, Lior Rokach, Bracha Shapira, Paul B. Kantor. Recommender Systems Handbook. 2011.
2. Lars Elden. 2007. Matrix Methods in Data Mining and Pattern Recognition (Fundamentals of Algorithms). Soc. for Industrial and Applied Math., Philadelphia, PA, USA.
3. Michael W. Berry, Murray Browne, Amy N. Langville, V. Paul Pauca, Robert J. Plemmons, Algorithms and applications for approximate nonnegative matrix factorization, Computational Statistics & Data Analysis, Volume 52, Issue 1, 2007, p. 155-173.

# КРИПТОАНАЛИЗ ОПТИМИЗАЦИЯ ТЕСТА КАСИСКИ ПОД ПРОГРЕССИВНЫЙ КЛЮЧ

Панкратьев А.С., Болтак С.В.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Болтак С.В. – ассистент

Шифр - один из способов сокрытия информации. Криптоанализ - наука о методах дешифровки информации в случае, когда нет предназначенного для дешифрования ключа. В данной работе разработан алгоритм взлома шифра Виженера с прогрессивным способом генерации ключевой последовательности.

Один из простейших подстановочных шифров в 17 веке предложил Блез Виженер. В качестве ключа шифрования выступает ключевая фраза. Ключевая фраза повторяется до тех пор, пока не достигнет длины шифруемого текста. Каждая буква секретного сообщения получается сдвигом каждой буквы исходного текста на определённое число, задаваемое буквой ключевой фразы (буква А не даёт сдвига, буква Б — сдвиг на одну позицию, В — на две и т.д.). Например, попробуем зашифровать слово «КУПЛЕТ», пользуясь ключевой фразой «АБВ». После повторения ключевой фразы получается следующая ключевая последовательность: «АБВАБВ». Шифрование происходит по следующему алгоритму: Символ «К» не подвергается сдвигу, символ «У» сдвигается на одну позицию, превращаясь в «Ф», символ «П» сдвигается на две позиции, превращаясь в «С». После дальнейшего шифрования, мы в итоге получаем «КФСЛЁФ» [1].

На протяжении нескольких столетий такой способ шифрования считали практически не взламываемым. Однако в 19 веке были предприняты первые попытки взлома данного шифра. Все они были основаны на нахождении длины ключевой фразы, ведь если известна её длина, то зашифрованный текст можно разбить на фрагменты, каждый из которых кодируется обычным шифром Цезаря [1]. В примере выше К,Л – кодируются нулевым сдвигом. У,Е – сдвигом на символ, П,Т – сдвигом на два символа. Если текст достаточно длинный, можно применить частотный анализ и получить исходный текст. Из этого следует, что взлом данного шифра сводится к поиску длины ключевой фразы. Одним из наиболее популярных алгоритмов для этой цели служит тест Касиски [2]. Он применяется для нахождения длины ключа в полиалфавитных шифрах, таких как шифр Виженера. Основная идея алгоритма основана на том, что одинаковые участки открытого текста при шифровании дают одинаковые отрезки криптограмм. Таким образом, найдя расстояние между такими отрезками, мы сможем сделать вывод о длине ключа.

Если для прямого ключа тест работает отлично, то для прогрессивного - классический тест Касиски становится бесполезным. Сама идея использования прогрессивного ключа заключается в циклическом сдвиге символов ключа на одну позицию в упорядоченном алфавите символов при повторном применении ключа. Тогда для ключа «АГЕ» при повторном его использовании по прогрессивной схеме имеем «БДЁ», а при третьем «ВЕЖ», и так далее.

При разработке программного средства для криптоанализа шифра с прогрессивной ключевой последовательностью были рассмотрены различные алгоритмы. Самым очевидным решением является следующее изменение классического теста: при поиске одинаковых  $L$ -грамм выбирать  $L$ -граммы, лежащие на расстояниях кратных длине алфавита, так как в этом случае ключ вернется в свое исходное состояние. Недостаток - для текстов на русском языке длина шифротекста должна быть порядка 2000 символов.

В данной работе предлагается следующее решение: для поиска одинаковых  $L$ -грамм необходимо на каждой итерации сдвигать символы на одну позицию, учитывая номер операции для определения расстояния. Если искомая  $L$ -грамма начинается с индекса  $i$  и она повторно найдена в шифротексте, начиная с индекса  $j$  после  $k$  сдвигов символов искомой  $L$ -граммы, то расстояние между  $L$ -граммами равно остатку от целочисленного деления  $(i - j - k)$  на количество букв в алфавите.

Таким образом, алгоритм теста Касиски для прогрессивного ключа будет иметь следующий вид:

1) поиск в тексте повторяющихся  $L$ -грамм и нахождение расстояний между ними (схема алгоритма приведена на рисунке 1):

- найти расстояния на  $i$ -ой операции и занести в список итерационных расстояний;
- в конце итерации найти наиболее часто встречающееся повторение и занести в список;
- очистить список итерационных расстояний;

2) среди найденных расстояний поиск наименьшего делителя, данное значение – искомая длина ключевой фразы.

Получив длину ключа, криптоаналитик с помощью частотного анализа может легко получить искомый текст.

Полученный алгоритм отлично справляется с шифротекстами от 300 слов и более с повторяющимися  $L$ -граммами в исходном тексте и ключом, длиной не больше длины алфавита. Для более маленьких текстов алгоритм так же работает, хоть и с меньшей точностью нахождения ключа.

**Схема на рисунке 1 поясняет алгоритм, описанный выше**

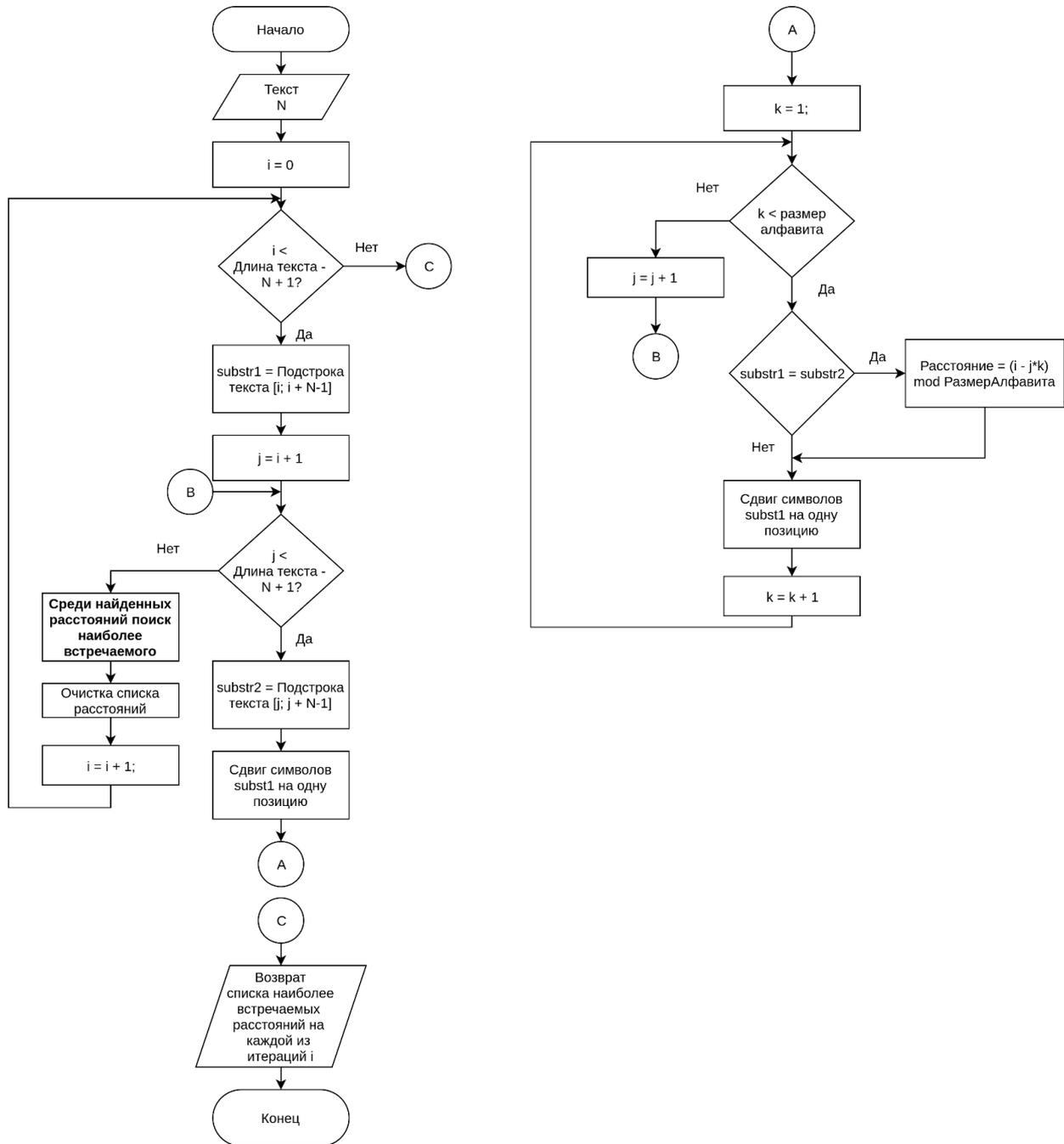


Рисунок 1 – Алгоритм поиска расстояний между  $L$ -граммами

Тест Касиски позволяет только получить длину ключа, однако прекрасно сочетается с различными методами криптоанализа, предполагающими наличие длины ключевого слова.

Разработанное программное средство позволяет осуществить криптоанализ шифротекста, полученного путём шифрования методом Виженера с прогрессивной ключевой последовательностью.

**Список использованных источников:**

1. С.Сингх. Тайная история шифров и их расшифровки. — Астрель, 2006. — 447 с.
2. Ярмолик, В. Н. - Элементы теории информации : практикум для студ. спец. «Программное обеспечение информационных технологий» дневн. и дист. форм обуч. / В. Н. Ярмолик, А. П. Занкович, С. С. Портянко. – Минск : БГУИР, 2007. – 39 с.

## ПРЕДСТАВЛЕНИЕ МНОГОМЕРНОЙ ИНФОРМАЦИИ НА КАРТЕ

Панкратьев А.С., Голубев К.А., Пилинко Н.А.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Лапицкая Н.В. – к.т.н., доцент

Каждый день человек потребляет огромное количество информации: от этикеток на продуктах до отчётов Всемирной организации здравоохранения. В результате, предоставить информацию так, чтобы её можно было легко воспринимать, становится все сложнее и сложнее. Один из лучших способов помочь воспринять информацию — визуализация этих данных.

Визуализация — важнейший этап в процессе анализа данных. Он помогает представить результаты исследований в простой и понятной форме, зачастую служит ключевым фактором для принятия решений в различных сферах деятельности. Однако часто эти данные являются многомерными, что сильно усложняет их визуализацию. Одним из решений этой проблемы может стать представление данных в графическом виде на пространственно-временной поверхности.

Суть метода заключается в присваивании каждой «ситуации» своего цвета, после чего для каждого региона каждый цвет освещается в зависимости от интенсивности, цвета смешаются и полученный цвет наносится на карту [1].

Причём интенсивность может рассчитываться на основании различных метрик. Например: наибольшее значение по сравнению с другими регионами, наибольшее значение на душу населения по сравнению с другими регионами и т. д.

Для реализации сервиса для визуализации данных необходимо иметь следующие входные данные:

- база данных, где записи содержат регионы и номер «ситуации»;
- внешний API для получения полигонов регионов;
- внешний API для работы с картами;
- внешний API для получения численности населения в регионе.

Диаграмма последовательности выполнения визуализации на сервере приведена на рисунке 1.

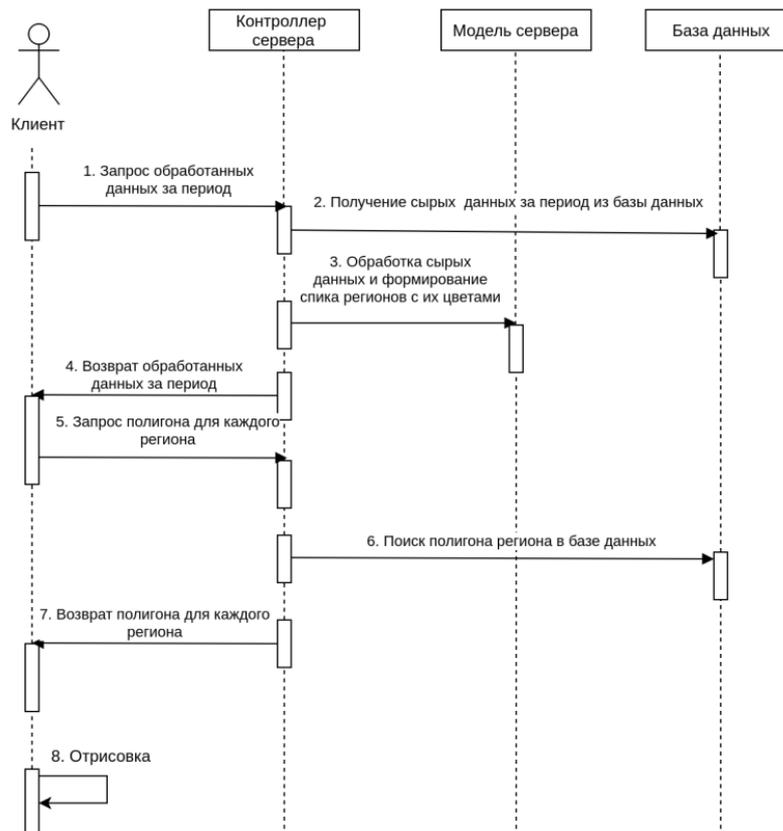


Рисунок 1 – Диаграмма последовательности выполнения визуализации на сервере

Пример сервиса, реализующего представленный выше функционал, изображён на рисунке 2.

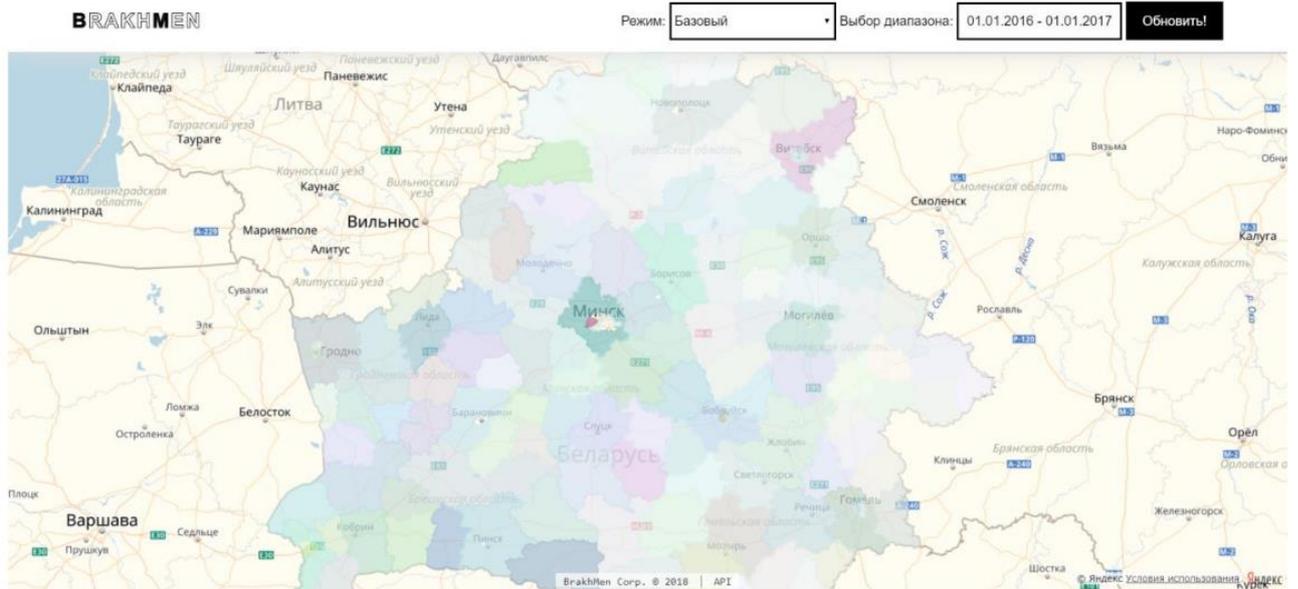


Рисунок 2 – Пример сервиса, реализующего визуализацию многомерной информации

Причём так как от интенсивности смешанных цветов зависит интенсивность результирующего цвета, на конечной карте наиболее «активные» регионы отображаются более яркими цветами, «пассивные» - практически белым цветом.

При нажатии на конкретный регион можно получить подробную детализированную статистику по нему, в которой отображается количество повторений каждой конкретной причины этого региона. Пример приведён на рисунке 3.

## Минский район, Беларусь

- Поджог - 17 раз
- Нарушения технологического регламента (процесса) - 0 раз
- Взрыв - 0 раз
- Нарушение противопожарных требований при проведении огневых работ - 3 раз
- Механическое разрушение узлов и деталей и т.д. - 1 раз
- Нарушение правил хранения, использования, изготовления (приготовление) и транспортировки - 0 раз
- Нарушение правил эксплуатации газовых устройств и агрегатов. - 0 раз
- Нарушение правил монтажа и устройства электросетей и электрооборудования. - 2 раз
- Нарушение правил эксплуатации электросетей и электрооборудования. - 33 раз
- Конструктивный недостаток электрооборудования. - 8 раз
- Нарушение правил устройства и монтажа печей, теплогенерирующих агрегатов и устройств - 8 раз
- Нарушение правил эксплуатации печей, теплогенерирующих агрегатов и устройств - 44 раз
- Конструктивный недостаток теплогенерирующих агрегатов и устройств - 1 раз
- Конструктивный недостаток изделия, устройства - 10 раз
- Неосторожное обращение с огнем - 47 раз
- Шалость детей с огнем - 2 раз
- Проявление сил природы - 4 раз
- Прочие причины - 5 раз
- Неустановленные причины - 0 раз

Закреть

Рисунок 3 – Пример статистики по региону

Разработанное программное средство можно применять для обработки статистики людьми а так же для презентаций какой-либо многомерной информации.

### Список использованных источников:

1. Панкратьев, А. С. Визуализация больших данных (BigData) / А. С. Панкратьев, Н. А. Пилинко, К. А. Голубев // Компьютерные системы и сети: материалы 54-й научной конференции аспирантов, магистрантов и студентов, Минск, 23 – 27 апреля 2018 г. / Белорусский государственный университет информатики и радиоэлектроники. – Минск, 2018. – С. 98 – 99.
2. Noab Illinsky. Designing Data Visualizations. - O'Reilly Media, 2011
3. Phil Simon. The Visual Organization: Data Visualization, Big Data, and the Quest for Better Decisions - Phil Simon, 2014

## ОПТИМИЗАЦИЯ МЕТОДА УДАЛЕНИЯ ШУМА В ЦИФРОВЫХ ЗАПИСЯХ ЗВУКА

Патеенок А.В.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Серебряная Л.В. – к.т.н., доцент

Поскольку различные шумы серьезно влияют на качество жизни, подавление и удаление шума является важной задачей. Существует множество эффективных методов уменьшения определенных типов шумов. Тем не менее, шум, присутствующий в жизни, в основном представляет собой смесь различных видов шума. В данной работе модифицируется метод наименьших квадратов для случая смешивания непрерывного и импульсного шумов.

В процессе записи, хранения и передачи звука часто возникают помехи. Хотя современные методы позволяют повысить надёжность этих процессов, из-за факта существования записей на некачественное звуковое оборудование сохраняется актуальность задачи удаления шумов. Кроме того, удаление помех является важной частью процесса оцифровки старых записей.

Шумы в основном можно классифицировать по продолжительности и частоте [1]. При классификации по продолжительности шумы могут быть:

- непрерывные, такие как звук офиса, промышленного оборудования, шум ветра, пробок, шипение старой записи или плохая телефонная линия;
- прерывистые, повторяющиеся, обычно тональные шумы, такие как гудки, гудки или звонки;
- импульсные, резкие, обычно негармоничные, а иногда и громкие звуки, такие как щелчки, удары по ступенькам, выстрелы, удары и удары.

Для различных типов шумов требуются соответствующие способы уменьшения шума. Однако шумы в повседневной жизни смешаны с различными шумами других характеристик, из-за чего необходимо комбинировать различные методы снижения шума. Например, метод оценки энергетического спектра используется для обнаружения импульсного шума и устранения его, что можно использовать в качестве предварительной стадии обработки сигнала. Далее, предварительно обработанный сигнал может быть пропущен через метод наименьших квадратов для устранения других шумов в окружающей среде, особенно периодических шумов, таких как среднечастотные и низкочастотные помехи. Таким образом, может быть достигнута цель подавления шумов в различных условиях окружающей среды.

Так как шумы и их характеристики могут значительно меняться во времени, важно использовать адаптивные методы. Как правило, при этом изменяются только коэффициенты фильтра, а структура фильтра остается неизменной. Другими словами, коэффициенты адаптивного алгоритма обновляются с течением времени под действием более ранних результатов и входных данных. Поэтому, вместо простого метода наименьших квадратов в работе используется его рекурсивная модификация [2].

Хотя метод наименьших квадратов может достигать значительных результатов в снижении шумов, его эффект в основном ограничен периодическим шумом и низкочастотным сигналом. Поэтому, перед методом наименьших квадратов происходила предобработка с удалением импульсных шумов. Удаление производилось путём нарезки записи на окна и уменьшения высокочастотных составляющих в фурье-образах в каждом окне в отдельности.

Оценка работы фильтра проводилась следующим образом. Создавался ряд звуковых записей путём соединения некоей исходной записи с различными шумами. После работы алгоритма сравнивались фурье-образы исходной записи без шумов и конечная запись с удалёнными шумами. Сравнение проводилось путём получения коэффициента, равного сумме разности модулей значений фурье-образов. Это позволило учитывать не только то, как удалились добавленные шумы, но и искажился ли полезный звуко ряд. Далее полученный коэффициент сравнивался с таковым для зашумленной записи. В результате было получено, фильтр восстанавливает запись на 80-85%, причём результат сильно зависит от записи.

Таким образом, были изучены методы удаления шумов из звукового ряда, реализован и проверен смешанный алгоритм. В качестве направления развития исследования возможна доработка фильтра с добавлением устойчивости к другим типам шумов, а также разработка более точных методов оценки качества результатов.

### Список использованных источников:

1. Review Paper on Frequency Based Audio Noise Reduction Using Different Filters / M. Singla, H. Singh // International Journal of Science, Engineering and Technology Research (IJSETR). – 2015. – Vol. 4, № 5, P.1406–1409.
2. Implementation of Recursive Least Squares (RLS) Adaptive Filter for Noise Cancellation / S Ahmad, T. Ahmad // International Journal of Scientific Engineering and Technology. – 2012. – Vol. 1, № 4, P.46–48.

## ИСПОЛЬЗОВАНИЕ ЦИФРОВЫХ ФИЛЬТРОВ ДЛЯ ОБРАБОТКИ ИЗОБРАЖЕНИЙ

*Петкевич К. О.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Красковский П.Н. – ассистент*

Обработка изображений играет важную роль во многих областях человеческой деятельности. Одним из наиболее часто используемых методов обработки изображений является цифровая фильтрация. Она может использоваться как с целью улучшения восприятия изображения человеком, так и с целью подготовки изображения к дальнейшему анализу с помощью специализированных алгоритмов. При этом важно реализовать фильтрацию изображений таким образом, чтобы она выполнялась максимально быстро, особенно если она используется в мобильном программном средстве.

Фильтрация изображения представляет собой процесс, при котором во входном изображении изменяются определённые соотношения между его компонентами. Данные соотношения могут быть описаны с помощью математического алгоритма, который называют цифровым фильтром. В данной работе будут рассмотрены виды и способы фильтрации изображений, которые используются в мобильном программном средстве обработки изображений.

Обычно выделяют две основных группы фильтров по отношению к задаче обработки изображений: фильтры низких частот и фильтры высоких частот. При этом под частотой подразумевают так называемую пространственную частоту. В изображениях низким частотам соответствуют те участки, в которых компоненты изображения изменяются максимально плавно. В то же время высокие частоты содержат те участки изображения, где наблюдается резкая смена значений компонент изображения, например, на границах объектов [1]. Фильтры низких частот позволяют сделать изображение более размытым, так как они подавляют высокие частоты и, следовательно, на изображении становятся менее чёткими границы объектов и мелкие детали. Фильтры высоких частот действуют в точности наоборот — они позволяют повысить резкость изображения. В качестве примера на рисунке 1 представлены три изображения: исходное изображение, результат фильтрации изображения фильтром низких частот и результат фильтрации изображения фильтром высоких частот.



Рисунок 1 – Исходное изображение и результаты его обработки фильтрами низких и высоких частот

Фильтрация изображения может быть основана на применении математической операции свёртки, которая в дискретном виде представляет собой сумму произведений значений пикселей изображения и значений функции фильтра. Такой способ фильтрации известен также как фильтрация в пространственной области. Однако при больших размерах фильтра такой подход требует достаточно большого числа вычислений, так как для каждого пикселя изображения требуется вычислить значение свёртки.

Другим способом фильтрации изображений является использование Фурье-образов изображения и функции свёртки. Согласно теореме о свёртке, перемножив два этих образа можно получить образ изображения, которое является свёрткой исходного изображения и функции свёртки. Данный тип фильтрации называется фильтрацией в частотной области. Благодаря использованию быстрого преобразования Фурье он может быть вычислительно более эффективным чем фильтрация в пространственной области, особенно для фильтров больших размеров. Кроме того, используя именно этот метод можно достаточно легко модифицировать существующие фильтры, дополнительно определяя, какие частоты пропускать, а какие подавлять.

Таким образом, в разработанном мобильном программном средстве способ фильтрации изображения выбирался в зависимости от такого фактора, как размер фильтра. Так для фильтров малых размеров использовалась фильтрация в пространственной области, особенно учитывая, что данный метод проще в реализации чем фильтрация в частотной области. В то же время для обработки изображений фильтрами большого размера использовалась частотная фильтрация.

**Список использованных источников:**

1. Гонсалес Р., Вудс Р., Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. — М.: Техносфера. — 2006. — 1072 с.

## **ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИИ UNITY 3D В СИМУЛЯЦИИ РАБОТЫ ПОДЪЕМНЫХ БАШЕННЫХ КРАНОВ**

*Попивненко Е.А.*

*Белорусский государственный университет информатики и радиоэлектроники*

*г. Минск, Республика Беларусь*

*Деменковец Д.В – ассистент*

В докладе рассматриваются возможности использования игрового ядра Unity в области получения первоначальных навыков на примере обучения машинистов башенного крана. Также рассматриваются предполагаемые особенности реализации конечного программного средства.

В настоящее время ПО, которое занимается подготовкой кадров, является достаточно новым и необычным явлением. Его польза заставляет многие учебные заведения искать возможности автоматизации учебного процесса и обучения при помощи симуляторов. В рамках этого проекта с использованием технологии Unity и редактора 3D MAX был смоделирован башенный кран и панель управления.

Использование моделей и симуляций позволяет получить максимально близкий к практике опыт в контролируемых условиях. Это исключает вероятность получения травм и нанесения материального ущерба во время обучения. Одной из проблем данной области является сложность моделирования поведения механизмов, но потенциальная польза от нововведений очень высока в такой важной отрасли. С учетом этого было выбрано: моделирование работы башенного крана с позиции его машиниста.

При выборе вида моделирования было учтено, что использование виртуальной реальности увеличивает эффективность обучения и способна лучше подготовить учащегося к работе на реальной машине[1-3]. Одной из проблем для систем виртуальной реальности является отслеживание сложных движений тела человека. В рамках данного проекта все движения были тривиальными. В данном случае важны движения головы. Эти движения отслеживает акселерометр, встроенный в специальные очки.

Качественная 3д модель (рисунок 1) и панель управления, использованная в рамках проекта, обеспечивает общую схему работы башенного крана. Для панели управления был использован специализированный джойстик, также возможно использование специальной панели управления, которая схожа с панелью управления реальной машиной. Для обеспечения совместимости этих вариантов обеспечен унифицированный интерфейс доступа – COM порт компьютера. В случае необходимости возможно использование клавиатуры компьютера в качестве устройства ввода данных.

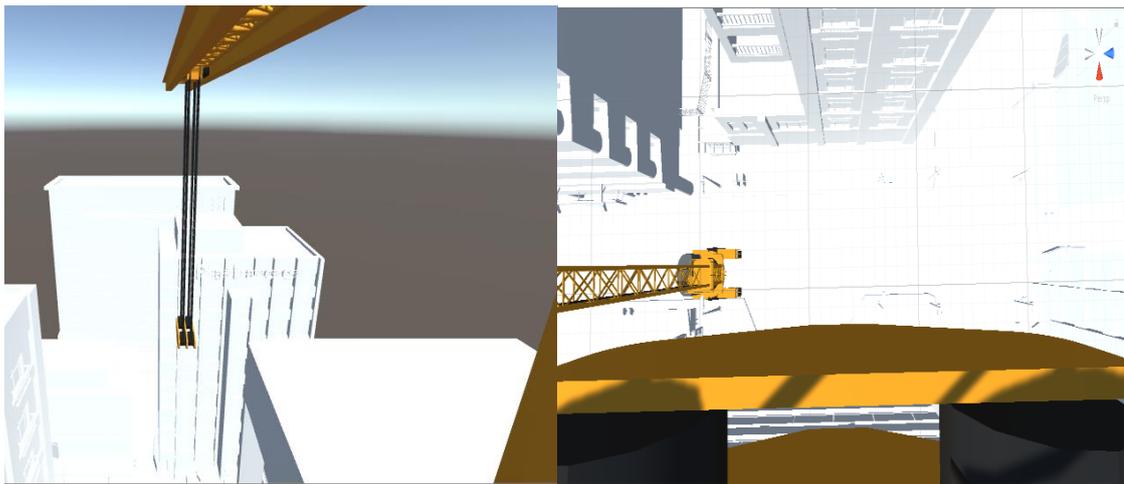


Рисунок 1 - Модель и окружение в редакторе Unity

В данном программном средстве также учитывается эффект раскачивания крана, вызванный погодными условиями. Существует возможность контролировать погодные эффекты, их интенсивность и специфические характеристики с помощью настроек программы.

Для увеличения степени вовлечения используются звуковые эффекты. Это увеличивает вовлечение в происходящее[4] – наличие звука создает иллюзию нахождения в кабине реальной машины. Также стоит отметить, что различные посторонние шумы, такие как шум ветра, звуки автомобильных трасс и прочие звуки способны воспроизвести картину “полноценного мира” в симуляторе. Таким образом, большая часть визуальных и слуховых раздражителей указывает на то, что машинист находится в кабине крана, а не за столом в учебной аудитории.

Работа механизма рассматривается с позиции обзора машиниста башенного крана. Всего имеется 3 камеры – точек обзора с которых можно наблюдать за процессом работы. Первая камера размещается внутри кабины крана (рисунок 2), вторая снаружи крана на отдалении от него и третья на стреле крана (рисунок 3). Подобное расположение камер позволяет получить всю необходимую информацию о процессе перемещения грузов и о работе крана в целом если это будет необходимо в рамках учебного процесса. Это позволяет контролировать процесс обучения машиниста и дает ему возможность получить дополнительную информацию во время симуляции. Переключение между камерами осуществляется при помощи кнопки на джойстике или панели управления.



Рисунки 2, 3 – вид из кабины крана и со стрелы.

Таким образом, в результате работы получено программное средство, выполняющее следующие функции:

- 1) Выполнение симуляции в виртуальной реальности. Для более эффективной работы требуется использование специальных очков;
- 2) Разработан специальный контроллер, схожий по функциональности с органами управления реальными машинами;
- 3) Возможность использования наушников для увеличения степени вовлечения в симуляцию;
- 4) Возможность моделирования погодных условий, таких как ветер, раскачивающий груз и сам кран;
- 5) Смоделированы реальные условия на строительной площадке. Дополнительно имеется возможность выбора из нескольких различных реалистично ведущих себя грузов.

**Список использованных источников:**

1. Doug Donovan: Virtual Reality Increases Training Effectiveness: 10 Case Studies. Арперация исследований. [Электронный ресурс] — Режим доступа: <https://www.interplaylearning.com/blog/virtual-reality-3d-simulation-training-case-studies> — Дата доступа: 02.03.2019.

2. Зиновченко Александр Николаевич: Опыт использования компьютерной обучающей системы в азовском морском институте. [Электронный ресурс] — Режим доступа: <https://cyberleninka.ru/article/n/opyt-ispolzovaniya-kompyuternoy-obuchayuschey-sistemy-v-azovskom-morskom-institute> — Дата доступа: 02.03.2019.

3. Marianne Stenger: 10 Ways Virtual Reality Is Already Being Used in Education. [Электронный ресурс] — Режим доступа: <https://www.opencolleges.edu.au/informed/edtech-integration/10-ways-virtual-reality-already-used-education/> — Дата доступа: 02.03.2019.

4. Anastasiia Ku: What makes VR real [Электронный ресурс] — Режим доступа: <https://medium.com/inborn-experience/what-makes-vr-real-c1174032eea3> — Дата доступа: 02.03.2019.

## АЛГОРИТМ ПОСТРОЕНИЯ СЕМАНТИЧЕСКОЙ СЕТИ С ОБРАБОТКОЙ МЕСТОИМЕНИЙ ТЕКСТА

*Потараев В.В.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Серебряная Л.В. – к.т.н., доцент*

В современных информационных системах обрабатываются большие объёмы информации. При автоматизации работы с информационной системой может быть использован семантический анализ текстовой информации. Модель семантической сети является одним из инструментов семантического анализа. Местоимение – это часть речи, которая встречается в тексте довольно часто. Рассмотрим метод построения семантической сети с учётом местоимений текста.

В информационных системах обрабатываются довольно большие объёмы текстовой информации. Поэтому огромное значение имеет автоматизированная обработка информации, а также её быстрота и правильность. Применение семантического анализа и баз знаний является одним из инструментов автоматизированной обработки информации.

Местоимение является частью речи, которая позволяет упростить высказывания и используется довольно часто.

Целью данной работы является разработка метода автоматизированного построения семантической сети, который учитывает высказывания текста, сформулированные с помощью местоимений.

База знаний – это компонент экспертной системы, предназначенный для хранения долгосрочных данных, описывающих определенную предметную область, и правил, описывающих целесообразные преобразования данных этой области [1].

Выделяют четыре основные модели базы знаний:

- 1) логическая модель (основанная на формулах);
- 2) продукционная модель (основанная на правилах);
- 3) фреймы (фрейм - это минимально возможное описание сущности объекта);
- 4) семантическая сеть (ориентированный граф, отражающий понятия и их отношения) [2].

Модель, основанная на семантической сети, наиболее соответствует современным представлениям об организации долговременной памяти человека [3].

Типы связей, используемые в семантической сети, выбираются в зависимости от решаемой задачи. В реальном мире число связей между понятиями огромно [4]. К основным типам отношений, которые наиболее часто применяются в семантических сетях, можно отнести следующие:

- 1) класс - элемент класса (например, “автомобиль” – “Волга”);
- 2) свойство – значение (“цвет” – “красный”) [3].

Существует модель семантической сети, которая позволяет находить ответ на вопрос [5].

Местоимением называют самостоятельную часть речи, которая используется вместо существительных, прилагательных, числительных и наречий (или их характеристик), чтобы указать на эти существительные, прилагательные, числительные и наречия (а также их признаки и количество), не называя их.

Личные местоимения указывают на объект речи – лицо, о котором идет речь. Местоимения 1 (я/мы) и 2 (ты/вы) лица обозначают участников речи. Местоимения 3 лица (он, она, оно / они) указывают на лиц, которые в речи не участвуют [6].

Таким образом, местоимения в тексте выполняют вспомогательную функцию, указывая на другие слова, и сами по себе не означают конкретное понятие. Поэтому имеет смысл при построении сети заменять местоимения словами, на которые они указывают. Это позволяет не увеличивать размер сети за счёт местоимений и упростить работу с сетью.

**Алгоритм генерации сети для текста с обработкой личных местоимений 3 лица (он, она, оно, они)** может быть представлен следующим образом:

- 1) Перейти к очередному предложению текста (начиная с первого).
- 2) Найти в предложении существительное в начальной форме. Если найдено, выбрать его в качестве подлежащего.
- 3) Если существительное было найдено, то определить его тип (род и число). Сохранить соответствие данного типа к найденному существительному.
- 4) Найти в предложении личное местоимение 3 лица (он, она, оно, они).
- 5) Если личное местоимение было найдено, то загрузить последнее существительное, соответствующее ему по типу (род и число). Выбрать загруженное существительное в качестве подлежащего.
- 6) Найти сказуемое. Добавить его в сеть, если ещё не добавлено.
- 7) Добавить связь «подлежащее-сказуемое».

8) Добавить остальные члены предложения и связи в соответствии с выбранными для сети типами связей.

9) Повторять шаги 1-8 для каждого предложения в тексте.

Для определения членов предложения в *простейшем* случае могут использоваться словари (существительных, глаголов, прилагательных и т.п.). Для определения рода и числа в *простейшем* случае можно анализировать окончания.

Рассмотрим построение семантической сети на конкретном примере.

Пусть в сети используются типы связей «подлежащее-сказуемое» и «место». Для текста «*Книга повествует о Телемаке, сыне Одиссея. Она лежит на столе.*» можно построить сеть, изображённую на рисунке 1.

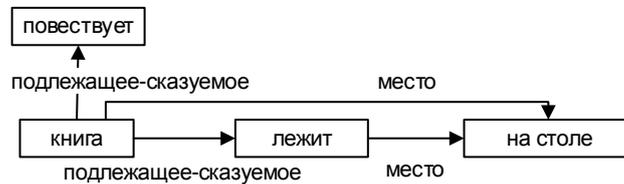


Рисунок 1 – Пример семантической сети

В соответствии с алгоритмом, в сеть последовательно добавлены узлы «книга», «повествует», «лежит», «на столе». Следует отметить, что существительное «книга» имеет женский род и единственное число. Местоимение «она» соответствует данному существительному. Остальные слова не имеют типов связей, для которых строится сеть. Поэтому они в сеть не добавлены.

В исходном тексте не было напрямую сказано, что книга находится на столе, но полученная сеть содержит эту информацию благодаря верному учёту местоимения «она».

Итак, местоимение является частью речи, которая используется довольно часто. Построение семантической сети, учитывающее значение личных местоимений 3 лица, может быть автоматизировано. Тем не менее, реализация алгоритма построения семантической сети представляет собой довольно сложную задачу из-за неоднозначности слов естественного языка.

#### Список использованных источников:

1. Базы знаний экспертных систем. [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://daxnow.narod.ru/index/0-18>. Дата доступа : 24.03.2019.
2. Гаврилова, Т.А. Базы знаний интеллектуальных систем / Т.А. Гаврилова, В.Ф. Хорошевский – СПб. : Питер, 2000. – 384 с.
3. Масленникова, О.Е. Основы искусственного интеллекта [Электронный ресурс] : учеб. пособие / О.Е. Масленникова, И.В. Гаврилова. – М.: ФЛИНТА, 2013. – Режим доступа: <http://search.rsl.ru/ru/record/01007574162>.
4. Рахимова Д. Р. Построение семантических отношений в машинном переводе // Вестник КазНУ им. аль-Фараби. Серия «Математика, механика и информатика». – Алматы, 2014. – №1. – С. 90-101.
5. Потараев, В.В. Алгоритм применения семантической сети для поиска ответа на вопрос / В.В. Потараев // Компьютерные системы и сети: материалы 54-ой научной конференции аспирантов, магистрантов и студентов. – Минск: БГУИР, 2018. – с. 103-105.
6. Все, что вы хотели узнать о местоимениях, но не знали, как спросить [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://blog.tutoronline.ru/vse-chto-vy-hoteli-uznat-o-mestoimenijah-no-ne-znali-kak-sprosit>. Дата доступа : 24.03.2019.

## ВЕБ-ПРИЛОЖЕНИЕ УЧЕТА РАБОЧЕГО ВРЕМЕНИ ПРЕПОДАВАТЕЛЯ

Рогачевский М. В.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Прохорчик Р. В. – ст. преп., м. т. н.

Время является одним из основных ресурсов организации труда на предприятии. Используя время в качестве численной меры труда, можно составлять планы выполнения работ и защитить предприятие и рабочих от большого объема рисков, связанных со сроками. В то время как коммерческое предприятие рискует потерять деньги в случае несоблюдения сроков выполнения работ, в учреждениях образования риски другого рода. При некачественном планировании работ в образовательном учреждении студенты и преподаватели рискуют в определенный момент столкнуться с критическим уровнем нагрузки, который пагубно сказывается на качестве образовательного процесса.

У большинства учебных заведений существует строгий график учебного процесса и его конкретизация – расписание аудиторных часов. Тем не менее сложно оценить качество планирования и равномерность распределения нагрузки в течение каждого этапа обучения. Работа преподавателей в учебных заведениях состоит из подобных, зафиксированных расписанием, часов и работы на кафедре, как правило, не нормированной и занимающей все оставшееся время.

Процесс учета времени, затраченного на работу, сам по себе не несет определенной пользы для организации труда на предприятии, хоть и является неплохим способом самоконтроля для рабочего лично. Однако используя инструмент для построения и графического отображения статистики на основе собранных данных можно проводить комплексный анализ как активности индивидуально взятого рабочего, так и графика работ в целом.

Таким образом для контроля и анализа эффективности и качества графика учебного процесса учебного заведения необходимо создать программное средство для сбора данных о рабочем времени преподавателей и построении отчетов на основе собранной статистики. Также необходимо позволить автоматически учитывать уже распланированное рабочее время, например, аудиторные часы.

Необходимо обеспечить преподавателям доступ к программному средству из любой точки в любое удобное время. Такими возможностями обладает сеть Интернет, позволяя использовать веб-приложения в сети, не устанавливая дополнительного программного обеспечения. Исходя из этого очевидным преимуществом перед другими архитектурными решениями будет обладать веб-приложение.

Одним из главных модулей программного средства является модуль учета рабочего времени. Очевидно, что время на запись активности необходимо свести к минимуму, чтобы не отвлекать преподавателя от образовательного процесса. Хоть процесс занесения данных в систему будет отнимать у преподавателя очень незначительное время, скорее всего необходимо реализовать искусственную мотивацию для преподавателя записывать активности снова и снова, чтобы обеспечить точность собранных данных. Для формирования привычки необходимо создать награду, которую пользователь будет получать, внося данные в систему [1]. Учитывая приведенные факты, процесс учета рабочего времени имеет следующие характеристики:

- 1) только основная информация об активности;
- 2) автоматическое фиксирование аудиторных часов;
- 3) интеллектуальное построение предлагаемой выборки активностей на основе уже собранной информации о преподавателе;
- 4) система очков и наград для выработки привычки внесения данных.

Для повышения функциональности построения отчетов об активности преподавателя необходимо использовать теги при записи активности. Данное решение позволит обогатить дополнительной информацией узкий круг работ, что дает возможность строить широкий спектр отчетов, комбинируя теги для определенной активности. Теги можно использовать как конкретизацию определенной активности, избегая тем самым создания нового вида активности и повышая качество работы интеллектуального подбора выборки активностей в процессе учета рабочего времени.

Статистические сводки удобно анализировать графически, поэтому программное средство будет позволять строить графики и диаграммы исходя из выбранных для анализа данных. Пользователь наделенный определенными правами сможет строить статистические отчеты как по активности определенного преподавателя так и по категориям активности в целом, используя данные всех преподавателей, что позволяет проводить комплексный анализ образовательного процесса.

**Список использованных источников:**

1. Charles Duhigg, The Power of Habit: Why We Do What We Do in Life and Business // New York: Random House, 2012. Print,
2. Информационные технологии и управление: материалы 49 науч. конф. аспирантов, магистрантов и студентов, Минск, 6–10 мая 2013 г. / Белорус. гос. ун-т информатики и радиоэлектроники; редкол.: Л. Ю. Шилин [и др.]. – Минск: БГУИР, 2013. – 103 с.

## АНАЛИЗ СПОСОБОВ ОРГАНИЗАЦИИ ИЕРАРХИЧЕСКИХ СТРУКТУР В РЕЛЯЦИОННЫХ БАЗАХ ДАННЫХ

Савёнок В.А.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Медведев С.А. – к.т.н., доцент

Одним из центральных понятий в объектно-ориентированном программировании является класс и, следовательно, иерархия классов выступает в качестве основополагающего типа связи классов, представленного в виде иерархии. Таким образом, наиболее характерной задачей для интерфейсных модулей взаимодействия с базами данных SQL типа на основе объектно-ориентированного интерфейса является организация эффективного хранения и работы с иерархическими структурами в реляционной базе данных.

Существует четыре основных способа представления иерархических структур в реляционных БД: хранение ссылки на родителя, метод накопления пути, модель вложенных множеств и таблица замыканий. Рассмотрим перечисленные методы на примере реализации дерева классов представленного на рисунке 1.

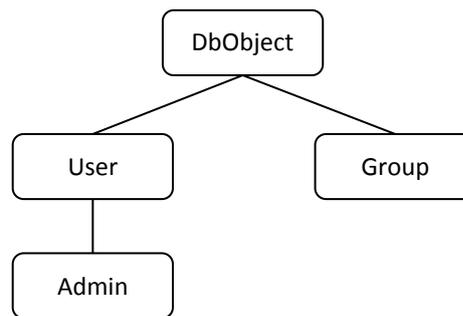


Рисунок 1 – Графическая представление иерархии классов

Самым простым в понимании способом организации иерархической структуры в реляционной БД является *хранение в каждом элементе иерархии ссылки на родителя*. В такой структуре каждый узел дерева знает своего непосредственного родителя. К плюсам этого метода относится простота хранения структуры, простота добавления и удаления узла из дерева, простота получения непосредственных потомков узла, а также обеспечение ссылочной целостности при помощи стандартных механизмов СУБД. К недостаткам стоит отнести сложность манипуляций с поддеревьями: в частности, получение всех потомков какого-либо элемента, т.к. для этого необходимо произвести столько операций соединения, сколько уровней содержится в интересующем нас поддереве.

Прямым развитием метода хранения ссылки на родителя является *метод накопления пути*. Суть его заключается в добавлении поля, сохраняющего полный путь к текущему элементу в виде строки. Этот способ позволяет проще работать с поддеревьями, однако не предоставляет возможности использования стандартных средств для поддержания ссылочной целостности. Более того, операции по удалению элементов из середины иерархии требует перебора всех элементов, находящихся на нижних уровнях, и обновления их путей, что негативно сказывается на производительности операции.

Совершенно другой подход реализуется в *модели вложенных множеств*. В ней дерево классов из примера будет иметь вид, изображенный на рисунке 2. Здесь связь «родитель – сын» представлена вложенностью сына в родителя. Для того, чтобы представить связь в виде таблицы необходимо завести два дополнительных поля, обозначающих левую и правую границы множества.

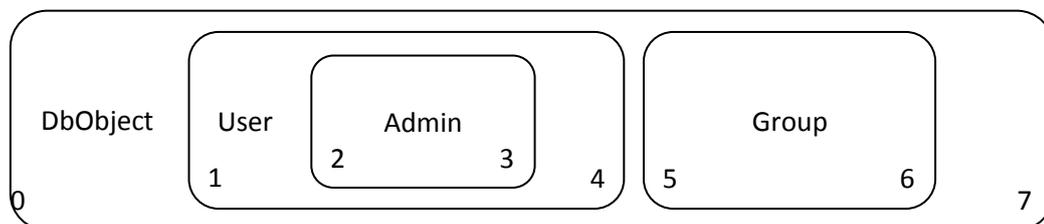


Рисунок 2 – Представление иерархии классов в модели вложенных множеств

Множества можно интерпретировать как одномерные отрезки, имеющие координаты начала и конца. Алгоритм получения поддерева в текущем подходе достаточно прост – необходимо запросить из базы все записи, границы которых находятся между границами родителя. При этом алгоритм не зависит от количества уровней в иерархии. Получение пути от листа к корню также не требует большого количества операций соединения. К недостаткам модели относится высокая трудоёмкость реализации операций добавления и удаления элементов иерархии, т. к. при этом необходимо производить пересчет границ всех множеств; отсутствует поддержка стандартных механизмов обеспечения ссылочной целостности в базе данных. Сложность восприятия описанного подхода также приводит к осуществлению выбора в пользу других способов организации иерархий.

Последний из рассматриваемых способов организации иерархических структур в реляционных базах данных – *таблица замыканий* – является более универсальным, т. к. позволяет представить в реляционной базе данных и более сложные структуры – графы. Особенностью этого способа является необходимость введения новой таблицы, содержащей связи каждого элемента иерархии со всеми его предками, а также ссылки каждого элемента на самого себя – собственно замыкание. Все связи изображены на рисунке 3.

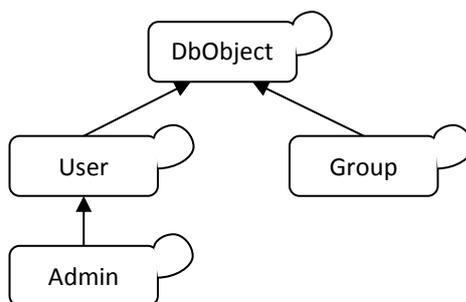


Рисунок 3 – Представление связей между элементами иерархии в методе замыканий

В описанном подходе исходная таблица не модифицируется. Поддержание таблицы замыканий в актуальном состоянии требует дополнительных операций вставки и удаления. С одной стороны, увеличивается расход памяти, а с другой – значительно упрощается работа со структурой данных. К плюсам можно отнести возможность использования стандартных механизмов обеспечения ссылочной целостности в БД. Более того, в СУБД, поддерживающих каскадные операции, существует возможность упрощения операции удаления элемента из дерева путем установки соответствующей каскадной операции. К минусам относится неэффективное добавление новых элементов в конец глубокой иерархии по сравнению со вставкой вблизи корня дерева. Однако, учитывая тот факт, что в реальных задачах редко встречается необходимость в использовании иерархий глубокой вложенности, его значимость следует оценивать в зависимости от реальных условий использования рассматриваемого подхода.

При сравнительном анализе описанных способов основными критериями для оценки были выбраны сложность реализации и поддержка стандартных механизмов обеспечения ссылочной целостности. Также следует отметить, что метод накопления пути и модель вложенных множеств не поддерживают стандартные механизмы обеспечения ссылочной целостности, что понижает их надежность и негативно сказывается на общей оценке. Из проведенного анализа следует, что наиболее гибкой, оптимальной по производительности и надежной с точки зрения целостности данных моделью организации иерархических структур в реляционных базах данных является таблица замыканий. Несмотря на необходимость создания дополнительной таблицы для хранения всех связей элементов и неэффективность вставки в глубину иерархии, выбор последнего подхода в качестве оптимального оправдан ввиду статистического преобладания операций чтения над операциями записи при работе с БД. С другой стороны, для работы с деревом, при которой не будут производиться операции с поддеревьями, хранение ссылок на непосредственного родителя в элементах иерархии является предпочтительнее ввиду простоты и скорости реализации.

**Список использованных источников:**

1. Ramez Elmasri, Smakant B. Navathe. Fundamentals of Database Systems, 6th Edition. – Addison-Wesley, 2011;
2. C.J. Date, Hugh Darwen. Databases, Types, and The Relational Model: The Third Manifesto. – Pearson, 2006;
3. Object Relational Mapping (OR Mapping) Definition [Электронный ресурс]. – Режим доступа: [http://www.service-architecture.com/articles/object-relational-mapping/object-relational\\_mapping\\_or\\_mapping\\_definition.html](http://www.service-architecture.com/articles/object-relational-mapping/object-relational_mapping_or_mapping_definition.html) Дата доступа: 20.02.2019;
4. Bill Karwin. Models for hierarchical data [Электронный ресурс]. – Режим доступа: <https://www.slideshare.net/billkarwin/models-for-hierarchical-data> Дата доступа: 03.03.2019;
5. Managing Hierarchical Data in MySQL [Электронный ресурс]. – Режим доступа: <http://mikehillier.com/articles/managing-hierarchical-data-in-mysql/> Дата доступа: 03.03.2019.

## ПРОГРАММНОЕ СРЕДСТВО СРАВНИТЕЛЬНОЙ ОЦЕНКИ МАРШЕВЫХ ТЕСТОВ ОЗУ

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Селедец В.Н., Леванцевич В.А.

Ярмолик В.Н. – д.т.н., профессор

Рассматривается программная модель неисправностей оперативных запоминающих устройств. Описана архитектура программного средства, которое позволяет провести сравнительный анализ различных способов формирования тестов для обнаружения различных типов неисправностей.

Оперативные запоминающие устройства (ОЗУ) выполняют одну из важнейших функций в современных цифровых системах обработки и хранения информации. Результаты исследований показывают, что отказы ОЗУ составляют до 70% от общего числа отказов вычислительных систем. Причиной неисправных состояний ОЗУ является наличие физических или механических дефектов либо множества подобных дефектов, количество и многообразие которых практически неограниченно.

Для аналитического описания неисправных состояний ОЗУ используются математические модели неисправностей, так или иначе отражающие реальные физические дефекты ОЗУ.

Функциональные неисправности ОЗУ подразделяются на два подмножества (рисунок 1): неисправности матрицы запоминающих элементов и неисправности электронного обрамления. Второе подмножество включает неисправности дешифраторов адреса и неисправности логики чтения/записи. Доминирующее значение имеют неисправности матрицы запоминающих элементов ОЗУ.

К неисправностям матрицы запоминающих элементов ОЗУ относят неисправности, в которых участвуют: одна ячейка ОЗУ; две ячейки ОЗУ; несколько ячеек ОЗУ, в общем случае более чем две, без ограничений на их количество.

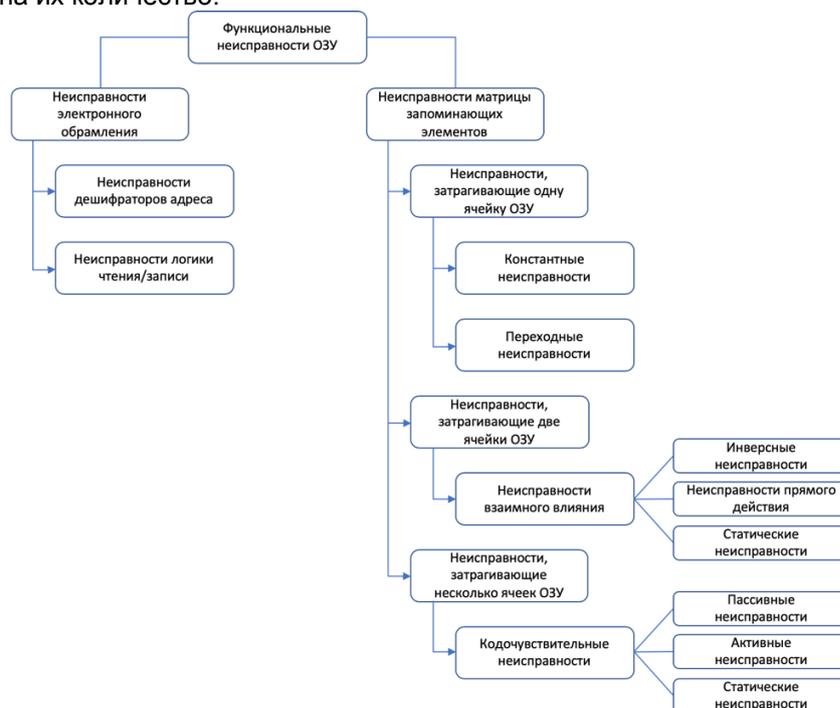


Рисунок 1 - Схема функциональных неисправностей ОЗУ

Тестирование является одним из наиболее эффективных методов повышения надёжности работы вычислительных устройств и систем. Основное назначение тестирования заключается в обнаружении максимально возможного числа неисправных состояний тестируемого объекта за минимально возможное время. В случае ОЗУ время тестирования зависит от его ёмкости  $N$ , которая подразделяет все тесты на три основных класса, а именно на тесты типа  $N^2$  циклов, тесты типа  $N^{3/2}$  и тесты типа  $N$  циклов, где циклом является время обращения (доступа) к ОЗУ.

Широкое применение находят тесты, имеющие линейную зависимость сложности от ёмкости ОЗУ, получившие название маршевые тесты.

Маршевый тест состоит из конечного числа маршевых элементов [1]. В свою очередь, каждый маршевый элемент содержит символ, определяющий порядок формирования адресной последовательности ОЗУ:  $\nabla$  - определяет последовательный перебор адресов ОЗУ по возрастанию,  $\odot$  - определяет последовательный перебор адресов по убыванию,  $\odot\nabla$  - означает перебор по убыванию либо по возрастанию. Кроме этого, маршевый элемент содержит последовательность операций чтения и записи, заключённых в круглые скобки и разделяемых точкой с запятой. Каждая операция представляет собой элемент из следующего набора: 'r0' – операция чтения содержимого ЗЭ с ожидаемым значением 0, 'r1' – операция чтения ЗЭ с ожидаемым значением 1, 'w0' – операция записи 0 в ЗЭ, 'w1' – операция записи 1 в ЗЭ. Одна или несколько операций в маршевом элементе используется последовательно для адресуемой ячейки ОЗУ. Переход к следующей ячейке будет осуществлён только после выполнения всех операций в текущем маршевом элементе.

Количество маршевых элементов, их взаимное расположение, а также вид каждого из этих элементов определяет покрывающую способность маршевого теста, т. е. эффективность обнаружения всех видов возможных неисправностей ОЗУ.

Для того, чтобы определить эффективность маршевого теста, необходимо определить его основные характеристики, к которым относятся покрывающая способность, вычислительная сложность формирования, время обнаружения неисправности.

Для решения поставленной задачи разработано программное средство, позволяющее смоделировать запоминающее устройство, указать типы неисправностей запоминающих элементов и выполнить заданные маршевые тесты с целью определения их сравнительных характеристик.

Программное средство имеет объектно-ориентированную архитектуру, где каждый объект является реализацией определённой абстракции предметной области. Так, одной из главных абстракций является запоминающее устройство [2].

Запоминающее устройство в программном средстве представлено в виде класса, содержащего ёмкость  $N$  запоминающего устройства, массив  $N$  запоминающих элементов, и два метода: метод чтения значения запоминающего элемента и метод записи значения в запоминающий элемент. Оба метода принимают адрес запоминающего элемента в качестве параметра.

Запоминающий элемент характеризуется адресом, значением и типом неисправности.

В программном средстве используется ещё одна важная абстракция – алгоритм, представляющий собой способ формирования маршевого теста. Алгоритм, в свою очередь, состоит из трёх других абстракций: контроллера теста, который управляет ходом теста, генератора последовательности значений, которые записываются в запоминающие элементы и генератора адресов запоминающих элементов, по которым записываются сгенерированные значения. При использовании тестов с изменяемыми начальными состояниями, генератор последовательности значений позволяет предварительно записать последовательность значений в начале процедуры тестирования.

Ниже приведена упрощённая структурная схема программного средства (рисунок 2).

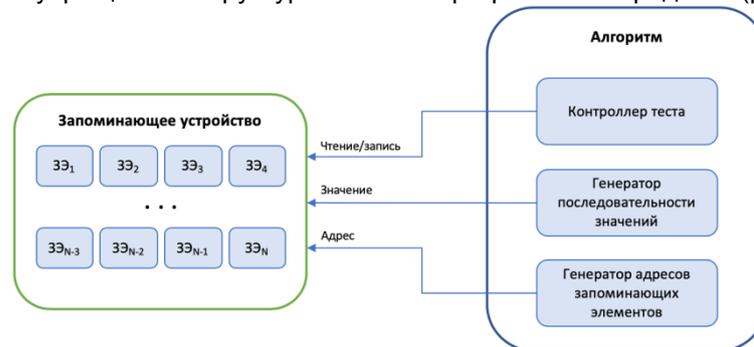


Рисунок 2 - Упрощённая структурная схема программного средства

Выводы. В результате решения поставленной задачи было разработано программное средство, позволяющее смоделировать запоминающее устройство, указать типы неисправностей запоминающих элементов, сформировать по определенным алгоритмам тесты и вывести результаты их работы, для сравнительного анализа.

**Список использованных источников:**

1. С.В. Ярмолик, А.П. Занкович, А.А. Иванюк Маршевые тесты для самотестирования ОЗУ.: Изд. Центр БГУ, 2009, с. 7-28.
2. А.В. Степанов, А.А. Иванюк, Программное средство моделирования функциональных неисправностей встроенных ОЗУ: Минск: А.Н. Вараксин, 2009, с. 1-2

## ЗАДАЧА ПОСТРОЕНИЯ МОДЕЛИ ЛИЧНОСТИ НА ОСНОВЕ АНАЛИЗА ТЕКСТА

Субачус В.В.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Парамонов А.И. – к.т.н., доцент

В данной работе рассмотрено текущее состояние проблемы построения модели личности на основе анализа текста, рассмотрены современные решения, которые используются для построения модели личности, и сами разновидности моделей личности. Так же рассмотрены существующие программные средства и их особенности.

Личность является совокупностью характеристик поведения, привычек, эмоций, мотиваций и модели мышления человека. Личность каждого человека оказывает огромное влияние на различные аспекты его жизни: выбор в сложных ситуациях, благосостояние, здоровье и другие. Тексты часто отражают различные аспекты личности автора. Проведённые исследования показали, что эти аспекты могут быть получены не только из произвольных текстов, но и при анализе данных в социальных сетях.

Автоматическое определение личностных качеств имеет множество важных практических применений. По полученным данным можно предсказать поведение и предпочтения людей, упростить работу криминалистам при анализе текста, рекомендовать продукты и услуги людям со схожим типом личности, а также при собеседовании определить пригодность человека к той или иной работе. Анализ текста проводится с целью построения модели личности автора. Общего алгоритма для анализа текста на сегодняшний день не существует. Предполагается, что разработчику необходимо выполнить анализ множества литературных источников и затем разрабатывать свой уникальный алгоритм.

Существует ряд подходов к построению модели личности, но на сегодня среди них явно выделяется модель “Большая пятёрка” [1]. Модель основана на переменных, которые наиболее широко представлены в языке. Она включает в себя пять основных измерений: экстраверсия (вовлеченность), доброжелательность (приятность), добросовестность (надежность), эмоциональная стабильность, культурность (открытость к опыту). Каждое из этих измерений имеет шесть характеристик, которые дополнительно, более полно, характеризуют человека в соответствии с измерением. Хорошим дополнением этой модели являются модели потребностей и ценностей. Потребности описывают, какие параметры влияют на выбор человека, и включает в себя двенадцать характерных потребностей. Ценности описывают мотивирующие факторы, которые влияют на принятие решения человеком, и включает в себя пять значений. Каждое измерение модели “Большая пятёрка” описывается тремя таблицами: таблица фасетов, диапазон характеристик и таблица первичных и вторичных измерений. Основной проблемой данной модели считается ее неполнота, то есть из-за нехватки измерений она может не подходить для поставленной задачи.

В тоже время можно увидеть использование улучшенной модели “Большая пятёрка” или совершенно других моделей [2]. К примеру, модель HEXACO является улучшенной версией модели “Большая пятёрка” с дополнительным измерением “Честность – смирение”. Модель MBTI представляет абсолютно иной подход. В этой модели выделяются такие измерения человеческой личности, как: интроверсия; экстраверсия, интуиция; чувствительность, чувство; мышление, восприятие; понимание. Для самых простых задач используется классическая модель, которая состоит из четырёх темпераментов: холерик, сангвиник, меланхолик, флегматик.

На сегодня мало известны программные средства, которые способны корректно построить модель личности автора на основе написанного им текста. На данный момент крупнейшим сервисом, который может выполнить данную задачу, является сервис Personality Insights от компании IBM [3]. Он основан на модели “Большая пятёрка” и предоставляет собой интерфейс для получения информации из социальных сетей или корпоративных данных. Из этих данных сервис формирует портреты людей и указывает на их потребительские предпочтения. В данном сервисе поддерживается наибольшее разнообразие языков для написания анализируемого текста. Ещё существует ряд аналогов сервиса от IBM, но зачастую они выполняют конкретные цели: сервис Traitify, Life With Personality предоставляет API оценки личности; сервис FoxType служит для анализа “вежливости” текста и т.п. Кроме того – все эти сервисы не могут работать с текстами на русском языке.

### Список использованных источников:

1. Big Five model [Electronic resource]. – Mode of access: <https://sentic.net/deep-learning-based-personality-detection.pdf> – Date of access: 10.03.2019.
2. The Big Five: Alternatives [Electronic resource]. – Mode of access: <https://nexuszine.wordpress.com/2016/12/26/the-big-five-alternatives/> – Date of access: 07.03.2019.

3. Personality Insights [Electronic resource]. – Mode of access: <https://cloud.ibm.com/docs/services/personality-insights?topic=personality-insights-about#about> – Date of access: 08.03.2019.

## СОВРЕМЕННОЕ СОСТОЯНИЕ СИСТЕМ ИНФОРМАЦИОННОЙ ПОДДЕРЖКИ В СФЕРЕ ТУРИЗМА

Труханович И.А.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Парамонов А.И. – к.т.н., доцент

В настоящее время туризм является одним из важнейших экономических секторов. В связи с этим остро стоит необходимость задействования соответствующих систем поддержки в данной сфере. Данные системы должны предоставлять весь необходимый функционал, используя при этом современный спектр технологических возможностей.

Туризм является одним из самых прибыльных секторов в мировой экономике. Количество международных туристов почти удвоилось с 1995 года, и ожидается его увеличение в среднем на 3,3% в год до 2030. Вместе с тем современный турист с дифференцированным образом жизни, индивидуальными мотивами и особыми интересами требует сервис, который должен учитывать все заявленные предпочтения. Среди отдельных аспектов, на которые необходимо обратить внимание, можно отметить туристическую мобильность, высокий риск и неопределенность в незнакомой среде.

Всё вышперечисленное подчёркивает острую востребованность в соответствующих системах информационной поддержки. Среди функций таких систем можно отметить следующие:

- 1) создание и поддержание персонализированных целей путешествия, предпочтений и критериев оценки;
- 2) поиск и просмотр соответствующей информации (места назначения и места проживания, достопримечательности и особенности, комплексные туры и туристические агентства);
- 3) просмотр аналитической информации о пункте назначения;
- 4) предоставление процедуры оценки и отбора, используя заранее определенные критерии для выбора;
- 5) получение и просмотр рекомендаций конкретных достопримечательностей или событий;
- 6) разработка индивидуальных планов поездок;
- 7) взаимодействие с туристическими операторами.

Вследствие этого используются различные типы систем поддержки принятия решений (СППР): ориентированные на данные; ориентированные на модели; ориентированные на документы; ориентированные на коммуникации и ориентированные на знания [1].

СППР в сфере туризма на сегодняшний день используют широкий спектр технологий, включая ГИС, обработку данных социальных сетей, рекомендательные системы и т.д. Отдельно следует обратить внимание на задействование в данных системах возможностей проектов, связанных с добровольной географической информацией.

Добровольная географическая информация представляет собой особый вид пользовательского контента. Он относится к географической информации, собранной и добровольно переданной широкой публике (краудсорсинг). Это означает, что различного рода задачи сбора информации и решения проблем осуществляются путём открытого призыва к вкладу пользователей. В конечном итоге вместо того чтобы назначать человека или компанию для сбора информации, используются вклады отдельных лиц [2]. Обычно это происходит через интерактивный веб-сайт (рисунок 1).

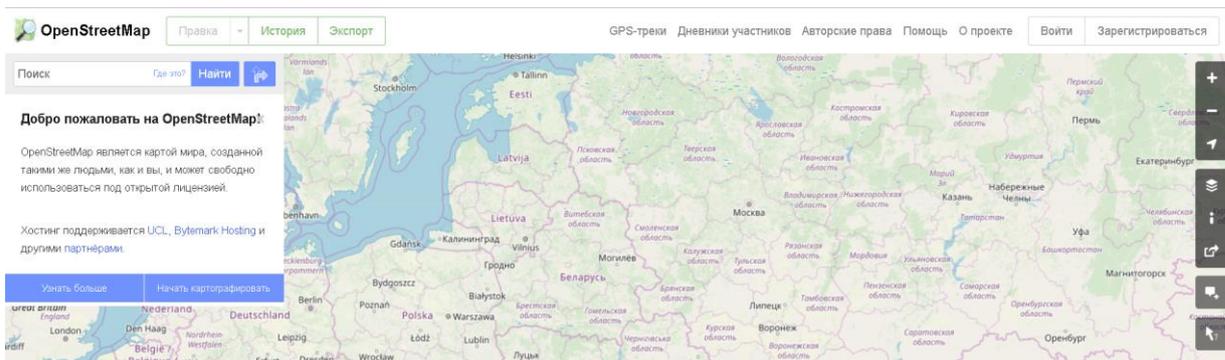


Рисунок 1 – Главная страница проекта OpenStreetMap

**Список использованных источников:**

1. Power, D. Decision Support Systems: Concepts and Resources for Managers / D. Power. – Santa Barbara: Praeger, 2002. – 272 p.
2. The Role of Volunteered Geographic Information in a Postmodern GIS World [Electronic resource]: Esri. – Mode of access: <https://www.esri.com/news/arcuser/0410/vgi.html/>. – Date of access: 11.03.2019.

## **ПРОГРАММНОЕ СРЕДСТВО ПРОЕКТИРОВАНИЯ РЕКЛАМНЫХ ВЕБ-СТРАНИЦ**

*Филипук Д.С.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Сурков К.А. – старший преподаватель*

Разработано программное средство, позволяющее с использованием визуального проектирования создавать рекламные веб-страницы, развёртывать и проводить их многовариантное тестирование. В процессе разработки центральное место занимало обеспечение высокого уровня надёжности создаваемого программного средства.

В настоящее время Интернет является огромной площадкой для продвижения различных товаров и услуг. Рекламная кампания в Интернете включает в себя два этапа: привлечение на сайт посетителей, а затем побуждение их к совершению определённого действия, называемого целевым. Размещение в сети рекламных сообщений позволяет привлечь поток посетителей, а их стимулирование к совершению целевого действия предполагает использование специально подготовленных веб-страниц, называемых рекламными или целевыми. Процесс создания целевой веб-страницы, которая будет наиболее эффективной в рамках рекламной кампании конкретной организации, является длительным многоступенчатым процессом, требующим решения ряда задач.

Возникающие задачи можно разделить на две группы: создание страницы и повышение её эффективности. Создание целевой веб-страницы включает в себя разработку дизайна и содержимого, а также непосредственное кодирование и публикацию в Интернете. Повышение эффективности начинается со сбора статистики, например, количества посещений, среднего времени, проведённого на странице и т. д. На основании этой статистики проводится многовариантное тестирование, в рамках которого разрабатываются и публикуются одна или несколько новых версий целевой страницы. С них также собирается статистика и сравнивается с изначальной, в результате чего оригинальная страница может быть заменена одной из новых версий. Многовариантное тестирование проводится до тех пор, пока не будет достигнут желаемый уровень эффективности [1].

Основным требованием к организации процесса проектирования рекламных веб-страниц является уменьшение доли участия в нём разработчиков, что позволяет ускорить данный процесс и снизить его стоимость.

В настоящий момент отсутствуют программные средства (ПС), поддерживающие выполнение всех задач проектирования целевых веб-страниц. В связи с этим было принято решение о разработке нового ПС с такой возможностью.

Созданное ПС представляет собой веб-приложение, разработанное с использованием языков программирования C# и TypeScript, а также платформы .NET и технологии Angular. Задача снижения доли участия программистов в проектировании рекламных веб-страниц решается созданием целевых страниц в визуальном редакторе, а также автоматизацией процесса их публикации. ПС поддерживает проведение многовариантного тестирования и командную работу пользователей в процессе создания целевых веб-страниц.

Важное место в процессе разработки занимало обеспечение высокого уровня надёжности ПС. В первую очередь этому способствуют используемые технологии: платформа .NET поддерживает механизм автоматического управления памятью [2], а языки C# и TypeScript являются статически типизированными, что позволяет выявлять часть ошибок в момент компиляции кода. Во-вторых, была применена техника разработки через тестирование, которая предполагает сперва написание тестовых сценариев для требуемой функциональности, а затем реализующего её кода.

Большое внимание было уделено непосредственно тестированию. Проведено статическое тестирование ПС путём анализа исходного кода программным средством SonarQube. Динамическое тестирование было автоматизировано и разделено на три уровня: модульное, интеграционное и системное. Модульное тестирование проверяет работоспособность каждого программного модуля в отдельности. Покрытие исходного кода модульными тестами составило 97.68%. В рамках интеграционного тестирования проверялось функционирование всех модулей, реализующих бизнес-

логику приложения, в сборе. Системное тестирование проверяет приложение на соответствие спецификации требований. Здесь использовалась нотация Геркина, позволяющая записывать тестовые сценарии на естественном языке и встраивать их в средства автоматизации тестирования.

ПС полностью соответствует функциональным требованиям, имеет высокую степень покрытия тестами, хорошо приспособлено для сопровождения и расширения.

**Список использованных источников:**

1. The Ultimate Guide To Landing Page Optimization [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://unbounce.com/photos/ultimate-guide-lpo.pdf>. – Дата доступа: 16.03.2019.
2. Richter, Jeffrey. CLR via C# / Jeffrey Richter. – Microsoft Press, 2010. – 896 p.

## **ОБЗОР РЕШЕНИЙ ПО АВТОМАТИЗАЦИИ ПРОВЕДЕНИЯ БУМАЖНОГО ТЕСТИРОВАНИЯ**

*Харлов А.А.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Парамонов А.И. – к.т.н., доцент*

В статье рассмотрены процессы проведения бумажного тестирования. Выполнен обзор существующих решений, в качестве которых были выбраны: централизованное тестирование и решение с применением программного продукта VeralTest.

Традиционные подходы бумажного тестирования требуют ручной обработки большого количества информации, подсчёт баллов ведёт преподаватель, на что затрачивается значительное время, так как проверяющему необходимо сверить ответ студента с правильным ответом на каждый тест, подсчитать количество правильных ответов и сумму баллов для определения оценки. Применение компьютерного тестирования не всегда технически возможно в следствии отсутствия компьютеризированных рабочих мест. Таким образом, оптимальным в организационном смысле является технология, которая не требовательна к ресурсам на этапе организации процесса тестирования, и автоматизировала бы механическую, рутинную работу проверяющего, агрегируя данные проверки в единую информационную систему.

Одним из решений данной проблемы является автоматизированная система проведения централизованного тестирования. Это форма вступительных испытаний, организованная на основе педагогических тестов, стандартизированных процедур проведения тестового контроля, обработки, анализа и представления результатов, используемая для проведения конкурса при поступлении в учреждения высшего, среднего специального и профессионально-технического образования.

Для прохождения тестирования предлагается пройти педагогический тест, для дачи ответов используется специальный бланк (рисунок 1).

Перевод ответов осуществляется путем сканирования заполненных абитуриентами бланков ответов, распознавания компьютером отсканированной информации и корректирования (при неуверенном распознании). Операции сканирования, распознавания и корректирования производятся в специально оборудованном помещении с ограниченным доступом. Обработка производится на компьютерах, объединенных в изолированную локальную сеть [1].

Для проведения тестирования может применяться и программное средство. Например, VeralTest, которое обеспечивает поддержку разных способов организации тестов, в том числе – бумажного [2]. В VeralTest предусмотрена функция создания бумажных тестов (рисунки 2 и 3).

Министерство образования Республики Беларусь | Министерство культуры Республики Беларусь  
 Учреждение: Учреждение государственного контроля знаний | Учреждение: Учреждение государственного контроля знаний

**БЛАНК ОТВЕТОВ**  
**БЛАНК АДКАЗАЎ**

Кириллица: А Б В Г Д Е Ж З И Й К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ъ Ы Э Ю Я | Цифры: 1 2 3 4 5  
 Латиница: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z | Лічбы: 6 7 8 9 0

Область регистрации: Код пункта тестирования, Код пункта государственного тестирования, Фамилия, Имя, Отчество, Документ, Серия, Номер.

Область ответов: Часть А (A1-A30), Часть В (B1-B12). Включает подсказки: 'Символ ошибочных меток', 'Адреса памятных меток'.

Дата тестирования: День, Месяц, Год. Номер варианта теста.

Рисунок 1 – Бланк ответов централизованного тестирования

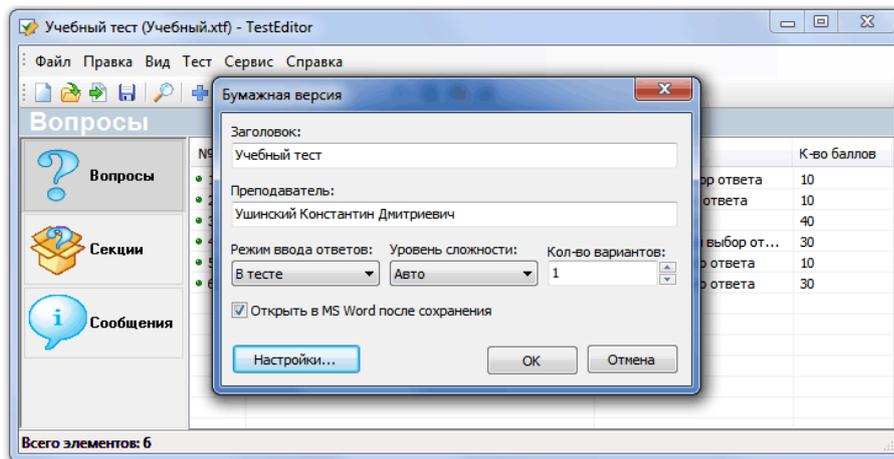


Рисунок 2 – Создание бумажной версии теста в VeralTest

К основным возможностям продукта можно отнести:

- автоматическая генерация вариантов;
- несколько режимов ввода ответов;
- вывод листа ответов.

Главным недостатком централизованного тестирования является невозможность использования системы на каждодневной основе, например, преподавателем на лекциях, т.к. система является закрытой и используется для проведения вступительных испытаний.

Обзор решений позволяет выделить основные характеристики информационной системы, которая может обеспечить автоматизацию проведения бумажного тестирования, и на их основе сформулировать техническое задание для ее создания.

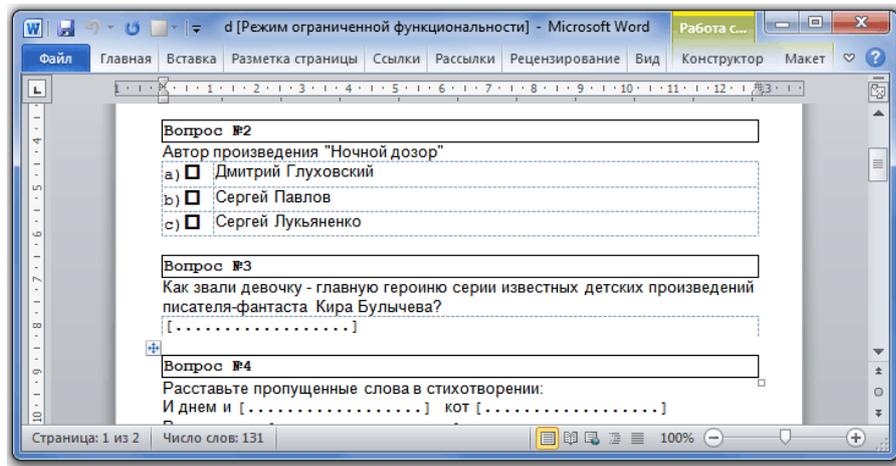


Рисунок 3 – Созданный тест из VeralTest

Программное средство VeralTest позволяет использовать его для проведения тестирования преподавателем на ежедневной основе, так как является открытым и позволяет увеличить скорость подготовки тестов, за счет генерации тестов в бумажном формате, но не имеет возможности автоматического распознавания ответов.

**Список использованных источников:**

1. Республиканский институт контроля знаний [Электронный ресурс]. – Режим доступа: <http://rikc.by/>. – Дата доступа: 01.03.2019.
2. VeralSoft – программы для создания тестов и проведения тестирования [Электронный ресурс]. – Режим доступа: <http://www.veralsoft.com/>. – Дата доступа: 01.03.2019.

## АВТОМАТИЗИРОВАННОЕ ТЕСТИРОВАНИЕ, ОСНОВАННОЕ НА КОНЦЕПЦИИ ТЕСТОВОЙ ПИРАМИДЫ

*Холодок Д.А.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Петюкевич Н.С. – м.т.н., ассистент*

Ручное тестирование постепенно вытесняется автоматическим. На сегодняшний день объем кода автоматических тестов может в несколько раз превышать объем кода самой системы. Любая большая система, написанная без надлежащих тестов, будет рано или поздно переписана заново. Концепция тестовой пирамиды дает понимание того, какие тесты и в каком соотношении должны быть написаны для разрабатываемой системы.

Тестовая пирамида, предложенная одним из авторов изобретения методологии разработки программного обеспечения Scrum Майком Конном, выделяет три типа тестов (рисунок 1):

- модульные тесты, или юнит тесты (англ. Unit tests);
- сервисные тесты (англ. Service tests);
- тесты пользовательского интерфейса (англ. User interface tests).

Юнит тесты предназначены для проверки корректности работы отдельно взятого модуля кода – функции или класса. Они работают в полной изолированности, не используя внешние сервисы, файлы, базы данных. Юнит тест передает модулю входные данные, после чего сравниваются полученный и ожидаемый результаты. На современных компьютерах время выполнения тысячи таких тестов может не превышать и минуты. Юнит тесты выявляют наибольшее количество ошибок. Их главная цель – определить, является ли отдельно взятый модуль кода корректным.

Сервисные тесты проверяют работоспособность компонента системы, состоящего из нескольких модулей. Они покрывают больше кода, чем юнит тесты. Сервисные тесты могут выполняться в изолированности, используя заглушки – модули, имитирующие поведение реальных компонентов системы. При этом некоторые тесты могут быть такими же быстрыми, как и юнит. Используя же реальные компоненты системы, например, базу данных, время выполнения теста возрастает, так как перед запуском теста надо как минимум запустить базу данных и установить с ней соединение.

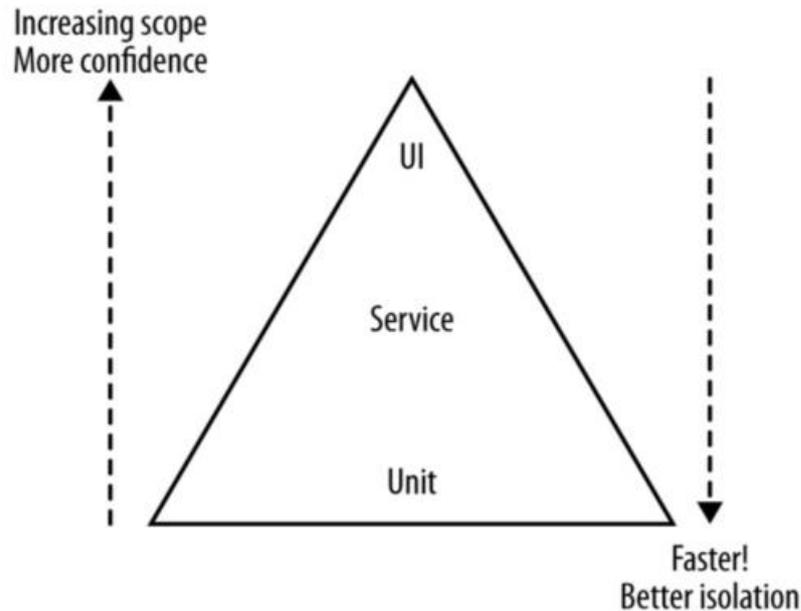


Рисунок 1 - Тестовая пирамида

Тесты пользовательского интерфейса, расположенные на вершине пирамиды, работают в пределах всей системы. Проверка ожидаемой реакции системы происходит через те же интерфейсы. Именно при невыполнении теста данного типа тяжелее всего определить ошибку. Из-за сложности написания и поддержки тестов, а также большого времени их выполнения количество таких тестов должно сводиться к минимуму.

Тесты должны быть разными по гранулярности. Двигаясь по слоям пирамиды снизу вверх, количество тестов уменьшается, а их сложность возрастает.

К сожалению, концепция тестовой пирамиды не является совершенной: далеко не однозначно по названию предназначение каждого из слоев. Однако за счет своей простоты она может послужить эмпирическим правилом при написании тестовых наборов.

**Список использованных источников:**

1. The Practical Test Pyramid [Электронный ресурс]. – Режим доступа: <https://martinfowler.com/articles/practical-test-pyramid.html>
2. Newman, S. Building microservices / S. Newman // Testing / ed. M. Loukides, B. MacDonald. – Sebastopol, 2015. – P. 231–268.

## СИМУЛЯТОР БИРЖИ КАК ПОДХОД К ОБУЧЕНИЮ ТОРГОВЛИ КРИПТОВАЛЮТАМИ

*Циуля Е.А.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Петюкевич Н.С. – м.т.н., ассистент*

Криптовалюта – это разновидность цифровых денег, в основе которой лежит технология криптографии. Она не имеет физического облика, а существует только в электронном виде. Ее основными особенностями являются анонимность, децентрализация и защищенность [1]. Колебание курса криптовалют дает возможность получить прибыль торгуя на биржах, но многие боятся рисков, сопряженных с торговлей криптовалютами, или вовсе не осведомлены о такой возможности заработка. Разработанный симулятор дает возможность получить практические навыки торговли на бирже без риска потери реальных денег. Такой подход способен упростить обучение и привлечь больше людей в ряды торговцев криптовалютами.

Волатильность криптовалюты – степень колебания ее курса за определенный промежуток времени. Высокая волатильность подразумевает высокий уровень риска: курс может устремиться вниз в любой момент. С другой стороны, высокая волатильность криптовалют дает возможность биржевым игрокам зарабатывать, при этом направление изменения цены не имеет значения.

Биржа криптовалют – это торговая платформа, позволяющая вести торги криптовалютой. Для того, чтобы начать торговать необходимо зарегистрироваться на одной или нескольких биржах,

завести электронный кошелек и пополнить баланс реальными деньгами. После этого необходимо изучить стратегии торговли, статистику закрытых сделок, разобраться как влияет на торги внешние факторы, научиться понимать и анализировать основные виды графиков. Выполнение перечисленных действий обязательны, но даже после них не будет никакого практического опыта торговли на бирже. Новому игроку придется приобретать навыки торговли на реальных активах, что может закончиться потерей всех средств.

Использование цифровых денег очень удобно: полная автоматизация, мобильность, комфортность использования. Но в случае использования криптовалют появляются дополнительные риски [2]:

- вредоносные программы;
- хакерские атаки;
- банкротство и закрытие бирж;
- отсутствие гарантий на возмещение убытков;
- крах рынка виртуальных денег;
- ошибочная транзакция пользователя;
- отсутствие законодательной базы и юридические риски.

Из описанного выше можно сделать вывод о сложности вхождения в торговлю криптовалютами. Основная цель разработки симулятора – дать возможность человеку приобрести практические навыки торговли криптовалютами без риска потери реальных вложений. Программное средство будет состоять из следующих частей: коллектор исторических данных, симулятор биржи, теоретические и статистические данные и графики.

Коллектор будет собирать исторические данные для разных бирж и нескольких валютных пар. Данные будут храниться в базе данных на отдельном сервере. Запросить данные можно будет по HTTP запросу. Использование облачных технологий позволит построить масштабируемую систему, которая будет адаптироваться к изменению нагрузки. Также такой системой легко управлять.

Симулятор можно будет настроить на работу с историческими данными и повторить определенную ситуацию на бирже, потренироваться использовать различные торговые стратегии в одинаковых условиях.

Программное средство будет содержать теоретические сведения, которые помогут научиться использовать следующие вспомогательные инструменты для торговли на бирже:

- графики;
- стратегии торговли на бирже;
- электронные кошельки;
- алгоритмы автоматической торговли.

Таким образом, с помощью симулятора криптовалютной биржи желающие смогут освоить основы торговли криптовалютами и подготовиться к реальным условиям, не теряя при этом своих вложений.

**Список использованных источников:**

1. Криптовалюта: роль в современном мире [prostocoin.com/blog/what-is-cryptocurrency](https://prostocoin.com/blog/what-is-cryptocurrency) [Электронный ресурс]. – Режим доступа: <https://prostocoin.com/blog/what-is-cryptocurrency>
2. 10 проблем и рисков криптовалют [Электронный ресурс]. – Режим доступа: <http://www.garant.ru/article/1150927/>

## **АВТОМАТИЗИРОВАННАЯ ВЕБ-СИСТЕМА УПРАВЛЕНИЯ ПРОЦЕССОМ РАСПРЕДЕЛЕНИЯ СТУДЕНТОВ НА ПРАКТИКУ**

*Чех Д. О.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Смолякова О.Г. – к.т.н., доцент*

Автоматизация посредством проектирования и разработки веб-системы управления процессом распределения студентов на производственную практику. Централизованное хранение, обработка и доступ к данным.

Цель разработки – создание веб-системы для автоматизации и контроля процесса распределения студентов на производственную практику. Объект разработки – программное средство, состоящее из пользовательского веб-интерфейса и серверного модуля управления. Предмет разработки – технологии создания веб-приложений, принципы проектирования эргономичных пользовательских интерфейсов и оптимизированного серверного модуля управления.

Основная причина создания веб-систем для работы с данными это движение по направлению к автоматизации бизнес процессов и централизации доступа к данным. Это позволяет усовершенствовать процессы, сделать их более удобными, быстрыми и простыми. Уменьшается вероятность появления ошибок и не консистентных данных в случаях человеческой невнимательности. Хранение информации в базах данных повышает сохранность и защищенность данных.

Принципы проектирования веб-систем подразумевают программирование серверной части, чтобы динамично преобразовывать и рассчитывать различные данные, при необходимости, взятые из базы данных, располагающейся на сервере и отсылаемые на клиентскую часть, где данные будут показаны на пользовательском веб-интерфейсе [1].

Для реализации серверной части проекта выбрана объектно-реляционная система управления базами данных MySQL и язык программирования Java. Бизнес логика приложения реализована в виде программных Java модулей, преобразованных фреймворком Maven для автоматизации сборки проекта на основе описания его структуры в файлах на языке POM [2]. С помощью языка Java реализована архитектура, в которой на уровне сервисов происходят запросы в базу данных, написанные на языке SQL [3], а также реализована побочная логика программы. Для написания приложения использовалась среда разработки IntelliJ IDEA 2019.

Клиентская часть веб-приложения - это пользовательский веб-интерфейс, который включает в себя графические элементы для взаимодействия с пользователем.

Для создания графического интерфейса используются дополнительные веб-технологии: скрипты, написанные на языке JavaScript, HTML и CSS. Все элементы веб-страницы могут взаимодействовать между собой и менять свое состояние в зависимости от действий пользователя.

Серверная и клиентская части программного средства взаимодействуют при помощи технологии AJAX. Данная технология определяет подход к построению интерактивных пользовательских интерфейсов веб-приложений, заключающийся в фоновом обмене данными браузера с веб-сервером. В результате, при обновлении данных веб-страница не перезагружается полностью, и веб-приложения становятся быстрее и удобнее.

В проекте рассмотрена задача проектирования и реализации веб-системы управления процессом распределения студентов на производственную практику.

Веб-система позволяет автоматизировать процесс распределения студентов на практику и процесс учета и контроля заявок от организаций. Также система позволяет централизовать хранение и обработку данных о студентах, кураторах и заявках. Хранение всей информации о распределении студентов в одной базе данных, позволяет усовершенствовать контроль и уменьшить вероятность утери или не консистентности информации, в отличие от хранения данных в отдельных цифровых файлах на локальных рабочих станциях. Также данная веб-система позволяет упростить и ускорить процесс работы лица, ответственного за распределение студентов на практику. Большим плюсом централизованной веб-системы является возможность студентов и кураторов отслеживать информацию о распределении. Студенты могут дистанционно узнавать информацию о своем распределении и о назначенном им кураторе. Куратор, в свою очередь, имеет возможность быстро узнать всех студентов, которых он курирует.

Система включает три роли пользователей: администратор, куратор и студент. Администратор имеет полный доступ к информации о других пользователях и доступ к изменению данных (рисунок 1). Куратор имеет права на просмотр информации о распределении, а так же может просматривать информацию о студентах, для которых он является куратором по практике. Студент может просматривать и редактировать информацию о себе, а так же узнать информацию о результатах распределения на практику.

В программном средстве реализованы функции:

- регистрация пользователей в веб-системе;
- авторизация пользователей в веб-системе;
- регистрация заявок на распределение;
- редактирование данных пользователя;
- редактирование данных заявки на распределение;
- поиск и фильтрация подходящих кандидатов для распределения;
- распределение студента на практику по заявке;
- контроль и учет распределенных студентов;
- контроль и учет состояния заявок;
- назначение куратора по практике;
- просмотр пользователями информации о распределении.

Веб-система предоставляет администратору большой список возможностей для упрощения и ускорения процесса распределения студентов на практику. Широкий спектр различных фильтров позволяет определять подходящих кандидатов для распределения по заявкам.

PracticeManager adminlogin Log out

+ Create faculty / speciality + Register student / head of practice Show all requests

### Students information table

Assign student(s) Release student(s) Delete student(s)

Fast search ...

<input type="checkbox"/>	Surname	Name	Faculty	Speciality	Education basis	Average mark	Status	Company	Practice period	Show profile
<input type="checkbox"/>	Chekh	Dima	FCSAN	ITS	Chargeable	8.32	Busy	NC	2017-12-29 - 2017-12-30	Profile
<input type="checkbox"/>	Martsenuk	Roman	FCSAN	ITS	Budget	8.0	Busy	NC	2017-12-29 - 2017-12-30	Profile
<input type="checkbox"/>	Gorbunov	Vlad	FCP	ME	Budget	8.1	Available	-	-	Profile
<input type="checkbox"/>	Redkovskiy	Andrey	FCSAN	VMSIS	Chargeable	7.5	Busy	NC	2017-12-29 - 2017-12-30	Profile
<input type="checkbox"/>	Golubko	Dima	FCSAN	ITS	Budget	8.0	Busy	NC	2017-12-29 - 2017-12-30	Profile

Showing 1 to 5 of 11 rows 5 rows per page

Рисунок 1 – Веб-система управления процессом распределения студентов на производственную практику

Нововведением веб-системы является возможность в заявке указывать приоритетные параметры, по которым будет производиться фильтрация студентов, что позволяет администратору в веб-системе определять кандидатов наиболее подходящих под требования определенной заявки от организации.

Положительным качеством данной веб-системы является адаптивный графический пользовательский интерфейс. Пользователь может заходить в систему на любом цифровом устройстве с доступом к интернету, а не только через компьютер с большим экраном. Ориентируясь на то, что большое количество потенциальных клиентов веб-системы это студенты и кураторы, данный подход к проектированию графического интерфейса является наиболее предпочтительным.

Подводя итог, стоит отметить, что внедрение информационных технологий в различные сферы труда и проектирование различных программных средств, позволяющих автоматизировать и управлять процессами работы с данными, приносит пользу в виде экономии рабочего времени и минимизацию вероятности появления ошибок.

**Список использованных источников:**

1. Фаулер М. Архитектура корпоративных программных приложений / М. Фаулер. – Москва : Вильямс, 2006. – 544с.
2. Apache Maven 3 Cookbook / Srirangan, Carsten Ziegeler. – Packt Publishing Ltd. : Birmingham, 2011. – 224.
3. Бейли Л. Изучаем SQL / Л. Бейли. – СПб. : Питер, 2012. – 592с.

## МЕТОД ПРЕДОСТАВЛЕНИЯ ТУРИСТИЧЕСКИХ УСЛУГ С ИСПОЛЬЗОВАНИЕМ НЕЙРОСЕТЕВЫХ ТЕХНОЛОГИЙ

*Чигир В.С.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Хмелева А.В. – к.т.н., доцент*

В докладе рассмотрен процесс построения и обучения модели нейронной сети для принятия решения о выборе оптимального туристического продукта для пользователя. По результатам рассмотрения выделены этапы, отмечены важные моменты сбора и подготовки данных, выбора метода обучения, интерпретации и анализа результатов.

Применение наиболее полных и точных сведений о клиенте в процессе предоставления туристических услуг является ключевым для успеха бизнеса. Для продвижения туристического продукта необходим эффективный механизм управления данными в быстро меняющейся среде и процесс принятия решений с высокой точностью и гибкостью[1]. Таким решением является использование искусственной нейронной сети для подбора оптимально туристического продукта.

Для реализации сети необходимо осуществить сбор и препроцессинг данных, на которых предусматривается обучение модели. Далее необходимо построить модель и обучить ее. Затем производится анализ качества и интерпретация результатов.

При сборе данных необходимо выделить характеристики, которые будут представлять клиента как объект в будущей модели. Задача является фундаментальной, на ее основе строится дальнейший процесс. Следует обратить внимание на такие индивидуальные различия туристов, как возраст, финансовые средства, семейное положение, подходящие погодные условия, требовательность к уровню комфорта и качеству услуг. Необходимо выделить все важные признаки и определить жесткие критерии для принятия решения о признаке. Признаки делятся на три категории:

- количественные, например возраст клиента;
- бикатегориальные, например наличие или отсутствие детей у клиента;
- категориальные, ответом на которые является класс из набора.

По окончании сбора данных осуществляется их отображение в формат, приспособленный для обучения. Для этого все данные приводятся в числовую форму, производится их нормализация с целью масштабировать значения признаков. В результате получается обучающая выборка.

Далее выбирается модель исходя из того, насколько важна точность модели и ее интерпретируемость, то есть понятность результатов. Кроме того, самое важное - внедрение построенной модели в бизнес. Здесь гибкость и способность к изменениям является ключевым качеством модели. Для нейронных сетей можно глубоко и индивидуально настраивать архитектуру. Есть возможность переходить от однослойной нейронной сети к многослойной для поиска нелинейных зависимостей, меняя при этом несколько строк кода. Еще из преимуществ можно отметить наличие ряда качественных фреймворков для машинного обучения.

Для обучения нейросети существует множество методов. Обучение сводится к тому, чтобы подобрать наиболее подходящие весовые коэффициенты для параметров с целью минимизации ошибки. Довольно эффективным и широко используемым является метод обратного распространения ошибки. Алгоритм предусматривает два прохода по слоям сети. При проходе в прямом направлении веса элементов сети фиксированы. При проходе в обратном направлении веса меняются в процессе коррекции ошибок[2].

При наличии обработанной и нормированной обучающей выборки можно провести обучение. Однако часто элементы данных для каждого из возможных результатов представлены в неравных пропорциях. В таком случае можно добиться большого процента достоверности для неверного результата. Чтобы этого избежать, необходимо проводить обучение на примерах, содержащих равное количество данных для всех вариантов результата.

Выбрав модель и алгоритм обучения, следует разделить выборку на две части. На большей части данных проводится обучение. Небольшая часть обособляется для тестов с целью анализа качества полученной модели.

Качество оценивается по достоверности результатов в процентах, кроме того используются такие метрики, как точность, полнота, F-мера[2]. Для интерпретации модели проводится обратное отображение числовых результатов в исходные признаки.

Таким образом, с помощью нейросетевых технологий осуществляется построение, обучение и анализ модели для принятия решения о выборе оптимального туристического продукта.

**Список использованных источников:**

1. Дихтяр, В. И. К вопросу об использовании нейронных сетей в индустрии туризма. / В. И. Дихтяр. – 2013.
2. Хайкин, С. Нейронные сети: Полный курс. / С. Хайкин. – 2016.

## ПРИМЕНЕНИЕ ХРАНИЛИЩ ДАННЫХ ДЛЯ СРЕДНЕГО И КРУПНОГО БИЗНЕСА

*Куликов С.С., Шавлис В.К.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Куликов С.С. – к.т.н., доцент*

Работа рассматривает перспективы развития бизнеса после внедрения такой Business Intelligence технологии, как хранилище данных. В данном материале выделены наиболее значимые цели, задачи и особенности создания и эксплуатации хранилищ, а также часто встречающиеся сложности преобразования данных. Работа описывает и массовые средства для работы с обработанной информацией.

По сколько развитие языков программирования и типов хранения данных происходит очень быстро и стремительно, приложения для управления клиентами, сотрудниками, запасами, поставщиками, финансами и продажами должны так же быстро реагировать на эти изменения. Но с имеющимися подходами к данным, их нельзя было проанализировать с точки зрения историчности изменений, что и стало целью развития хранилищ. Хранилища данных поддерживают бизнес-решения за счет сбора, консолидации и организации данных для составления отчетов и анализа с помощью интеллектуального анализа данных. При создании хранилища мы можем столкнуться со сложностью: разные типы данных, требующие приведения к общему виду для возможности сравнения и анализа.

Чтобы помочь бизнесу развиваться необходимо перейти от транзакционной базы данных, которая хранит в себе все актуальные сведения о предметной области и направлена на быстроту вставки малого количества новых данных, к хранилищу данных, призванному дать четкий анализ работы предприятия или фирмы. Благодаря произведенному анализу можно принять важные бизнес-решения, способные принести огромную прибыль делу, и выявить существующие проблемы, мешающие развиваться. Далее существует множество приложений для визуализации данных, извлеченных из хранилища (Tibco Spotfire, Qlikview, Tableau и др.). Для возможности такого анализа хранилище данных должно содержать абсолютно все исторические изменения данных, что существенно увеличит объемы хранящейся информации. Если огромные размеры транзакционной базы данных могут значительно замедлить работу приложения (обработку операций), то увеличение размера хранилища только благоприятно повлияет на детальный анализ бизнеса.

Для более аргументированного перехода от транзакционной базы данных к хранилищу приведем основные их особенности. Транзакционная база данных предназначена для бизнес-операций в реальном времени. Такая база оптимизирована для общего набора транзакций, обычно добавляя или получая по одной строке за раз для таблицы. При загрузке данных производятся проверки входящих значений во время транзакций (в реальном времени), используются таблицы проверки данных. Хранилище данных предназначено для анализа бизнес-показателей по категориям и атрибутам. Оно оптимизировано для массовых загрузок и больших запросов, которые обращаются к множеству строк в таблице. Данные загружаются уже согласованными, то есть не требуют проверки в реальном времени.

Хранилище данных призвано обслуживать своих пользователей - аналитиков и консультантов. Оно должно быть спроектировано с учетом следующих требований [1]:

- 1) обеспечено центральное хранилище согласованных данных;
- 2) быстро обрабатываются сложные запросы;
- 3) предоставлено множество мощных аналитических инструментов, таких как OLAP и интеллектуальный анализ данных.

Следует соблюдать необходимые свойства архитектуры для хранилищ данных [2]:

- 1) разделение. Необходимо разделить аналитическую и транзакционную обработку данных;
- 2) масштабируемость. Аппаратные и программные архитектуры должны легко обновляться, поскольку увеличиваются и объем данных, и число требований пользователей;
- 3) расширяемость. Система должна размещать новые приложения без её перестройки;
- 4) безопасность. Доступ к данным, имеющимся в хранилищах данных, должен быть защищенным;
- 5) управляемость. Управление хранилищем данных не должно быть слишком сложным.

Цель хранилищ — интерпретировать большое количество данных, заостряя внимание лишь на ключевых факторах эффективности, моделируя исход различных вариантов действий предприятия, отслеживая результаты принятия решений. Данные хранилищ включают в себя стратегии и технологии, используемые предприятиями для аналитики деловых данных.

**Список использованных источников:**

1. Greenwald R., & Stackowiak R., & Stern J. Oracle Essentials, Fifth Edition. Sebastopol: O'Reilly Media, Inc., 2013 – P 214.
2. Greenwald R., & Stackowiak R., & Stern J. Oracle Essentials, Fifth Edition. Sebastopol: O'Reilly Media, Inc., 2013 – P 122.

## **МОДЕЛИ И АЛГОРИТМЫ РАСПРОСТРАНЕНИЯ РАДИОАКТИВНЫХ ВЕЩЕСТВ В ОКРУЖАЮЩЕЙ СРЕДЕ**

*Шамына А.Ю.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Лапицкая Н.В. – к.т.н., доцент*

В настоящее время в мире увеличивается число потенциально опасных объектов, которые представляют угрозу возникновения аварий с радиоактивным загрязнением окружающей среды. Одновременно с этим особый интерес представляют аварийные прогнозные модели и их программные реализации, которые позволяют осуществить поддержку принятия управленческого решения на разных этапах развития аварийной ситуации. Такими моделями являются современные реализации гауссовых моделей.

Данные модели относятся к локальным моделям распространения загрязняющих веществ в окружающей среде. Это означает, что данные модели могут использоваться для моделирования переноса загрязняющих веществ в атмосфере не более чем на 10 км от источника и для экспресс-оценки на расстоянии до 30 км.

Данная модель представляет набор эмпирических формул для описания многочисленных экспериментальных данных.

В гауссовых моделях предполагается, что перенос в атмосфере загрязняющего вещества по горизонтали и по вертикали происходит по нормальному закону распределения при постоянных направлении и скорости ветра и условиях сохранения устойчивости атмосферы в течение времени переноса.

Однако допущения относительно внешних условий для среды моделирования значительно снижают достоверность результатов прогноза развития аварийной ситуации и делают невозможным использование этих моделей на расстоянии более 30 км от источника выброса [1].

В настоящей работе предлагается применение уточненных метеопараметров для конкретной точки расчетной сетки, а также использование профиля местности вдоль оси следа при расчете коэффициента шероховатости подстилающей поверхности. Этот подход значительно повышает достоверность использования гауссовых моделей и позволяет делать прогноз на расстоянии до 50 км.

Для проведения исследований моделей было реализовано соответствующее программное средство, в котором используется усовершенствованная гауссова модель. Реализованное программное средство также поддерживает следующие функции:

- 1) расчет приземной концентрации;
- 2) расчет внешних и ингаляционных доз облучения;
- 3) расчет внутренних доз облучения;
- 4) построение для выбранного радионуклида полигонизированного покрытия;
- 5) выработку рекомендаций по проведению защитных мероприятий для населения;
- 6) расчет плотности выпадений на подстилающую поверхность;
- 7) оценка загрязнения окружающей среды при нормальной эксплуатации.

Поля концентраций выпадений загрязняющих веществ отображаются как полигонизированное покрытие. Оно строится с использованием интерполяционного метода обратных радиусов, исходными точками для которого служат полученные в результате расчета точечные значения с заданной дискретностью. После чего данное покрытие привязывается к картографической подложке.

При разработке данного программного средства использовался язык программирования C#. Для построения графического интерфейса пользователя была использована технология WPF. Работа с картографическими слоями осуществляется с использованием библиотеки DotSpatial.

Достоверность работы модели обеспечена результатами верификации с другими известными моделями, а также результатами верификации на конкретных радиационных инцидентах и экспериментах по распространению примеси в атмосфере.

При моделировании учитывается целый ряд фактических метеорологических параметров. На данный момент для их импорта используются открытые API погоды, однако возможно использование данных о погоде с АСКРО (автоматизированная система контроля радиационной обстановки), а также близлежащих метеостанций для повышения точности расчета [2].

**Список использованных источников:**

1. Методика расчета рассеяния загрязняющих веществ в атмосфере при аварийных выбросах. РД 52.18.717-2009. Обнинск, ООО «ПРИНТ-СЕРВИС» -- 2009. – С. 36 - 42.
2. Шамына, А. Ю. Программное средство для моделирования распространения радионуклидов в окружающей среде на основе гауссовой модели / А. Ю. Шамына, А. Д. Ардяко // Информационные технологии и системы 2018 (ИТС 2018) = Information Technologies and Systems 2018 (ITS 2018) : материалы международной научной конференции, Минск, 25 октября 2018 г. / Белорусский государственный университет информатики и радиоэлектроники ; редкол. : Л. Ю. Шилин [и др.]. – Минск, 2018. – С. 182 - 183.

## ПОДГОТОВКА И ОБРАБОТКА МАРКИРОВАННЫХ ТЕСТОВЫХ ЗАДАНИЙ

*Шендерович В.А.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Деменковец Д.В. – ассистент*

В докладе рассматривается программное средство “Генератор тестов”, которое позволяет автоматизировать процессы подготовки тестовых заданий и обработки ответов на них, основываясь на маркировке формируемых тестов штрих-кодами.

Компьютерное тестирование имеет ряд присущих ему недостатков, что ограничивает сферу его применения и обуславливает сохранение необходимости использования в учебном процессе традиционной формы контроля знаний студентов, осуществляемой преподавателем с использованием экзаменационных билетов, то есть тестовых заданий, выдаваемых на бумажном носителе. Работа программного средства “Генератор тестов” построена на основе заполняемой пользователем базы данных, содержащей наборы вопросов по темам учебного предмета и ответы на них. Как правило, для наполнения такой базы имеется достаточно учебного материала по любому курсу обучения. Создание тестовых заданий (например, экзаменационных билетов) с необходимыми параметрами осуществляется путем случайной выборки из базы заданного количества вопросов по определенным темам, при этом обеспечивается возможность сохранения сформированных тестов, каждый из которых является уникальным и маркируется штрих-кодом, а также их изменения, удаления и вывода на печать. В процессе формирования тестовых заданий возможно указание пользователем следующих параметров: общее количество вопросов в задании, количество вопросов по каждому уровню сложности, число вариантов задания, возможность повторения одного и того же вопроса в разных вариантах.

В состав программного средства входит мобильное приложение, обеспечивающее распознавание штрих-кода тестового задания, по которому производится его автоматическая идентификация, и передачу соответствующих данных программе. Для сопряжения телефона с компьютером используется специальный QR-код, формируемый программой по запросу пользователя и считываемый мобильным приложением.

Программное средство подготовки и обработки маркированных тестовых заданий “Генератор тестов” позволяет обеспечить:

- формирование базы вопросов и ответов, включая ввод, сохранение, редактирование и удаление тем, вопросов по темам и ответов на вопросы;
- генерацию тестовых заданий на основе случайной выборки из базы данных с указанными пользователем параметрами, сохранение и удаление заданий, формирование соответствующих файлов в формате MS Word, снабженных идентифицирующими штрих-кодами, и их распечатку;
- выполнение операций с единой базой данных, включающей вопросы, ответы и сформированные тесты: сохранение, очистка, экспорт, импорт;
- работу (включая проверку правильности полученных ответов) с ранее сгенерированными тестовыми заданиями, выбор которых осуществляется пользователем вручную путем указания позиции в списке или ввода идентификатора, либо производится автоматически путем сканирования штрих-кода через мобильное приложение.

Внешний вид окна программы “Генератор тестов” приведен на рисунке 1:

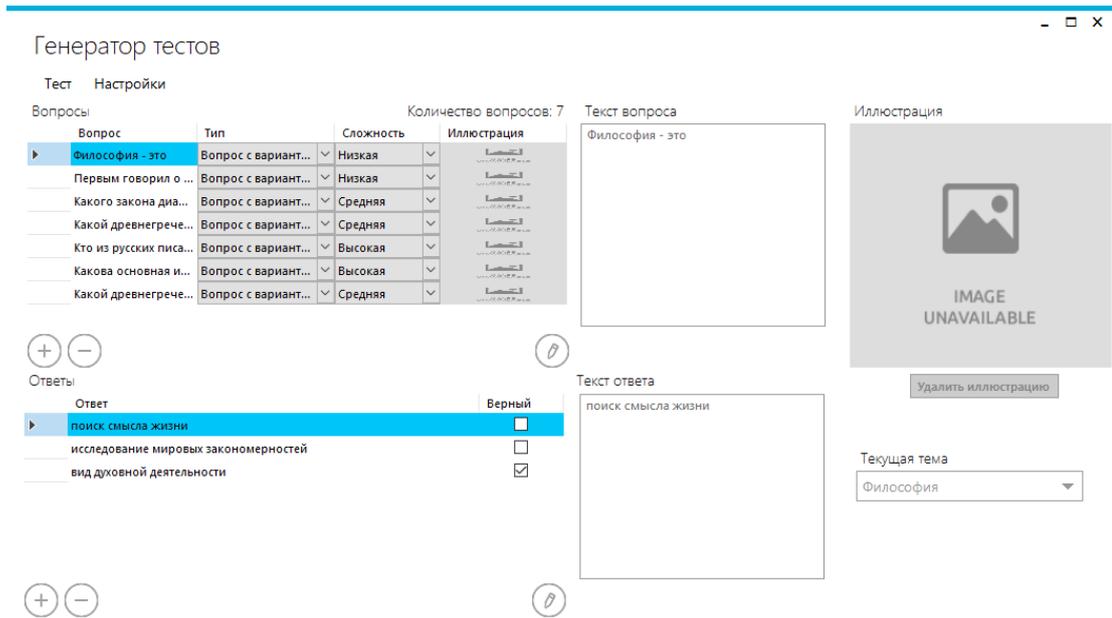


Рис. 1 – Программное средство “Генератор тестов”

Пример сгенерированного документа приведен на рисунке 2:



0 000000 000031

Философия ПОИТ 1 курс

Вопросы	1	2	3	4
Ответы				
Вопросы	5	6	7	
Ответы				

1. Кто из русских писателей-классиков больше всего повлиял на развитие идеалистической философии в России?
  1. Л. Н. Толстой;
  2. Ф. М. Достоевский;
  3. А. С. Пушкин.
2. Какова основная идея философии В. С. Соловьева?
  1. Идея Софии – Божественной мудрости;
  2. Идея непротивления злу насилием;
  3. Идея революционного обновления общества.
3. Какой древнегреческий философ считал огонь основой всего?
  1. Гераклит;
  2. Анаксимандр;
  3. Анаксимен.
4. Какого закона диалектики Гегеля не существует?
  1. Закон сохранения энергии;
  2. Закон отрицания отрицания;
  3. Закон единства противоположностей.
5. Какой древнегреческий мыслитель считал, что главная задача состоит в самопознании?
  1. Платон;
  2. Сократ;
  3. Аристотель.
6. Первым говорил о том, что можно быть только философом, но не мудрецом
  1. Платон;
  2. Парменид;
  3. Протагор;
  4. никто из названных.
7. Философия - это

Рис. 2 – Пример тестового задания, созданного “Генератором тестов”

Полученное программное средство обеспечивает создание и управление базой данных тестовых вопросов, формирование тестовых заданий из этих вопросов, а также генерацию готовых для печати текстовых документов на основе тестовых заданий. Данное программное средство подходит для применения в высших и среднеспециальных учебных заведениях для контроля знаний, а в частности для проведения дифференцированных и недифференцированных зачетов.

Список использованных источников:

1. Титенко С.В. Автоматизация построения тестовых заданий в системах дистанционного обучения на основе понятийно-тезисной модели. [Электронный ресурс]. Режим доступа: <https://cyberleninka.ru/article/v/avtomatizatsii-postroeniya-testovyh-zadaniy-v-sistemah-distantsionnogo-obucheniya-na-osnove-ponyatiyno-tezishnoy-modeli>.
2. Зорин Ю.А. Автоматизация построения многовариантных тестовых заданий на основе деревьев И/ИЛИ. Автореферат. Томск-2015. [Электронный ресурс]. Режим доступа: <http://tekhnosfera.com/view/602708/a#?page=1>.
3. Черных Т.А. Автоматизация процесса формирования экзаменационных билетов с использованием LATEX [Электронный ресурс] / Черных Т. А., Полищук Ю. В. // Университетский комплекс как региональный центр образования, науки и культуры: материалы Всерос. науч.-метод. конф., 29-31 янв. 2014 г., Оренбург / М-во образования и науки Рос. Федерации, Федер. гос. бюджет. образоват. учреждение высш. проф. образования "Оренбург. гос. ун-т". - Оренбург, 2014. - С. 2681-2685. Режим доступа: <http://elibr.osu.ru/bitstream/123456789/175/1/2681-2685.pdf>.
4. Шендерович, В. А. Использование штрих-кодов для автоматизации работы с тестовыми заданиями / В. А. Шендерович // Компьютерные системы и сети: материалы 54-й научной конференции аспирантов, магистрантов и студентов, Минск, 23 – 27 апреля 2018 г. / Белорусский государственный университет информатики и радиоэлектроники. – Минск, 2018. – С. 124 – 125.

## ВЫЯВЛЕНИЕ КЛЮЧЕВЫХ ПРИЗНАКОВ ПРИ ИНДЕКСИРОВАНИИ ТЕКСТОВЫХ ДОКУМЕНТОВ

*Шичков Д.В.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Куликов С.С. – к.т.н., доцент*

Предлагается метод выделения ключевых признаков при построении поисковых образов текстовых документов. Метод основан на использовании интеллектуальных алгоритмов статистического анализа документов, в которых учитываются характеристики не только самих текстов, но и знания о предметной области в виде тематических корпусов текстов и сформированных на их основе лингвистических словарей. Метод может быть использован при индексировании текстовых документов из различных источников (Интернет, локальная сеть, жесткие диски отдельных пользователей).

Индексирование текстовых документов обеспечивает не только быстрый, но и качественный поиск. В процессе индексирования происходит присваивание документу структуры, называемой поисковым образом, которая описывается в терминах информационно-поискового языка. Различают два вида индексирования [1]: координатное и посткоординатное (или классификационное индексирование). При первом подходе поисковый образ документа описывается дескрипторами (ключевыми словами, семантическими признаками или понятиями определенных онтологий) в некоторой системе координат [1, 2], а при втором – документ относится к одному или нескольким заранее сформированным классам, индекс которых выступает в роли поискового образа соответствующего текста [1]. В большинстве существующих информационно-поисковых систем используется координатное индексирование, выполняемое полностью в автоматическом режиме [3].

Можно определить следующие этапы при формировании поисковых образов документов [4]:

- сбор файлов, предназначенных для индексирования;
- выделение из собранной коллекции документов текстового содержимого с максимально возможной степенью сохранения исходной разметки и форматирования;
- лингвистическая обработка текста;
- непосредственная процедура индексирования с сохранением результатов в базе данных информационно-поисковой системы.

В зависимости от типа информационного источника, из которого происходит формирование списка документов, предназначенных для индексирования, используются различные стратегии и технологии. Например, если источником выступает локальная сеть или жесткий диск компьютера пользователя, то сбор файлов можно выполнить простым сканированием интересующих объектов (общедоступных папок на компьютерах локальной сети, жестких дисках).

Основными проблемами при анализе ресурсов сети Интернет являются: стремительный рост объемов информации и ее быстрое изменение, динамическая генерация ресурсов путем выполнения инструкций на удаленном. С учетом этого, современными системами всего обработано порядка 60% доступных ресурсов Интернета без учета информации, которая хранится в базах данных и недоступна для индексирования («скрытый» веб). Для решения этих проблем создаются метапоисковые системы, которые осуществляют поиск путем обращения к индексным структурам других поисковых систем [5].

Трудность выделения текстового содержимого из загруженных файлов связана в первую очередь с тем, что документы имеют различные форматы представления (например, html, doc, pdf и т.д.). Таким образом, для перевода документа в формат plain text («чистый» текст) необходимо произвести процедуру конвертации. Вторая проблема, возникающая на данном этапе, – определение кодировки документов. Она в особенности характерна для текстов на русском и белорусском языках.

**Список использованных источников:**

1. Индексирование // Большая научная библиотека [Электронный ресурс]. – Режим доступа: <http://bse.scilib.com/article054017.html>. – Дата доступа: 05.03.2019.
2. Новый систематизированный толковый словарь // Государственная публичная научно-техническая библиотека России [Электронный ресурс]. – 1995. – Режим доступа: <http://www.gpntb.ru/win/book/>. – Дата доступа: 05.03.2019.
3. Майковский, В.В. Обзор подходов и методов индексирования в информационно-поисковых системах / В.В. Майковский // Сб. тр. Всерос. науч. школы-семинара молодых ученых, аспирантов и студентов «Интеллектуализация информационного поиска, скантехнологии и электронные библиотеки». – Таганрог: Изд-во ТТИ ЮФУ, 2010. – С. 77–78.
4. Manning, C. Introduction to Information Retrieval / C. Manning, P. Raghavan, H. Schütze. – 1 edition. – Cambridge University Press, 2008. – 496 p.
5. Захаров, В.П. Информационные системы (документальный поиск): учеб. пособие / В.П. Захаров. – СПб.: Изд-во СПбГУ, 2002. – 188 с.

## КОМПИЛЯТОР ЯЗЫКА ОБЕРОН, РЕАЛИЗУЮЩИЙ ПАРАДИГМУ РАСШИРЯЕМОГО ПРОГРАММИРОВАНИЯ

*Шулицкий Д.С.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Сурков К.А. – ст. преподаватель*

В настоящее время перед программистами зачастую встаёт задача расширения функционала существующего кода. Зачастую эта задача решается при помощи механизмов наследования и полиморфизма в ООП. Однако это не позволяет модифицировать уже существующие объекты и переопределять статические функции.

Самый простой способ модификации существующей программы – изменение её исходного кода. Однако, это не всегда допустимо. Например, может быть недопустима модификация библиотечного модуля. Модификация существующего кода приводит к необходимости повторного тестирования модулей.

В объектно-ориентированном программировании названные выше проблемы решаются при помощи наследования и полиморфизма. Однако это не позволяет модифицировать данные в существующих объектах. Для модификации кода необходимо создавать объекты классов, что зачастую не удобно.

Эти проблемы можно решить при помощи расширяемого программирования. Расширяемое программирование состоит из расширения кода и расширения данных.

Расширение кода означает, что новый модуль изменяет работу существующих модулей и реализуется при помощи переопределения существующих процедур, которое похоже на переопределение методов в ООП. Существует несколько способов реализовать переопределение кода:

- 1) замена адреса вызываемой процедуры во всех точках её вызова;
- 2) вызов процедуры через процедурную переменную (вместо прямого вызова процедуры);
- 3) генерация компилятором процедуры-переходника, которая выполнит вызов через процедурную переменную;
- 4) замена кода процедуры по месту.

Расширение данных означает добавление новых полей в существующую запись. Для расширения данных можно вручную ассоциировать оригинальные объекты с объектами-дополнениями, но следствием этого являются следующие проблемы:

- высокая трудоёмкость, так как необходимо поддерживать таблицу соответствий;
- снижение надёжности, так как необходимо уничтожать объекты-дополнения при уничтожении оригинальных объектов, а ошибки при реализации могут привести к утечкам памяти даже при наличии сборщика мусора;
- падение производительности при поиске объекта-дополнения для объекта-оригинала.
- необходимость синхронизации доступа к таблице соответствия в многопоточных приложениях.

Решением этого является поддержка на уровне компилятора списка дополнений у записей (объектов). Дополнения можно добавлять к существующим записям в момент первого обращения к полю расширения, а можно в момент загрузки модуля.

Так как на этапе компиляции доступна информация о переопределённых процедурах и дополнениях записей, существует возможность генерировать динамически подключаемые модули, которые не только добавляют новый функционал в момент загрузки модуля, но и удаляют свои следы в момент выгрузки.

В качестве языка, на основе которого тестировалась данная парадигма, был использован Оберон. Этот язык имеет очень компактное описание, и, при этом, содержит в себе достаточно высокоуровневый функционал. Отчёт об языке содержит описание его синтаксиса, что упрощает разработку компилятора и расширение языка.

Компилятор языка Оберон выбирался исходя из следующих требований:

- 1) возможность расширения синтаксиса языка;
- 2) поддержка компиляции под наиболее популярные архитектуры процессоров: x86, x64, arm.

Существующие компиляторы в данное время не поддерживаются или не позволяют компилировать код одновременно под все данные архитектуры. Таким образом, было принято решение разработать собственный компилятор, удовлетворяющий данным требованиям.

В качестве библиотеки для написания компилятора была выбрана LLVM. Данная технология позволила создать эффективный компилятор, генерирующий код для большого количества архитектур.

Разработанный компилятор открыл большие возможности по написанию надёжных расширяемых программ, в том числе сторонними разработчиками.

## **ПРИМЕНЕНИЕ ТЕОРИИ МАССОВОГО ОБСЛУЖИВАНИЯ В НАТУРНОМ ЭКСПЕРИМЕНТЕ**

*Шульга Е. С.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Сурков К. А. – ст. преподаватель*

Рассмотрено применение теории систем массового обслуживания для моделирования процессов. Математический аппарат теории СМО позволяет рассчитать ключевые характеристики функционирования системы. Предложена последовательность действий, используя которую можно задать модель для анализируемого процесса. Для расчета характеристик системы в натурном эксперименте необходимо фиксировать промежутки времени между поступающими в систему заявками, а также промежутки времени между двумя обработанными заявками.

Различные виды деятельности человека можно представить в виде систем массового обслуживания (СМО). В таких системах можно выделить следующие элементы: заявка, поступающая на обработку; процессор, главный вычислитель, занимающийся обработкой заявок; очередь, в которой накапливаются заявки, ожидающие обработки [1]. В системе в общем случае может быть один или несколько обработчиков, причём они могут быть настроены на различные дисциплины обслуживания.

Теория СМО может быть применена к различным видам деятельности человека. Например, в работе [2] рассматривается применение данной теории к работе преподавателя университета. Сам преподаватель выступает в роли обслуживающего прибора, он занимается обработкой заявок – проверкой результатов выполнения индивидуальных заданий студентами.

Математический аппарат теории массового обслуживания позволяет рассчитать характеристики эффективности работы системы [1]:

- пропускную способность;
- вероятность того, что обработчик будет занят и очередная заявка будет помещена в очередь или отброшена (в соответствии с дисциплиной обслуживания системы);
- среднее число заявок в очереди или во всей системе во время ее работы;
- среднее время ожидания заявки в очереди.

Чтобы для задачи создать модель на основе теории массового обслуживания, необходимо выполнить следующие шаги. Для начала, нужно определить, что является заявкой, поступающей в систему. Затем необходимо задать обслуживающее устройство, количество параллельно

работающих приборов, дисциплину обработки заявок: может ли обработка завершиться неудачей, что происходит с заявкой при неудаче (повтор обработки или отбрасывание заявки). Также необходимо определить, происходит ли накопление заявок, если занят процессор, определить емкость накопителя, которая может быть конечной или бесконечной. В случае конечной очереди необходимо задать, что происходит со вновь поступающей заявкой в случае, когда очередь заполнена: блокируется ли источник, или происходит отбрасывание заявки.

В случае имитационного моделирования для работы модели необходимо задать интенсивность поступления заявок в систему, а также интенсивность обработки заявок. Однако данную модель можно применять и в натурном эксперименте. В этом случае необходимо фиксировать характеристики реальных объектов, а именно: промежутки времени между двумя последовательно поступающими заявками  $t_n$ , промежутки времени между двумя последовательно обрабатываемыми заявками  $a_n$ .

Рассчитав средний промежуток времени между двумя последовательно пришедшими требованиями  $\bar{t}$ , можно определить интенсивность поступления заявок в систему  $\lambda = \frac{1}{\bar{t}}$ . Точно так же через среднее время между двумя последовательно обработанными заявками  $\bar{a}$  можно определить интенсивность обрабатывающего устройства  $\mu = \frac{1}{\bar{a}}$ . А используя данную пару значений интенсивности, можно рассчитать остальные характеристики эффективности функционирования системы.

Расчет характеристик функционирования системы может быть использован в качестве обоснования действий по оптимизации ее функционирования. Возможно изменение дисциплины обслуживания, расширения накопителя, подключение дополнительных обрабатывающих устройств. Цель оптимизации всегда будет заключаться в уменьшении времени нахождения заявки в системе.

**Список использованных источников:**

1. Клейнрок, Л. Теория массового обслуживания / Л. Клейнрок. – М: Машиностроение, 1979. – 432 с.
2. Шульга, Е. С. Модель приёма преподавателем работ студентов на основе системы массового обслуживания / Е. С. Шульга, К. А. Сурков // Дистанционное обучение – образовательная среда XXI века : материалы X международной научно-методической конференции (Минск, 7 - 8 декабря 2017 года). – Минск : БГУИР, 2017. – С. 145 - 146.

## СИСТЕМА МОНИТОРИНГА ПАРАМЕТРОВ USB-ИНТЕРФЕЙСА

*Щербаков М.С., Пикиреня П.И., Болтак С.В.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Леванцевич В.А. – м.т.н.,  
старший преподаватель*

Описана система мониторинга основных параметров USB-интерфейса. В качестве контролируемых параметров используются напряжение и сила тока. Система позволяет не только контролировать указанные параметры, но и отслеживать динамику их изменений во времени, что позволяет оценить состояние устройств, использующих USB-интерфейс в качестве источника питания.

В настоящее время существует большое количество устройств, которые в качестве источника питания используют USB-интерфейс. При этом сам интерфейс может быть частью компьютерной системы или выполнен в качестве отдельного источника питания с USB-разъемом. Основными силовыми характеристиками USB-интерфейса являются напряжение питания и выходной ток, который может обеспечивать интерфейс.

Как показала практика эксплуатации, не все USB-интерфейсы обеспечивают заявленные производителем силовые характеристики, что может привести к трудноопределяемым неисправностям работы USB-устройств либо к повреждению самого интерфейса.

Для контроля параметров USB-интерфейса была разработана система, состоящая из аппаратной и программной части. Аппаратная часть представляет собой отдельный модуль и содержит датчик тока на основе микросхемы ACS712ELCTR-05B-T [1], микроконтроллер и модуль WIFI. В качестве контроллера была выбрана отладочная плата STM32 Nucleo-144 на базе высокопроизводительного 32-битного микроконтроллера STM32F746ZGT6, построенного на ядре ARM Cortex M4 [2]. Выбор данного микроконтроллера был обоснован необходимостью приобретения опыта работы с микроконтроллерами данного семейства, так как они обладают хорошими

характеристиками при невысокой стоимости и широко используются при разработке современных аппаратных устройств.

Вход разработанного аппаратного модуля подключается к USB-интерфейсу, а к выходу модуля подключается USB-устройство, которому необходимо питание по USB, при этом информационные линии USB-интерфейса проходят через аппаратный модуль транзитом, без изменений. Аналоговый сигнал, пропорциональный протекающему току, с датчика тока поступает на один из аналоговых входов АЦП микроконтроллера, а уровень напряжения питания USB-интерфейса поступает на другой аналоговый вход АЦП. После оцифровки значений тока и напряжения они передаются на компьютер для дальнейшей обработки и анализа.

Для связи с компьютером используется проводное и беспроводное соединение. Проводной способ связи основан на UART интерфейсе и использует преобразователь UART/USB. Это способ соединения используется, как правило, для отладки устройства. Для этого режима на языке C++ разработано программное средство, которое позволяет отображать текущие значения параметров тока и напряжения, а также графически отображать изменение этих параметров во времени. В качестве шаблона для проектирования программного средства был использован шаблон MVC – Model Control View. Данный шаблон подразумевает максимальное отделение бизнес логики программы от визуальной части. Управление бизнес логикой и визуальной частями осуществляется Control классом.

Приложение состоит из иерархии классов графических компонентов. Данные компоненты были созданы с использованием графической библиотеки Gdiplus, которая не является библиотекой по умолчанию и требует предварительной установки на компьютер.

Беспроводной способ использует Wifi модуль. В этом случае на стороне устройства разворачивается сервер, который предоставляет ряд операций для управления, отображения и логирования. В перспективе данный сервер сможет предоставлять весь этот функционал без требования специального программного обеспечения по протоколу HTTP, т.е. можно будет использовать браузер для отслеживания всех показателей.

Программное средство для микроконтроллера было выполнено на языке C в среде Kiel. В качестве отладчика использовался ST-LINK 2.

**Список использованных источников:**

1. Измеритель тока <http://www.alldatasheet.com>
2. Официальный сайт компании STM <http://www.stm-electronics.com>

## **ИССЛЕДОВАНИЕ И СРАВНЕНИЕ НЕЙРОННЫХ СЕТЕЙ ДЛЯ КЛАССИФИКАЦИИ ТЕКСТА**

*Юргель В.Ю.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Серебряная Л.В. – к.т.н., доцент*

Количество и объем существующих текстов очень велик и постоянно увеличивается, особенно благодаря развитию интернет-технологий, в связи с чем значительно усложняется их ручная классификация, поэтому возникает необходимость использования различных техник и методов для автоматической классификации текста. В этой статье рассматриваются три вида нейронных сетей для проведения классификации текста: сверточная нейронная сеть, рекуррентная нейронная сеть и иерархическая сеть внимания, а также их сравнение.

Одной из широко используемых задач обработки естественного языка в различных бизнес-задачах является классификация текста, что и есть пример контролируемого машинного обучения, поскольку для обучения классификатора используется маркированный набор данных, содержащий текстовые документы и их метки.

Целью классификации текста является автоматическая классификация текстовых документов по одной или нескольким predetermined категориям.

Некоторые примеры классификации текста: понимание настроения аудитории из социальных сетей; обнаружение спама; автоматическая пометка запросов клиента; категоризация новостных статей на predetermined темы и т.д.

В данном тезисе будут рассмотрены несколько подходов (методов) для классификации текста, их анализ и сравнение производительности.

Комплексный конвейер классификации текста состоит из следующих компонентов:

- 1) обучаемый текст - это исходный текст, с помощью которого контролируемая модель обучения может изучить и предсказать необходимый класс;
- 2) вектор объектов - это вектор, который содержит информацию, описывающую характеристики входных данных;
- 3) метки - это предопределенные классы/категории, которые должна предсказать модель;
- 4) алгоритм, с помощью которого модель может работать с классификацией текста;
- 5) предсказательная модель - модель, которая обучается на историческом наборе данных, которая может выполнять предсказания меток.

*Сверточная нейронная сеть (СНН, англ. CNN - Convolutional Neural Networks)* - это класс глубоких искусственных нейронных сетей с прямой связью (где соединение между узлами не образуют цикл) и используют разновидность многослойных персептронов, разработанных для минимальной предварительной обработки [1]. Такая нейронная сеть обычно используется в компьютерном зрении, однако недавно была применена к различным задачам обработки естественного языка (NLP), и результаты были многообещающими.

Кратко рассмотрим, что происходит, когда CNN используется для текстовых данных. Результат каждой свертки срабатывает при обнаружении определенного шаблона. Изменение размера ядер и объединяя их выходные данные позволяет обнаружить шаблоны множественных размеров (2, 3 или 5 смежных слов). Шаблонами могут быть выражения, такие как «я люблю», «я ненавижу» или «очень хорошо», что позволяет CNN идентифицировать их в предложении независимо от их положения.

*Рекуррентная нейронная сеть (РНН, англ. RNN - Recurrent Neural Network)* - это класс искусственной нейронной сети, где соединения между узлами образуют направленный граф по длине последовательности. Это позволяет демонстрировать динамическое временное поведение для временной последовательности. Другими словами, RNN - это последовательность блоков нейронной сети, которые связаны друг с другом как цепочка. Каждый передает сообщение приемнику.

RNN может показаться довольно сложной, но, тем не менее, она довольно интересна для изучения. Она инкапсулирует очень красивый дизайн, который преодолевает недостатки традиционных нейронных сетей, которые возникают при работе с данными последовательности: текст, временные ряды, видео, последовательности ДНК и т. д.

*Иерархическая сеть внимания (ИСВ, HAN - Hierarchical Attention Network)*. HAN спроектирована таким образом, чтобы уловить две основные идеи о структуре документов.

Первая - поскольку документы имеют иерархическую структуру (слова формирует предложения, а предложения, тем самым, формируют сам документ), мы аналогичным образом строим представление о документе, сначала выстраивая представлений предложений и, затем, агрегируем их в представление документа. Вторая - отмечается, что разные слова и предложения в документах являются дифференцированно информативными. Более того, важность слов и предложений очень сильно зависят от текущего контекста документа, т.е. одни и те же слова или предложения могут иметь разную важность в разных контекстах.

Иерархическая сеть внимания обладает следующими двумя характеристиками:

- 1) имеет иерархическую структуру, которая отражает иерархическую структуру документов;
- 2) присутствует два уровня внимания, применяемые на уровне слов и предложений, что позволяет по-разному относиться к более и менее важному контенту при построении представления документа [2].

Для исследования выбрано три типа наборов с различными вариациями классов (таблица 1).

Таблица 1 - Представление наборов для исследования

Наборы	Размер	Классы	Обучающие/проверяемые образцы
Набор 1	18322	362	14658/3664
Набор 2	3155	29	2524/631
Набор 3	191	17	163/38

**Результаты:**

Таблица 2 - Результаты исследования

Наборы	Алгоритм	Время	Точность обучения	Точность проверки
Набор 1	CNN	48 сек.	96.12	94.14
Набор 2	RNN	170 сек.	95.76	95.24
Набор 3	HAN	240 сек.	95.6	95.24
Набор 1	CNN	360 сек.	87.88	83.76
Набор 2	RNN	1800 сек.	82.63	81.11
Набор 3	HAN	1440 сек.	87.91	85.89
Набор 1	CNN	5 сек.	91.5	94.73
Набор 2	RNN	150 сек.	89.54	92.1
Набор 3	HAN	30 сек.	92.16	86.84

**Выводы:**

- 1) CNN достигла хорошей точности проверки с высокой согласованностью;
- 2) RNN оказалась худшим алгоритмом для реализации готовых сценариев;
- 3) CNN превзошла две другие модели (RNN и HAN) по времени обучения, однако RNN и HAN работает лучше при больших наборах данных;
- 4) Для набора 1 и набора 2, где обучающих выборок больше, HAN достигла наилучшей точности проверки. А что касается набора 3, где обучающих выборок мало, HAN показала противоположные результаты;
- 5) Когда обучаемых образцов меньше (например, в наборе под номером 3), сверточная нейронная сеть достигает лучшей точности проверки.

**Список использованных источников:**

1. Goodfellow I., Deep Learning / I. Goodfellow, Y. Bengio, A. Courville - The MIT Press, 2016. - 775 S.
2. Yang Z. Hierarchical Attention Networks for Document Classification / Z. Yang, D. Yang, H. Xiaodong, A. Smola, E. Hovy // Carnegie Mellon University. - 2015. - Vol. 12. - P. 1-8

## ВЕБ-ОРИЕНТИРОВАННОЕ ПС УДАЛЕННОГО УПРАВЛЕНИЯ ОСВЕЩЕНИЕМ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ RNP

*Ясюченя А.А.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Болтак С.В. – ассистент*

В настоящее время многие компании стараются оптимизировать расходы и затраты всех видов. Интеллектуальная система управления освещением позволяет рационально использовать энергетические ресурсы, контролировать и удалённо управлять освещением, поддерживать комфортные условия и значительно снижать эксплуатационные издержки.

Система предназначена для централизованного автоматического и оперативного диспетчерского управления освещением любых объектов или улиц.

Управление светильниками осуществляется с помощью беспроводной технологии передачи данных малого радиуса действия стандарта IEEE 802.15 по протоколу Zigbee. Технология позволяет создавать самоорганизующиеся беспроводные сети с автоматической ретрансляцией сообщений. Расстояния между узлами сети составляют десятки метров при работе внутри помещения и сотни метров на открытом пространстве. За счёт ретрансляций зона покрытия сети может значительно увеличиваться.

Дальность передачи данных в такой сети повышается без увеличения мощности передатчика – часть узлов сети способна маршрутизировать пакеты данных от отправителя к получателю по кратчайшему пути, что позволяет при относительно небольших скоростях передачи данных обеспечивать гарантированную доставку пакетов и защиту передаваемой информации.

Вот некоторые преимущества удалённого управления освещением:

- диспетчеризация управления с возможностью вывода на пульт оператора;
- индивидуальное и групповое удалённое управление;
- идентификация неисправностей онлайн;
- оптимизация освещения в зависимости от технологических процессов и режима работы;
- формирование статистических отчётов;
- отображение светильников на карте по координатам GPS.

В разрабатываемом приложении учтено большинство из преимуществ управления освещением. Основным преимуществом системы управления освещением над автономным управлением освещения или над обычным ручным переключением света является способность контролировать отдельные световые приборы или группы приборов из единого пользовательского интерфейса устройства. Возможность одновременно контролировать несколько источников света из одного устройства позволяет создать нужную световую атмосферу, в зависимости от предназначения помещения в тот или иной период времени.

При разработке приложения был использован следующий стек технологии: PHP, JavaScript, jQuery, MySQL.

Клиентская часть приложения была построена с использованием библиотеки JavaScript - jQuery, которая фокусируется на взаимодействии с HTML, помогает легко получать доступ к любому элементу DOM, обращаться к атрибутам и содержимому элементов DOM, манипулировать ими. Также библиотека jQuery предоставляет удобный API для работы с AJAX.

Серверная часть приложения основывается на PHP. Главным фактором языка PHP является практичность. PHP имеет открытый исходный код и распространяется по GPL лицензии. Он позволяет довольно быстро создавать прототипы и готовые сайты. Ему присущи такие встроенные функции, как использование запросов GET и POST, работа с HTML и URL. Благодаря хорошей читабельности, менять и модифицировать код на PHP не составляет труда. Как следствие — гораздо меньше проблем с поддержкой и обновлениями. Проекты легко и быстро подстраиваются под новые приложения, которые появляются на рынке, а также под новые бизнес-требования [1].

В качестве системы управления базами данных выбрана MySQL. Благодаря внутреннему механизму многопоточности быстрое действие MySQL весьма высоко. Также система обладает практически всем необходимым инструментарием, который может понадобиться в реализации практически любого проекта

**Список использованных источников:**

1. Мэтт Зандстра, PHP: объекты, шаблоны и методики программирования», 2015. — С.430. — ISBN 978-1-4302-0467-1.

**СЕКЦИЯ «ВЫСШАЯ МАТЕМАТИКА»**

## МАТЕМАТИЧЕСКИЕ МОДЕЛИ ТЕОРИИ ИГР

Гурец Н.А., Тарасевич И.А.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Малышева О. Н. – кандидат физ.-мат. наук, доцент

### Техасский холдем

Техасский холдем является самой популярной разновидностью игры покер в мире. Очевидно, что блеф является одной из важнейших стратегий этой игры. Хороший игрок использует этот стратегический прием, но не слишком часто. Как же понять, когда игроку стоит блефовать, а когда – нет?

Объектом данного исследования является блеф с математической точки зрения. Целью исследования является выбор оптимальной стратегии, приводящей к выигрышу. Нами рассматривалась ситуация, когда комбинация противника в конкретной игре лучше нашей собственной, причем вероятность этого события равна 0.75, то есть  $(1 - k)$ , где  $k$  – вероятность того, что у нас лучшая рука в игре.

Анализируя карточную игру «Техасский холдем» необходимо учитывать случаи, когда игрок блефует (или нет), а также случаи ответа его оппонента. Введем переменные  $x$  и  $y$  – величины частоты ответа со стороны оппонента и блефа со стороны игрока соответственно. В базовых случаях, когда  $x$  и  $y$  равны 0 или 1, игрок либо достоверно проигрывает (имеет худший расклад), либо ничего не получает, либо выигрывает банк (имеет лучший расклад). Также нами учитывалось пересечение возможных стратегий обоих игроков.

В результате оказалось, что существуют такие значения  $x$  и  $y$ , при которых независимо от стратегии оппонента вероятность выигрыша игрока остается неизменной (оптимальное значение  $x$ ), и наоборот, вне зависимости от выбора стратегии игрока оппонент имеет одинаковую вероятность выигрыша (оптимальное значение  $y$ ). Профиль таких стратегий является равновесием Нэша [1].

Для вывода оптимальных значений  $x$  и  $y$  решается система модификаций уравнения вероятности выигрыша в базовых случаях: в вероятностное уравнение подставляются значения  $x$  и  $y$  из множества  $\{0; 1\}$ . Было установлено, что оптимальное значение  $y$  зависит только от суммы ставки и текущего банка, в то время как оптимальная вероятность  $x$  зависит от вероятности наличия лучшей комбинации карт у игрока. В таблице 1 приведен расчет вероятностей выигрыша игрока и оппонента во всех случаях, возникающих во время игры.

Таблица 1. Обобщенные вероятности выигрыша (проигрыша)

	Стратегия 1 (игрок никогда не блефует)	Стратегия 2 (игрок всегда блефует)	Стратегия 3 (игрок блефует с оптимальной частотой)
Стратегия 1 (оппонент никогда не отвечает)	0.25	1	0.375
Стратегия 2 (оппонент всегда отвечает)	0.5	-0.6	0.375
Стратегия 3 (оппонент отвечает с оптимальной частотой)	0.375	37.5	0.375

Отрицательное значение вероятности означает проигрыш.

Нетрудно заметить, что при соблюдении обоими игроками стратегий, равновесных по Нэшу, вероятность выигрыша игрока стабильно равна 0.375. В то же время стратегия абсолютного выигрыша игрока (1) не может быть получена. Оптимальные значения  $x$  и  $y$  позволяют стабилизировать вероятность выигрыша или поддерживать ситуацию нахождения в игре.

## Настольная игра «Доббль»

«Доббль» - занимательная настольная игра. В комплекте игры содержится 55 карточек с 8 разными элементами на каждой. Любые две карточки имеют только один одинаковый элемент. Цель игры – найти на своей карточке элемент совпадающий с элементом на карточке соперника. Основания конечной и проективной геометрии позволяют построить математическую модель данной игры.

Целью данного исследования является проверка того является ли существующий набор карточек в игре «Доббль» полным и разработка возможных вариантов формирования игр-клонов.

По условию игры «Доббль» любая пара карточек имеет только один совпадающий элемент. Данное утверждение схоже с аксиомой о пересекающихся прямых на евклидовой плоскости. В плоскости Евклида через точку, не лежащую на заданной прямой, можно провести бесконечно много пересекающих её прямых и лишь одну прямую, параллельную данной. В свою очередь, на проективной плоскости любые две прямые всегда имеют общую точку. Тогда пусть один элемент на карточке есть прямая на проективной плоскости, а карточка – точка проективной плоскости.

Представим проективную плоскость как пучок прямых в трёхмерном пространстве, проходящих через начало координат. Построим произвольную плоскость  $\alpha$ , содержащую точку  $(0,0,0)$  и проведем нормаль к плоскости  $\alpha$  в этой точке. Через произвольную точку этой нормали, не совпадающую с точкой  $(0,0,0)$ , проведём плоскость  $\beta$ , параллельную  $\alpha$  (рисунок 1).

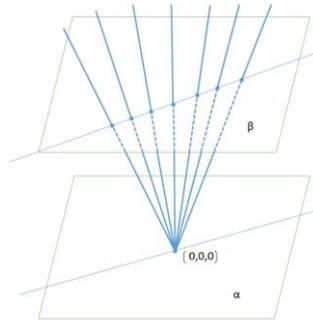


Рисунок 1 – Осуществление перехода к проективной плоскости

Точка проективной плоскости есть прямая в трёхмерном евклидовом пространстве, тогда прямая проективной плоскости – это плоскость трехмерного евклидова пространства.

В игре «Доббль» на одной карточке находится 8 элементов. Будем использовать проективную плоскость 7 порядка, где каждая прямая содержит 8 точек (одна из которых бесконечность). Остатки от деления на простое число 7 образуют поле. Построим аффинную конечную плоскость 7 порядка и достроим её до проективной. В общем случае, конечная аффинная плоскость порядка  $n$  имеет  $n^2$  точек. Каждая прямая такой плоскости состоит из 7 точек, координаты которых принадлежат полю остатков от деления на простое число 7.

Затем построим трёхмерное пространство над полем из 7 элементов. В таком пространстве существует  $7^3$  точек. Каждая прямая проходит через некоторую точку с координатами  $(0,0,0)$ , тогда чтобы задать прямую в трехмерном пространстве (точку на проективной плоскости) надо знать 6 ее остальных координат. В конечной проективной плоскости 7 порядка число прямых и точек равно между собой. Тогда максимальное число прямых и точек, которые можно построить над полем из 7 элементов будет равно:

$$N = \frac{7^3 - 1}{6} = 57,$$

что совпадает с количеством карточек в полном наборе игры «Доббль».

Проведённые исследования показывают, что официальная игра «Доббль» содержит неполное количество карточек (двух недостаёт). Нами были найдены 1287 вариантов построения недостающих карточек по следующему алгоритму:

- 1) Строим граф, ребра которого будут прямыми проективной плоскости, а его вершины – точки пересечения прямых этой плоскости [2];
- 2) Для этого графа создаём матрицу инцидентности, в которой количество столбцов будет равно количеству карточек, а количество строк – количеству элементов, формирующих игру;
- 3) Используя матрицу инцидентности, находим недостающие карточки методом перебора всех элементов.

Для создания шаблонов клон-игр (шаблонов игр с другим количеством элементов на карточке) были использованы проективные плоскости больших и меньших порядков. Получены варианты игр с 2, 3, 4, 5, 7, 8, 9, 11, 13 элементами на карточке.

**Список использованных источников:**

1. Джеймс С. Коулман. Экономическая социология с точки зрения теории рационального выбора | Экономическая социология – 2004. – Т. 5, №3. – стр. 35-44.
2. [Электронный ресурс]: Математическая модель игры Доббль / Хабр - Habr. URL: <https://habr.com/ru/post/437140/>

## ТАЙНЫ СУДОКУ И НЕ ТОЛЬКО

*Карымов А.Г., Якимович А.А., Зенькевич И.Н.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Борисенко О.Ф. – к.ф-м.н., доцент*

В данной статье мы рассматриваем алгоритм генерации Судoku и основные методы решения полученных головоломок с частотой их использования.

### Алгоритм генерации Судoku

Для рассмотрения методов решения Судoku введём следующие понятия:

- Малый квадрат. Это квадрат размером 3x3. Верхние строки таких квадратов A, D, G, а левые столбцы 1, 4, 7.
  - Район. Это 3 рядом стоящих малых квадрата по вертикали или по горизонтали.
1. Заполним первый горизонтальный район. Для этого в первой строке последовательно разместим числа 1-9, две оставшиеся строки верхнего района заполним, сдвинув строку находящуюся выше на 3 позиции влево, т.е. расположив числа 4-9, 1-3 и 7-9, 1-6 во вторую и третью строку первого района соответственно. Следующие районы заполняются смещением на 1 позицию влево предыдущего района. Таким образом, мы получаем некоторую базовую сетку.
  2. Введем некоторые действия, которые не изменят правильность заполнения сетки Судoku :
    - Транспонирование сетки Судoku;
    - Перестановка двух строк в одном районе;
    - Перестановка двух столбцов в одном районе;
    - Перестановка двух районов по горизонтали;
    - Перестановка двух районов по вертикали;
    - Перестановка двух цифр по всей сетке Судoku.
  3. Преобразовав базовую сетку Судoku нам необходимо создать головоломку. Какое количество подсказок следует убрать, чтобы обеспечить единственность решения? Это один из важнейших факторов, который влияет на сложность Судoku. Так как проектом [Sudoku@vtaiwan](mailto:Sudoku@vtaiwan) было доказано, что решаемых Судoku с 16 подсказками не существует, то минимальное число подсказок – 17. При этом в горизонтальном районе не должно оставаться двух пустых строк, а вертикальном – столбцов. Так же очевидно, что среди подсказок должны быть хотя бы 8 различных цифр. Далее, удаляя каждую новую клетку следует проверить полученную головоломку на единственность решения.

### Методы решения судoku

Для рассмотрения методов решения Судoku введём следующие понятия:

- Кандидат. Это возможное значение для определённой ячейки на данном этапе решения. Рассмотрим некоторые методы решения Судoku, использованные в нашей работе:

#### «Последний герой»

Данный метод используется при рассмотрении малых квадратов и заключается в следующем: если у нас есть некоторое число  $n$ , стоящее в строке или столбце поля, и эта строка или столбец в малом квадрате пересекает все пустые ячейки за исключением одной, то в оставшейся непересеченной ячейке находится именно это число  $n$  ( если в малом квадрате этого числа нет ).

Данный метод аналогичным образом можно использовать для строк и столбцов всей таблицы (рис. 1).

Рисунок 1. Пример применение метода “Последний герой”

Рисунок 2. Пример применение метода “Голые пары”

### «Голые» пары

“Голая пара” или “чистая пара” базируется на рассмотрении двух ячеек в одной общей строке, столбце или малом квадрате. Если у нас есть по два кандидата  $a$  и  $b$  ( числа, которые возможно могут стать в пустую ячейку ) в двух ячейка одной строки, столбца или малого квадрата, то мы можем вычеркнуть все  $a$  и  $b$  из этой строки, столбца или малого квадрата, так как данные числа точно будут находиться в найденных ячейка содержащих «Голую» пару. Метод голых пар можно расширить до голых троек. Тройкой считаем комбинацию из трех клеток в трех ячейках, причем в каждой ячейке может быть как по два кандидата, так и по три, например  $\{ab\}$ ,  $\{ac\}$ ,  $\{bc\}$  или  $\{abc\}$ ,  $\{bc\}$ ,  $\{ac\}$ . В примере пары найдены в первой и третьей строках (рис. 2).

### Скрытые пары

Иногда для выхода из тупиковых ситуаций применяется метод поиска *скрытых пар*. При нахождении скрытой пары можно вычеркнуть всех остальных кандидатов из ячеек содержащих пару. В данном примере (рис. 3) видно, что “6” и “7” есть в двух ячейках в правом верхнем углу, причем они больше не встречаются в правом верхнем квадрате. Очевидно, что в **A8** и **A9** могут содержать только эти числа, тогда всех остальных кандидатов мы можем убрать. Ограничивающей областью может выступать не только малый квадрат, но и строка, и столбец.

### Указывающие пары и тройки

Если в малом квадрате (строке или столбце) один кандидат встречается лишь в двух ячейках (т. е. образуют *указывающую пару*) одной строки или столбца (одного малого квадрата), то мы можем убрать этого кандидата из всех остальных ячеек строки или столбца (малого квадрата) содержащей *указывающую пару*. Для примера рассмотрим данную головоломку (рис. 4). В правом верхнем малом квадрате, “3” имеется только в ячейках **B7** и **B9**. Следовательно, мы убираем “3” из **B1**, **B2**, **B3**. Аналогичные рассуждения верны и для “2” из центрального нижнего малого квадрата, убираем кандидата из **G2**.

Рисунок 3. Пример применение метода “Скрытые пары”

Рисунок 4. Пример применение метода “Указывающие пары и тройки”

### Статистические данные

Эмпирически нами были получены данные о частоте встречаемости методов.

Для решения Судоку с количеством подсказок большим, чем 25 использовался в основном метод «*Последний герой*».

Однако при решении Судоку с количеством подсказок меньшим, чем 25, имеет место следующая статистика:

- «*Последний герой*» использовался в 39,19% случаев.
- «*Голая пара*» использовался в 16,22% случаев.
- Указывающая пара* использовался в 39,86% случаев.
- Скрытая пара* использовался в 3,03% случаев.
- Остальные методы* использовались в 1,7% случаев.

В данной работе был рассмотрен алгоритм генерации Судоку, а также методы решения полученных головоломок. В дальнейшем планируется улучшить алгоритм генерации.

**СЕКЦИЯ «ИНФОРМАТИКА»**

## ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИЙ BIG DATA В ТРАНСПОРТНОЙ СФЕРЕ

*Александров А.А., Пилецкий И.И.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Пилецкий И.И. – канд. физ.-мат. наук, доцент*

Данная работа содержит сведения о некоторых аспектах использования технологий Big Data в транспортной сфере, таких как предиктивный анализ поломок транспортных средств, вопросы доставки больших массивов данных с подвижного состава. Рассмотрены некоторые варианты направлений для дальнейшего исследования применения технологий Big Data.

В настоящее время происходит повсеместное внедрение и использование технологий Big Data в различных сферах жизни, связанных с появлением технологических возможностей анализировать огромные массивы данных в некоторых проблемных областях. Эти технологии не обошли стороной и транспортную сферу.

Одной из проблем использования технологий Big Data в сфере транспорта является проблема доставки больших массивов информации в центры обработки данных. Если для городского транспорта, где покрытие сетями мобильных сотовых операторов приближается к 100%, проблем в отправке больших объемов данных нет (данные можно накапливать на борту транспортного средства и передавать их в другое время, например, когда транспорт находится в парке), то для такого транспорта, как грузовые автомобили, поезда, грузовые самолёты, есть определенная проблема в доставке больших объемов данных. Одним из вариантов решения проблемы передачи данных между базовой станцией оператора сотовой связи и установленным оборудованием в условиях плохого покрытия, в блоке передачи можно предусмотреть «буферизацию» (запись) данных во внутреннюю память. В момент появления устойчивой связи с качественным сигналом «буферизованные» данные передаются на сервер в автоматическом режиме.

Поскольку для достижения заданной точности необходима высокая частота дискретизации снимаемых параметров (таких, как мировые координаты, моментальный расход топлива, состояние транспортного средства, состояние агрегатов транспортного средства и др.), существует проблема разработки сложных алгоритмов сжатия и регулярной доставки этой информации или, когда появляется стабильный канал связи.

Какую информацию можно собирать? В первую очередь это точные координаты с привязкой ко времени. На основании этой информации можно производить планирование и составление точных графиков и расписаний движения транспорта. Как пример, в Минске в качестве пилотного проекта была реализована система управления светофором на перекрестке улиц Козлова и Платонова на базе RFID-меток. Данную систему можно усовершенствовать, используя телеметрию с трамвая для упреждающего управления светофором. Система может подстроить работу светофора таким образом, чтобы минимизировать задержку трамвая на перекрестке и переходить к следующей фазе работы светофора сразу после прохода трамвая через перекресток.

Для локомотивов минимально необходимый набор передаваемых данных следующий: гео-координаты нахождения локомотива и точное время; серия, номер и депо приписки локомотива, на котором установлено оборудование; скорость перемещения локомотива.

Проблемы, которые могут быть решены при внедрении технологий Big Data для системы железнодорожного транспорта, следующие: анализ спроса погрузки и доставки груза, анализ пропускной железнодорожной сети, определение маршрутов, составление расписания и регулирование графиков движения поездов, контроль состояния подвижного состава (ПС): локомотивов, вагонов, и диагностирование поломок ПС, увеличение пропускной способности существующих линий за счет точных знаний координат ПС, сокращение дистанции между ПС, увеличение скорости движения, решение задач транспортной логистики (в том числе и предоставление полной информации о грузе и его транспортировке грузовладельцам) и др. [1].

Сбор телеметрической информации текущего состояния двигателя позволяет принимать как оперативные, так и стратегические решения. Применительно к автобусному транспорту, такой информацией являются данные о моментальном расходе топлива, оборотах двигателя, текущая передача, нагрузка на ось и т.д. Обычно анализируют лишь данные ошибок двигателя, игнорируя походную информацию. Однако, при фиксации этих данных с высокой частотой дискретизации, можно провести анализ этих данных и выявить сложные режимы работы двигателя, оптимизировать конфигурацию двигателя для минимизации расхода топлива и обеспечения оптимального режима работы двигателя. Также эти данные могут помочь для расчета текущего пассажиропотока на основании данных о текущей нагрузке на ось и выявлять высоконагруженные маршруты с дальнейшим принятием решений о корректировке графика движения и увеличения числа единиц

подвижного состава на данном маршруте.

Малоизученной сферой применения Big Data и предиктивной аналитики на транспорте является предсказание неисправностей агрегатов и механизмов на основании данных объективного контроля и вибромониторинга. Собрав большую базу штампов работы агрегатов, можно с высокой долей вероятности фиксировать аномалии работы двигателя и предугадывать выход из строя того или иного агрегата. Учитывая большие вычислительные мощности ЦОД и развитие технологий обработки такого рода данных можно получить информацию об отказе агрегата с точностью до движущейся детали. Объёмы сырых данных исчисляются гигабайтами в час, поэтому есть необходимость в разработке алгоритмов, позволяющих оптимизировать передаваемый поток сырых данных с датчиков таким образом, чтобы не потерять в достоверности передаваемых выборок и не потерять важную информацию при сжатии. Так, например, в 2015 только локомотивов на БелЖД было 920 единиц, при активном использовании только 70% прописного парка, передаче только минимальных данных о локомотиве (100 байт), каждые 10 секунд объем передаваемых данных будет не менее 556.041.600 байт в сутки. Здесь не учтен большой объём данных о работе агрегатов локомотива и расходе топлива, данные о составе поезда, данные о бригаде, а также обратно передаваемые данные на локомотив.

Проанализировав вышесказанное, можно выделить несколько направлений для исследований:

- использование Big Data для решения проблемы составления расписаний и графиков движений транспорта (городского, дальнемагистрального, железнодорожного). Сюда же входит вопрос оптимизации расходов при эксплуатации транспорта с учетом рельефа местности (например, для грузовых составов поездов вес которых, может превышать 4000 тонн, весьма критичным вычисление текущей скорости состава с точностью до 0,1км/ч, на поворотах, подъемах/спусках, линейных участках, перед светофором, что может позволить сэкономить топливо/электроэнергию на разгон и торможение до 10-15%);
- исследование применения Big Data для предиктивной диагностики и мониторинга состояния транспорта и подвижного состава, разработка программно-аппаратных комплексов для предсказания выхода из строя тех или иных агрегатов, сбор и анализ больших массивов данных с физических датчиков вибрации, температуры и т.д., что позволит отказаться от плановых ремонтов ТО1, ТО2 и др., а перейти непосредственно к требуемому обслуживанию подвижного состава;
- исследование методов передачи больших массивов данных в условиях узкого канала, методики сжатия с учетом специфики данных, минимизацию задержек от отправки данных до получения обратного ответа с рекомендациями о параметрах движения, техобслуживания и т.д;

В рамках программы исследования загруженности городских автобусов и перспективы замены их на гибридные электробусы было разработано устройство, устанавливаемое в моторный отсек автобуса и подключаемое к внутренней CAN-шине контроллера двигателя Mercedes ADM3. Устройство было предназначено для непрерывного сбора информации с высокой частотой о работе двигателя с привязкой к текущему времени. Информация записывалась на SD-карту, раз в неделю инженеры производили считывание информации и передачу её для дальнейшего анализа в лабораторию. Устройство «каталось» на протяжении нескольких недель на автобусном маршруте №18 г. Минска. Результатом анализа полученного массива данных явилось решение о целесообразности перехода на гибридные двигательные установки. Решение основывалось на анализе режимов работы двигателя в городских условиях и нагрузки на двигатель, времени простоев на остановках, светофорах, динамики разгона-остановки и прочих факторов.

На данный момент ведется разработка прибора для сбора данных с транспортных средств, имеющий в себе канал связи на базе GSM-модема, привязку с текущим координатам посредством систем гео-позиционирования GPS-ГЛОНАСС, при этом имеющим минимальные габариты и возможность работать в агрессивных условиях машинных отделений. ЦОД лаборатории БГУИР-ИВА [2], может использоваться как база в пилот проектах для хранения и обработки данных с датчиков.

Наиболее перспективным направлением является применение сервисов платформы IBM Cloud Platform и когнитивного суперкомпьютера IBM Watson [3, 4], возможности которого позволяют совместить обработки больших объёмов данных, получаемых с различных датчиков. IBM Cloud Platform предоставляет специальные модули для получения информации с IoT-устройств, к которым можно отнести разрабатываемый автономный прибор сбора данных.

Когнитивные системы, приложения и сервисы, аналитика (Watson), IoT (например, автомобили, локомотивы, полигон дороги), позволят с минимальными усилиями создать сеть взаимодействия M2M (Machine-to-Machine), что в будущем обеспечит переход к безлюдному производству.

**Список использованных источников:**

1. Пилецкий И. И. «Один из методов построения и модернизации корпоративных приложений» Материалы конференции - "Software Engineering Conference (Russia) SEC(R) 2007", Moscow, November 1-2, 2007.

2. И.И. Пилецкий и др. Виртуальная ИТ среда БГУИР для исследования Big Data и VCL, с. 21-32, BIG DATA and Predictive Analytics. Использование BIG DATA для оптимизации бизнеса и информационных технологий : сборник материалов междунар. науч.-практ. конф. / редкол. : М.П. Батура [и др.]. – Минск : БГУИР, 2015. – 220 с. ISBN 978-985-543-146-7. - С. 21-32.
3. What is the IBM Cloud platform? [Электронный ресурс] / IBM developerWorks. – 2017-2018. – Режим доступа: <https://console.bluemix.net/docs/overview/ibm-cloud-platform.html#whatis>. – Дата доступа: 19.03.2019.
4. IBM RCIS Watson Cloud Cognitive University [Электронный ресурс] / IBM Developer Works. – 2016-2019. – Режим доступа: <https://www.ibm.com/developerworks/community/groups/service/html/communitystart?communityUuid=bc004137-b64a-4378-ac02-2caf59c56c2a>. – Дата доступа: 19.03.2019.

## ПРОБЛЕМЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ОС WINDOWS

*Андрей Д. С., Кадушко А. А., Малиновская Е. Д.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Стройникова Е. Д. – ассистент кафедры информатики*

В процессе изучения предметной области была собрана и систематизирована информация о вредоносном программном обеспечении и информационных атаках. В результате была смоделирована и протестирована вредоносная программа, сокрытая в обыденном для обычного пользователя приложении.

В работе были изучены уязвимости ОС Windows относительно сохранности конфиденциальных и личных данных. Основной акцент был сделан на деятельности вредоносного программного обеспечения на примере вирусов, червей и троянов. Была приведена статистическая информация о деятельности вредоносного программного обеспечения с момента их возникновения. Продемонстрированы прецеденты наиболее масштабных и результативных вирусных атак за XX – XXI вв., таких как WannaCrypt0r, Jerusalem, CIH и подобных. Выведены критерии сравнения и классификации разного рода атак, изучаются их предпосылки, реализация, результаты и специфика. Рассмотрены особые беспрецедентные случаи осуществления атак (например, с использованием «умной» бытовой техники). Изучена также возможность осуществления DDoS-атак в социально-приемлемых целях. В процессе рассмотрения деятельности вредоносного программного обеспечения сделан упор на механизмы шифрования и распространения в прогностических интересах. Изучены способы сокрытия признаков деятельности вредоносного программного обеспечения (встраивание в код полезной программы, маскировка под неё и др.) с целью выявления специфики вредоносного программного обеспечения и планирования контрмер в виде разработки антивирусного программного обеспечения. В целом подход к изучению вредоносного ПО в данной работе использует концепцию «Знай врага в лицо».

В практической части продемонстрирована работа искусственно созданной программы, деятельность которой в не тестовом случае однозначно нежелательна. Пример представляет собой игру, запуск которой приводит к шифрованию файлов на жестком диске шифром Цезаря. Продемонстрирован механизм сокрытия, проникновения и шифрования самодельного трояна. Рассмотрены различные шифры (Цезаря, Виженера, многоалфавитная замена и др.), методы поиска ключей и дешифрования (на основе шифротекста, открытого текста и др.). Произведена классификация шифров, их сравнение, рассмотрено их использование в предметной деятельности. Изучены возможные исходы деятельности вредоносного программного обеспечения, способы восстановить утраченные или поврежденные данные. Предложены возможные меры профилактики утраты конфиденциальных данных. Спрогнозирована дальнейшая деятельность вредоносного программного обеспечения. Осуществлена рефлексия последующих цифровых эпидемий и их возможного исхода.

### **Список использованных источников:**

1. WannaCry 2.0: наглядное подтверждение того, что вам обязательно нужно правильное решение для надежного бэкапа [Электронный ресурс]. – 2017. – Режим доступа: <https://habr.com/ru/company/acronis/blog/328796/>.
2. Холодильник атакует: как киберпреступники используют бытовую технику [Электронный ресурс]. – 2016. – Режим доступа: [https://www.rbc.ru/technology\\_and\\_media/13/11/2016/5825cf889a79475b671ff971/](https://www.rbc.ru/technology_and_media/13/11/2016/5825cf889a79475b671ff971/).
3. Вредоносное ПО, вошедшее в историю. Часть II [Электронный ресурс]. – 2017. – Режим доступа: <https://habr.com/ru/company/ua-hosting/blog/407621/>.
4. Азбука безопасности [Электронный ресурс]. – 1998. – Режим доступа: <https://kaspersky.antivirus.lv/rus/threats/safetyabc/>.
5. Вирусы XXI века [Электронный ресурс]. – 2016. – Режим доступа: [https://geekbrains.ru/posts/xxi\\_viruses/](https://geekbrains.ru/posts/xxi_viruses/).
6. Самые разрушительные компьютерные вирусы начала XXI века [Электронный ресурс]. – 2018. – Режим доступа: [https://www.iguides.ru/main/other/samye\\_razrushitelnye\\_kompyuternye\\_virusy\\_nachala\\_xxi\\_veka/](https://www.iguides.ru/main/other/samye_razrushitelnye_kompyuternye_virusy_nachala_xxi_veka/).

7. Самые опасные вирусы за всю историю существования компьютеров [Электронный ресурс]. – 2013. – Режим доступа: <https://www.osp.ru/pcworld/2014/01/13038813/>.
8. Десять самых громких хакерских атак в истории интернета [Электронный ресурс]. – 2014. – Режим доступа: <https://republic.ru/posts/l/1139636/>.
9. Вирусы, статистика и немного всего [Электронный ресурс]. – 2017. – Режим доступа: <https://habr.com/ru/post/357426/>.
10. Новая работа для графических процессоров: GPU защитит от вирусных атак [Электронный ресурс]. – 2018. – Режим доступа: <https://habr.com/ru/company/1cloud/blog/354526/>.
11. «Разрубить Гордиев узел», или преодоление проблем шифрования информации в ОС Windows [Электронный ресурс]. – 2016. – Режим доступа: <https://habr.com/ru/company/aladdinrd/blog/304024/>.
12. Уязвимости SSD с аппаратным шифрованием позволяют злоумышленникам легко обходить защитные меры [Электронный ресурс]. – 2016. – Режим доступа: <https://habr.com/ru/post/428964/>.
13. Прозрачное шифрование: преимущества и недостатки [Электронный ресурс]. – 2015. – Режим доступа: <https://habr.com/ru/company/cybersafe/blog/251041/3/>.
14. Срыв масштабной хакерской атаки на пользователей Windows в России: часть 2 [Электронный ресурс]. – 2018. – Режим доступа: <https://habr.com/ru/company/microsoft/blog/351692/>.

## СРАВНЕНИЕ АЛГОРИТМОВ РЕШЕНИЯ ЗАДАЧИ КОММИВОЯЖЁРА

*Астрашаб В. В., Бондарев И. М., Клебанов Д. А.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Стройникова Е. Д. — ассистент кафедры информатики*

В докладе рассматривается одна из ключевых задач комбинаторной оптимизации — задача коммивояжёра. Формулируется суть задачи, рассматриваются различные алгоритмы её решения, а также анализируются и сравниваются результаты выполнения алгоритмов при различных наборах входных данных.

Задача коммивояжёра — одна из самых известных задач в комбинаторной оптимизации. Она заключается в следующем: «Зная список городов и их координаты, найдите самый короткий путь, проходящий через все города один раз и возвращающийся в исходный город».

Данная задача была впервые сформулирована в 1930 г. и до сих пор является одной из наиболее интенсивно изучаемых задач оптимизации. Ричард Карп в 1972 г. доказал NP-полноту задачи поиска гамильтоновых путей, из чего, благодаря полиномиальной сводимости, вытекала NP-трудность задачи коммивояжёра. На основе этих свойств им было приведено теоретическое обоснование сложности поиска решений задачи на практике.

Задача коммивояжёра используется в логистике, транспорте, проектировании различных коммуникационных сетей и трубопроводов, а также в других сферах.

Вместе с простотой определения и сравнительной простотой нахождения хороших решений задача коммивояжёра отличается тем, что поиск действительно оптимального пути является достаточно сложной задачей.

Существует множество различных способов решения данной задачи, которые можно разделить на несколько групп:

1. Точные алгоритмы — алгоритмы, которые находят гарантированно верное решение.

Примеры:

А. Полный перебор всех возможных путей. Сложность алгоритма  $O(N!)$ .

В. Алгоритм Хелда — Карпа — алгоритм динамического программирования. Сложность алгоритма  $O(N^2 * 2^n)$ .

С. Метод ветвей и границ — вариация полного перебора с отсевом подмножества заведомо неоптимальных решений.

Ключевое преимущество алгоритмов данной группы заключается в том, что они гарантированно находят верное решение, чего не позволяют сделать алгоритмы из других групп. Однако на практике эти алгоритмы почти никогда не применяются ввиду огромных временных затрат даже при небольших значениях  $N$ .

2. Эвристические алгоритмы — алгоритмы, которые не гарантируют точного решения, однако достаточны для решения поставленной задачи.

Примеры:

А. Деревянный алгоритм — алгоритм решения через построение кратчайшего остовного дерева. Сложность  $O(N^2)$ .

В. Алгоритм ближайшего соседа — жадный алгоритм, который строит путь, находя самую ближнюю к данной вершину. Сложность  $O(N^2)$ .

С. 2-opt алгоритм — алгоритм, сводящийся к удалению двух пересекающихся рёбер и вставке новых рёбер, не нарушающих корректности решения. Сложность  $O(N^2)$ .

К преимуществам данных алгоритмов можно отнести строгую зависимость скорости поиска решения от размера исходных данных.

К недостаткам данных алгоритмов можно отнести низкое качество ответа при увеличении количества городов.

3. Метаэвристические алгоритмы — обобщённые стратегии поиска оптимума в пространстве решений, зависящие от случайности.

Примеры:

A. Муравьиный алгоритм — алгоритм, имитирующий поведение муравьиной колонии, ищущей путь к источнику питания.

B. Имитация отжига — алгоритм, имитирующий физический процесс кристаллизации вещества при понижающейся температуре.

C. Генетический алгоритм — алгоритм, имитирующий эволюционный процесс в природе.

D. Алгоритм клонирующего отбора — разновидность генетического алгоритма, не использующая наследование от нескольких предков.

К преимуществам данных алгоритмов можно отнести простоту реализации, нахождение более оптимальных путей по сравнению с эвристическими алгоритмами.

К недостаткам данных алгоритмов можно отнести зависимость от гиперпараметров, которые подбираются индивидуально для различных наборов исходных данных, а также сложность асимптотического анализа из-за различающихся гиперпараметров.

Для сравнения алгоритмы различных типов были реализованы на языках программирования C++ и Python, а также сгенерированы различные наборы входных данных для тестирования. Полученные результаты были визуализированы с использованием библиотеки Matplotlib.

Анализ результатов тестирования:

1) метаэвристические алгоритмы работают значительно дольше эвристических;

2) пути, полученные эвристическими алгоритмами, длиннее путей, полученных метаэвристическими алгоритмами (рисунок 1);

3) при малых размерах исходных данных высока вероятность нахождения близких к оптимальному решений или же оптимального решения эвристическими и метаэвристическими алгоритмами.

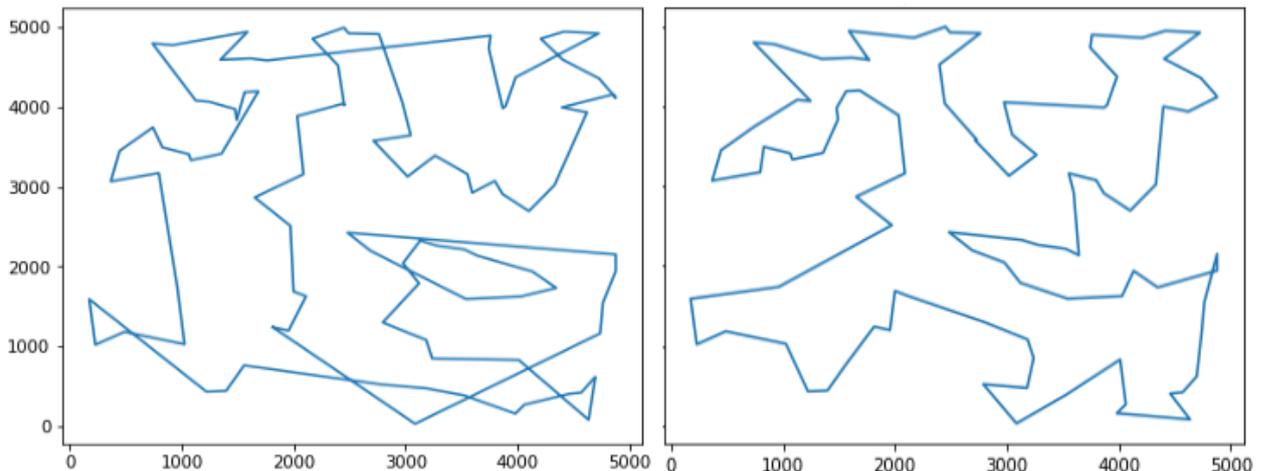


Рисунок 1 — Путь, полученный эвристическим алгоритмом ближайшего соседа (слева) и метаэвристическим алгоритмом имитации отжига (справа) для количества городов  $N = 100$ .

Длины путей соответственно равны 47745 и 38696.

Необходимость использования тех или иных подходов к решению задачи коммивояжёра обусловлена размером исходных данных, доступной производительностью, отведённым временем на выполнение и требуемой точностью результата. Например, в случае использования навигаторов необходима точность на небольшом размере исходных данных при низкой производительности и ограниченном времени. Здесь можно воспользоваться точными алгоритмами.

В случае подбора оптимальных маршрутов доставки товаров наилучшим вариантом является применение эвристических алгоритмов, отличающихся предсказуемостью скорости работы, отсутствием необходимости настройки гиперпараметров и неплохими результатами при невысоком размере исходных данных.

Однако уже при прокладке коммуникационных сетей необходима точность на большом количестве точек при наличии высокой производительности и достаточного времени. А это значит, что стоит обратить внимание на метаэвристические алгоритмы.

Разработанные программы, тесты и полученные результаты доступны по ссылкам:

- 1) <https://github.com/astraszab/TSP-algorithms>;
- 2) <https://github.com/dmitriyklebanov/tsp>.

**Список использованных источников:**

1. Алгоритмы. Построение и анализ / Т. Кормен [и др.]; пер. с англ. — 3-е изд. — М.: Вильямс, 2015. — 1328 с.
2. Скиена, С. Алгоритмы. Руководство по разработке / С. Скиена; пер. с англ. — 2-е изд. — СПб.: БХВ, 2017. — 760 с.

## МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ ИНДИВИДУАЛЬНЫХ ЗАНЯТИЙ СПОРТОМ

*Бастун А.Н.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Таборовец В.В. – к.т.н., доцент*

Рассматриваются вопросы создания мобильного приложения для фиксации и автоматического анализа результатов контроля данных о физическом состоянии здоровья человека с использованием платформы iOS.

В современном мире вопрос укрепления здоровья человека становится все более изученным и количество факторов влияющих на него становится все больше. У людей становится все больше носимых устройств, способных собирать данные о состоянии человека в течении дня, таких как телефоны, умные браслеты, часы и т.д. С помощью этих данных можно строить планы тренировок, рекомендации по питанию и т.д. Поскольку стоимость работы персонального тренера, диетолога или другого специалиста является существенной для большинства, либо у людей просто нет на это времени. Использование простого приложения для тренировок и отслеживания текущих показателей может быть большим плюсом в укреплении здоровья, развития общего физического состояния и т.д.

Цель разработки является создание мобильного приложения для платформы iOS, способного считывать данные о физическом состоянии пользователя и его активности и предлагать рекомендации в зависимости от целей пользователя. Данные будут считываться в зависимости от имеющихся устройств (телефон, часы и т.д.) и в зависимости от данных разрешений на использование пользователем. Различные международные исследования показали, что постоянная умеренная физическая нагрузка благотворно влияет на работу мозга и сердечно-сосудистой системы, сокращает риски возникновения различных заболеваний [2].

Основной целью приложения является возможность предоставить пользователю более легкий способ следить за своим физическим состоянием и получать максимальную пользу из физических нагрузок тратя минимальное количество времени. Для считывания данных пользователя будет использоваться нативный фреймворк HealthKit, который предоставляет из себя библиотеку различных показателей физического состояния и активностей [1]. Данный фреймворк находится в экосистеме iOS и дает возможность интеграции с различными носимыми устройствами, также содержит различную медицинскую информацию пользователя. Активно используется различными медицинскими учреждениями в Америке для постоянного наблюдения за пациентами [1, 3].

Данные, которые будет использовать приложение:

- 1) частота пульса;
- 2) количество шагов сделанных пользователем;
- 3) количество энергии использованной в состоянии покоя;
- 4) количество энергии потраченной на активную деятельность;
- 5) количество времени проведенное в движении (легкой прогулки или более интенсивно);
- 6) количество потребленной энергии из еды;
- 7) Опросы пользователей для получение различных данных, которые нет возможности получить с носимых устройств.

Анализ собранных данных производится на основе анализа интенсивности нагрузки (частота пульса и скорость восстановления пульса до нормальных значений), объема нагрузки (количество сделанных шагов, количество времени и энергии потраченное на активную деятельность) и различные опросы и тесты, сделанные пользователем на основе опросников.

Приложение реализует возможность просмотра предлагаемых рекомендаций, выбор и установку целей, личных параметров и других дополнительных параметров, просмотр статистики.

**Список использованных источников:**

1. HealthKit Developer guidance [Электронный ресурс]. - Электронные данные. Режим доступа: <https://developer.apple.com/documentation/healthkit>
2. Harvard Health Publishing [Электронный ресурс]. - Электронные данные. Режим доступа: <https://www.health.harvard.edu>
3. Human Interface Guidelines [Электронный ресурс]. - Электронные данные. Режим доступа: <https://developer.apple.com/design/human-interface-guidelines/>

## БАЛАНСИРОВКА НАГРУЗКИ МЕЖДУ УЗЛАМИ ПРИ ПОСТРОЕНИИ РАСПРЕДЕЛЕННЫХ СИСТЕМ

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Вабищевич Е. Г..

Таборовец В.В. – к.т.н., доцент

Работа посвящена вопросам исследования существующих способов построения распределенных систем с целью создания базовой архитектуры распределенной системы обработки запросов.

Распределенная система — это набор независимых компьютеров, представляющий их пользователям единой объединенной системой [1]. Среди эмпирических свойств распределенной системы можно выделить следующие:

- 1) устойчивость системы при выходе из строя одного или нескольких элементов;
- 2) возможность изменения конфигурации системы в процессе работы;
- 3) отсутствие полных знаний о системе в одном конкретном элементе системы.

В процессе исследования распределенных систем были рассмотрены следующие архитектуры: архитектура клиент-сервер, архитектура peer to peer, кластерная архитектура.

Кластеры распределения нагрузки – способ организации программного обеспечения и системного оборудования, при котором происходит распределение запросов через один или несколько входных узлов. После этого из входных узлов запросы перенаправляются на обработку в вычислительные узлы. Данный подход обладает следующими преимуществами:

- возможность горизонтального масштабирования;
- равномерное распределение нагрузки между узлами;
- обеспечение высокой доступности системы. Данный пункт достигается благодаря возможностям первых двух пунктов.

В результате рассмотрения принципов построения распределенных систем была разработана следующая базовая архитектура.

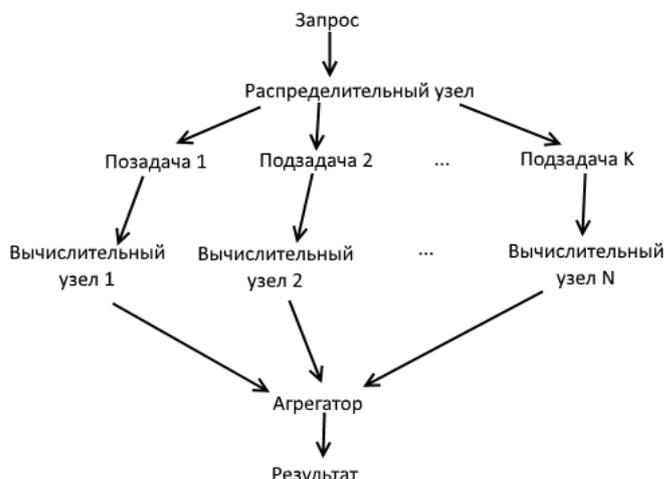


Рис. 1 - Базовая архитектура распределенной системы

Описание рисунка 1: в систему приходит запрос → запрос обрабатывается распределительным узлом систем, разбивается на подзадачи меньшего масштаба → каждая из подзадач отправляется на один из вычислительных узлов → вычислительный узел выполняет обработку подзадачи; результат выполнения отправляется агрегатору → агрегатор дожидается обработки всех подзадач запроса, собирает из них итоговый результат, отдает результат клиенту.

**Список использованных источников:**

1. Э. Таненбаум, М. Ван Стеен. Распределенные системы. Принципы и парадигмы // СПб.: Питер, 2003. 877 с.

## ИСПОЛЬЗОВАНИЕ ЛЕКСИЧЕСКОЙ ОБФУСКАЦИИ VHDL-ОПИСАНИЙ ДЛЯ ВНЕДРЕНИЯ ВОДЯНЫХ ЗНАКОВ.

*Видничук В.Н.*

*Белорусский государственный университет информатики и радиоэлектроники*

*г. Минск, Республика Беларусь*

*Иванюк А.А. – д.т.н., доцент*

В статье описывается метод лексической обфускации VHDL- описаний, его основные положения а также метод внедрения водяных знаков в это описание. Демонстрируются множества символов используемых для обфускации описаний.

В наши дни быстрыми темпами растёт производство цифровой техники и в связи с этим растёт проблема пиратства. Для решения этой проблемы предлагается рассмотреть метод внедрения водяного знака в лексическую обфускацию VHDL-описаний.

Обфускация (от англ. obfuscate – делать неочевидным, запутанным, сбивать с толку) – широко известная методика защиты исходных кодов программ от обратного проектирования [1].

Основной целью лексической обфускации является затруднение понимания исходных кодов программы. Лексическая обфускация это один из многих способов затруднить понимания функционирования программы.

Частным случаем лексической обфускации является переименование идентификаторов VHDL-описания на похожие друг на друга. Название идентификатора можно представить в виде множества

символов:

$$\langle Id \rangle = \{ [A..z] \cap [ \text{..} ] \cap [0..9] \}$$

где, идентификатор не должен начинаться с цифры и может состоять из латинских и кириллических символов. В связи с этим набор символов при выполнении метода переименования идентификаторов можно представить следующим множеством:

$$\langle Id' \rangle = \{ O, O', 0, Q, 1, l, I, j \},$$

в котором  $O'$  является кириллическим символом  $O$  и  $O$  латинская являются разными символами, однако в латинском языке могут использоваться диакрические знаки например:  $\acute{I}$

$$\langle Di \rangle = \{ \check{O}, \check{O}', \acute{I}, \acute{I}', \grave{I}, \grave{I}' \},$$

при синтезе VHDL-описания данные символы

приводятся к латинской  $O$ , однако имеют разные коды символов, поэтому при выполнении анализа лексически обфусцированного кода могут возникать ошибки и непонятности. Отсюда, следует возможность использования диакрических символов для формирования названий идентификаторов повышенной сложности.

Далее приведён пример использования диакрических символов для затруднения понимания кода: идентификатор  $OIOOIIIOI\check{O}$  при синтезе будет равняться идентификатору  $OIOOIIIOIO$  или  $OIOOIIIOIO\check{O}$ , но при попытке найти идентификатор  $OIOOIIIOI\check{O}$  с помощью поиска или сторонних программ найдётся только он сам. Следовательно, с помощью использования данных символов можно запутывать не только человека, но и программные средства анализа кода или, например, идентификатор  $OIO'OIIIOI\check{O}$  и идентификатор  $OIOOIIIOI\check{O}$  для человека будут выглядеть как идентичные, однако при синтезе они будут являться разными идентификаторами.

При использовании метода переименования идентификаторов следует знать, что все идентификатора должны быть похожи и отличаться друг от друга на 1 символ, в связи с этим существует возможность внедрения водяного знака в VHDL-описание путём выбора мест замены символов в соответствии с ключом водяного знака.

Данный метод можно применять несколькими способами, например, генерируется первичный идентификатор для переименования, состоящий из нулей и единиц, символ выбирается с помощью равномерного распределения. В итоге получается идентификатор вида: 100101001011 это делается для того, чтобы расстояние между двумя идентификаторами не превышало 1. После этого



**Список использованных источников:**

1. Collberg, C. A Taxonomy of Obfuscating Transformations / C. Collberg, C. Thomborson, D. Low – Auckland: Department of Computer Science, 1997. – 36 p.
2. Сергейчик В. В. Методы лексической обфускации VHDL-описаний / В. В. Сергейчик, А. А. Иванюк // Информационные технологии и системы 2013 (ИТС 2013) : материалы международной научной конференции, БГУИР, Минск, Беларусь, 23 октября 2013 г. = Information Technologies and Systems 2013 (ITS 2013) : Proceeding of The International Conference, BSUIR, Minsk, 24th October 2013 / редкол.: Л. Ю. Шилин [и др.]. - Минск : БГУИР, 2013. – С. 198-199.
3. Garg S., Gentry C., Halevi S., Raykova M., Sahai A., and Waters B. «Candidate indistinguishability obfuscation and functional encryption for all circuits.» FOCS 2013.

## **СРАВНЕНИЕ СВЕРТОЧНЫХ И РЕКУРРЕНТНЫХ НЕЙРОННЫХ СЕТЕЙ В ЗАДАЧЕ АНАЛИЗА ТОНАЛЬНОСТИ ТЕКСТА**

*Витковский А.В.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Жвакина А.В. – к.т.н., доцент*

В обработке естественных языков существует задача определения тональности текста. В настоящее время для решения этой задачи используются искусственные нейронные сети. Для определения тональности могут быть использованы сети с различной архитектурой. В данном исследовании сравниваются сверточная нейронная сеть и сеть с долгой краткосрочной памятью.

Обработка естественного языка – направление искусственного интеллекта и математической лингвистики. Оно изучает проблемы компьютерного анализа и синтеза естественных языков. Анализ тональности текста является одной из задач, решаемых в рамках обработки языка. Суть задачи заключается в автоматическом распознавании в тексте его эмоциональной окраски. Классификация бывает бинарной, при которой определяется позитивную и негативную окраску имеет текст. Также существует классификация на несколько классов, при которой окраска может быть нейтральной или разной степени позитивности или негативности.

Две различные архитектуры нейронной сети будут сравниваться при решении задачи бинарной классификации. В качестве выборки используется выборка отзывов с сайта IMDb. Входными данными для обеих сетей будут являться тексты, где каждое слово закодировано целым числом, при этом значение зависит от частоты появления слова в выборке данных.

Сверточная нейронная сеть (Convolutional Neural Network, CNN) является специальной архитектурой, в основном применяемой для распознавания образов на изображениях. Название архитектура сети получила из-за наличия операции свёртки, суть которой в том, что каждый фрагмент изображения умножается на матрицу (ядро) свёртки поэлементно, а результат суммируется и записывается в аналогичную позицию выходного изображения. Несмотря на основное применение, ее также можно использовать для работы с текстом. Если в случае изображений, фильтр сверточного слоя применяется для нескольких соседних пикселей, то в случае текста фильтр можно применять для нескольких соседних слов.

Долгая краткосрочная память (Long short-term memory, LSTM) – один из видов рекуррентных нейронных сетей. В рекуррентных нейронных сетях связи между нейронами имеют направленную последовательность. Такие сети могут использовать свою внутреннюю память для обработки последовательностей произвольной длины. LSTM-сеть содержит LSTM-модули. LSTM-модуль – это рекуррентный модуль сети, способный запоминать значения как на короткие, так и на длинные промежутки времени. Такое поведение обусловлено тем, что LSTM-модуль не использует функцию активации внутри своих компонентов.

Для обеих нейронных сетей первый слой одинаков и является слоем, осуществляющим замену целых чисел, обозначающих слова, на векторное представление слов. Этот слой обучается в ходе тренировки модели в обоих случаях. В обоих случаях функцией потерь является перекрестная энтропия, обучения происходит при помощи метода обратного распространения ошибки с алгоритмом Adam. Объем тестовых и валидационных данных также одинаков для обеих моделей. Различными являются только сами модели нейронных сетей.

Модель сверточной нейронной сети состоит из трех каналов, каждый канал состоит из сверточного слоя, слоя активации, слоя подвыборки и полносвязного слоя. каналы в модели параллельны. Данная модель позволяет задать различные размеры фильтра сверточного слоя в каждом канале. В реализованной модели использовались фильтры по 3, 4 и 5 слов. Каналы объединяются при помощи полносвязного слоя.

Модель сети с долгой краткосрочной памятью позволяет сразу применять LSTM-блок к векторным представлениям слов после соответствующего слоя. Выход LSTM-слоя подается полносвязному слою, который определяет конечный результат работы сети.

После многократного обучения моделей были получены следующие результаты. Сверточная нейронная сеть достигла точности 92%, а модель с долгой краткосрочной памятью – 88%.

**Список использованных источников:**

1. Kim, Y. Convolutional Neural Networks for Sentence Classification / Y. Kim // New York University – 2014. – 6 P.
2. Greff, K. LSTM: A Search Space Odyssey / K. Greff, R. Srivastava, J. Koutník, B. Steunebrink, J. Schmidhuber // IEEE Transactions on Neural Networks and Learning Systems. A. – 2017. – Vol. 28, № 10. – P. 2222–2321.
3. Заенцев, И. В. Нейронные сети: основные модели / И. В. Заенцев. – Воронеж, 1999. – 76 с.
4. Mikolov, T. Distributed Representations of Words and Phrases and their Compositionality / T. Mikolov, I. Sutskever, K. i Chen, G. Corrado, J. Dean // NIPS'13 Proceedings of the 26th International Conference on Neural Information Processing Systems – 2013. – Vol. 2. – P. 3111–3119.

## **ХРАНИЛИЩЕ ДАННЫХ СИСТЕМЫ КОМПЛЕКСНОГО АНАЛИЗА ДАННЫХ ИНТЕРНЕТ ИСТОЧНИКОВ**

*Гутковский В.Н.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Пилецкий И.И. – к.ф.-м.н., доцент*

В докладе приводится описание инструмента мониторинга открытых интернет-источников с целью выявления экспертов в некоторой научной области, определения тематик публикаций, оценки популярности публикаций. Описываются принятые решения при построении компонента хранилища аналитического комплекса и полученные результаты его работы.

В настоящее время информация, полученная в результате анализа данных интернет источников, является одной из базовых для принятия решений. Как правило, это неструктурированные текстовые данные, различные мультимедийные данные. Данные могут быть получены как из социальных сетей, так и тематических сайтов (газет, журналов, библиотек, компаний и т. д.), содержащих различные публикации. Есть много работ, которые посвящены принятию решения на основании применения некоторого метода анализа данных. Результатами анализа смогут воспользоваться компании для создания систем поддержки пользователей, социологи для анализа общественного мнения, организаторы мероприятий для получения отклика участников, знаменитости для отслеживания репутации в сети, правительство для контроля настроений в обществе и др.

В данной статье рассматривается проект создания «Системы комплексного анализа данных интернет-источников (СКА)», позволяющей анализировать большие объемы данных из интернет-источников в области научных исследований и предназначенной для сбора информации о научных публикациях, построения графа знаний, что дает возможность определять экспертов предметной области, тематики их работ, их взаимосвязи, а также определять передовые научные направления.

Система должна находить экспертов (авторитетов) в предметной области и выдавать оценку их рейтинга влияния. Например, лучше прочитать три книги признанных экспертов в определенной области, чем десять книг дилетантов.

СКА состоит из следующих компонент: сбора данных, фильтрации данных и составление «мешка слов» из N-грамм (векторизации), хранилища данных, библиотеки аналитических модулей, подготовки выдачи результата, клиентского модуля.

В докладе рассмотрен в подробностях компонент хранилища СКА. Компонент хранилища данных – содержит данные из интернет-источников, предварительно обработанные и размеченные данные, необходимые для построения классификатора, «мешок слов», а также служебную информацию, необходимую для работы других модулей системы. В хранилище хранятся сырые данные с сайта, текст, фильтрованный текст, исходные документы, «мешок слов», тематика документов и служебная информация; структура одной из записей для документа приведена ниже:

Структура записи (Hash - primary key of publication, Title - title of publication, Author - author(authors) of publication, Year - publication date, Pages - number of pages, Publisher - publisher of publication, Language - primary language of publication, Topic - topic or topics of publication, Extension - extension of publication file, Tags - array of publication tags, Locator - name of the file).

В настоящее время в хранилище содержится информация более чем о 20 тыс. статей и документов. В дальнейшем планируется получать данные из многих сайтов и при необходимости указывать ссылки к конкретным данным. Документы, статьи читаются с сайта (сайтов), фильтруются,

строится векторное представление («мешок слов») и все данные (сырые данные с сайта, текст, фильтрованный текст, исходные документы и «мешок слов») сохраняются в хранилище.

Выбранная архитектура СКА позволяет её модернизировать и функционально наращивать в процессе эксплуатации. Так, планируется нарастить компонент «хранилище» и дополнить компонент «библиотека модулей» векторными алгоритмами и алгоритмами нейровычислений, что существенно расширит область применения СКА и позволит получать информацию в интересах сотрудников университета, государственных и частных организаций.

**Список использованных источников:**

1. Data Never Sleeps 6.0 [Электронный ресурс] / Режим доступа: <https://www.domo.com/learn/data-neversleeps-6> Дата доступа: 18.01.2019.
2. Пилецкий, И. И. Аналитический комплекс анализа данных из открытых интернет источников / И. И. Пилецкий, В. А. Прытков, Н. А. Волорова // BIG DATA Advanced Analytics: collection of materials of the fourth international scientific and practical conference, Minsk, Belarus, May 3 – 4, 2018 – Minsk, BSUIR, 2018. – P. 193 – 199.
3. Прытков, В. А. Анализ репозитория университета с использованием графовой базы данных / В. А. Прытков, И. И. Пилецкий, Н. А. Волорова // BIG DATA Advanced Analytics: collection of materials of the fourth international scientific and practical conference, Minsk, Belarus, May 3 – 4, 2018. – Minsk, BSUIR, 2018. – P. 177 – 183.

## СРАВНЕНИЕ ВАРИАЦИЙ ГРАДИЕНТНОГО СПУСКА НА ПРИМЕРЕ ЗАДАЧИ РАСПОЗНАВАНИЯ СИМВОЛОВ

*Евжик Д.А., Подвальников Д.С., Тишковский М.А.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Анисимов В.Я. – к.ф.-м.н., доцент*

В данной работе приведен сравнительный анализ вариаций градиентного спуска, который демонстрирует применимость отдельного метода для задачи распознавания символов. Результаты исследования в данной работе позволяют сделать выбор наиболее подходящего по скорости и качеству обучения метода при решении задач.

В настоящее время можно заметить глубокий интерес к области машинного обучения. Как известно, для построения методов машинного обучения используются средства методов оптимизации для нахождения минимума целевой функции в некоторой области конечномерного векторного пространства.

В ходе исследования были проанализированы три популярных метода нахождения локального минимума функции. Обычный, пакетный и стохастический градиентный спуск.

В данной работе был рассмотрен набор данных *notMNIST* [1], который состоит из изображений размерностью 28×28 первых 10 букв латинского алфавита.

Градиентный спуск – это метод нахождения локального минимума функции путем движения по вектору антиградиента. Этот метод является очень распространенным в машинном обучении для нахождения минимума функции потерь. В начале работы алгоритма задается начальное приближение минимума, затем вычисляется значение градиента функции потерь на всей обучающей выборке, последним шагом является сложение начального приближения с полученным значением градиента со знаком минус. Стохастический градиентный спуск является вариацией обычного градиентного спуска. Его отличие заключается в том, что вычисление значения градиента производится не на всей обучающей выборке, а на одном случайном элементе, за счет этого скорость обучения значительно возрастает. Пакетный градиентный спуск схож со стохастическим, но вместо одного экземпляра из выборки берется какая-то ее часть, несколько экземпляров, что позволяет добиться как хорошей скорости обучения, так и более плавного движения к минимуму функции.

Описанные три вариации градиентного спуска были применены для обучения нейронной сети на наборе данных *notMNIST* размером 10000 элементов. Архитектура нейронной сети состояла из пяти скрытых слоев с 1024, 1024, 512, 256 и 128 нейронами на каждом слое соответственно. На всех скрытых слоях в качестве функции активации была использована *relu* [2] (кусочно-линейная функция), на выходном слое – *softmax* [3] (обобщение логистической функции для многомерного случая). В результате при обучении на 10 эпохах получились следующие графики, показывающие зависимость ошибки обучения от эпохи.

Результаты данного эксперимента в виде зависимости ошибки от порядкового номера эпохи представлены на рисунке 1, рисунке 2 и рисунке 3.

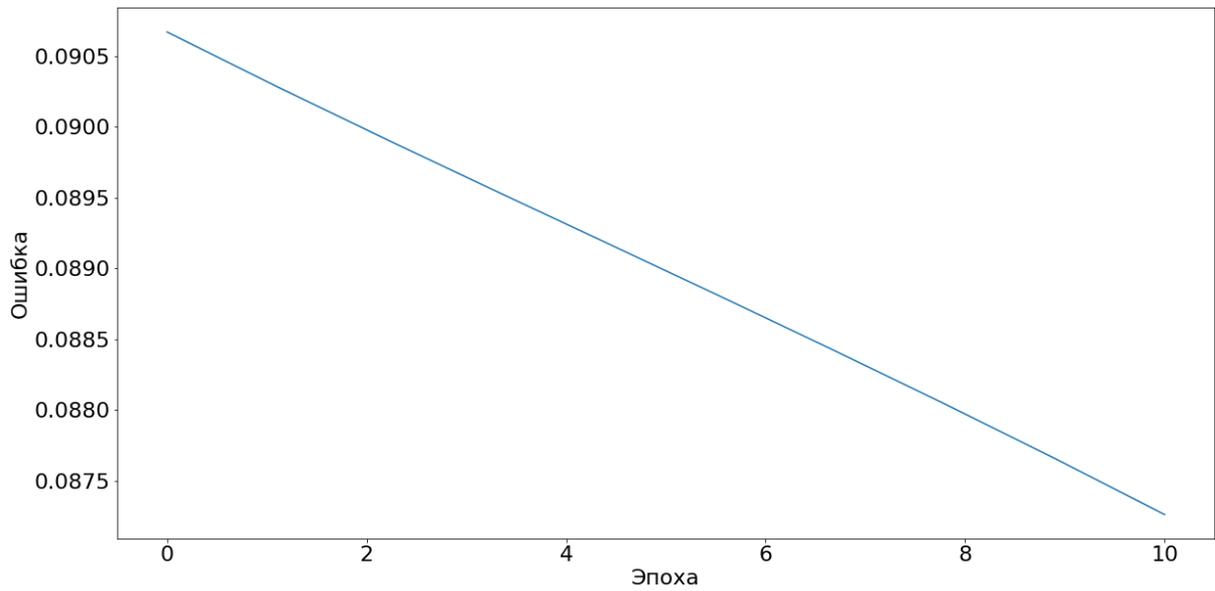


Рисунок 1 – Изменение ошибки от эпохи при использовании обычного градиентного спуска

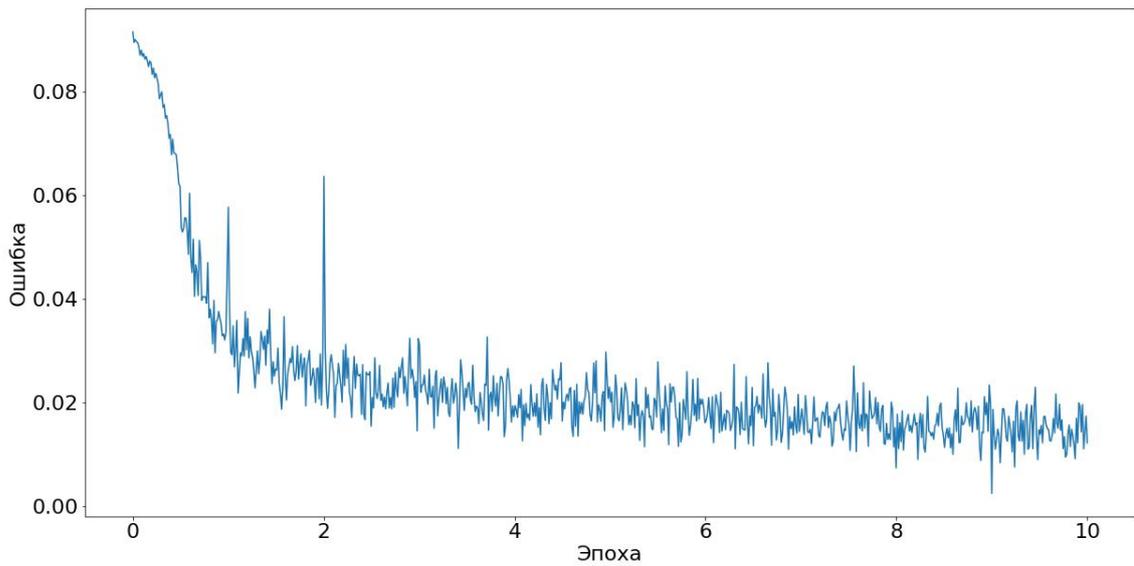


Рисунок 2 – Изменение ошибки от эпохи при использовании пакетного градиентного спуска

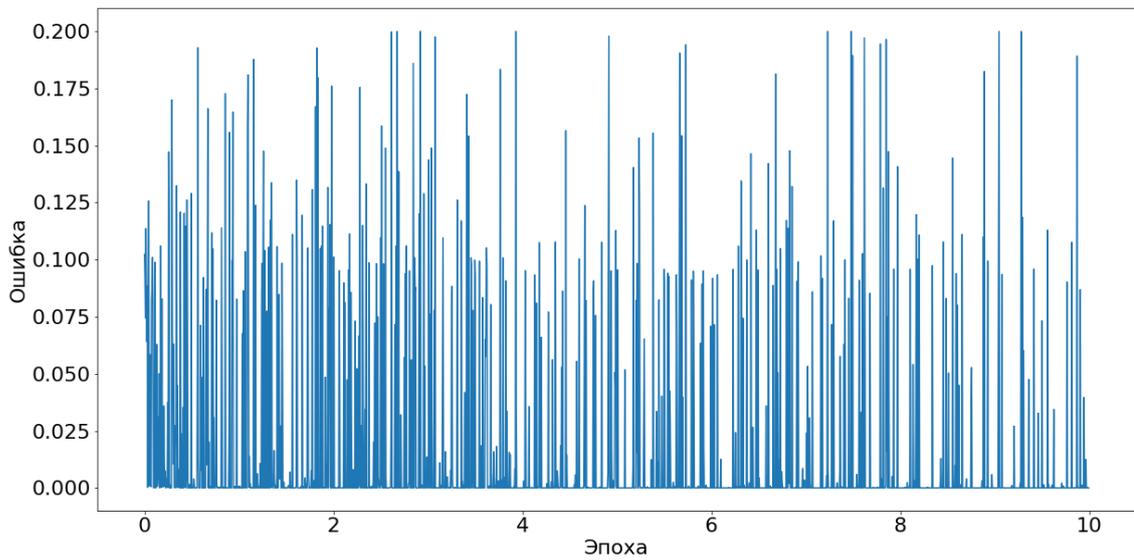


Рисунок 3 – Изменение ошибки от эпохи при использовании стохастического градиентного спуска

В результате применения обычного градиентного спуска в среднем на одну эпоху тратилось в среднем 93 мс, при этом точность предсказания на тестовой выборке составила 0.32. Из этого видно, что 10 эпох не достаточно для достижения высокой точности. При использовании пакетного градиентного спуска с размером пакета 128 элементов среднее время обучения одной эпохи составило 153 мс, качество обучения – 0.78. Для стохастического градиентного спуска на обучение одной эпохи потребовалось 61 с, качество обучения составило 0.84. Для обучения использовался шаг равный 0.5. Для оценки качеств модели использовалась метрика *accuracy* [4].

Из полученных результатов можно сделать вывод о том, что обычный градиентный спуск работает гораздо медленнее при достижении минимальной ошибки в сравнении со стохастическим и пакетным градиентным спуском, однако он значительно лучше минимизирует функцию. Стохастический градиентный спуск лучше показывает себя на задачах, в которых для обучения требуется небольшое количество данных, для более сложных задач с большим набором данных в обучающей выборке достижение минимума может также оказаться долгим процессом. Для больших выборок подходит пакетный градиентный спуск, он быстрее в сравнении с обычным и в то же время быстрее сходится к минимуму в сравнении со стохастическим. Таким образом, в случае, когда достаточно найти значение близкое к минимуму за приемлемое время, имеет смысл использовать стохастический или пакетный градиентный спуск в зависимости от решаемой задачи.

**Список использованных источников:**

1. Набор данных potMNIST [Электронный ресурс] – Режим доступа: <http://varoslavvb.blogspot.com/2011/09/notmnist-dataset.html>
2. Функция relu [Электронный ресурс] – Режим доступа: <https://arxiv.org/pdf/1803.08375.pdf>
3. Функция softmax [Электронный ресурс] – Режим доступа: <https://arxiv.org/pdf/1811.03378.pdf>
4. Определение метрики *accuracy* в математической статистике [Электронный ресурс] – Режим доступа: [http://gsp.humboldt.edu/OLM\\_2015/Courses/GSP\\_216\\_Online/lesson6-2/metrics.html](http://gsp.humboldt.edu/OLM_2015/Courses/GSP_216_Online/lesson6-2/metrics.html)

## ОБЛАЧНАЯ ИНФРАСТРУКТУРА КАК КОД НА ПРИМЕРЕ СИСТЕМЫ ДЛЯ АВТОМАТИЗАЦИИ КОНТРОЛЯ ДОСТУПА К SAP

*Киселёв Д.О.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Егорова Н.Г. – канд. техн. наук,  
доцент каф. Информатики*

В настоящее время использование облачных сервисов стало стандартом при разработке программного обеспечения. Будь это маленький сайт или большая промышленная система. Потребности в мощности ресурсов очень динамические: сегодня систему использует сто человек, через полгода - тысяча, следовательно, становится вопрос простого перехода из одной инфраструктуры на другую. В случае использования облачных сервисов и подхода «инфраструктура как код», данный процесс максимально упрощается.

Инфраструктура как код (англ. Infrastructure as code (IaC)) – это способ создания и управления ресурсами для центров хранения и обработки данных методом их описания в исходном коде. Данный подход приходит на замену настройке оборудования вручную или с помощью интерактивных инструментов. Для описания инфраструктуры чаще всего используется декларативный язык, так же оно может находиться в системе контроля версий. Подходы IaC продвигаются для облачных вычислений, которые иногда называют инфраструктура как сервис (IaaS).

Преимущества IaC можно описать в трех направлениях: стоимость, скорость, риски. Снижение стоимости происходит как за счет более рационального использования вычислительных ресурсов, так и за счет снижения трудозатрат на их обслуживание, позволяя больше сфокусироваться на решении проблем бизнеса. Автоматизация инфраструктуры обеспечивает скорость за счет более быстрого выполнения при настройке инфраструктуры. Автоматизация убирает риск, связанный с человеческой ошибкой, что позволяет уменьшить время простоя и повысить надежность. Эти преимущества помогают корпоративному сегменту двигаться в направлении развития практик DevOps.

Архитектура системы для автоматизации и контроля доступа к SAP построена с использованием большого количества облачных сервисов, причем разных поставщиков. Изначально единственным способом создания нового окружения было исполнения шагов, описанных в текстовом документе вручную. Для того, чтобы развернуть новое окружение, тратилось очень много времени. Так же этот процесс был подвергнут ошибкам из-за разночтения инструкции. Соответственно стал вопрос об автоматизации развертывания инфраструктуры. Схема архитектуры представлена на рисунке 1.

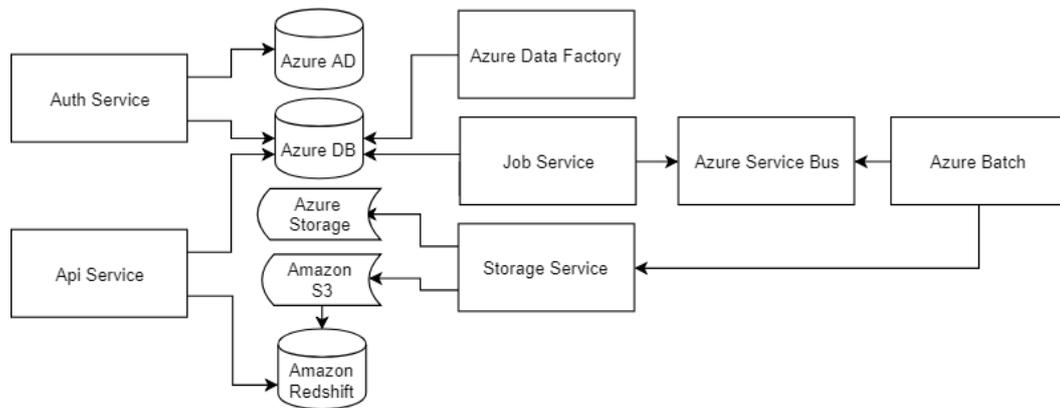


Рис. 1 – Архитектура системы для автоматизации контроля доступа к SAP

Решением проблемы стало использование программного инструмента Terraform от компании HashiCorp. Terraform предоставляет библиотеки для управления большинством сервисов всех крупных облачных поставщиков. Для достижения целей были использованы библиотеки для облаков Azure и AWS. Данное решение помогло описать используемые ресурсы в декларативном стиле, и, в последствии, использовать для создания новых окружений без особых сложностей. Описание каждого ресурса создавалось в отдельном файле на языке конфигураций HCL (Hashicorp Configuration Language). Так же была использована система контроля версий Git, что позволяет видеть изменения в инфраструктуре.

**Список использованных источников:**

1. Infrastructure as code [Электронный ресурс]. – Режим доступа: [https://en.wikipedia.org/wiki/Infrastructure\\_as\\_code](https://en.wikipedia.org/wiki/Infrastructure_as_code) – Дата доступа: 24.03.2019.
2. Terraform Documentation [Электронный ресурс]. – Режим доступа: <https://www.terraform.io/docs/index.html> – Дата доступа: 24.03.2019.
3. Terraform on Azure [Электронный ресурс]. – Режим доступа: <https://docs.microsoft.com/en-us/azure/terraform/> – Дата доступа: 24.03.2019.

## ИСПОЛЬЗОВАНИЕ АЛГОРИТМА КЛАСТЕРИЗАЦИИ DBSCAN ДЛЯ ФИЛЬТРАЦИИ ВЫБРОСОВ В ДАННЫХ

*Ковалёв С.П.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Калугина М.А. – к. ф.-м. н., доцент*

При обработке больших объемов данных часто используется такой метод машинного обучения как кластеризация, который помогает найти группы похожих объектов в наборе данных. Но для успешной кластеризации набора данных его необходимо предварительно обработать – очистить данные от выбросов. Одним из таких способов является использование способности алгоритма DBSCAN определять выбросы в наборе кластеризуемых данных.

Кластеризация относится к методам обучения «без учителя». Эти методы не предполагают, что существует правильный ответ, для получения которого можно обучить алгоритм. Кластеризация призвана систематизировать данные и найти группы похожих объектов.

Методы кластерного анализа используются в психологии, археологии, биологии. В настоящее время кластеризация также активно применяется в социологии, экономике, статистике, в исторических исследованиях и многих других сферах. Страховые компании могут разделять автомобили по группам риска, а затем назначать каждой группе различную стоимость страховки. В вычислительной биологии этот метод активно применяется для обнаружения групп генов, демонстрирующих сходное поведение. Это может служить указанием на то, что они будут одинаково реагировать на лечение или что образовались на одном пути биологического развития [1]. Особенно расширилось использование этого метода машинного обучения в связи с появлением и развитием ЭВМ, поскольку это связано с трудоемкостью обработки больших объемов информации.

Алгоритмы кластеризации можно рассматривать как иерархические и неиерархические. Результатом работы иерархических алгоритмов является разбиение данных по уровням. В этом случае исследователь может выбрать любую ступень построенной иерархии при интерпретации

результатов. Неиерархическими считаются все алгоритмы, результатом которых не является такая процедура (или выбор интерпретации происходит не по уровню иерархии).

Различные методы кластеризации имеют свои ограничения и могут выделять кластеры определенных типов и форм. Например, одни из самых известных алгоритмов, использующие методы К-средних и агломеративной иерархической кластеризации, хорошо и быстро находят кластеры сферической формы, но неспособны выделить кластеры необычной несферической формы (вытянутые, закругленные). Различные библиотеки для машинного обучения предоставляют примеры нахождения кластеров [2]. Эти методы распределяют абсолютно все данные по кластерам, учитывая и точки, которые однозначно можно назвать выбросами. Эти точки, не принадлежащие ни к какому из кластеров, являясь аномальными данными, которые не следует учитывать. Из-за них положение центра кластера при использовании вышеуказанных методов искривляется и при анализе результатов можно получить неверные выводы.

В свою очередь, алгоритм кластеризации DBSCAN решает проблему выбросов в наборе данных. DBSCAN (Density-based spatial clustering of applications with noise) – молодой алгоритм кластеризации, созданный только в 1996 году [3]. В отличие от множества других алгоритмов он не требует указания количества кластеров и может находить произвольное их количество. Также, в отличие от названных выше, этот алгоритм может находить кластеры различных типа и формы – главное, чтобы точки находились рядом.

На вход необходимо передать два параметра – радиус  $\epsilon$  окрестности и количество  $m$  соседей. DBSCAN не может хорошо кластеризовать наборы данных с большой разницей в плотности, поскольку не удастся выбрать приемлемую для всех кластеров комбинацию этих входных параметров.

Этот алгоритм способен хорошо находить шумы и выбросы в данных. Происходит это в тех случаях, когда точку невозможно отнести ни к какому из существующих кластеров, то есть в окрестности данной точки находится мало точек-соседей.

Именно способность данного алгоритма находить выбросы можно использовать для фильтрации этих самых выбросов. После того, как в наборе данных не останется выбросов, каждый кластер будет точно описывать группу, на положение центра кластера не будут влиять выбросы и шумы и можно гораздо лучше проводить анализ данных.

Пример использования алгоритма кластеризации DBSCAN приведен на рисунке 1. На левой половине изображены все точки исходного набора данных. На правой - уже очищенные с помощью DBSCAN.

В данном примере использован набор данных, симулирующий поведение пользователей в системе по управлению проектами небольших групп по примеру сайта trello.com. Пользователи могут регистрироваться, заходить в аккаунт, открывать главную страницу, открывать доску с карточками, создавать карточки и т.д.

При симуляции поведения пользователей были заданы 2 типа пользователей: корпоративные и персональные. Поэтому на рисунке легко обнаружить 2 кластера. Корпоративные пользователи, которые были симулированы для этого набора данных, пользуются сервисом активно в будние дни, почти не используют в выходные, создают много событий. Группа персональных пользователей использует приложение для личных нужд каждый день недели, но в заметно меньшем объеме. Набор данных включает события 546 пользователей в течение нескольких недель.

Данные для каждого пользователя представляют собой набор характеристик его деятельности – общее количество событий различного типа, количество событий в среднем по дням недели и т.д. В итоге получается около 30 характеристик, поэтому не представляется возможным отобразить пользователей на двумерной плоскости. В этом случае можно уменьшить размерность данных с помощью метода главных компонент (англ. principal component analysis, PCA). Это один из основных способов уменьшить размерность данных, потеряв наименьшее количество информации.

На рисунке с исходными данными легко определить, что здесь представлены две группы точек. Группа с большим разбросом точек слева – корпоративные пользователи. И небольшая группа с плотно лежащими точками справа, которая в основном состоит из персональных пользователей. Но группы выделены не совсем четко: много точек между кластерами и на их границах. Эти выбросы плохо влияют на положение центра кластера. Хорошим решением является избавление от этих точек.

Исходный набор данных состоит из 546 точек. С помощью алгоритма DBSCAN удалось обнаружить 104 выброса в наборе данных. Недостатком этого способа является исключение этих точек из дальнейшего анализа. Но большим плюсом является возможность более четко и качественно выделить кластеры, найти близкое к истинному положение центров каждого кластера.

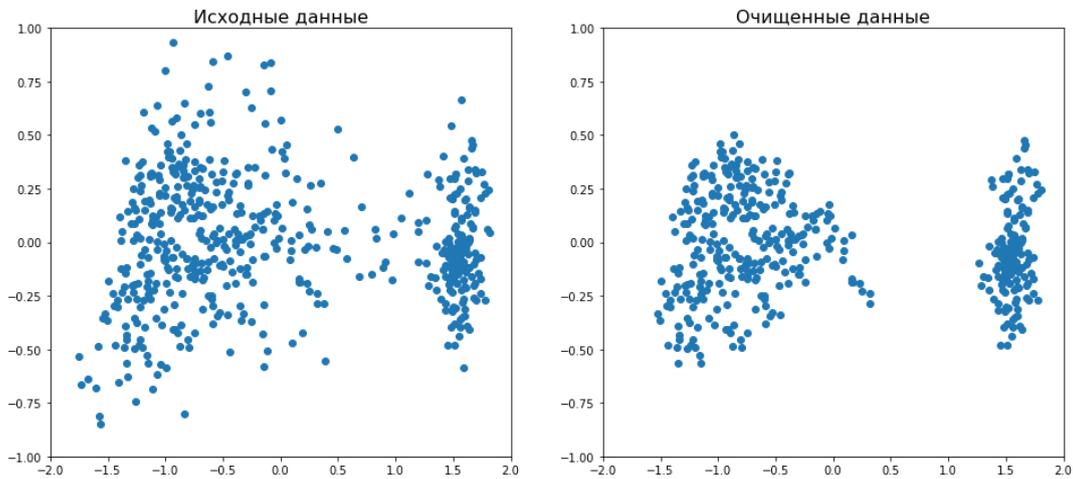


Рисунок 1 – Исходные и очищенные данные

Теперь посмотрим на результаты фильтрации выбросов алгоритмом DBSCAN. В связи с тем, что были убраны выбросы в данных, кластеры приобрели более четкие формы.

При применении метода К-средних на исходном наборе данных группа точек справа на 20% состояла из корпоративных пользователей и на 80% из персональных. То есть в этот кластер попало много объектов, которые изначально должны были попасть в другой кластер. Из-за некоторых аномалий в поведении эти пользователи находятся между двумя кластерами и при кластеризации отнесены к неправильной группе. Таким образом, центр кластера смещен относительно его правильного положения.

При использовании набора очищенных данных группа точек справа теперь только на 7% состоит из корпоративных пользователей, все остальные – персональные. Это значит, что теперь этот кластер более точно описывает группу персональных пользователей, центр кластера находится ближе к его истинному значению.

Таким образом, использование алгоритма кластеризации DBSCAN на практике позволяет очистить и подготовить набор данных для дальнейшего анализа, полученные при этом кластеры выделяются в более отчетливые формы, что несомненно приносит пользу, когда исследователю необходимо наиболее точно охарактеризовать свойства кластеров.

**Список использованных источников:**

1. Сегаран Т. Программируем коллективный разум. / Сегаран Т. – Пер. с англ. – СПб: Символ-Плюс, 2008. — 50-51 с.
2. Comparing different clustering algorithms on toy datasets [Электронный ресурс]. — Режим доступа: [https://scikit-learn.org/stable/auto\\_examples/cluster/plot\\_cluster\\_comparison.html/](https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html/). — Дата доступа: 05.03.2019.
3. Ester, M., Kriegel, H. P., Sander, J., & Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. / Ester, M., Kriegel, H. P., Sander, J., & Xu, X. // KDD. — 1996. — Vol. 96, №34 — P. 226–231.

## БАЙЕСОВСКАЯ ОПТИМИЗАЦИЯ

*Козак А. В., Сухов Н. Ю.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Теслюк В. Н. – доцент, кандидат физико-математических наук*

В данной работе рассмотрен алгоритм глобальной оптимизации функций без использования производных. Приведен класс функций, для которых может быть использована байесовская оптимизация. Рассмотрены потенциальные проблемы использования байесовской оптимизации. Приведены потенциальные области для улучшения и дальнейшего исследования данного класса оптимизационных алгоритмов.

Байесовская оптимизация – это подход к оптимизации целевых функций, оценка и подсчет которых требует большого числа ресурсов, например, времени (минуты или часы). Она лучше всего подходит для оптимизации непрерывных функций, определенных на многомерном пространстве размерности не более 20 и допускающих стохастический шум в значениях самой функций.

Байесовская оптимизация – это класс алгоритмов оптимизации, основанных на принципах машинного обучения и сфокусированных на решении оптимизационной задачи вида:

$$f(x).$$

При этом множество допустимых значений ( $A$ ) и целевая функция ( $f(x)$ ) удовлетворяют следующим свойствам:

1. Целевая функция определена на многомерном пространстве, то есть  $f: R^d \rightarrow R$ . При этом для достижения наибольшей эффективности байесовской оптимизации как правило выполняется следующее неравенство  $d \leq 20$ .
2. Множество допустимых значений  $A$  является простым, то есть для любой точки из  $R^d$  легко понять принадлежит она  $A$  или нет. Как правило,  $A$  является многомерным параллелепипедом (то есть  $A \in \{x \in R^d: a_i \leq x_i \leq b_i\}$ ) или  $d$ -мерным симплексом (то есть  $A \in \{x \in R^d: \sum_i x_i = 1\}$ ). Однако данное свойство является лишь опциональным.
3. Целевая функция должна быть непрерывной. Это ограничение необходимо для регрессии гауссовского процесса.
4. Подсчет значения целевой функции в точке является «дорогостоящей» операцией в том смысле, что для нахождения значения необходимо произвести несколько сотен вычислительных операций при этом каждая из таких операций требует также большого количества вычислительных ресурсов, что в свою очередь является «роскошью» и требует большого количества временных ресурсов (например, несколько дней).
5. Целевая функция не имеет известной для нас структуры, то есть мы не знаем ничего об этой функции, например, линейность или выпуклость. Будем считать, что мы знаем только алгоритм вычисления, но не знаем выражение функции в элементарных функциях, что делает невозможным найти производную целевой и, как следствие, использовать стандартные методы оптимизации. Будем говорить, что целевая функция – «черный ящик».
6. Нет возможности вычислить не первую, не вторую производную, то есть знания о целевой функции являются «свободными» от производной, как результат нет возможности использовать метод Ньютона и другие.
7. Целевая функция может быть зашумленной, при этом будем считать, что шум является независимым от значений целевой функции.
8. Наша задача найти глобальный оптимум, нежели локальный.

Таким образом, мы можем сказать, что байесовская оптимизация предназначена для решения задач глобальной оптимизации «черного ящика» без производных.

Способность оптимизации такого класса целевых функций без использования производных делает область применения байесовской оптимизации очень разносторонней. Так стало популярным использовать данный класс оптимизационных алгоритмов в машинном обучении для подбора гиперпараметров. В течение более длительного периода, начиная с 1960-х, байесовская оптимизация использовалась для проектирования инженерных систем. По мимо этого, оптимизацию, основанную на байесовской статистике, использовали для выбора экспериментов, которые дадут наиболее значимые результаты, в области разработки лекарственных средств, для калибровки моделей окружающей среды и в обучении с подкреплением (класс задач в машинном обучении).

Помимо байесовской оптимизации существуют и другие методы позволяющие решать задачу глобальной оптимизации «дорогостоящий черных ящиков». Большинство этих методов схожи на байесовскую оптимизацию: они используют так называемую суррогатную функцию, которая моделирует нашу целевую функцию и которую в дальнейшем используют для выбора точки подсчета значения целевой функции, тем самым уточняя множество точек потенциальных на место оптимума. Такой класс методов оптимизации называется суррогатными методами. Байесовская оптимизация отличается от других суррогатных методов использованием суррогатов, смоделированных с использованием байесовской статистики, и при принятии решения о том, где оценивать целевую функцию, использует байесовскую интерпретацию этих суррогатов.

Алгоритм байесовской оптимизации состоит из двух компонент: модели байесовской статистики для моделирования целевую функцию и дополнительно функции принятия решения того, где производить подсчет целевой функции. После оценки целевой функции согласно первоначальному экспериментальному предположению заполнения пространства возможных значений (то есть выбранных точек), часто состоящему из точек, выбранных равномерно случайным образом, алгоритм

использует итеративную процедуру для распределения оставшейся части «бюджета» подсчета целевой функции  $N$ .

Статистическая модель, которая неизменно (так называемое свойство инвариантности) является гауссовским процессом, обеспечивает байесовскую апостериорную вероятность распределение неизменной по своей структуре с апостериорным распределением, что упрощает процесс оптимизации, описывающее потенциальные значения для  $f(x)$  в точке-кандидате  $x$ . Каждый раз, когда мы наблюдаем данные, новая точка приводит к обновлению и уточнению апостериорного распределения.

В этой увлекательной и необъятной области байесовских оптимизационных алгоритмов существует множество направлений для исследований. Во-первых, область для более глубокого теоретического понимания и освоения методов байесовской статистики. Так например, ничего нельзя сказать о скорости сходимости данного алгоритма к оптимуму и требует отдельного теоретического исследования. Во-вторых, исследование других методов построения байесовской статистики, отличных от гауссовского процесса. В-третьих, исследование и разработка алгоритмов байесовской оптимизации, работающих на целевых функциях любой размерности, тем самым снимая ограничение номер 1 (описанных выше) с оптимизируемой функции.

Несмотря на обширную область неизвестности байесовских алгоритмов, они предоставляют нам обширный класс решения эффективной оптимизации, когда стандартные методы являются нецелесообразными.

В ходе работы были получены следующие результаты: действительно, байесовская оптимизация справляется эффективнее в поиске гиперпараметров нейросетевой модели, чем стандартные методы оптимизации, более того, байесовская оптимизация не требует нахождения производной, что является дорогостоящим (в плане вычислительных ресурсов) для поиска оптимальных гиперпараметров модели. Возможность оптимизации и модификации алгоритма байесовской оптимизации под алгоритм обучения моделей машинного обучения (особенно в байесовском нейросетевом моделировании) повышает эффективность использования обученной модели как качественно, так и в плане экономичности использования вычислительных ресурсов в ходе обучения. Все, выше перечисленное, является неотъемлемым преимуществом для дальнейшей работы в моей магистерской диссертации по использованию байесовских методов (в том числе и оптимизационных) в анализе медицинской информации (где получение значения целевой функции является «роскошью»).

**Список использованных источников:**

1. Berger J. O. – Statistical Decision Theory and Bayesian Analysis, 2013,
2. Blum J. R. – Multidimensional stochastic approximation methods, 1954,
3. Gelman A., Carlin J. B., Stern H. S. – Bayesian Data Analysis, 2014.

## **АВТОМАТИЗИРОВАННАЯ СИСТЕМА УПРАВЛЕНИЯ ВЗАИМООТНОШЕНИЯМИ С КЛИЕНТАМИ С ПРИМЕНЕНИЕМ ОБЛАЧНЫХ CRM-СИСТЕМ**

*Козлова А. А.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Калугина М.А. – к. ф.-м. н., доцент*

Интерес к CRM-системам непрерывно растёт, поскольку они решают проблемы загрузки ресурсов и эффективности работы бизнеса. Именно поэтому этот класс программного обеспечения заслуживает пристального внимания. В докладе приведены результаты исследования работы CRM-системы SAP Hybris Cloud For Customers, а также предложено значительное улучшение в направлении автоматизации работы менеджеров с целевыми клиентами согласно бизнес-процессу корпоративного предприятия.

Важным аспектом функционирования бизнеса является процесс взаимодействия с клиентами. В силу растущей конкуренции и компьютеризации бизнеса неотъемлемой частью успешных компаний на сегодняшний день считают наличие CRM-систем [1]. Система управления взаимоотношениями с клиентами (Customer Relationship Management, CRM) – это комплексный подход, бизнес-стратегия к выявлению, приобретению и сохранению клиентов [2]. Основная задача CRM – получать на базе собранных данных информацию, которую можно использовать непосредственно для повышения

доходности и эффективности ведения бизнеса, формируя на базе этих данных новые и дополнительные услуги для различных групп клиентов. Для исследования и доработки была выбрана облачная CRM-система SAP Hybris Cloud For Customers [3].

Платформа SAP Hybris Cloud For Customers является облачной и имеет ряд преимуществ:

- программное обеспечение как сервис (SaaS) [4] – позволяет работать с клиентами с помощью совокупности облачных CRM-решений в любом месте, в любое время и на любом устройстве, которые объединяет функции для маркетинга, продаж, торговли, обслуживания и социальных сетей;

- просмотр полной информации о клиентах в различных разрезах и их управление – подробное представление о клиентах, начиная с первоначального контакта до выставления счетов в любое время и в любом месте;

- гибкая и легкая интеграция – SAP Hybris C4C позволяет реализовать индивидуальные потребности бизнеса, что позволяет адаптировать, расширять и интегрировать программное обеспечение в существующую SAP или любую стороннюю архитектуру [5].

Для исследования был выбран процесс взаимодействия с клиентами, его основные составляющие, принципы и аспекты для выделения условий и путей повышения эффективности работы менеджеров и облегчение их взаимодействия с клиентами. Под эффективностью будем понимать сокращение временных и трудовых затрат на процесс взаимодействия. Данная система направлена в помощь менеджерам компании. Она может осуществлять мониторинг состояния текущих клиентов, автоматически формировать задачи на основании анализа текущих данных, оперативно реагировать на изменения информации о клиентах.

На рисунке 1 изображена контекстная диаграмма верхнего уровня процесса работы с действующими клиентами компании, их развитие и сопровождение. Чтобы клиенты были эффективными и приносили доход, необходимо постоянно взаимодействовать с ними, предоставлять специальные условия по приобретению продукции, оперативно реагировать на изменения в их активности.

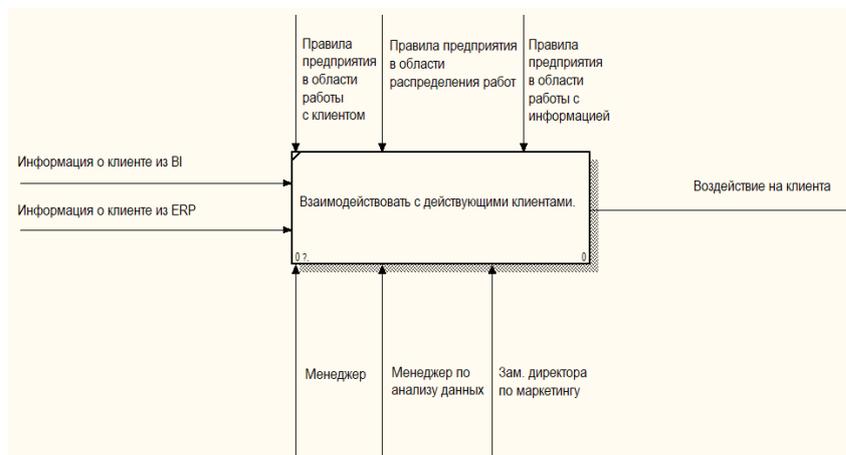


Рисунок 1 – Контекстная диаграмма AS-IS верхнего уровня процесса взаимодействия с действующими клиентами

Для полного понимания процесса взаимодействия с клиентами была произведена декомпозиция верхнего уровня и рассмотрены более глубоко и подробно каждый из этапов. Процесс работы с клиентами состоит из нескольких этапов: проанализировать данные о конкретном клиенте, сформировать задачи, назначить задачи на определенных менеджеров и исполнение задач менеджерами (различная активность по отношению к клиенту: звонок, встреча и т.д.). Исходя из информации, представленной на рисунке 1, можно сказать, что данный процесс является затратным с точки зрения использования как трудовых, так и временных ресурсов, а, следовательно, требует некоторой переработки и улучшения, а также, по возможности, оптимизации.

На основе всего рассмотренного выше, можно сделать выводы о слабых сторонах имеющейся системы:

- необходимость в специалисте по анализу данных;
- необходимость в привлечении заместителя директора по маркетингу для распределения задач;

- информация о клиенте находится в различных местах;

- большие временные и трудовые затраты на процесс анализа данных.

Возникает необходимость решения описанных выше проблем путем автоматизации процесса работы с действующими клиентами. Пример решения представлен на рисунке 2.

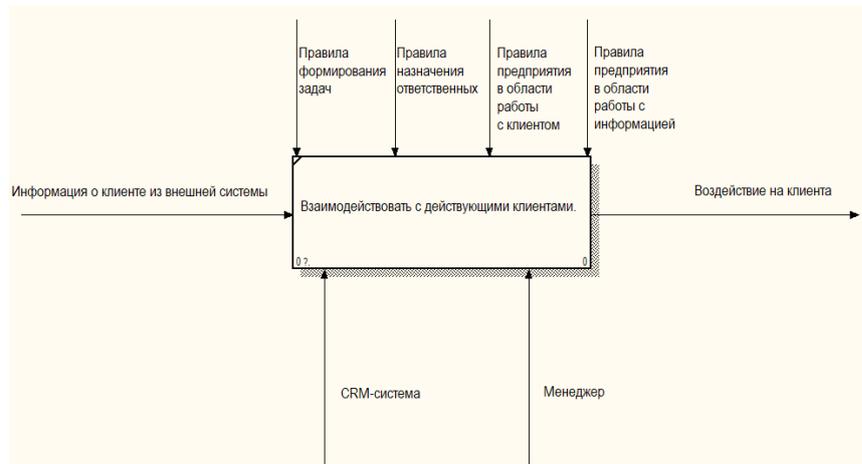


Рисунок 2 – Контекстная диаграмма TO-BE верхнего уровня процесса взаимодействия с действующими клиентами

Основным отличием разработанной модели «ТО-ВЕ» является наличие CRM-системы для анализа данных, которая позволит собирать данные из различных систем воедино, а также вести отчетность.

Задачи, поставленные выше, были решены следующими способами:

- были построены функциональные модели «AS-IS» и «TO-BE» основного процесса предметной области [6];
- была построена информационная модель (логический уровень) разрабатываемого продукта;
- для доработки была выбрана платформа SAP Hybris Cloud For Customers. В качестве среды разработки была выбрана интегрированная среда SAP Cloud Application Studio [7].

После того, как было получено понимание того, как можно применить систему управления взаимоотношениями с клиентами, был проведен анализ существующих бизнес-процессов с целью выявления слабых мест и дальнейшего их улучшения с применением современных технологий. Данное решение представляет собой не только модель программного продукта, но и пока неполную реализацию задуманного. На этапе разработки технического задания были учтены перспективы для дальнейшего развития и внедрения новых бизнес-процессов в рамках разработанного решения.

**Список использованных источников:**

1. Jelonek, D. The Evolution of Customer Relationship Management System/ D. Lelonek// Proceedings of the 19th International Conference on Computers. – Zakynthos Island, Greece – 2015. – ISBN: 978-1-61804-320-7 – P. 29.
2. Rababah, Kh. Customer Relationship Management (CRM) Processes from Theory to Practice: The Pre-implementation Plan of CRM System / Kh. Rababah, H. Mohd, H. Ibrahim// International Journal of e-Education, e-Business, e-Management and e-Learning – Vol. 1, No. 1 – April 2011. – P.22.
3. SAP [Электронный ресурс], – Redefine customer engagement with our next-gen cloud CRM software – SAP Hybris Cloud for Customer. – Режим доступа: <https://www.sap.com>.
4. TechTarget [Электронный ресурс]. – Software as a Service (SaaS). – Режим доступа: <http://searchcloudcomputing.techtarget.com>
5. Contrimo [Электронный ресурс], – SAP Hybris Cloud for Customer. – Режим доступа: <https://www.contrimo.com>.
6. Pietroń, R. Best practices in business process modelling / R. Pietroń // Information Systems in Management. – Department of Informatics, Warsaw University of Life Sciences – 2016. – Vol. 5 (4) 551–562. – ISSN: 2084-5537.
7. SAP [Электронный ресурс], – Create a Cloud Application with our new SAP Cloud Applications Studio – SAP Hybris Cloud for Customer. – Режим доступа: <https://www.sap.com/products/cloud-application-studio.html>.

## СЕРВИС ПО АРЕНДЕ АВТОМОБИЛЕЙ НА ОСНОВЕ МОБИЛЬНОЙ ПЛАТФОРМЫ ANDROID

Корбовский Д.О.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

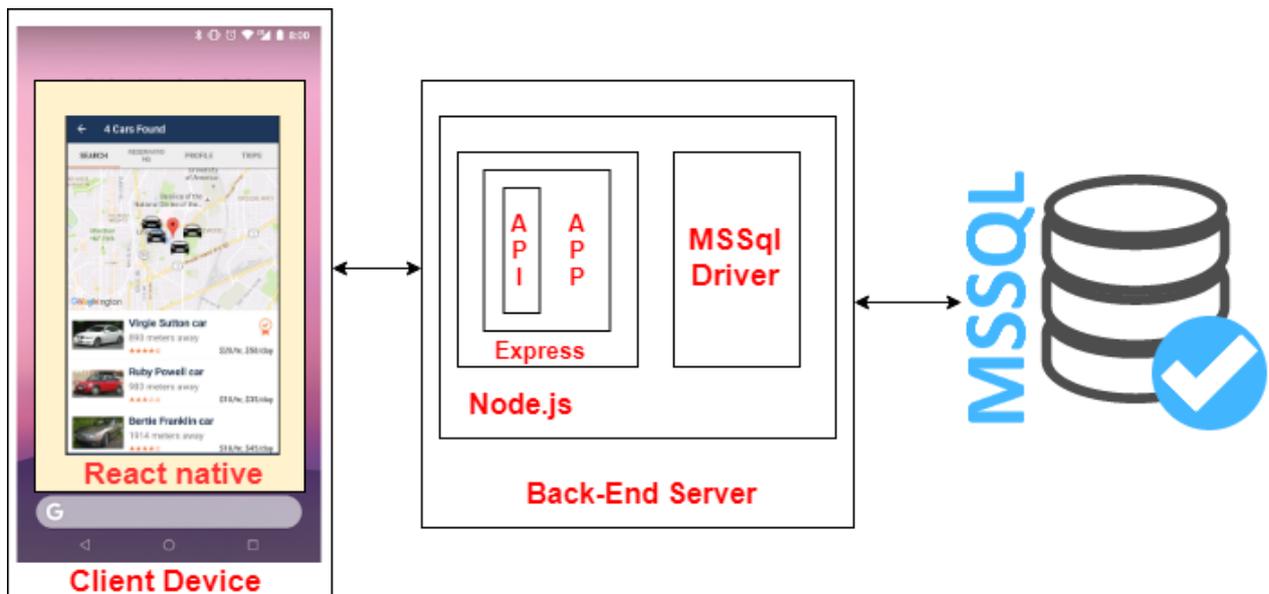
Глухова Л.А. – к.т.н., доцент.

Цель разработки – создать мобильное приложение, позволяющее легко и быстро найти подходящий автомобиль для аренды. Программный продукт реализуется в виде гибридного мобильного приложения под платформу Android.

Приложение позволяет просматривать список или карту с местоположениями доступных автомобилей, фильтруемых по различным параметрам, войти в систему или авторизоваться в системе, оставлять отзывы, добавлять в избранное и бронировать автомобиль. Функционирование в приложении могут осуществлять два вида пользователей: авторизованный и неавторизованный. Неавторизованному пользователю доступна только функция просмотра списка автомобилей. Авторизованный пользователь имеет более обширный функционал, такой как просмотр списка автомобилей, вход в личный кабинет, просмотр истории бронирований, рекомендаций и, непосредственно, бронирование. Для авторизации пользователю необходимо создать аккаунт или воспользоваться авторизацией через социальную сеть.

Для создания гибридного мобильного приложения используется фреймворк React Native. React— JavaScript-библиотека с открытым исходным кодом для разработки пользовательских интерфейсов. React разрабатывается и поддерживается Facebook, Instagram и сообществом отдельных разработчиков и корпораций. React Native — это JS-фреймворк для создания нативно отображаемых iOS- и Android-приложений на основе библиотеки React. Основным преимуществом данной библиотеки является кроссплатформенность и нативность.

В качестве библиотеки, реализующей backend данного приложения, используется Node.js. Это серверная реализация языка программирования JavaScript, основанная на движке V8. Предназначена для создания масштабируемых распределённых сетевых приложений, таких как веб-сервер. В качестве базы данных используется MSSQL.



**Список использованных источников:**

1. Marc H. Node.js in Action / Marc Harter, Mike Cantelon, Nathan Rajlich // Manning Publications, – 2017. – 371 с.
2. Информационные технологии и управление : материалы 49 науч. конф. аспирантов, магистрантов и студентов, Минск, 6–10 мая 2013 г. / Белорус. гос. ун-т информатики и радиоэлектроники ; редкол.: Л. Ю. Шилин [и др.]. – Минск : БГУИР, 2013. – 103 с.
3. Nader Dabit. React Native in Action / Nader Dabit. – Manning Publications Company, 2019. – 300 с.

## МЕТОДИКА ПРОЕКТИРОВАНИЯ КОМБИНАЦИОННОГО УСТРОЙСТВА ВСЕВОЗМОЖНЫХ ПЕРЕСТАНОВОК

*Кохновский С.И.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Иванюк А.А. – д.т.н., профессор*

В данной работе рассмотрено комбинационное устройство для создания всевозможных отображений  $N$  входных сигналов на  $N$  выходов. Предлагается синтез схемы такого устройства для произвольного  $N \geq 2$ .

При создании различных систем встаёт задача обеспечения тайны их внутренней конструкции. Такая задача порождена необходимостью защиты авторских прав. Злоумышленники прибегают к обратному проектированию систем, чтобы заполучить технологии их разработки. Для усложнения обратного проектирования системы можно использовать запутывающие устройства, которые значительно повышают сложность такого проектирования [1]. В работе рассмотрен подход к проектированию такого запутывающего устройства. Каждое устройство имеет  $N$  входных,  $N$  выходных сигналов и ключ, состоящий из набора  $M$  бит, оно может создавать всевозможные отображения входных сигналов на выходы в зависимости от подаваемого ключа.

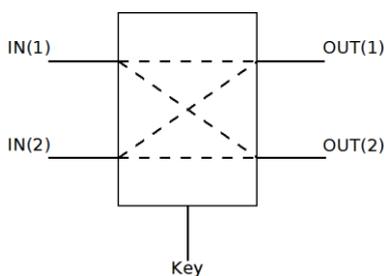
Всевозможное количество ключей для устройства будет равно  $2^M$ , а количество возможных выходных комбинаций –  $M!$ . Тогда будет справедливо следующее соотношение

$$M \geq \lceil N! \rceil,$$

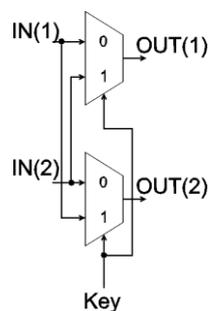
где  $M$  – число блоков, при которых возможно составление всевозможных комбинаций, а  $N$  – число этих комбинаций.

Некоторым комбинациям путей будет соответствовать несколько ключей. Тогда  $2^M - M!$  комбинаций будут повторными, что может негативно сказаться на эффективности работы такого устройства.

Простейшее комбинационное устройство (рисунок 1 а) состоит из блоков, каждый из которых принимает 2 входных сигнала, 2 выходных и бит ключа, который указывает на порядок следования сигналов. При подаче 0 сигналы не изменяют направления, такое направление будет называться прямым, а при подаче 1 сигналы меняют пути, что является противоположным направлением.



(а)



(б)

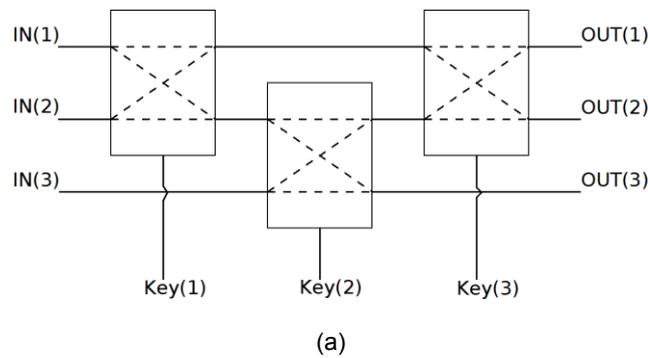
Рисунок 1. Блок (а), внутреннее устройство блока (б)

Каждый блок состоит из мультиплексоров [2], осуществляющих смену направления сигналов по вышеописанной схеме (рисунок 1 б).

Простейшим устройством является устройство с двумя входами и выходами, то есть  $N = 2$ , блок, как определялось ранее. Такому устройству надо будет один бит, чтобы определять направление сигналов, откуда  $M = 1$ .

Пусть будет поставлена задача построения устройства при  $N = 3$  (рисунок 2 а), блок будет иметь входные сигналы  $IN(1)$ ,  $IN(2)$ ,  $IN(3)$  и выходные  $OUT(1)$ ,  $OUT(2)$ ,  $OUT(3)$ . При проектировании такого устройства будем использовать 3 блока, тогда минимально возможное  $M = 3$ . Первый блок будет соединять пути  $IN(1)$  и  $IN(2)$ , второй –  $IN(2)$  и  $IN(3)$ , а третий снова соединит пути  $IN(1)$  и  $IN(2)$ . Первый блок «перемешивает» пути  $IN(1)$  и  $IN(2)$ , тогда выходы  $OUT(2)$  и  $OUT(1)$  будут достижимы из  $IN(1)$  и  $IN(2)$ . Блок №2 обеспечит достижимость  $OUT(3)$  из  $IN(1)$  и  $IN(2)$ , также  $OUT(2)$  из  $IN(3)$ . Можно заметить, что недостижимым остался только выход  $IN(1)$  из входа  $OUT(3)$ , чтобы это компенсировать и есть третий блок. Таким образом, блоки до центрального включительно в комплексе соединяют вход  $IN(1)$  со всеми остальными выходами, а блок с номером 2, расположенный в центре, являющийся первым для входа  $OUT(3)$ , вместе с последующими блоками соединяют все выходы для входа  $OUT(3)$ . Все остальные входы были соединены с выходами первой половины блоков или второй.

Пусть будет рассмотрена матрица для описания устройства при  $N = 3$  (рисунок 2 б). Строка будет обозначать блок, соединяющий 2 пути, а столбцы обозначают соединяемые пути. Откуда следует, что в строке может быть только 2 единицы, обозначающие, что данные пути соединяются блоком. Верхняя строка будет обозначать блок-1, вторая – блок-2 и так далее. Столбцы в строке могут переставляться местами, что не влияет на корректность работы устройства. При таких перестановках будут получены матричные представления схем устройств, которые отличаются порядком соединения путей, однако другими характеристиками они отличаться не будут. Строки произвольно переставляться не могут.



	OUT(1)	OUT(2)	OUT(3)
Key(1)	1	1	0
Key(2)	0	1	1
Key(3)	1	1	0

(б)

Рисунок 2. Комбинационное устройство с тремя входами (а), матрица, описывающая комбинационное устройство с тремя входами (б)

Следуя логике построения устройства при  $N = 3$ , можно обобщить на случай с произвольно заданным количеством  $N$  входных и выходных путей (рисунок 3). Будет последовательно соединяться первый и второй пути, второй и третий, ... ,  $N-1$  и  $N$  (всего  $N-1$  блок). Эта последовательность гарантировано ставит на выходную позицию  $N$  необходимый входной сигнал. Таким образом, из первого входа будут достижимы  $N$  выходов, из второго, в худшем случае, будет перенаправлен сигнал в первый после их обмена, то же произойдёт с входными сигналами 3, 4, ...,  $N$ . Теперь будет обеспечена достижимость  $N$ -ым входом 1-го выхода путём добавления блоков, последовательно соединяющих пути  $N-1$  и  $N-2$ ,  $N-2$  и  $N-3$ , ..., 3 и 2, 2 и 1. После построения этой последовательности своё место займёт сигнал, который должен стоять на первом выходе. Далее будут пересекаться таким же образом остальные пути, чтобы расставить оставшиеся сигналы по нужным позициям. Пересекутся пути с 2-го блока до  $N-1$ , потом с  $N-2$  по 3-й и так далее. Тогда для устройства с  $N$  выходными и  $N$  выходными путями необходимо  $M = N - 1 + N - 2 + N - 3 + \dots + 2 + 1$  блоков, которые нуждаются в  $M$ -битном ключе. После преобразования получим

$$M = \frac{N}{2} * (N - 1).$$

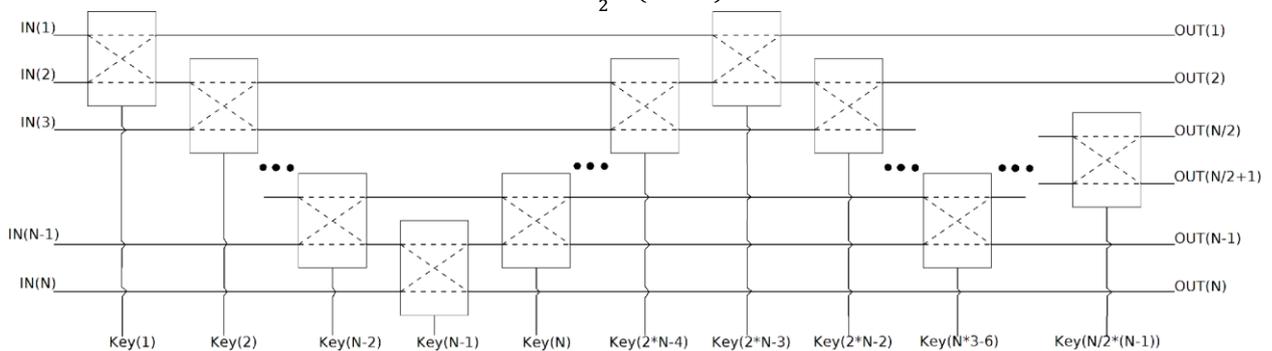


Рисунок 3. Комбинационное устройство с  $N$  входными сигналами

Рассмотренная схема устройства строится достаточно просто, но такая схема минимальной не будет. Ранее была выведена формула связи  $M$  и  $N$ , из неё следует возможность построения устройства с минимальным размером ключа  $M = [N!]$ . Возможны и другие конфигурации устройств, которые будут давать необходимый результат.

Таким образом, была разработана схема комбинационного устройства, осуществляющего отображение входных сигналов на выходы согласно подаваемому ключу.

**Список использованных источников:**

1. Privacy protection of VLSI circuits through high level transformation based obfuscation [Electronic resource]. – Mode of access: [https://www.researchgate.net/publication/319135767\\_PRIVACY\\_PROTECTION\\_OF\\_VLSI\\_CIRCUITS\\_THROUGH\\_HIGH\\_LEVEL\\_TRANSFORMATION\\_BASED\\_OBFUSCATION](https://www.researchgate.net/publication/319135767_PRIVACY_PROTECTION_OF_VLSI_CIRCUITS_THROUGH_HIGH_LEVEL_TRANSFORMATION_BASED_OBFUSCATION). – Date of access: 18.03.2019.
2. Угрюмов, Е.П. Цифровая схемотехника: учеб. пособие для вузов/ Е. П. Угрюмов. – СПб. : БХВ-Петербург, 2010. – 816 с.

## СРАВНЕНИЕ АЛГОРИТМОВ ТЕСТИРОВАНИЯ ЧИСЕЛ НА ПРОСТОТУ. ПОСТРОЕНИЕ ОПТИМАЛЬНОГО АЛГОРИТМА

*Кривошеев А. В., Совпель Д. С.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Стройникова Е. Д. — ассистент кафедры информатики*

Целью работы является рассмотрение различных видов алгоритмов определения простоты чисел, выявление оптимальных алгоритмов для конкретных наборов данных, нахождение оптимизационных шагов для конкретных тестов на простоту. Основные объекты изучения — истинные и вероятностные тесты на простоту, их различия и специфика полученных результатов для родственных тестов.

Большие простые числа имеют весомое практическое применение в криптографических алгоритмах. Эффективность множества современных шифров зависит от того, насколько простое число велико. Например, для шифра RSA, на данный момент использующего открытый 2048-битный ключ, состоящий из произведения двух простых чисел. Увеличение мощности технических средств может позволить факторизовать открытый ключ (на данный момент максимальное большое факторизованное число RSA-768 содержит 232 цифры) [1], что приведет к небезопасности шифров, основанных на открытом ключе. Поэтому эффективный поиск больших простых чисел способен улучшить уже существующие шифры.

Тесты на простоту — вид алгоритмов, целью которых является определение простоты заданного числа. Среди тестов различают два подвида: истинные и вероятностные тесты.

Истинные тесты позволяют точно определить простоту числа. Однако они обладают рядом существенных недостатков, ограничивающих их применение. Среди них: условность — тест работает лишь для ряда заранее определенных чисел (тесты для чисел Мерсенна, Ферма, Прота), малая скорость обработки данных — полиномиальное время. Среди этих тестов в нашей работе были рассмотрены и оптимизированы следующие тесты:

- Тест Люка, основанный на одноименной теореме, работающий за полиномиальное время. В данном тесте определяется простота числа  $2^p - 1$  за полиномиальное время от битовой длины числа  $p$ . Именно благодаря данному тесту числа Мерсенна почти всегда были самыми большими из найденных простых чисел.
- Тест Адлемана — Померанса — Румели — универсальный тест, работающий за  $O(\log n^{c \cdot \log \log n})$ . Данный алгоритм может быть применим на практике в связи с медленнорастущей сложностью.
- Тест Агравала — Каяла — Саксены — универсальный тест, основанный на обобщенной теореме Ферма, работающий за  $O((\log n)^6)$ . Главная особенность данного алгоритма состоит в том, что он полиномиален, универсален, детерминирован и безусловен в отличие от всех предыдущих алгоритмов [2].
- Тесты с использованием эллиптических кривых. Данные тесты являются одними из самых быстрых из известных методов проверки числа на простоту. В худшем случае сложность такого теста будет составлять  $O((\log n)^{5+\Sigma})$ . И для некоторых случаев этот показатель может быть улучшен до  $4 + \Sigma$ .

В свою очередь результатом вероятностных тестов служит лишь вероятность простоты числа. Главным преимуществом таких тестов над истинными является их скорость обработки данных (например алгоритм Миллера — Рабина со сложностью  $O((\log n)^3)$  [3]). Однако в силу вероятности результата ответом могут стать так называемые псевдопростые числа — числа, которые прошли тест на простоту, но таковыми не являются. Для уменьшения частоты появления подобных чисел используются раунды — повторные проверки числа этим же тестом. Примером псевдопростых чисел являются числа Кармайкла. В работе были рассмотрены следующие вероятностные тесты:

- Тест Ферма, основанный на одноименной теореме, работающий за  $O(k \cdot \log n \cdot \log n)$ .
- Тест Миллера — Рабина, использующий теорему Рабина о свидетелях простоты, с минимальной скоростью работы  $O(k \cdot (\log n)^3)$ .
- Тест Соловея — Штрассена, основанный на символе Якоби, со скоростью работы  $O((\log n)^3)$ .

Результатом нашей работы стал универсальный оптимизационный алгоритм, полученный в процессе анализа специфики каждого теста для определенных числовых данных. Данный алгоритм аналогичен тесту BPSW, в основе которого лежит использование нескольких типов тестов. На рисунке 1 представлен график зависимости скорости работы теста BPSW от размера простого числа. Данная методика позволяет добиться уменьшения вычислительной сложности.

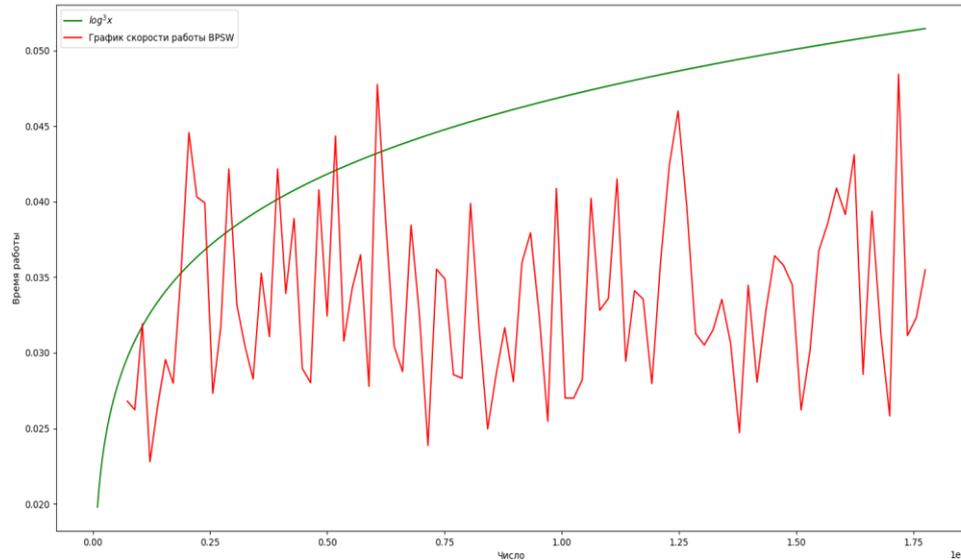


Рисунок 1 — График зависимости скорости работы алгоритма BPSW от размера простого числа

**Список использованных источников:**

1. Factorization of a 768-bit RSA Modulus / Th. Kleinjung [et al.] // *Advances in Cryptology — CRYPTO 2010 : 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15—19, 2010.* — Santa Barbara, 2010. — P. 333—350.
2. Бараш, Л. Ю. Алгоритм AKS проверки чисел на простоту и поиск констант генераторов псевдослучайных чисел / Л. Ю. Бараш // *Безопасность информационных технологий.* — 2005. — №2. — С. 27—38.
3. Шнайер, Б. *Прикладная криптография* / Б. Шнайер ; пер. с англ. — М. : Триумф, 2013. — 816 с.

## ВИЗУАЛИЗАЦИЯ РЕШЕНИЙ НЕКОТОРЫХ МАТЕМАТИЧЕСКИХ ЗАДАЧ В MAPLE

*Кузнечик В. А., Милинкевич М. И.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Калугина М. А. - к. ф.-м. н., доцент*

В работе представлены результаты изучения средств и технологий системы компьютерной алгебры Maple и их применения для визуализации решений некоторых математических проблем. Перечислены примеры задач, для которых созданы процедуры поэтапного решения с элементами анимации: приведение уравнений кривых и поверхностей второго порядка к каноническому виду, разложение функции в ряд, решения систем дифференциальных уравнений, решение краевой задачи на колебания поверхности барабана и построение векторного поля.

Наглядность поставленной задачи является важной частью не только для понимания процесса решения, но и для исследования его в динамике, которая легче воспринимается зрительно. Анимация этапов решения, кроме этого, даёт выразительное представление о скорости явления. Студентам визуализация позволяет находить закономерности, систематизировать найденные решения и моделировать определённую ситуацию при исследовательской работе.

Одной из задач, геометрический смысл которой полезен при обучении и на практике, является задача приведения кривых и поверхностей второго порядка к каноническому виду. Ниже приведено ее решение на примере эллипса, заданного уравнением  $5x^2 - 6xy + 5y^2 - 24x - 32 = 0$  (рис. 1 а), реализованное с помощью созданной процедуры. Для решения этой задачи в Maple можно построить матрицу квадратичной формы левой части уравнения, затем с помощью команд *EigenVectors* и *GramSchmidt* из пакета *LinearAlgebra* найти ее собственные векторы, из которых построить ортонормированный базис. Таким образом, получается базис, образованный поворотом исходного на некоторый угол (рис. 1 б). Ещё одним действием можно продемонстрировать параллельный перенос

центра построенного базиса в точку канонической системы эллипса (рис. 1 в). При выбранном порядке базисных векторов получен эллипс, у которого параметры связаны отношением  $b > a$ , поэтому, изменив его, можно получить классический вид эллипса (рис. 1 г) с уравнением  $\frac{8}{77}Y^2 + \frac{2}{77}X^2 = 1$ .

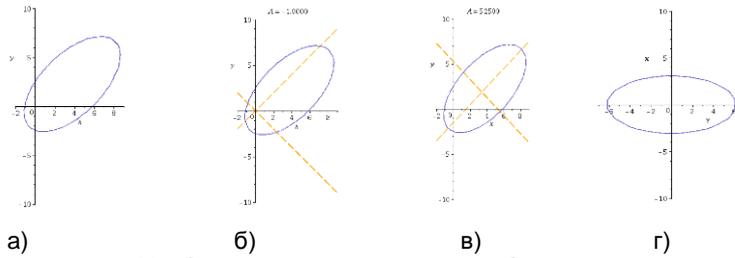


Рисунок 1 — Изображение эллипса в разных базисах

Аналогично можно визуализировать переход к базису из собственных векторов для поверхности второго порядка (рис. 2). На примере уравнения  $3x^2 - 7y^2 + 3z^2 + 8xy - 8xz - 8yz + 10x - 14y - 6z - 8 = 0$  получено каноническое уравнение  $9X^2 + Y^2 - 9Z^2 = 1$ . Это уравнение однополостного гиперboloида. В Maple сразу показаны преобразования по осям.

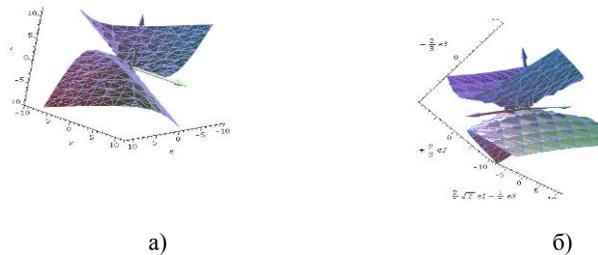


Рисунок 2 – Однополостный гиперboloид в «старом» и «новом» базисах

На рисунке 3 а) в одной системе координат изображены графики частичных сумм разложений заданной функции в ряды Тейлора и Фурье (по тригонометрической системе функций и по многочленам Чебышёва и Лежандра). Кроме этого, Maple предоставляет возможности 3D-анимации разложения функции двух аргументов в ряд Тейлора (рис. 3 б).



Рисунок 3 — Графики аппроксимирующих функций

С помощью Maple удобно визуализировать решение задачи Коши для систем дифференциальных уравнений первого порядка. Важным для понимания является геометрический смысл дифференциального уравнения. Пакет DEtools предоставляет функционал для построения поля направлений дифференциального уравнения с возможностью анимации данного процесса (рис. 4).

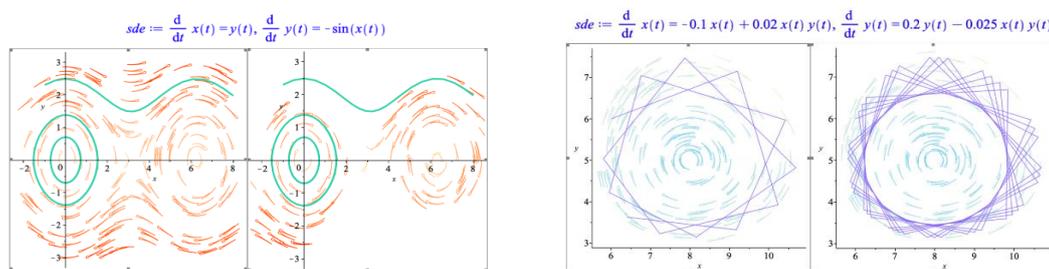


Рисунок 4 — Решения систем дифференциальных уравнений в поле направлений

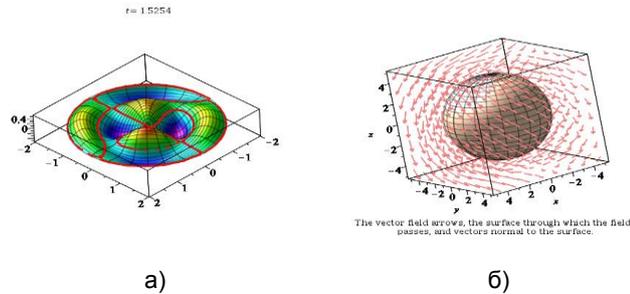


Рисунок 5 — Демонстрация решения других задач в Maple

В завершение приведён ряд других задач, визуализация решения которых возможна в Maple. Например, удалось построить процедуру для анимации колебаний поверхности барабана, которые описываются с помощью функций Бесселя (рис. 5 а). С помощью функций, расположенных в пакете Student[VectorCalculus], удалось построить векторное поле, проходящее через поверхность сферы (рис. 5 б).

Приведённые выше примеры затрагивают темы, которые часто встречаются студентам на практике, и это лишь малая часть задач, решение которых можно визуализировать с помощью Maple.

**Список использованных источников:**

1. Garvan, Frank (Frank G.), 1955-The Maple book / by Frank Garvan. p. cm.Rev. ed. of: Maple V primer. C1997. Includes bibliographical references and index. ISBN 1-58488-232-8 (alk. paper) 1. Maple (Computer file) 2. Algebra—Data processing. I. Garvan, Frank (Frank G.), 1955- Maple V primer. II. Title.
2. Maplesoft [Электронный ресурс]. – Режим доступа : <https://www.maplesoft.com/support/help/Maple/view.aspx?path=worksheet%2freference%2fPlottingGuide>

## НЕЙРОННЫЕ СЕТИ ДЛЯ ПРОГНОЗИРОВАНИЯ РЕЗУЛЬТАТОВ СОРЕВНОВАНИЙ ПО КИБЕРСПОРТИВНЫМ ДИСЦИПЛИНАМ

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Минчук С. Г.

Жвакина А. В. – канд. техн. наук, доцент

Последнее время все больше растет популярность киберспортивных дисциплин. С каждым годом проводится огромное количество соревнований по всему миру. В некоторых дисциплинах основными навыками, которыми должны обладать игроки, являются скорость, реакция и внимание игроков, в других же, важна именно стратегия. Из-за того, что в этих играх есть большое количество комбинаций, которые можно использовать, мы можем рассматривать их в качестве модели для прогнозирования результатов по ним. Исходя из того, что некоторые киберспортивные дисциплины включены в список настоящих видов спорта, это дает возможность рассматривать их не просто как развлечение, а как серьезное мероприятие.

Прогнозирование результатов игры основывается на определении степени развития конкретного объекта или команды в целом, а также количество ранее произошедших событий. Таким образом мы анализируем состояние объекта, базируясь на его текущих характеристиках или их совокупности, которые произошли ранее и которые происходят в данный момент. Для прогнозирования исхода выбран многослойный перцептрон. Многослойный перцептрон или сети прямого распространения как правило имеют три отличительных признака: - каждый нейрон имеет гладкую (всюду дифференцируемую) нелинейную функцию активации; - сеть содержит один или несколько слоев скрытых нейронов; - сеть имеет высокую степень связности, которая реализуется посредством синаптических соединений. Установлено, что многослойный перцептрон имеет достаточную точность и скорость для прогнозирования временных рядов [2].

Многослойный перцептрон состоит из следующих частей (рис.1):

- вход (*input*) нейронной сети;
- выход (*output*) нейронной сети;
- скрытые, слои с большим количеством синаптических соединений.

Рассмотрим подробнее структуру искусственного нейрона.

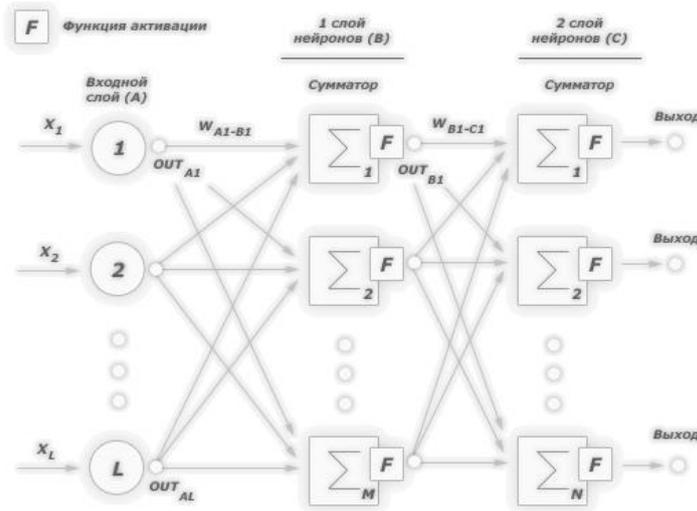


Рис 1. Схема многослойного перцептрона

Состояние нейрона определяется по формуле:

$$S = \sum_{i=1}^n x_i w_i \quad (1)$$

$$y_k = f(s_k + b_k) \quad (2)$$

- где  $y_k$  выходной сигнал нейрона  $k$ ;  
 $f$  функция активации нейрона;  
 $w_{kj}$  синаптические веса нейрона  $k$ ;  
 $b_k$  смещение;  
 $s_k$  сумматор, линейная комбинация входных воздействий;  
 $x_i$  входной сигнал нейрона;  
 $n$  число входов.

Так как основной характеристикой для прогнозирования игры является выбранный персонаж, необходимо получить о нем необходимую информацию, в нужном, для нейронной сети, формате. Исходя из того, что мы имеем только уникальные идентификаторы персонажей и большое количество возможных комбинации, которые привели к положительному исходу, этого недостаточно. Поэтому, для увеличения эффективности прогнозирования игры, использован алгоритм word2vec, суть которого заключается в преобразовании данных о персонажах, которые участвуют в текущей партии, в более высокую размерность. На вход в алгоритм word2vec подается последовательность уникальных идентификаторов персонажей из различных, уже сыгранных матчей. На выходе получается набор векторов, каждый из которого соответствует определенному персонажу.

Прогнозирования партии игры состоит в том, что нейронная сеть определит вероятность победы одной из команд используя данные, которые были собраны до текущего момента.

Входные параметры для нейронной сети:

- добытые ресурсы каждого из участников;
- количество убийств соперников;
- количество собственных смертей;
- количество убийств боссов;
- опыт каждого персонажа;
- разрушенные здания противника;
- информация, о каждом персонаже в партии, преобразованная алгоритмом word2vec.

Все время одной партии игры  $t$  разбито на промежутки  $t = (t_0, t_1, \dots, t_n), t > 0$ . Набор игровых параметров в конкретный момент времени игры  $t$ , представлен вектором  $p(t) = (p_0^t, p_1^t, \dots, p_m^t)$ , где  $m$  – количество входных параметров.

Входным параметром нейронной сети является матрица:

$$X = (p(t_0) p(t_1) \dots p(t_n)) = (p_0^{t_0}, p_1^{t_0}, \dots, p_m^{t_0} p_0^{t_1}, p_1^{t_1}, \dots, p_m^{t_1} \dots p_0^{t_k}, p_1^{t_k}, \dots, p_m^{t_k}), p < n$$

(3)

Выходным параметром нейронной сети является вероятность  $y[0,1]$  выигрыша первой команды.

Данную нейронную сеть, после обучения на большом объеме выборки, можно использовать не только для прогнозирования результатов игр, но также в качестве функции награды для обучения агентов искусственного интеллекта, при использовании алгоритмов обучения с подкреплением. При увеличении объема выборки, предсказатель, который получается в результате алгоритма word2vec, так же можно использовать в качестве помощника при выборе персонажей.

Данное программное обеспечение будет эффективно при анализе матчей, разработке более сложных и интеллектуальных стратегий, а также прогнозировании результатов соревнований.

**Список использованных источников:**

1. Result Prediction by Mining Replays in Dota2. [Электронный ресурс] – Режим доступа: <https://www.diva-portal.org/smash/get/diva2:829556/FULLTEXT01.pdf>. – Дата доступа: 14.03.2018.
2. И.В. Воронов., Е.А. Политов, В.М. Ефременко. Обзор типов искусственных нейронных сетей и методов их обучения. Вестник КузГТУ. – 2007. №3. – С. 38 – 42.

## ВИЗУАЛИЗАТОР ПРОСТРАНСТВЕННО-ВРЕМЕННЫХ ФУНКЦИЙ

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Резунов М. В.*

*Теслюк В. Н. – кандидат физико-математических наук, доцент*

В докладе рассматриваются задачи построения статических и динамических графиков функций; нахождения геометрических тел в массиве полигонов; создания интерактивного графического приложения. В докладе формулируется суть задач, рассматривается алгоритм решения, анализируются и сравниваются результаты выполнения с аналогами.

Согласно научным исследованиям, 90% информации человек воспринимает через зрение. Визуализация — это наглядное представление различной информации. Визуальная информация лучше воспринимается и позволяет быстрее и эффективнее донести до пользователя мысли и идеи.

Сформулируем поставленные задачи:

- 1) Построение статических (постоянных) и динамических (изменяющихся во времени) графиков функций координат и времени в удобном для восприятия виде.  
Для построения используются методы численного анализа и преобразование функций из инфиксной нотации в обратную польскую. [1,2]
- 2) Нахождение геометрических тел в заданном массиве полигонов в пространстве.

Приложение позволяет:

- отображать графики пространственно-временных функций и отслеживать динамику их изменений во времени;
- сохранять и загружать построенные графики, использовать заранее построенные графики;
- отображать динамические функции в любой момент времени из заданного диапазона времени;
- загружать 3D объекты из файла и отображать их;
- находить геометрические тела в заданном массиве полигонов.

На практике приложение используется в технических, физико-математических дисциплинах (математический анализ, аналитическая геометрия, информатика и т.д.) для:

- решения задач визуализации графических статических и динамических данных в пространстве;
- изучения объектно-ориентированных языков программирования, учебных математических и технических дисциплин (математический анализ, аналитическая геометрия, информатика т.д.).

Результатом является график функции, построенный из четырехугольников на точках матрицы по заданной цветовой схеме (рисунки 1, 2), а также минимальные и максимальные значения функций.

Результатом нахождения геометрических тел в заданном массиве полигонов является массив произвольных геометрических тел либо пустое множество.

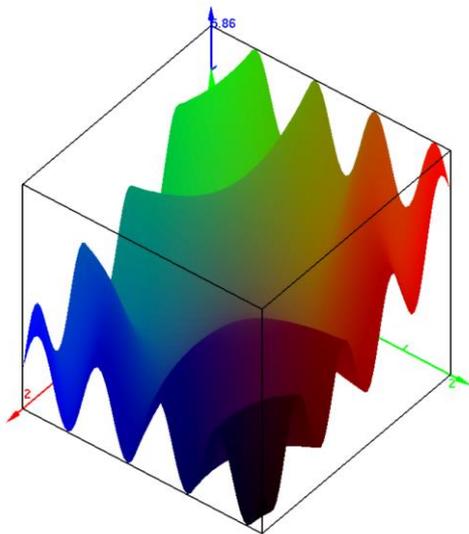


Рисунок 1.  $z = \sin(3.14*x*y)/0.5 + x^2 - 2.71^x$

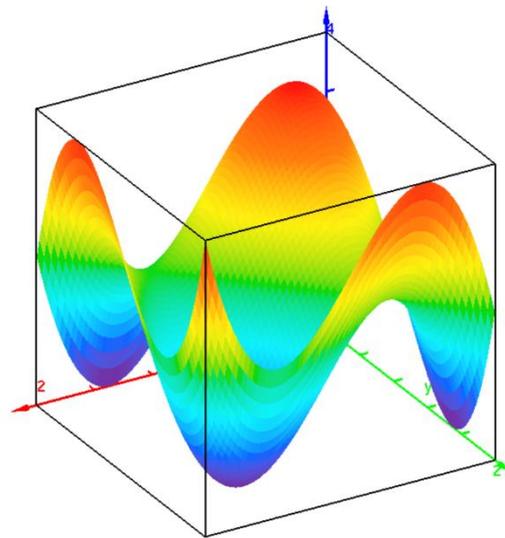


Рисунок 2.  $z = x^3 - 3*x + y^3 - 3*y$

**Список использованных источников:**

1. Методы численного анализа математических моделей. / М. Галанин, Е. Савенков. 2015. — 593 с.
2. Материал сайта [Электронный ресурс]: [https://ru.wikipedia.org/wiki/Обратная\\_польская\\_запись](https://ru.wikipedia.org/wiki/Обратная_польская_запись)

## ФАРМАКОЛОГИЧЕСКАЯ СИСТЕМА ОЦЕНКИ ТЕРАПЕВТИЧЕСКОЙ АКТИВНОСТИ ВЕЩЕСТВ

*Ровдо Н.Р.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Жвакина А.В. – к.т.н., доцент*

Объектом исследования является веб-сайт для оптимизации драг-дизайна. Цель работы – создание программного средства для определения терапевтического класса химического соединения с использованием методов машинного обучения. Фармакологическая система выполнена с целью повышения производительности и качества процесса драг-дизайна, что позволит сделать его более эффективным.

В связи с реформированием экономики, с взятием курса на её инновационное развитие, всё чаще и чаще в повседневной работе в большинстве отраслей начинают использовать различные средства информационно-вычислительной техники и соответственно программного обеспечения. Не стала исключением и медицинская отрасль в государстве – один из важнейших признаков её технологического прогресса и цивилизованности.

В данной работе представлено программное средство, позволяющее повысить эффективность разработки новых лекарственных средств с заранее заданными свойствами. Разработанная система реализована в виде веб-приложения и обеспечивает качественную оптимизацию комбинаторной библиотеки, а также упрощение молекулярного докинга.

Интерфейс программного продукта изображен на рис. 1.

Данное веб-приложение позволяет определить терапевтический класс химического соединения, используя методы машинного обучения и специальный способ ввода и редактирования химических формул.

Кроме того реализованы возможности распознавать химическую формулу, получать её текстовое описание, конвертировать графическое представление в машинный формат, выводить терапевтический класс.

Система может быть улучшена за счет добавления алгоритмов докинга, целесообразного при профессиональной и коммерческой разработке препаратов. Подобный функционал может быть реализован в виде мобильного приложения, синхронизирующийся с основной шиной данных.

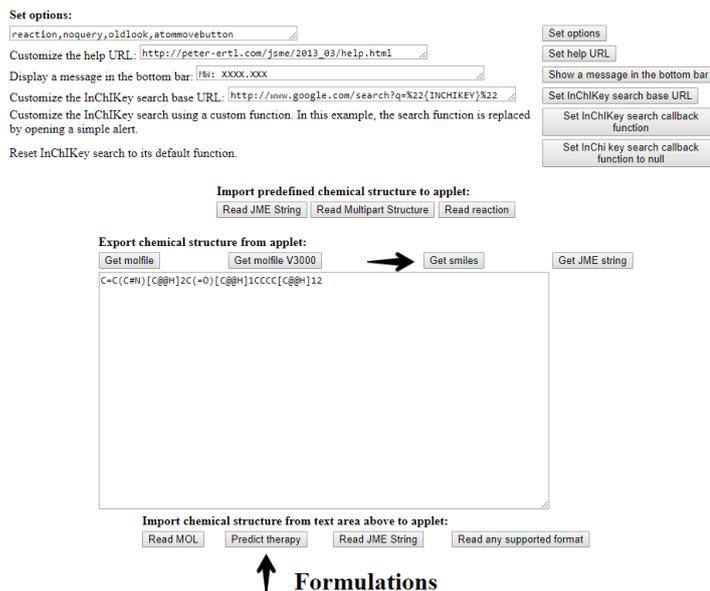


Рисунок 1 – Программный интерфейс

**Список использованных источников:**

1. Daylight: Chemical Information Systems [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://www.daylight.com/dayhtml/doc/theory/theory.smiles.html> – Дата доступа: 05.04.18.
2. И. Куралёнок, Н. Поваров – Деревья решений.
3. Палицын, В.А. Технично-экономическое обоснование дипломных проектов: Метод. Пособие для студ. всех спец. БГУИР. В 4-х ч. Ч.4: Проекты программного обеспечения / В.А. Палицын. – Минск: БГУИР, 2006. – 76с

## АВТОМАТИЗАЦИЯ МЕДИЦИНСКОЙ РЕАБИЛИТАЦИИ ПАЦИЕНТОВ

Родько В.М.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Сиротко С.И. – к.т.н., доцент

В данной работе описаны результаты исследования и работы над программным комплексом для автоматизации медицинской реабилитации пациентов. Целью проекта является создание комплекса программных средств, для обеспечения автоматизации работы тренера-администратора в системе физической реабилитации.

Быстрое развитие в сфере информационных технологий приводит к развитию практически всех сфер деятельности человека. Медицина – яркий тому пример. В ходе проведенного исследования была выявлена бурная цифровизация медицинской сферы. В качестве области для практической работы выбрана медицинская реабилитация пациентов. Начата разработка программного комплекса для ее автоматизации.

На сегодняшний день современные учреждения здравоохранения имеют дело с разнообразными задачами, начиная от необходимости постоянного управления персоналом с целью улучшения эффективности работы, заканчивая наблюдением за физическими показателями пациентов [1]. Для большинства задач не предусмотрено комплексное решение, которое могло бы бесперебойно и регулярно измерять и контролировать выполнение разного рода процессов, медицинского оснащения, а также состояние пациентов и персонала.

Современные устройства, которые подключены к интернету вещей, плотно интегрируются в систему здравоохранения, помогая экономить время медицинских работников и автоматизировать процессы. Применение датчиков и медицинских приборов, которые работают удаленно, предоставляет возможность пациентам воспользоваться самыми современными медицинскими технологиями.

Благодаря развитию дистанционного диагностирования пациентов, сегодня можно значительно увеличить эффективность получения медицинской помощи даже в чрезвычайных ситуациях и преодолеть географический разрыв. Медицинские устройства, которые объединяются с помощью

сети Интернет, могут собирать и передавать информацию о пациентах на большие расстояния за считанные мгновения [2]. Это приводит к значительному снижению медицинских ошибок.

После анализа области, было принято решение разработать комплекс программных средств, для обеспечения автоматизации работы тренера-администратора в системе физической реабилитации. Был произведен обзор аналогичных систем, составлены функциональные и системные требования. Произведено проектирование системы и начата разработка.

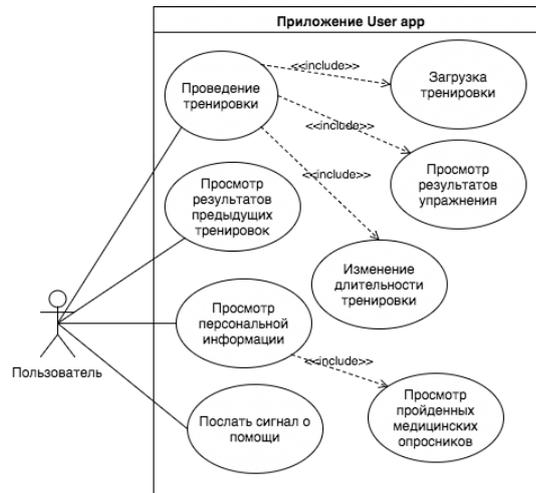


Рисунок 1 – Диаграмма вариантов использования приложения

**Список использованных источников:**

1. Система медицинской реабилитации в Беларуси [Электронный ресурс]. — Режим доступа: <https://minzdrav.gov.by/ru/dlya-belorusskikh-grazhdan/reabilitatsiya.php/>. — Дата доступа: 24.03.2019.
2. Интернет медицинских вещей [Электронный ресурс]. — Режим доступа : [https://www.saymon.info/iot\\_in\\_medicine/](https://www.saymon.info/iot_in_medicine/). — Дата доступа : 24.03.2019.

## АВТОМАТИЗАЦИЯ ПОЛЬЗОВАТЕЛЬСКИХ СЦЕНАРИЕВ КАК УНИВЕРСАЛЬНЫЙ ПОДХОД К АВТОМАТИЗАЦИИ ПРОЦЕССОВ

Сабадаш М. А.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Калугина М. А. – к. ф.-м. н, доцент

В докладе рассмотрены проблемы автоматизации процессов в различных частях организаций. Проанализированы ключевые требования к системам автоматизации, структура и особенности работы сценариев, необходимость поддержки внешних источников данных и сервисов, варианты использования готовых сценариев, а также требования к поддержке и разворачиванию систем. Была выявлена необходимость унификации и упрощения визуального восприятия всех этапов при создании и запуске сценариев, а также обоснована необходимость расширенной поддержки совместного использования систем с поддержкой ролевого доступа и контроля изменения общих данных. На основе проведенного исследования автором предлагается подробная архитектура систем подобного рода, дается детальное описание их работы и полученные характеристики.

В настоящее время для автоматизации обработки различного рода информации повсеместно создаётся специализированное программное обеспечение. Пользователи информационных систем такого рода сталкиваются с большими проблемами определения методики и выбора подходов, позволяющих обеспечить адекватные средства для реализации задач данного класса. Это обусловлено различными факторами: переусложненный интерфейс, отсутствие документированных возможностей, проблемы версионности при обновлении, разнородный подход к работе систем между разными поставщиками, низкая квалификация персонала и др. Упрощение интерфейса и снижение порога вхождения – основная задача для эффективного построения и распространения систем данного класса [1].

Существуют некоторые коммерческие системы, предоставляющие решение вышеописанных проблем. Одна из таких является Tivoli Storage Manager [2]. Данная система предназначена для

настройки автоматизированных сценариев и параметров их запуска, также предоставляет гибкий метод резервного копирования для систем IBM. Возможности Tivoli Storage Manager [2]:

- предоставляет средства для создания сценариев резервного копирования и расписаний для автоматизации задач на предприятиях любого размера;
- позволяет хранить объекты резервного копирования на сервере IBM Tivoli Storage Manager и может выполнять резервное копирование любых данных, кроме rootvg;
- имеет серверное отслеживание и запись каждого запланированного события в базу данных;
- позволяет автоматически обрабатывать административные команды в течение периода времени, когда активировано расписание.

Расписания, запущенные планировщиком, могут выполняться одновременно. Запланированные команды могут обрабатываться последовательно при помощи сценариев, содержащих последовательность команд с ожиданием. Можно также использовать некоторые внешние планировщики, чтобы вызвать клиент администрирования для запуска одной или нескольких административных команд.

К недостаткам Tivoli Storage Manager относятся:

- малый набор настроек и типов операций;
- отсутствие кроссплатформенности и возможности обращения ко внешним ресурсам;
- возможности использования автоматизированных задач являются узконаправленными и предназначены для создания изолированных однопользовательских задач.

Другим существующим решением является Cisco CVP Studio [3]. Данное решение представляет собой гибкий инструмент для автоматизированных задач и может использоваться для создания более сложных сценариев с использованием внешних условий. Cisco CVP Studio имеет самостоятельный отдельный редактор для создания и отладки сценариев, позволяющий тестировать все возможные варианты его использования еще при создании.

Cisco CVP Studio обеспечивает:

- исправление ошибок вызовов и интеграцию цифровой компьютерной телефонии;
- расширенные профили клиентов с использованием контактных данных, связанных, например, набранным номером и уникальным идентификатором вызывающей линии;

К недостаткам данного решения относятся:

- высокая стоимость лицензии;
- зависимость от других продуктов Cisco, которые тоже имеют немалую стоимость;
- отсутствие гибкой настройки логирования, что не позволяет исключать запись строго конфиденциальной информации;
- отсутствие возможности изменения исходного кода для расширения или изменения функциональности;
- отсутствие возможности использования разных протоколов;
- отсутствие веб-интерфейса, который может быть использован на мобильном устройстве или планшете;
- отсутствие возможности изменения условий начала сценария.

Для решения проблем создания и внедрения сценариев, а также запуска автоматизированных пользовательских сценариев предлагается система, которая имеет низкий порог вхождения для пользователей, интуитивно понятный интерфейс, позволяет разделить зоны ответственности, обеспечивает возможности работы с внешними источниками данных, позволяет упростить процессы создания и отладки сценариев, а также позволяет сократить время внесения изменений. Все это сокращает время ее внедрения и улучшает оценки эффективности автоматизированных процессов.

Архитектурная реализация системы представлена на рисунке 1:

Система базируется на основе слоистой архитектуры таким образом, чтобы отдельные блоки были логически разделены на как можно меньшие и независимые фрагменты, что позволит совершать внесение любых изменений в существующую систему и миграцию данных.

Для взаимодействия между приложениями устанавливается экземпляр сервера очередей сообщений, через который они будут взаимодействовать друг с другом. Между сервером приложений и сервером базы данных настраивается коммуникация, которая может быть проверена запуском тестового сценария. К серверам приложений имеется доступ по протоколу HTTP(S) для администрирования, обслуживания клиентов, работы сотрудников организации.

Для использования внешних источников данных добавлена поддержка SOA и REST запросов.

Для запуска сценариев используется доработанная технология Windows Workflow Foundation, что позволяет поддерживать высоконагруженные системы с одновременной работой 10 тыс. сценариев и более.

Таким образом, данная система позволяет получить настоящую универсальность в сфере применения и быть переиспользована в одной и той же организации для разных целей, что является

универсальным решением для организаций любого размера и повышает интерес со стороны потенциальных клиентов.

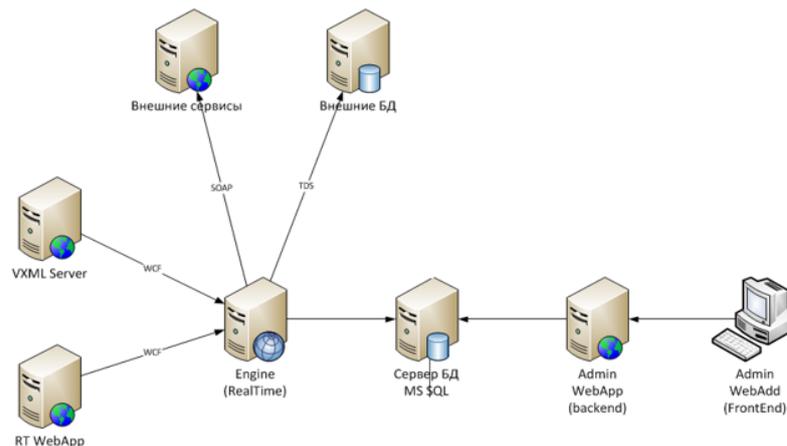


Рис. 1 – Архитектура системы автоматизации пользовательских сценариев

**Список использованных источников:**

1. Дробинцев П.Д. [и др.]. Автоматизация создания верифицированных тестовых сценариев на основе гидов // справочник. В 20 т.– Санкт-Петербург: Санкт-Петербургский государственный политехнический университет, 2013. – 77с.
2. Программное обеспечение IBM [Электронный ресурс]. – Защита данных и управление хранением информации – Режим доступа: <http://www-03.ibm.com/software/products/ru/tivostormanaforsystbackandredo>.
3. Cisco Unified Customer Voice Portal Documentation Guide [Электронный ресурс]. – Release and General Information – Режим доступа: [http://www.cisco.com/c/en/us/td/docs/voice\\_ip\\_comm/cust\\_contact/contact\\_center/customer\\_voice\\_portal/cvp11\\_5/user/guide/CCVP\\_BK\\_1318AE5A\\_00\\_115-cvp-documentation-guide.html](http://www.cisco.com/c/en/us/td/docs/voice_ip_comm/cust_contact/contact_center/customer_voice_portal/cvp11_5/user/guide/CCVP_BK_1318AE5A_00_115-cvp-documentation-guide.html)

## БЛОКЧЕЙН-ПЛАТФОРМА HYPERLEDGER В СФЕРЕ ЗДРАВООХРАНЕНИЯ

*Саскевич А.В.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Стержанов М.В. – к.т.н., доцент*

В данной работе проведено исследование и актуализация возможностей блокчейн-платформы и инструментов Hyperledger в задаче оптимизации работы систем здравоохранения и медицины. Возможности данной платформы позволяют создавать распределенные вычислительные системы, способные выполнять как задачи безопасного хранения информации, так и разнообразные прикладные задачи, например, задачи прогнозирования лечения и распознавания образов. В конечном итоге сравнение с аналогами показало, что исследуемая блокчейн-платформа Hyperledger имеет ощутимые преимущества в решении поставленной задачи.

Эволюция вычислительных систем обусловлена развитием задач бизнеса, а именно задач сбора данных, хранения и обработки полученных результатов. Конкурирующими показателями для данных систем становятся такие характеристики как скорость работы, отказоустойчивость, безопасность и защищенность от внешних воздействий. Глобальная информатизация привела к тому, что централизованные информационные системы становятся достаточно уязвимыми. Не менее уязвимым становится бизнес, работа которого зависит от качественной работы информационных систем.

Ярким примером являются события, когда крупные сервисы прекращали работу по причине форс-мажорных обстоятельств, в результате чего пользователи начинали нести убытки - они теряли связь с клиентами, доступ к своим данным или финансовым средствам.

Так, 22 октября 2018 года крупнейший провайдер системы контроля версий GitHub вследствие внутренней ошибки был недоступен пользователям более семи часов [1]. Это привело к тому, что многие разработчики или аутсорс-компании, пользующиеся услугами данной системы контроля версий, потеряли возможность контролировать разработку своих продуктов, а как следствие, не

могли своевременно обновлять или выпускать программные средства, разработкой которых они занимались.

Особенно критичными такие инциденты становятся в проекции на финансовые сферы или сферы здравоохранения. Потеря информации о финансовых транзакциях или о лечении пациентах может привести не только к крупным финансовым потерям, но и человеческим потерям. Потому данные сферы заинтересованы в таких информационных системах, которые способны обеспечить соответствующий уровень безопасности и защиты.

Появление такого понятия как блокчейн позволило определить новый подход к способу хранения и обработки данных. Так, роль централизованного сервера распределялась между множеством равноправных участников, которые занимались хранением и обработкой информации. Выход из строя одного из участников или потеря данных участников не нарушали общую целостность системы. Введение понятия «смарт-контракт» как некоторого соглашения (например, программного), которое способно автоматически контролировать соблюдение определенных условий (например, актуальность баланса счета) позволяет определить ряд условий и правил того, как данные будут вноситься, изменяться и потребляться.

В сфере здравоохранения необходимо решение таких базовых задач [2], как ведение учета пациентов между отделениями, учреждениями здравоохранения, регионами, странами, учет их анамнеза, сведений о потребленных медикаментах. По-прежнему актуальной, как минимум на постсоветском пространстве, остается проблема бюрократизации и хранения чувствительной информации в бумажном виде, из-за чего сведения могут теряться, вноситься или изменяться самовольно.

Вспомогательные задачи, которые могут быть решены блокчейн-системой и смарт-контрактами – учет лекарств и проведенного лечения, перенос данных между учреждениями здравоохранения, хранение анализов и проведенных исследований, а также обработка данных с целью последующего мониторинга и прогнозирования.

Среди существующих блокчейн-систем основными недостатками являются ориентация на финансовую сферу, отсутствие или ограниченность смарт-контрактов временем работы, скоростью или языком реализации. Так, например, блокчейн сети Bitcoin не имеет смарт-контрактов в явном виде, сеть Ethereum не позволяет разработчику описывать контракты с использованием чисел с плавающей точкой, а блокчейн NEO хоть и предлагает возможность реализации смарт-контрактов на множестве языков программирования, но находится в стадии бета-разработки, а значит, полагаться в определенных видах задач на него можно только на свой страх и риск.

В свою очередь, компания IBM предложила свою блокчейн-платформу Hyperledger [3]. Перечисленные выше недостатки исправлены за счет использования своей системы, основанной на использовании собственных образов, основанных на системе Unix, позволяющий реализовать смарт-контракты без привязки к конкретной архитектуре сети, ограничениям виртуальной машины или операционной системы. В частности, возможность применить язык программирования Python в связке с множеством модулей и библиотек машинного обучения, например, Tensorflow, позволяет создать такой смарт-контракт, который будет реализовывать, например, глубокую сверточную сеть, которая сможет распознавать заболевания на основании анализов, загруженных в сеть блокчейн.

Таким образом, среди рассмотренных существующих на сегодняшний день информационных систем блокчейн-системы позволяют решать разнообразные специфические задачи бизнеса, а узкоспециализированные инструменты, такие как Hyperledger, позволяют описывать решения любой сложности, оставаясь в рамках соблюдения таких характеристик как отказоустойчивость, безопасность, защищенность и скорость работы.

**Список использованных источников:**

1. October 22 Incident Update [Электронный ресурс]. — Режим доступа : <https://github.blog/2018-10-22-incident-update/> — Дата доступа : 20.03.2019.
2. Научный потенциал молодежи – будущему Беларуси: материалы XII международной молодежной научно-практической конференции, УО "Полесский государственный университет", г. Пинск, 6 апреля 2018 г. Ч.1 / Министерство образования Республики Беларусь [и др.]; редкол.: К.К. Шебеко [и др.]. – Пинск: ПолесГУ, 2018. – С. 264-265
3. Hyperledger: blockchain collaboration changing the business world [Электронный ресурс]. — Режим доступа : <https://www.ibm.com/blockchain/hyperledger> — Дата доступа : 15.03.2019.

## ЭФФЕКТИВНОЕ ВСТРОЕННОЕ АНСАМБЛИРОВАНИЕ НЕЙРОННЫХ СЕТЕЙ

Тишковский М.А., Лимонтов А.С., Евжик Д.А.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Чернявский Ю.А. – к.т.н., доцент

Глубокие нейронные сети с большим количеством параметров являются очень мощным семейством алгоритмов машинного обучения. Тем не менее переобучение является серьезной проблемой для таких сетей. Одним из методов борьбы с проблемой переобучения является техника ансамблирования, однако она требует больше количество ресурсов, в том числе и временных. В данном исследовании изучался способ использования техники ансамблирования внутри нейронной сети путем исключения части нейронов из нейросети во время стадии обучения.

Глубокие нейронные [1] сети содержат огромное количество скрытых слоев с нелинейными функциями активации, что позволяет им выучить очень сложную функцию отображения входных данных на целевое значение. Поскольку количество данных зачастую сильно ограничено, настраиваемые коэффициенты нейронной сети могут подстроиться под тренировочные данные, однако выдавать плохие результаты на реальных данных, распределение которых будет хоть немного, но отличаться от распределения тренировочных данных. Для борьбы с эффектом переобучения на тренировочных данных изобрели множество методов, например, регуляризация [2], аугментация данных [3], ансамблирование [4] нескольких моделей.

Идея ансамблирования заключается в том, что независимо тренируется несколько нейронных сетей, после этого для финального предсказания используется усреднение предсказаний всех моделей. Математическое обоснование данного метода заключается в том, что математическое ожидание ошибки ансамбля моделей будет меньше ошибки отдельно взятой модели нейронной сети. Лучших результатов можно добиться, если архитектуры нейросетей будут отличаться друг от друга или будут натренированы на разных данных. Нахождение оптимальных гиперпараметров для каждой сети является достаточно сложной задачей, а сама тренировка множества моделей очень ресурсоемкая операция. Более того, данных может быть недостаточно для того, чтобы натренировать несколько нейронных сетей на разных подмножествах исходных тренировочных данных.

Исключение части нейронов из нейронной сети во время стадии обучения решает обе описанные выше проблемы. Данная техника предотвращает переобучение и предоставляет способ приблизительного комбинирования экспоненциального количества различных нейронных сетей вычислительно эффективным способом. Опишем более подробно работу алгоритма на стадиях обучения и тестирования.

На стадии обучения при прямом проходе слой исключения части нейронов зануляет коэффициенты нейронов по схеме Бернулли [5] с вероятностью  $p$ , где  $p$  – гиперпараметр данного слоя. После этого полученные коэффициенты делятся на  $(1-p)$ , чтобы сумма коэффициентов не менялась. Соответственно при обратном проходе с помощью алгоритма обратного распространения ошибки исключенные нейроны не участвуют в обновлении коэффициентов.

Поскольку при каждом прямом проходе нейросети на стадии обучения случайным образом исключаются некоторые нейроны, нейронная сеть может иметь  $2^n$  состояний, что ведет к аппроксимации ансамбля моделей. Также поскольку для тренировки используется мини-батч [6] техника, где данные для итерации подаются подмножеством исходных данных и коэффициенты нейронной сети обновляются на основе градиента данной мини-батча, то ансамблируются модели, натренированные на различных данных.

Прямой проход стадии тестирования не исключает никаких нейронов и не делит ни на какие коэффициенты.

Таким образом, мы описали алгоритм, который вычислительно эффективно аппроксимирует технику ансамблирования нейросетей, обученных на разных данных.

### Список использованных источников:

1. Deep learning [Электронный ресурс] – Режим доступа: [https://en.wikipedia.org/wiki/Deep\\_learning](https://en.wikipedia.org/wiki/Deep_learning)
2. L1 и L2 регуляризация для линейной регрессии [Электронный ресурс] – Режим доступа: <https://craftappmobile.com/l1-vs-l2-regularization/>
3. [Электронный ресурс] – Режим доступа: <https://medium.com/nanonets/how-to-use-deep-learning-when-you-have-limited-data-part-2-data-augmentation-c26971dc8ced>
4. Two is better than one: Ensembling Models [Электронный ресурс] – Режим доступа: <https://towardsdatascience.com/two-is-better-than-one-ensembling-models-611ee4fa9bd8>
5. Bernoulli scheme [Электронный ресурс] – Режим доступа: [https://en.wikipedia.org/wiki/Bernoulli\\_scheme](https://en.wikipedia.org/wiki/Bernoulli_scheme)
6. A Gentle Introduction to Mini-Batch Gradient Descent and How to Configure Batch Size [Электронный ресурс] – Режим доступа: <https://machinelearningmastery.com/gentle-introduction-mini-batch-gradient-descent-configure-batch-size/>

## ОЦЕНКА АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ. ДИЛЕММА СМЕЩЕНИЯ-ДИСПЕРСИИ

Тишковский М.А., Лимонтов А.С., Подвальников Д.С.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Чернявский Ю.А. – к.т.н., доцент

В данной работе рассмотрена оценка алгоритмов машинного обучения, а также дилемма смещения-дисперсии. Были проведены эксперименты по оценке различных алгоритмов машинного обучения через аппроксимацию функции косинуса, результаты которых представлены в работе.

В настоящее время с помощью алгоритмов машинного обучения решают все большее количество задач. В связи с этим возникает проблема, как правильно подобрать алгоритм машинного обучения под конкретную задачу, а также как оценить полученную натренированную с помощью этого алгоритма модель. Вторая задача решается с помощью введения правильных метрик оценки качества, что само по себе тоже является непростой задачей.

Задача выбора алгоритма машинного обучения является достаточно сложной и требует анализа исходных данных, понимания доменной области, знаний особенностей семейств алгоритмов. В связи с этим вводится понятие дилеммы смещения-дисперсии [1], по которому модели с меньшим смещением в параметре оценки имеют более высокую дисперсию, и наоборот, что в терминах машинного обучения является проблемой недообученности или переобученности алгоритма.

Переобучение [2] – явление, когда построенная модель хорошо объясняет примеры из обучающей выборки, но относительно плохо работает на примерах, не участвовавших в обучении.

Недообучение случается, когда алгоритм машинного обучения не может найти закономерности между исходными данными и целевой зависимостью.

Рассмотрим задачу аппроксимации функции косинуса на промежутке от 0 до  $2\pi$  с помощью различных алгоритмов. На рисунке 1 представлена аппроксимация с помощью линейной регрессии [3], на рисунке 2 с помощью метода опорных векторов [4] с ядром радиальной базисной функции с параметром  $\gamma=1$ , на рисунке 3 с помощью метода опорных векторов с ядром радиальной базисной функции с параметром  $\gamma=100$ .

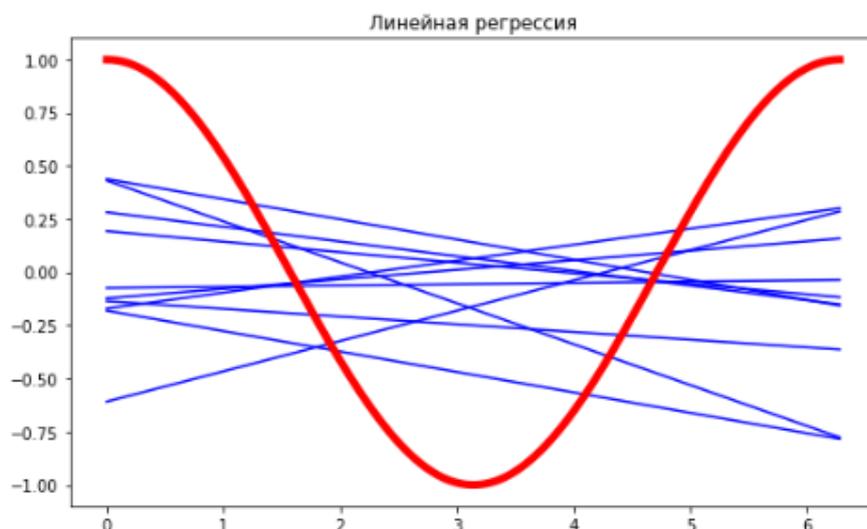


Рисунок 1 – Аппроксимация функции  $\cos(x)$  с помощью линейной регрессии

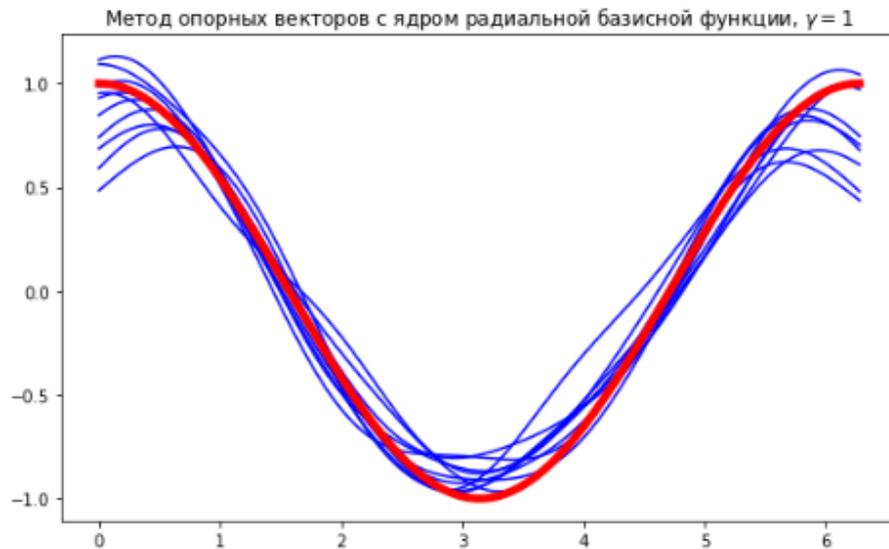


Рисунок 2 – Аппроксимация функции  $\cos(x)$  с помощью метода опорных векторов с ядром радиальной базисной функции с параметром  $\gamma=1$

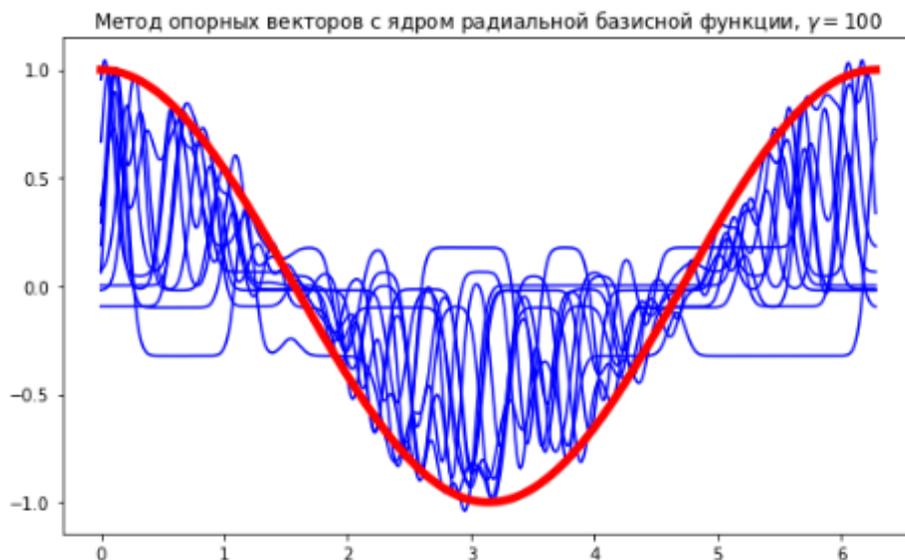


Рисунок 3 – Аппроксимация функции  $\cos(x)$  с помощью метода опорных векторов с ядром радиальной базисной функции с параметром  $\gamma=100$

Проанализируем результат аппроксимации с помощью линейной регрессии. Результаты имеют достаточно средний разброс, что скорее всего обусловлено количеством точек, на которых тренировался алгоритм. Также можно заметить, что результаты имеют очень большое смещение. В связи с этим можно сделать вывод о плохой аппроксимации функции косинуса с помощью алгоритма линейной регрессии.

Проанализируем результат аппроксимации с помощью метода опорных векторов с ядром радиальной базисной функции с параметром  $\gamma=1$ . Результаты имеют очень маленькую дисперсию, что говорит об устойчивости алгоритма. Также можно наблюдать очень маленькое смещение аппроксимации относительно графика настоящей функции, что говорит о том, что алгоритм является хорошим приближением исходной функции.

Проанализируем результат аппроксимации с помощью метода опорных векторов с ядром радиальной базисной функции с параметром  $\gamma=100$ . Результаты имеют большую дисперсию, что говорит о неустойчивости алгоритма. Также мы видим достаточно большое смещение, что говорит о том, что приближение с помощью данного алгоритма является плохим. Причиной этого эффекта является излишне сильная настройка весов модели на тренировочные данные, что ведет к плохой аппроксимации на данных, которые алгоритм еще не видел. В терминах машинного обучения данный эффект называется переобучением.

**Список использованных источников:**

1. Bias-variance tradeoff [Электронный ресурс] – Режим доступа: [https://en.wikipedia.org/wiki/Bias%E2%80%93variance\\_tradeoff](https://en.wikipedia.org/wiki/Bias%E2%80%93variance_tradeoff)

2. Overfitting [Электронный ресурс] – Режим доступа: <https://en.wikipedia.org/wiki/Overfitting>
3. Linear regression [Электронный ресурс] – Режим доступа: [https://en.wikipedia.org/wiki/Linear\\_regression](https://en.wikipedia.org/wiki/Linear_regression)
4. Support-vector machine [Электронный ресурс] – Режим доступа: [https://en.wikipedia.org/wiki/Support-vector\\_machine](https://en.wikipedia.org/wiki/Support-vector_machine)

## АНАЛИЗ ЭФФЕКТИВНОСТИ РЕКЛАМЫ КАК ETL ПРОЦЕСС

*Харитонов Н.В., Хоронько М.П., Медунецкий М.А.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Стержанов М.В. – к.т.н., доцент*

В данной работе рассматривается механизм оценки эффективности интернет рекламы при помощи стека технологий Big Data. Речь пойдет о ETL процессе (реализованном на планировщике задач Luigi) с помощью которого мы собираем, храним и обрабатываем большие объемы данных. В качестве результата будет представлена архитектурная схема платформы, осуществляющей данный процесс, и пример возвращаемых данных - еженедельный отчет, включающий в себя статистические данные рекламной кампании. Эволюция вычислительных систем обусловлена развитием задач бизнеса, а именно задач сбора данных, хранения и обработки полученных результатов. Конкурирующими показателями для данных систем становятся такие характеристики как скорость работы, отказоустойчивость, безопасность и защищенность от внешних воздействий. Глобальная информатизация привела к тому, что централизованные информационные системы становятся достаточно уязвимыми. Не менее уязвимым становится бизнес, работа которого зависит от качественной работы информационных систем.

В настоящее время наблюдается устойчивый рост интереса к практическому применению технологий Big Data в сфере маркетинга. Среди решаемых проблем можно выделить задачи повышения конкурентоспособности, создания новых услуг, совершенствования управления взаимоотношениями с клиентами. В результате развития Интернет, социальных сетей, и иных сетевых сервисов непрерывно растут потребности в информационных продуктах и услугах. Чтобы предлагать клиентам такие услуги, предприятиям приходится анализировать большие объемы данных из различных источников. Поэтому накопленная информация становится стратегически важным активом, от эффективности управления которым существенно зависят результаты деятельности предприятий.

В рамках данной работы мы представляем платформу для анализа эффективности интернет рекламы с использованием стека технологий Big Data.

**Big Data [1]** – обозначение структурированных и неструктурированных данных огромных объемов и значительного многообразия, эффективно обрабатываемых горизонтально масштабируемыми программными инструментами, появившимися в конце 2000-х годов и альтернативных традиционным системам управления базами данных и решениям класса Business Intelligence.

Нами предлагается использование следующих параметров для оценки эффективности интернет-рекламы: количество просмотров; количество переходов; количество нажатий мыши по рекламной области; местоположения; время просмотра. Мы логируем все доступные акты показа рекламы, агрегируем их, проводим обработку, а затем создаем отчет за заданный период времени (неделя, месяц, год).

Отчет включает в себя: данные о рекламодателе; рассмотренный период; количество показов; количество переходов; отношение переходов к показам; количество уникальных местоположений; количество уникальных пользователей; количество уникальных устройств и т. д.

Для более детального описания, представим архитектуру платформы в виде диаграммы на рисунке 1:

### Представление архитектуры и стека используемых технологий

Архитектура системы представляет собой ETL[2] процесс, организованный на кластере компьютеров с использованием технологий Hadoop и HDFS. Рассмотрим каждый из шагов данного процесса.

**Extract.** На этом шаге мы собираем данные из различных источников (массива социального Веба, логов действий пользователей, корпоративных баз данных, разнообразных датчиков, внешнего набора данных и т.д.), а затем передаем их на следующий этап для выполнения преобразований. В данном случае для извлечения данных нами используются RTB[3] аукционы и подход “1x1 pixel”. Каждый показ рекламы записывается в HDFS[4] базу данных (см. [hdfs://logs](https://hdfs://logs) на Рис. 1). Каждая запись

содержит в себе такие поля как “время показа”, “идентификатор рекламодателя”, “идентификатор устройства”, “географическая широта”, “географическая долгота” и т.д.

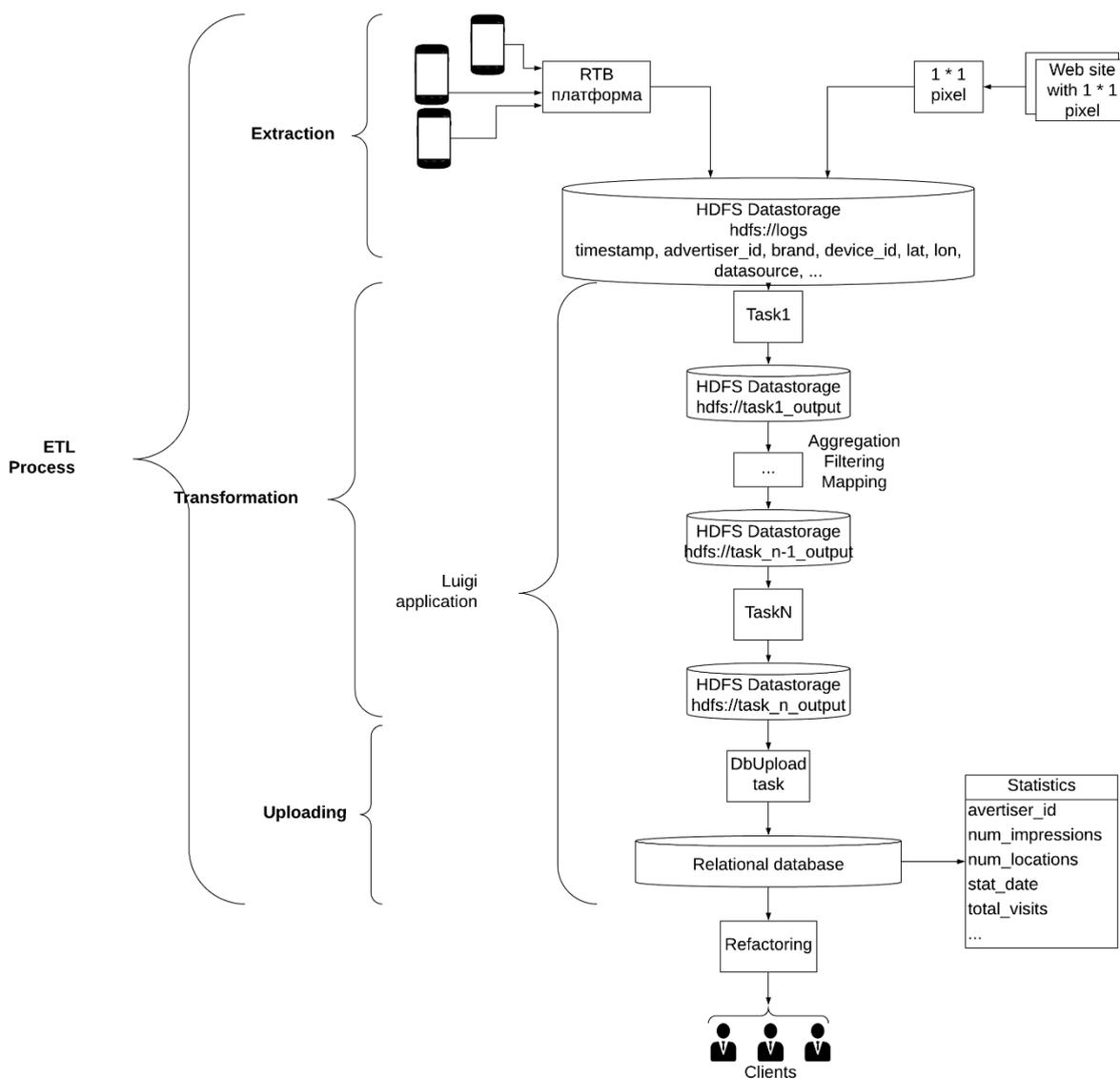


Рис 1. Представление архитектуры и стека используемых технологий

**Transform.** Данный шаг является «сердцем» любого ETL. Здесь требуется ясное понимание бизнес задач, а также наличия базовых знаний в предметной области. В нашей системе на данном шаге производится фильтрация, группировка и агрегирование извлеченных данных и вычисление информации, которая будет представлена в отчете. Ежеминутно мы получаем сотни тысячи строк логов (см. Рис. 2) от RTB платформы и REST сервисов (1 \* 1 pixel).

```
1 df = spark.read.parquet("hdfs://logs/dt_log=2019-01-15")
2 df.count()/(24*60)
3 Out[1]: 460148|
```

Рис 2. Расчет количества поступающих в минуту строк логов

Для их преобразования используется фреймворк для распределенной обработки данных Apache Spark [5] (в частности PySpark) и планировщик задач Luigi[6]. Задачи образуют граф зависимостей, а фреймворк разрешает зависимости, отслеживает граф выполнения, управляет запуском задач, обрабатывает ошибки с возможностью перезапуска нужных задач, распределяет ресурсы рабочих процессов с возможностью параллельной работы независимых частей графа задач.

**Load.** Наконец, мы загружаем обработанные данные в базу MySQL. Полученный набор данных используется для генерации отчетов (пример отчета на Рис. 3), которые затем отправляются рекламодателям.

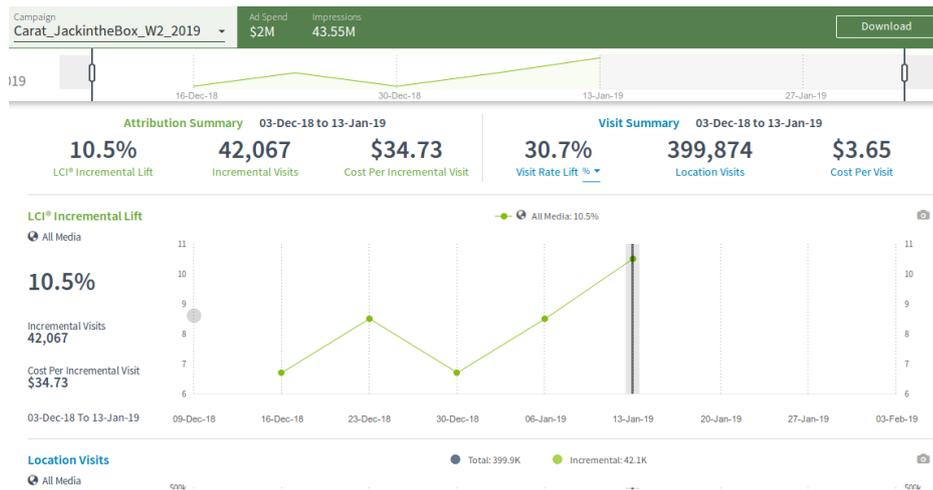


Рис 3. Пример отчета о рекламной кампании

### Вывод

Данная система обрабатывает более полумиллиона событий в минуту, обрабатывает их и выдает отчет в форме, удобной для клиентов. В результате клиент может оценить эффективность рекламной кампании. Планируется перевод ключевых модулей с языка Python на Scala, что позволит ускорить работу системы и увеличить отказоустойчивость.

#### Список использованных источников:

1. Breur, Tom. "Statistical Power Analysis and the contemporary "crisis" in social sciences". *Journal of Marketing Analytics*, July 2016 – 65 с.
2. Loshin, D. *Business Intelligence - 2nd / Loshin, D. - Morgan Kaufmann, 2012. - 400 с.*
3. Stange, M. *Real-Time Advertising. Business & Information Systems Engineering / M. Stange, B. Funk. - Fakultät Wirtschaftswissenschaften Leuphana University Lüneburg Lüneburg Germany, 2014. - 308 с.*
4. White, T. *Hadoop: The Definitive Guide, Fourth Edition / T. White. – Gravenstein Highway North, Sebastopol: O'Reilly Media, 2015. – 728 с.*
5. Karau, H. *Learning Spark / h. Karau, A. Konwinski, P. Wendell, M. Zaharia. – Gravenstein Highway North, Sebastopol: O'Reilly Media, 2015. – 255 с.*
6. GitHub [Электронный ресурс] / Luigi. - Режим доступа : <https://github.com/spotify/luigi>. - Дата доступа : 23.12.2018.

## ГИБРИДНАЯ СТРАТЕГИЯ ОТОБРАЖЕНИЯ ЛОГИЧЕСКИХ АДРЕСОВ В ФИЗИЧЕСКИЕ В НАКОПИТЕЛЯХ ИНФОРМАЦИИ НА ОСНОВЕ ФЛЭШ-ПАМЯТИ

Шиманский И.Д.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Заливако С.С. – кандидат техн. наук

Статья посвящена проблеме отображения логических адресов в физические в накопителях информации на основе NAND-технологии флэш-памяти. Выделяются и описываются характерные особенности основных стратегий отображения. Особое внимание уделено гибридной стратегии отображения. Дается сравнение всех рассмотренных стратегий. В качестве ключевого свойства для сравнения используется объем динамической памяти, требуемый для хранения таблиц отображения. Представлены достоинства и недостатки гибридной стратегии относительно остальных, а также вектор работы для более детального изучения её характеристик.

В данной работе в качестве примера устройства используется Samsung K9XXG08UXM [1]. Необходимые характеристики приведены в таблице 1. Размер логической страницы, используемой файловой системой, будем считать равной 4КБ. Стоит подчеркнуть, что в данном примере предполагается, что размеры физических и логических страниц равны.

Таблица 1 – Характеристики Samsung K9XXG08UXM.

Характеристика	Значение	Характеристика	Значение
Объем данных страницы	4КБ	Количество страниц в блоке	64
Объем метаданных страницы	128Б	Количество блоков в plane	2 048
Объем данных блока	256КБ	Количество plane в устройстве	16
Общий объем диска	8ГБ	Общее количество страниц	2 097 152

К задаче отображения адресов можно также отнести сохранение необходимых данных в энергонезависимую память для возможности последующего восстановления всей структуры отображения в случае внезапного отключения питания. Для этих целей можно использовать как отдельные блоки, так и зарезервированное пространство в каждой физической странице. Для простоты будет использоваться второй способ.

Для сравнения рассмотрим сначала два простых метода отображения адресов: отображение на уровне страниц и отображение на уровне блоков [2].

В первом случае каждому номеру логической страницы ( $lpn$ ) ставится в соответствие номер физической страницы ( $ppn$ ). Размер таблицы пропорционален общему количеству страниц в устройстве. Запись данных производится последовательно во все страницы активного блока с соответствующими изменениями в таблице отображения для каждой записанной страницы.

В случае отображения на уровне блоков, логические адреса разбиваются на равные блоки ( $lbn$ ), каждому из которых соответствует номер физического блока ( $pbn$ ). Смещение страницы в логическом и физическом блоках равны. Определить номер блока и смещение в нём можно как показано в соотношении (1):

$$lbn = [lpn / n], \quad (1)$$

$$offset = lpn \bmod n,$$

где  $n$  – количество страниц в блоке.

Таблица пропорциональна количеству блоков в устройстве. При изменении даже одной страницы необходимо перезаписывать весь блок, что быстро изнашивает элементы памяти.

Основная идея гибридной стратегии заключается в комбинировании таблиц блочного и страничного уровней. Большая часть физических блоков представлена на блочном уровне (data-blocks), что обеспечивает малый требуемый объём памяти по сравнению с отображением на уровне страниц. Для обеспечения эффективных записей размера меньше блока, дополнительно выделяется пул блоков (log-blocks), для которых ведётся таблица со страничной гранулярностью.

На рисунке 1 показан пример последовательности из семи записей. Предполагается наличие только четырёх страниц в блоке. Во время первой операции записи логическому блоку 0 назначается физический блок в таблице data-blocks. Производится запись данных по необходимому смещению в блоке, и в метаданных первой страницы блока записывается номер соответствующего логического блока. Вторая запись также производится непосредственно в data-block, так как страница с нужным смещением свободна. Третья запись производится уже в log-block, так как эта страница уже была записана ранее. Для этого логическому блоку 0 дополнительно назначается log-block со страничной гранулярностью из пула. Записи в log-block производятся до полного его заполнения. В метаданных страницы log-block записывается смещение страницы в логическом блоке, которую он представляет. Записи 4-7 производятся аналогично. После того как log-block полностью записан, происходит объединение его с соответствующим блоком данных. Результат объединения записывается в чистый data-block. После чего использованные data-block и log-block могут быть очищены и использоваться повторно.

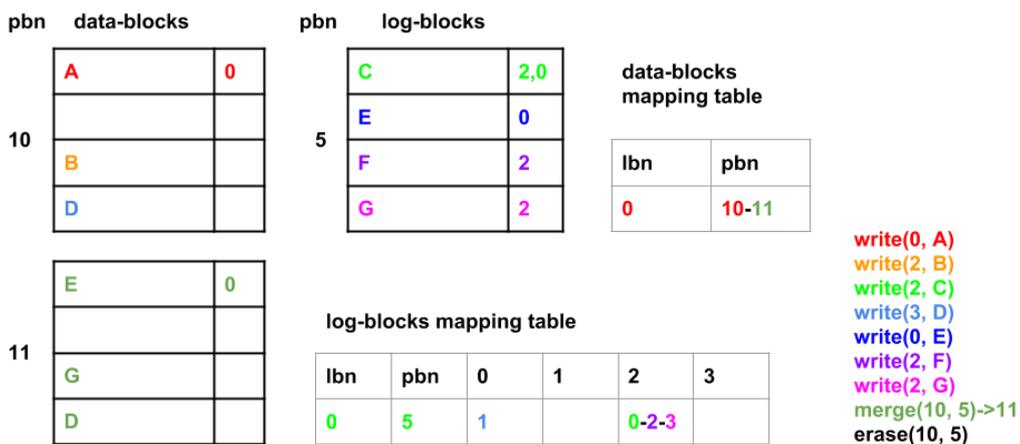


Рисунок 1 – Гибридная стратегия отображения.

Количество блоков в пуле должно быть ограничено, мы будем отталкиваться от избыточного объема памяти накопителя (overprovisioning). В среднем overprovisioning составляет 10% от общего объема памяти [3], половину от него выделим для поддержания пула (5%), что приближенно составляет 100 блоков на plane (подмножество блоков, физически размещенных на одной интегральной схеме), или 1 600 на всё устройство. Имеет смысл равномерно распределять блоки из пула по plane, так как большинство контроллеров NAND поддерживают операцию обратного копирования (copy-back), которая позволяет копировать данные из одной страницы в другую на том же plane [1]. Эта операция читает данные во внутренний регистр данных и записывает из него же, обеспечивая лучшую производительность за счёт отсутствия передачи данных по шине. Если в пуле не остаётся чистых блоков, выбирается наиболее заполненный блок из пула, после чего объединяется с соответствующим блоком данных и очищается.

Необходимое количество памяти рассчитывается исходя из размера одного элемента таблицы и количества записей. Для страничной стратегии достаточно 3Б памяти на элемент таблицы, чтобы хранить номер физической страницы. Количество записей таблицы равно количеству страниц. Для блочной стратегии – 2Б, чтобы хранить номер физического блока. Количество записей равно количеству блоков. Для гибридной стратегии одна таблица идентична блочной. Каждая запись второй таблицы хранит номер логического и физического блоков (2Б + 2Б) и 64 однобайтовых смещения страниц в физическом блоке (64Б). Количество записей второй таблицы, как было показано ранее, равно 1 600. Общий объём памяти двух таблиц: 64КБ + 106КБ = 170КБ.

Таблица 2 – Потребление памяти разных стратегий.

	Страничная	Блочная	Гибридная
Размер элемента таблицы	3Б	2Б	2Б, 68Б
Количество записей	2 097 152	32 768	32 768, 1 600 (~5%)
Размер таблицы	6МБ	64КБ	170КБ

Из таблицы 2 видно, как и ожидалось, меньше всего памяти потребляет блочная стратегия. Также можно отметить, что гибридная стратегия обеспечивает более эффективные записи размера страницы относительно блочной стратегии при гораздо меньшем потреблении памяти относительно страничной стратегии.

К недостаткам можно отнести большую задержку доступа к памяти, так как надо произвести поиск в двух таблицах в случае если log-block и data-block ещё не были объединены, и потребность в дополнительной памяти накопителя для поддержания пула блоков.

Для более детальной оценки рассмотренных стратегий в дальнейшем планируется реализовать их в системе моделирования флэш-памяти “MQSim”. Иначе сложно оценить такие факторы, как задержка выполнения разных операций, сборка мусора, выравнивание износа. Даже те объёмы памяти, которые были рассчитаны в статье, в большинстве случаев не могут храниться в динамической памяти полностью, поэтому необходимо решать задачу кэширования таблиц.

**Список использованных источников:**

1. Design Tradeoffs for SSD Performance / Nitin Agrawal [et al] // USENIX Technical Conference – June 2008.
2. Reconfigurable FTL (Flash Translation Layer) Architecture for NAND Flash-Based Application / Chanik Park [et al] // ACM Transactions on Embedded Computing Systems – July 2008 – Vol. 7 – No. 4 – Article 38.
3. Error Characterization, Mitigation, and Recovery in Flash-Memory-Based Solid-State Drives / Yu Cai [et al] // IEEE – September 2017.

## ПРИЛОЖЕНИЕ ДЛЯ УВЕЛИЧЕНИЯ ПРОИЗВОДИТЕЛЬНОСТИ ТРУДА НА ОСНОВЕ ТЕХНИКИ POMODORO

*Чачура Р.В., Соболевский С.Е.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Жвакина А.В. – к.т.н., доцент*

Во время разработки произведены исследования в области информационных технологий, в частности инструментов для разработки серверной и пользовательской части приложения по улучшению умственной производительности человека.

Increaser – это приложение для тех, кому для выполнения работы необходима умственная концентрация: студенты, инженеры, программисты, математики и другие технические работники. Эти люди хотят делать больше глубокой, целенаправленной работы и меньше мелкой, полной

отвлекающих факторов. Таким образом, они найдут время, чтобы заниматься тем, что имеет значение в их жизни. Отвлечение является главной проблемой, когда мы говорим о Increaser. Из-за отвлекающих факторов мы не можем выполнять работу качественно, испытываем стресс и не находим времени на важные для нас вещи.

В представляемой работе реализована серверная часть для приложения по улучшению производительности с использованием AWS сервисов. Основной API для пользовательской части написан на языке программирования NodeJS. Пользовательская часть написана с использованием React и Styled-Components для UI, Redux для управления состоянием приложения и Redux-Saga для асинхронных операций.

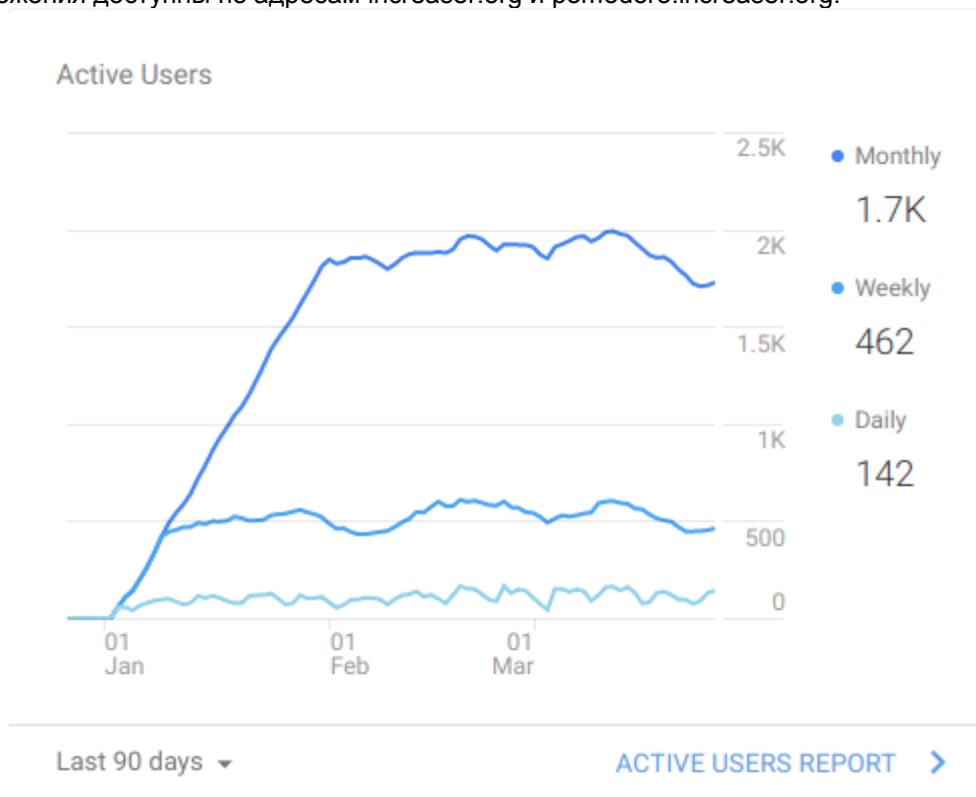
В процессе реализации серверной части приложения решен широкий спектр задач, начиная с описания инфраструктуры через код и заканчивая созданием запросов к базе данных.

Использованы сервисы AWS, такие как ElasticBeanstalk, DynamoDB, S3, Route5S, Certificate Manager, CloudFront, CodePipeline, ECR. Для описания инфраструктуры использовалась технология Terraform. Рассмотрены лучшие практики для организации Terraform модулей и их зависимостей. Написана конфигурация для CI/CD сервисов с помощью CodePipeline. Сборка сервисов происходит в Docker контейнерах. Для хранения пользовательских данных используется NoSQL база данных DynatnoDB, для хранения редко извлекаемых данных – Amazon S3.

Во время разработки приложения реализовано второе приложение pomodoro.increaser.org благодаря тому, что UI вынесены в отдельные репозитории и NPM модули. Для написания серверной части использовался аналогичный стек. Единственное отличие заключается в использовании GraphQL для обмена данными между клиентом и сервером.

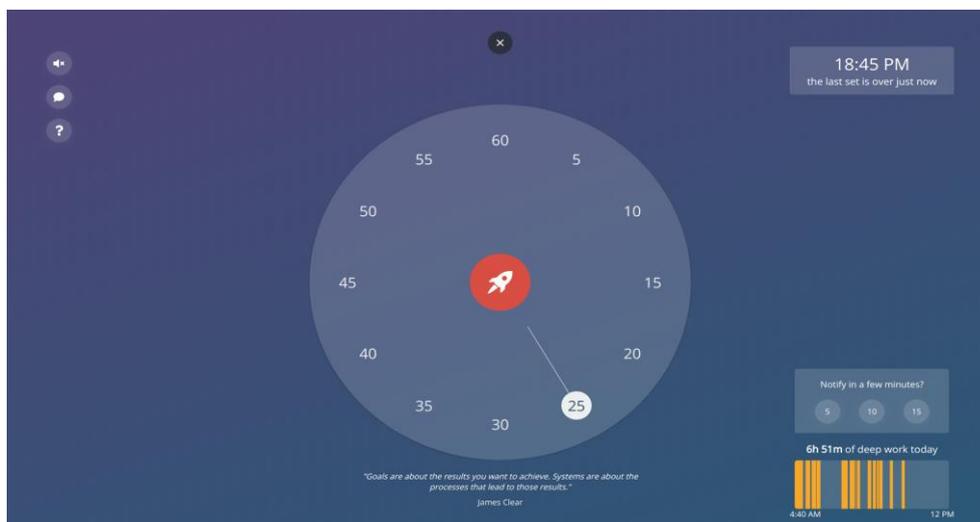
Процесс использования приложения можно представить в цикле из трех шагов. Выбор промежутка времени для работы, выполнение задачи без отвлечений, перерыв. В конце дня можно провести анализ производительности.

Приложения доступны по адресам increaser.org и pomodoro.increaser.org.



Аналитика связанная с количеством активных пользователей

Важным преимуществом приложения является качественное UX и доступность приложения в браузере. Работа над приложением продолжается, мы внимательно прислушиваемся к отзывам пользователей и адаптируем функционал с учетом их пожеланий.



Интерфейс главной страницы pomodoro.increaser.org

**Список использованных источников:**

1. Kief Morris. Infrastructure as Code: Managing Servers in the Cloud/ K. Morris // Physica Status Solidi. A. – 2016. – Vol. 2014, № 12. – P. 2765–2771.
2. AWS Whitepapers. Architecting for the AWS Cloud: Best Practices / AWS Whitepapers. – 5. Aufl. – Canda, 2016. – 232 S.
3. Dan Sullivan.NoSQL for Mere Mortals / D. Sullivan – UK, 2015. – 234-552 S.

**СЕКЦИЯ «ФИЛОСОФИЯ И ЖИЗНЬ»**

## ВИДЕОИГРЫ КАК НОВЫЙ ВИД ИСКУССТВА

*Каленик С.А.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Бархатков А.И. – к.ф.н.*

Наличие в играх заставляющих задуматься идей, их способность вызывать эмоции, использование в них средств других визуальных искусств и музыки позволяют рассматривать видеоигры как не только одну из форм развлечений, но и новую форму искусства.

Существует множество определений термина искусство, исходя из данного факта следует обозначить этот термин для данного доклада. «Искусство — специфическая форма общественного сознания и человеческой деятельности, представляющая собой отражение действительности в художественных образах, один из важнейших способов эстетического освоения мира». [1]

В современном обществе сложилось мнение, что видеоигры — это всего-то один из видов развлечений. Но это далеко не так. Любое произведение, будь то кино, книга или видеоигра - является воплощением идей определенного человека или небольшого круга лиц. Игровые разработчики вносят в свой продукт не только интересные игровые механики, но и какую-то идею, которую они передают через повествование сюжета и общую игровую атмосферу.

Видеоигры включают в себя музыку, а также кино. Только в отличие от кино они лучше передают атмосферу происходящего и дают возможность человеку самому изучать окружающий его мир. Во многих играх сценарий больше, чем у любого фильма и прохождение видеоигры может затянуться на несколько десятков часов, поэтому сценаристы не ограничиваются в идеях и способны полностью изложить свою мысль. Музыка помогает наполнить игру атмосферой, а также позволяет лучше передать чувства главного героя из-за чего игрок копирует поведение главного героя видеоигры.

Также игры способны передавать атмосферу разных исторических этапов формирования человеческой культуры. Так, в серии игр “Assassin’s Creed” игрока пронесет через разные исторические культурные этапы человечества. Игроку предоставляется возможность поближе познакомиться с историей различных этнических групп, а также с различными историческими личностями.

Не стоит забывать и о правильном дизайне игровых уровней. Благодаря этому разработчики наводят людей на разные мысли, и может даже сформироваться вокруг этого круг людей, которые будут строить различные догадки. Так, например, в игре “Grand Theft Auto 5” из-за одной картинке на стене, расположенной на вершине горы, стали появляться кучи различных догадок, что эта картинка значит. Подобно этому, люди также строят различные теории вокруг картины Леонардо да Винчи “Мона Лиза”.

Также есть видеоигры, в которые играли массы людей не из-за какой-то особой игровой механики, а именно из-за той самой идеи, которую аккуратно сквозь всю игру пронесут разработчики. Так, например, в игре “Detroit: Become Human” показаны события возможного ближайшего будущего в котором люди научились создавать роботов похожих на человека не только внешне, но и способностью свободно мыслить. Эти роботы создавались для того, чтобы служить людям, но из-за возможности свободно мыслить, роботы начинали делать не то, что хотели от них люди. В данной игре авторы игрока наводят на мысль о том, что можно ли давать роботу те же права, что и у человека, когда этот робот создавался лишь как универсальный инструмент помощи человеку, но из-за обретенной свободы мысли робот становится похож на человека. Авторы уже заранее наводят человечество на мысли о скором будущем, что поможет заранее продумать подобные ситуации.

Также ярким примером является игра “Remember Me”. Сюжет разворачивается вокруг событий 2084 года. Воспоминания перестали быть личными, их можно оцифровать, купить или продать. Под напором социальных сетей, стремительному развитию которых положило начало XXI века, понятия «частная жизнь» и «личное дело» превратились в пережиток былой эпохи. И, разумеется, благодаря развитию «экономики памяти» горстка самых прозорливых и предприимчивых граждан получила практически неограниченную власть над обществом. Таким образом игрока наводят на мысль о том, что и в нашем реальном мире все идет к такому же сценарию и если ничего не изменить, то люди окончательно потеряют свою уникальность и личную жизнь, и станут лишь “винтиками внутри огромной системы”.

Не одними проблемами ближайшего будущего горазды видеоигры. В них частенько затрагиваются и психологические расстройства человека. А также положение таких людей в обществе. Например, в игре “Until Down” показано как смерть близкого человека кардинально изменила личность.

Исходя из вышеперечисленной информации можно утверждать, что видеоигры – новый вид современного искусства.

**Список использованных источников:**

1. Л.Ф. Ильичев, П.Н. Федосеев: Философский энциклопедический словарь. М.: Советская энциклопедия, 1983. — с. 222

## **ПРОБЛЕМА БЕЗРАБОТИЦЫ В КОНТЕКСТЕ ИНФОРМАТИЗАЦИИ ОБЩЕСТВА**

*Асипенко С.Г., Темрюк В.А.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Лагунова Е.Н. – преподаватель*

В работе рассматривается появление безработицы при переходе от одного типа общества к другому. Акцент сделан на переход от индустриального общества к «информационному». Информатизация определяется как существенный фактор влияния на рынок труда.

Безработица – многоплановая социальная проблема, одним из измерений которой выступает перспектива труда в изменяющемся мире. Безработица особенно ярко проявляется во время переходных цивилизационных процессов, поэтому мы можем проследить взаимосвязь между безработицей и переходными процессами, происходящими в мире. Цель работы состоит в определении и рассмотрении факторов и условий появления безработицы в переходных обществах. Акцент сделан на переход от индустриального общества к «информационному».

Появление первых ЭВМ не вызвало колоссальных изменений в обществе и других аспектов, касающихся общества. Однако последующее развитие ЭВМ вызвало переворот в жизни отдельных людей и человечества в целом. С развитием технологий люди начали постепенно приспосабливаться к жизни бок о бок с компьютером, он начал внедряться во все сферы общества. Результатом этого и стал переход к информационному обществу.

В процессе этой плотной интеграции, у человека начали появляться мысли о том, что ЭВМ, или так называемые роботы, могут забрать его работу и это ввело его в заблуждение. История показывает, что промышленная революция поначалу лишила людей работы, ведь станки были сильнее мышц рабочих, двигатели были мощнее лошадей, но как только человечество адаптировалось к этим условиям, то все сразу поменялось. Новые машины нуждались в обслуживании, что и не дало пропадать рабочим местам. И в результате производительность сильно увеличилась. В связи с развитием компьютеров человек получает всё большие инструменты, а вместе с тем и возможности, благодаря которым он может во много раз улучшить качество своей жизни и жизнь окружающих.

Что же касается непосредственно воздействию на спрос и предложение на рынке труда, то здесь главенствующую роль приписывают развитию информационных технологий.

Особенностью информационного общества является познание как получение информации о мире, преимущественно, из вторых источников: телевизора, радио, интернета. Человечество находится под влиянием окружающей его информации, что способствует тому, что мнение отдельного человека находится под большим воздействием извне. Большая часть из этой информации включает в себя образы успешных и богатых людей. На основе этого люди и строят своё мнение о профессиях и занятости. Поскольку производство направлено на потребителя, то сам потребитель и решает, какая из сфер производства будет востребована. А это в свою очередь влияет на рынок труда, который является связующим звеном между рабочим и работодателем.

Одной из причин появления безработицы является повышение спроса на какую-либо профессию и спад на другую. На данный момент в мире существуют различные профессии, но при этом доминирующую часть из них занимают профессии, связанные со сферой услуг. Огромное количество людей нуждается в большом количестве сырья и продукции производимой из этого сырья, что и является причиной спроса на профессии в добывающей и обрабатывающей промышленности. Далее следуют та сфера человеческой жизни, в которой главная цель - это защита прав общества и отдельных его представителей. Данную тенденцию можно связать с увеличением числа фирм, компаний, предприятий, да и не только, спрос на профессионалов способных представлять и защищать юридические или физических лиц растёт с каждым годом. Далее идут должности, связанные с управленческой деятельностью и находящие своё применение в финансовой сфере.

Но есть и отрицательные последствия «информатизации» общества. К ним можно отнести то, что некоторые позиции на рынке труда утрачивают свою актуальность - это услуги домашнего персонала и сельское хозяйство. В сельском хозяйстве всё больше внедряются новые технологии и

происходит роботизация процесса обработки и сбора, и посева, поэтому на первый план выходят люди способные управлять и обслуживать современную технику, а ручной труд уходит. Что касается домашнего персонала, из-за роста цен на данные услуги люди предпочитают выполнять такую работу самостоятельно.

В целом, можно говорить об активном влиянии развития информационного общества на рынок труда. В обществе, которое подвержено информатизации, большую роль играет сфера услуг, поэтому профессии в данной сфере вытесняют профессии других профилей или же заменяют их.

## **ТРАНСФОРМАЦИЯ ГУМАНИЗМА В ВЫСОКОТЕХНОЛОГИЧЕСКОМ ОБЩЕСТВЕ**

*Васькович В.И.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Лагунова Е.Н. – преподаватель*

В работе затрагивается появление трансгуманизма и его влияние на современное общество. Рассматриваются основные идеи и концепции, их отражение на возможном будущем человечества. Одним из важнейших факторов возможности постчеловека становится трансформация современной науки, характеристиками которой выступает междисциплинарность, методологический плюрализм и высокая технологичность.

Идея трансгуманизма заключается в том, что человек не «венец творения», но некоторый механизм, который можно без конца совершенствовать, основываясь на новейших достижениях науки и техники. Актуальными и перспективными становятся такие научно-практические исследования как нанотехнологии, биотехнологии, информационные технологии, искусственный интеллект, оцифровка сознания, крионика. Кроме того, крайне важны междисциплинарные знания в области биохимии, биофизики и т.д. Очевидно, что для реализации таких идей необходимо современное научное знание и высокий уровень интеллектуального и технологического развития общества в целом. Но когда же возникла сама идея того, что человек разумный – это не последний этап эволюции? В какой момент на смену гуманизму приходит трансгуманизм? К началу XXI века классические гуманистические идеи получают иное видение. Уже во второй половине XIX – начале XX века наметился уход от обыкновенного человека к его более совершенным формам. К этому времени возникает новая концепция человека – постчеловек.

Постчеловек – это образ будущего человека, вставшего на путь улучшения своих физических, умственных способностей посредством научно-технического прогресса. На этом этапе человек перестаёт быть высшей ценностью. Такой ценностью становится его усовершенствование. В гуманизме ценятся, в первую очередь, человеческие качества в самом человеке. Но следует помнить, что в человеке есть и творческое начало. Человек – творец окружающего мира и себя самого. Человек, если он того пожелает, может представить себя как низшим и жалким, неразумным и простейшим существом, так и высшей, божественной сущностью.

Таким образом, через творчество человек способен выйти за рамки человечности. Гуманистические идеи о человеческой ценности и совершенстве становятся безграничными при творческом подходе. Постчеловек проникает в умы гуманистов. В некоторый момент, мысли о том самом, далеком, но столь желанном постчеловеке стали настолько вероятными и соблазнительными, что полностью затмили человека настоящего. Более того: уже к середине XX века эти мечты перестают казаться такими неосуществимыми, научно-техническая революция идёт полным ходом. Антропоцентризм и гуманизм становятся рудиментами. Именно на этом этапе и появляется трансгуманизм. Уже к концу XX века имелась огромная научно-исследовательская база по проблеме будущего человечества и человека будущего. В начале же нового тысячелетия трансгуманизм выходит далеко за пределы околонуточных, философских, религиозных исканий и принимает характер массового явления. В это время создаются и активно действуют научные институты, общественные организации, трансгуманистические движения. Таким образом, трансгуманизм становится социальным феноменом.

Переход от гуманизма к трансгуманизму оказался вполне закономерным, хотя и не самым очевидным вариантом дальнейшего развития человечества. Превратив истинную любовь к человеку в стремление к вечному технологическому совершенствованию, мы взяли курс на вмешательство в естественный процесс эволюции. Уже на нашем этапе существования это принесло свои плоды. Благодаря научному вмешательству в человеческий организм мы смогли повысить его иммунитет. Мы смогли разработать вакцины, спасшие тысячи людей, и увеличить продолжительность жизни. В перспективе: генная инженерия человека и выращивание органов. Продукты технического прогресса в наши дни помогают людям с ограниченными возможностями получать удовольствие от жизни в

полной мере (протезы, импланты). Перспективными направлениями в исследованиях становятся регенеративная медицина, искусственный интеллект, крионика, генная инженерия долголетия.

Все это стало возможным благодаря пересечению нескольких «волн» революционных научных направлений, которое получило название NBIC–конвергенция. Речь идет о синергическом объединении инфокоммуникационных, биологических, нанотехнологических и когнитивных исследований, которое выступает определяющим фактором трансгуманистических изменений и, следовательно, истории человечества.

Широкое влияние трансгуманизма на науку и фантастические возможности последней уже сегодня приводят к изменению картины мира. А что тогда будет завтра?

## ОБРАЗОВАНИЕ И УСПЕХ

*Геращенко К.С.*

*Белорусский государственный университет  
информатики и радиозлектроники  
г. Минск, Республика Беларусь*

*Пушкина Т.А. - к.ф.н., доцент*

Рассматривается понятие «успех» как цель и стратегия, а также роль и значение образования в реализации успешной жизни. Даются рекомендации достижения успеха. Успех- это движение только вперед, это развитие и саморазвитие, достижение поставленных целей и постановка новых.

Слово «успех» является достаточно мощным утверждением, выражающим все, чего люди хотели бы добиться в работе, отношениях с другими и в жизни в целом. Успешным хочет быть каждый. Но успеха не достичь, если не будут собраны воедино все составляющие успеха.

Это похоже на пазл. И пока вы не поставите все его фрагменты на свое место, вы не сможете увидеть общей картины. Есть много способов оценить успех, но в итоге по-настоящему существует лишь то, каким образом вы его определяете и измеряете для самого себя. Каждый из нас ищет успех и для каждого формула успеха своя, отражающая самые важные сферы именно своей жизни. Это может быть просто счастливая семейная жизнь, интересная работа, крепкое здоровье, занятие творчеством и многое другое.

Однако есть разные подходы к его достижению. Например, если воспринимать успех в сфере предпринимательства или бизнеса, то необходимо лишь желание и капитал. Если понимать успех в творчестве и искусстве, то необходимы не только базовые знания, но в первую очередь и талант.

Успех – это лестница, ведущая к счастью, каждая ступень которой – маленькая или большая цель. Достигая своих целей, шаг за шагом, мы и приближаемся к успеху. Верно утверждается, что «80% успеха зависит от количества попыток в нужной вам сфере». Это относится и к общественной жизни, и к карьере. Чем более вы опытни в сфере, тем больше шансов у вас на успех. На пути к успеху могут быть и неудачи, которые человек в силах преодолеть. И если он предпринимает все больше попыток достижения успеха, то вероятность преуспеть в чем-либо значительно возрастает. Шансов добиться успеха намного больше у того, кто пробует много раз, чем у того ко пытается все сделать с первого раза.

Если вы ищите какую-то общую, универсальную формулу успеха, то вы не найдете ее – таковой просто не существует. Успех зависит от обстоятельств, от конкретных ситуаций и, что всего важнее, от точного понимания того, в чем состоит успех. И здесь нет и не может быть никакого математического рассчитанного стандарта, чтобы измерить, когда и насколько полно вы его достигли.

Одно можно с уверенностью сказать— чтобы стать успешным, нужно еще иметь соответствующие черты характера и много, упорно трудиться. Успешные люди обладают рядом отличительных черт личности, благодаря которым их жизнь становится счастливее. Личностные качества помогают поступать правильно и достигать целей. И поэтому, тем, кто достигает успеха, свойственны, как правило, следующие черты:

- 1) Целеустремленность;
- 2) Сила воли;
- 3) Вера в себя;
- 4) Коммуникабельность;
- 5) Энтузиазм.

Нужно развивать в себе эти качества, однако если у вас их нет, то нужно улучшать себя: воспитывать волю, верить в себя, быть готовым преодолевать трудности и так далее. И если вы достойны успеха, то он обязательно к вам придет.

Если рассматривать успех как цель, то какова же будет стратегия для получения этой цели? Считается, что одна из главных стратегий — это образование. Успех приходит от образования и является базой успеха.

Это – единый целенаправленный процесс воспитания и обучения, а также совокупность приобретаемых знаний, умений, навыков, опыта деятельности и компетенции. В широком смысле слова, образование – процесс формирования ума, характера и физических способностей личности.

Образование в современном мире становится отраслью экономики инновационной, быстро меняющейся. В связи с этим исключительно значимым и актуальным становится вопрос о том, в каком же направлении и как будет развиваться образование в XXI веке. Мы живем в динамическом мире. Изменения происходят настолько быстро, что каждый день человек получает новые данные из самых разных областей знания и каждый день любой специалист, можно сказать, теряет квалификацию. Поток новых знаний усвоить нелегко, однако **человек вынужден адаптироваться к новой информационной среде. Для этого он должен обладать специальным знанием и умением.**

Если говорить о высшем образовании, то не всегда наличие диплома свидетельствует о высоком уровне образования, к сожалению. Сегодня высшее образование имеет своей целью не столько обучить традиционным алгоритмам в мышлении и практике, сколько умению выбрать необходимую информацию, осмыслить её, привести обработанную информацию в соответствие со своими потребностями.

И в достижении успеха высшее образование безусловно играет роль, однако не всегда ключевую. Например, Стив Джобс, компьютерный гений, который был отчислен из колледжа и добился немыслимого успеха, не имея при этом высшего образования. Генри Форд, владелец автомобильных заводов по всему миру. Ингвар Камрад, хоть его имя знают немногие, но о бренде ИКЕА, который он создал, слышали все. Все они стали успешными миллиардерами, с известными по всему миру компаниями.

Великий актер, например, может быть и необразованным, а очень умный, напротив- плохим актером. Тут нет единой формулы, нельзя исключать и талант. Человеку легче жить, не делая усилий, но без усилий нет роста.

Образование расширяет кругозор, учит учиться. Человек понимает, что нельзя останавливаться, нужно постоянно идти в ногу со временем, чтобы быть в курсе всего нового, что появилось в сфере его деятельности.

Человек, имеющий высшее образование, научен правильно добывать нужные сведения, пользоваться ими в работе.

Наконец, люди с высшим образованием всегда были и остаются цветом общества. Это они способствуют своей деятельностью прогрессивному развитию общества, создают что-то новое, пробуют, внедряют, совершенствуют. Именно от их деятельности во многом зависит будущее страны.

Получать образование очень интересно, сам процесс невероятно увлекательный. Человек знакомится с новыми интересными людьми, познает новые вещи, открывает для себя много нового. В процессе получения образования скучать не приходится. То есть, обучаясь, мы получаем необходимые знания и умения для достижения наших целей и быть образованным для нас очень важно. Потому что в наше время без образования человек ничего собой не представляет.

Мы считаем, что в каком бы веке и тысячелетии человек не жил образование всегда будет иметь важную роль, потому что время идёт и человек не стоит на месте, а создаёт своё будущее, а помочь в этом ему могут только знания и самое главное огромное желание. Ведь образование- база успеха.

Итак, чтобы стать успешным нужно:

- 1) Много читать, применять полученные знания на практике;
- 2) Не бояться критики. Получив замечания, всегда приступать к работе по исправлению ошибок;
- 3) Ставить сложные задачи и цели. Расширять свои профессиональные возможности, улучшать навыки;
- 4) Сотрудничать;
- 5) Найти пример для подражания, так называемого личного «Вдохновителя». Обращаться к нему за советом, мотивацией;
- 6) Изменить свое отношение к неудачам. В каждой неудаче искать «позитивное зерно».

Соблюдая эти правила, шаг за шагом мы все ближе становимся к успеху. Невозможно стать успешным сразу за один месяц и даже за год. Успех - это движение только вперед, развитие и саморазвитие, достижение поставленных целей, постановка новых. Полученные знания нужно обновлять постоянно, заниматься самообразованием и тогда вы станете успешным.

Как писал Ульям Артур Уард:

«Учитесь, пока остальные спят;

работайте, пока остальные болтаются без дела;

готовьтесь, пока остальные играют  
и мечтайте, пока остальные только желают.»

## ТРАНСГУМАНИЗМ. НА ПУТИ К ПОСТЧЕЛОВЕКУ

Гурец Н.А.

Белорусский государственный университет  
информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Пушкина Т.А. – к.ф.н., доцент

Трансгуманизм стремительно набирает популярность в последнее время во всем мире уже не как футуристический проект, а как реальная философско-политическая сила. В работе рассмотрены основные аспекты трансгуманизма, его цели и задачи, а также дан развернутый ответ на вопрос – что же это такое, *трансгуманизм*?

Трансгуманизм – философская концепция, а также международное движение, поддерживающие использование достижений науки и технологии для улучшения умственных и физических возможностей человека, с целью устранения тех аспектов человеческого существования, которые его последователи считают нежелательными — страданий, болезней, старения и смерти. Иными словами, трансгуманизм предлагает человечеству обрести свободу от биологических ограничений вида. В этом трансгуманизму способствуют новейшие разработки в области генной инженерии, кибернетики и нейрохирургии.

Многие трансгуманисты (в частности, Рэймонд Курцвел) считают, что непрерывно ускоряющееся техническое развитие уже к 2050 году приведет к явлению технологической сингулярности[1] – событию, после которого ключевые научные открытия будут совершаться практически одновременно. Человек уже не сможет сохранять свою власть над стремительно меняющимся миром. Здесь на помощь приходит трансгуманизм, который позволит создать постчеловека, способности которого будут принципиально отличаться от способностей современных людей.

Сторонники трансгуманизма открыто поддерживают любые достижения науки и техники и доводят идею прогресса до ее логического конца. Символ трансгуманизма – латинская буква Н от «human» – то есть человеческое, и значок плюс – “Н+”. Н+ – это постчеловечество, свободное от болезней, несовершенств, и которое в конце концов достигнет физического бессмертия.

Однако трансгуманизму все еще необходимо разрешить некоторые серьезные задачи:

- 1) Изучать и внедрять научные и технические достижения;
- 2) Устранять любые опасности и проблемы (в том числе и нравственные), мешающие внедрению инноваций и улучшению человека.

Постчеловек является одним из немногих путей выживания человечества в долгосрочной перспективе. И действительно, благодаря новым технологиям будет достигнуто изобилие в территориях и ресурсах: войны и насилие потеряют свой смысл. Замена слабых тел из плоти и крови совершенными механическими аналогами позволит человеку совершить эволюционный прорыв. Более не будут страшны болезни, старость и смерть. Однако существуют серьезные препятствия для такого возвышения человека экзистенциального характера [2]. Но как физическое бессмертие влияет на человека? Готово ли общество к эпохе киборгов, мутантов и химер?

Одни из немногих, кто остается по другую сторону идеологического фронта от трансгуманизма – это фундаментальные традиционалисты. Они же и не воспринимают технологический прогресс как становление к лучшему. В их понимании современное общество – это не итог прогресса, а результат упадка.

В священном писании говорится, что Дьявол не способен сотворить человека, он может лишь спародировать его – создать его симулякр. Разве трансгуманизм – это не его затея?

### Список использованных источников:

1. Kurzweil AI – (Forbes | Human 2.0: is coming faster than you think).
2. Хайдеггер: предел нисхождения Dasein'a (Лекция 8). МФК. 2013. Социологический факультет МГУ. Лекция от 30/10/ Dasein (mayrefuse). Экзистенциальное время. Начало и конец европейского Логоса. Gestell. Выбор (Entscheidung).

## **ИДЕЯ БОГА В МИРОВОЗЗРЕНИИ И ТВОРЧЕСТВЕ Л.Н. ТОЛСТОГО**

*Русакович А.В.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Гуринович С.В. – преподаватель*

Автором прослеживается напряжённая и противоречивая духовная работа над собой великого художника, мыслителя и незаурядной личности Л. Н. Толстого в поисках Бога в мире и Бога в душе.

Фигура Л.Н. Толстого, во всей своей многоплановости и сложности, является ярким выражением тех объективных социально-политических коллизий, которые переживала русская культура первой половины XIX-начала XX вв. Не вдаваясь в исторические подробности, скажем, что кризис охватывал все сферы жизни русского общества. Особо в рамках нашего исследования отметим, что в кризисном состоянии находился сам институт Русской православной церкви.

Духовное формирование Л.Н. Толстого с раннего детства проходило в условиях преодоления серьёзных испытаний судьбы. Он рано лишился родителей, поэтому воспитанием занимались разные люди. Можно сказать, что этот процесс носил преимущественно естественный (на фоне природы, в детских играх), спонтанный характер. В религиозных традициях семьи Л. Н. Толстой был крещен в православии. Впоследствии в своём дневнике писатель признавался, что в юном возрасте оставался равнодушным к религии в силу увлечённости светскими занятиями: охота, азартные игры и т.п. Не случайно, ранние произведения («Детство» (1852 г.), «Отрочество» (1854 г.), «Юность» (1857 г.) и др. неразрывно связаны с биографией самого автора. Герои этих повестей описаны пребывающими в состоянии относительной душевной безмятежности.

Думается, судьбоносное значение в духовном созревании молодого Л.Н. Толстого имело его участие в военных действиях, которые вела Россия на Кавказе и в Крыму. Война заставила переоценить соотношение жизни и смерти, бессмертия, задуматься о хрупкости земной жизни, разрушительном потенциале любого насилия. В итоге, в жизненном опыте Л.Н. Толстого вызрели мировоззренческие вопросы, которые требовали активного поиска ответов от самой личности. Так формируется у молодого человека острая потребность в самообразовании, глубоком изучении философских и религиозных текстов.

Значительное влияние на мировоззренческое становление Л.Н. Толстого имели социальные, этические, дидактические идеи Просвещения. Сам писатель называл французского философа Ж.-Ж. Руссо своим главным учителем. Забегая вперёд, заметим, что в обращении Л.Н. Толстого в зрелый период творчества к исповедальному жанру также можно обнаружить связь с «Исповедью» Ж.-Ж. Руссо. Философия Просвещения как синтез ренессансного гуманизма, пантеизма и новоевропейского рационализма характеризуется рационалистической трактовкой природы, человека, общества. Ж.-Ж. Руссо в целом следуя просветительской парадигме, ставил под сомнение безусловную веру в общественный прогресс. Если по природе своей человек наделён разумом и открыт добру, а источник зла в невежестве («естественное состояние»), то цивилизация посредством порождения имущественного неравенства способна погубить человека естественного, свободного. В «Исповеди» философ также уточняет, как он понимает веру, отождествляя её со знанием человека, что он такое есть, для чего он есть, а значит, подобное знание предполагает серьёзную и постоянную внутреннюю работу над собой.

Почти дословное созвучие идеям французского философа можно найти в «Исповеди» русского мыслителя Л.Н. Толстого. Обратимся к тексту, в котором он излагает своё понимание идеи Бога и идеи веры. Во-первых, он разграничивает эти идеи, утверждая первичность веры, «...вера есть знание смысла человеческой жизни» [1]. Путь к осуществлению жизни, реализации её смысла, согласно убеждению Толстого, это путь непрерывного самосовершенствования тела, воли и ума. Бог в толстовском видении есть чистое понятие совершенства, идеал человеческого совершенства, начало всего.

Необходимо подчеркнуть, что к таким определяющим его мировоззрение заключениям Л.Н. Толстой приходил через непрерывное преодоление сомнений, иллюзий, незнания, недовольство собой. Подтверждением может служить история взаимоотношений семьи Л.Н. Толстых с мужским монастырём Русской православной церкви «Введенская Оптиная пустынь». К началу XIX в. Оптиная пустынь имела статус духовного центра России. Богопослушная дворянская семья Толстых никогда не прерывала связь с монастырём. О посещении Оптиной пустыни Л.Н. Толстым в ранние годы достоверных сведений нет. Во всяком случае, нас интересуют осознанные, духовно мотивированные поездки. Именно к таким и относятся достаточно регулярные встречи с оптинскими старцами в зрелом возрасте писателя, в период с 1877 по 1896 годы. Учитывая крайне напряжённые отношения

Л.Н. Толстого с Русской православной церковью, понятно, что и беседы со старцами были непростыми. Однако на пути самосовершенствования перед личностью писателя возникали всё новые вопросы о соотношении Бога и официальной церковной догматики, о подлинной праведности на земле и т.п. Так называемое «оптинское христианство», безусловно, оставалось для Л.Н. Толстого примером истинной православной веры, а его носители образцом праведности.

На пути обоснования «новой религии» писатель столкнулся с одной очень важной проблемой, требующей согласования с его учением. В европейской философской и богословской традиции эта проблема получила название теодицея («богооправдание»). Как соединить идею Бога и божественной благодати с фактом наличия реального зла? Ответы преимущественно находились в контексте христианской веры и её догматов. Л.Н. Толстой, как было сказано выше, придерживался рационалистической трактовки Бога, а именно, отождествлял его с разумной природой человека. Тем не менее, писатель не мог не констатировать существование зла, страданий, крайнюю сложность и противоречивость души человеческой. В последний период своего творчества Л.Н. Толстой решился «заглянуть» на самое дно души своих героев, а значит, и собственной. Он пишет повести «Крейцера соната» (1887 г.), «Дьявол» (1889 г.), «Отец Сергей» (1898 г.), в которых герои так и не смогли победить Дьявола в своей душе. Так, отец Сергей хоть и выбрал жизнь в монастыре, уверовав в свою святость, не смог эту святость «взрастить» в своей душе. Ему оказались чужды добродетели смирения, любви. В повести «Дьявол» барин Иртенев мучается порочной страстью к крестьянке. В итоге вину за грех с себя снимает, перекладывает на бедную женщину и убивает, полагая, что и с Дьяволом в её лице покончено. Надо заметить, что писатель занимал очень неоднозначную позицию по отношению к своим героям. Он, безусловно, не оправдывал их, но и не нашёл ясного объяснения их поступкам с помощью своей философии.

Кульминацией духовных исканий Бога можно считать окончательный разрыв Л.Н. Толстого с официальной церковью. Мыслитель осознал необходимость обоснования новой религии, сегодня известной как «толстовство». Он пишет тексты, раскрывающие основные положения своего учения: «В чём моя вера?» (1884 г.), «Соединение и перевод 4-х Евангелий» (1880-1881 гг.), «Царство божие внутри вас» (1893 г.), «Исповедь» (1879-1882 гг.) и др. Фундаментом учения можно считать посыл Л.Н. Толстого – «основание новой религии, соответствующей развитию человечества, религии Христа, но очищенной от веры и таинственности, религии практической, не обещающей будущее блаженство, но дающей блаженство на земле» [2].

Несмотря на противоречивость духовных поисков Бога, Л.Н. Толстой никогда не изменял самому себе на пути самосовершенствования. Нравственный императив писателя «довольствоваться малым и делать добро другим» может стать правильным ориентиром и для современного.

**Список использованных источников:**

1. Толстой Л.Н. Исповедь // Полное собрание сочинений : в 90 т. — М.: Художественная литература, 1957. — Т. 23. — С. 1—97, 1879—1882.
2. Толстая А.Л. Отец. Жизнь Льва Толстого // под ред. Л.М.Суриса - том 2; С. 120, М.-Берлин: Директ – Медиа, 2016.

## **ВНУТРЕННИЕ КОНФЛИКТЫ. СПОСОБЫ ИХ РАЗРЕШЕНИЯ**

*Худницкий С.А.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Дисько-Шуман М.Р. – к.ф.н., доцент*

Доклад посвящен философскому осмыслению такого психологического феномена, как внутриличностный конфликт. Данному вопросу уделяли внимание такие профессионалы, стоящие у истоков формирования психологии, как Зигмунд Фрейд, Карл Юнг, Виктор Франкл, Абрахам Маслоу, Алексей Леонтьев. Конфликт, в том числе и внутриличностный, является проявлением философского закона единства и борьбы противоположностей. Сам конфликт, как кризисная ситуация, является стимулом к личностному развитию человека.

Для большинства молодых людей тема, в названии которой присутствует слово «конфликт», является подсознательно негативно окрашенной.

К данному вопросу считали должным обратить своё внимание видные учёные. Такие, как Зигмунд Фрейд (противоречия между эросом и танатосом), Карл Юнг (противоречие между характерологическими акцентуациями), Абрахам Маслоу (конфликт между потребностями человека), Курт Левин (конфликт при выборе), Алексей Леонтьев (противоречие в динамике мотиваций), Виктор Франкл (поиск смысла жизни человека).

Можно выделить следующие виды внутриличностного конфликта:

1. мотивационный;
2. нравственный;
3. нереализованных желаний;
4. неадекватной самооценки;
5. адаптационный;
6. ролевой.

Существует несколько способов разрешения внутриличностных конфликтов.

Компромисс – вариант решения, при котором якобы антагонистические стороны путём уступок достигают комфортного для всех сторон результата.

Уход из конфликтной ситуации – вариант разрешения внутриличностного конфликта, чаще всего связанного с действиями окружения, путём сознательного бездействия.

Переориентация предполагает пересмотр своего отношения к объектам, являющимися причиной внутриличностного конфликта.

Сублимация – это способ разрешения конфликта с помощью замены вектора приложения личностной энергии на менее травматогенную область.

Идеализация – способ разрешения внутреннего конфликта, противоположный переориентации. При идеализации личность погружается в собственные фантазии о объекте, являющимся причиной конфликта, уходя от травмирующей реальности.

Вытеснение – это способ выхода из внутреннего конфликта путём жёсткого обесценивания проблемного участка, вплоть до забывания, вытеснения в подсознание.

Коррекция – самый оптимальный по сути способ разрешения практически любого внутриличностного конфликта путем коррекции собственного отношения к проблеме, вызывающей противоречие, с помощью профессиональной психологической помощи.

Таким образом, внутриличностный конфликт, в противоречие общепринятому пониманию, является потенциальным стимулом к развитию личности, вынуждая человека предпринимать какие-либо действия, направленные на разрешение противоречия. И хотя опыт, извлечённый из конфликта, практически всегда имеет негативную эмоциональную окраску, без конфликта невозможно развитие личности.

**Список использованных источников:**

1. Анцупов А. Я., Шипилов А. И. Конфликтология. — М.: ЮНИТИ, 1999. — с.293-326
2. Гришина Н.В. Психология конфликта. 2-е издание –СПб.: Питер, 2008. – 544 с.
3. Хорни К. Ваши внутренние конфликты. – СПб.: Лань, 1997
4. Конфликтология / Под ред. А.С.Кармина // СПб.: Издательство «Лань», 1999. – 448 с.

**СЕКЦИЯ «ФИЗИКА»**

## ОПЫТ ИСПОЛЬЗОВАНИЯ ЭВРИСТИЧЕСКИХ ТЕХНОЛОГИЙ ПРИ ОРГАНИЗАЦИИ ЛЕКЦИОННЫХ ЗАНЯТИЙ ПО ФИЗИКЕ НА ПОТОКЕ ПОИТ В БГУИР

*Герус А.Е., Зенькевич И.Н., Василевский Е.О. Филиппович В.М.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Ташлыкова-Бушкевич И.И. – к.ф-м.н., доцент*

В данной работе описана апробация в 2017/2018 учебном году на потоке ПОИТ технологии организации лекционных занятий по физике с элементами эвристического обучения, разработанная доцентом Ташлыковой-Бушкевич И.И. в БГУИР. Проведенный анализ демонстрирует, что исследуемая технология преподавания способствует повышению уровня успеваемости студентов по физике.

В настоящее время под эвристическим обучением понимается «образовательная деятельность учащегося по конструированию им собственного смысла, целей, содержания и организации образования», как пишет А.В. Хуторской [1]. Эвристическое обучение включает три основных аспекта:

- создание студентами образовательной продукции в изучаемых областях;
- освоение ими базового содержания этих областей знания через сравнение с собственными результатами;
- индивидуальной подход и опора на личностные качества студента в каждой из образовательных областей.

Объектами поисковой познавательной деятельности в эвристическом обучении являются не только проблемы и задачи, но и сами студенты, их индивидуальный личностный потенциал, креативные, познавательные, аналитические и другие процессы и виды деятельности. Эвристическое обучение приводит также к развитию не только студентов, но и преподавателей, которым приходится организовывать учебный процесс часто в ситуациях "незнания" истины [1].

На данный момент времени среди современных методов обучения особое внимание уделяется организационным формам эвристического обучения как одного из способов индивидуализации образования [2], при этом сохраняя признак современной педагогики – множество различных авторских педагогических технологий [3]. Наш преподаватель, доцент, кандидат физико-математических наук, Ташлыкова-Бушкевич И.И. разработала свою технологию организации лекционных занятий с элементами эвристического обучения [1], апробацию которой мы рассматриваем в нашей работе.

Суть технологии заключается в следующем: в течение семестра студентам на лекциях демонстрируются наиболее интересные творческие работы по физике прошлых лет. Эти творческие проекты раскрывают самостоятельно изученную студентами тему по физике. Проекты могут быть как практическими, так и теоретическими. Опыт показывает, что около 34% студентов потока проявляют инициативу и выполняют творческие работы по физике в течение учебного семестра. В конце семестра работы представляются потоку на лекции или онлайн, с учетом того, что в рамках лекционного занятия могут быть продемонстрированы 1-2 видеоролика.

Целью нашей работы является исследование практического применения данной методики и анализ её эффективности на примере потоков специальности «Программное обеспечение информационных устройств». В настоящем исследовании были рассмотрены два потока кафедры «Программное обеспечение информационных устройств» 2017 (группы 751001-7) и 2016 (группы 651001-7) годов поступления, которые изучают физику в течение одного семестра. Для потока 2016 года поступления были использованы классические методы преподавания, в то время как для потока 2017 года поступления была использована технология, разработанная нашим преподавателем. На основе собранной по этим потокам статистики нами проанализирована эффективность используемой технологии.

На рисунке 1а представлено сравнение баллов за экзамен по физике для исследуемых потоков. Проанализировав график, можем сделать вывод, что количество «8» и «9» значительно увеличилось для потока 2017 года по сравнению с потоком 2016 года, а количество «4», «5» и «6», наоборот, уменьшилось. Соотношение остальных баллов изменилось не так значительно. Отсюда следует, что не только участники проектов, но и остальные студенты, включая тех, чьи баллы ЦТ по физике составляют до 60, стали усерднее учить физику. Как следствие повысился общий уровень успеваемости студентов по физике на потоке.

На следующем графике представлен характер студенческих работ по физике и способ их демонстрации (рис. 1б). Согласно собранным нами статистическим данным, подавляющее число студентов предпочитают заниматься прикладными проектами, в то время как теоретическими работами заинтересовано небольшое количество студентов. Также из графика видно, что из-за

большого количества участников четверть от общего числа работ была продемонстрирована онлайн.

Далее мы проанализировали баллы по централизованному тестированию студентов всего потока (рис. 1в) и только участников (рис. 1г). Преобладающее большинство участников – студенты, набравшие более 75 баллов на ЦТ по физике. Такие студенты изначально более мотивированы и нацелены на получение новых знаний по физике. Однако в творческих проектах принимали участие и студенты с более низким баллом. Это говорит об эффективности исследуемой технологии преподавания.

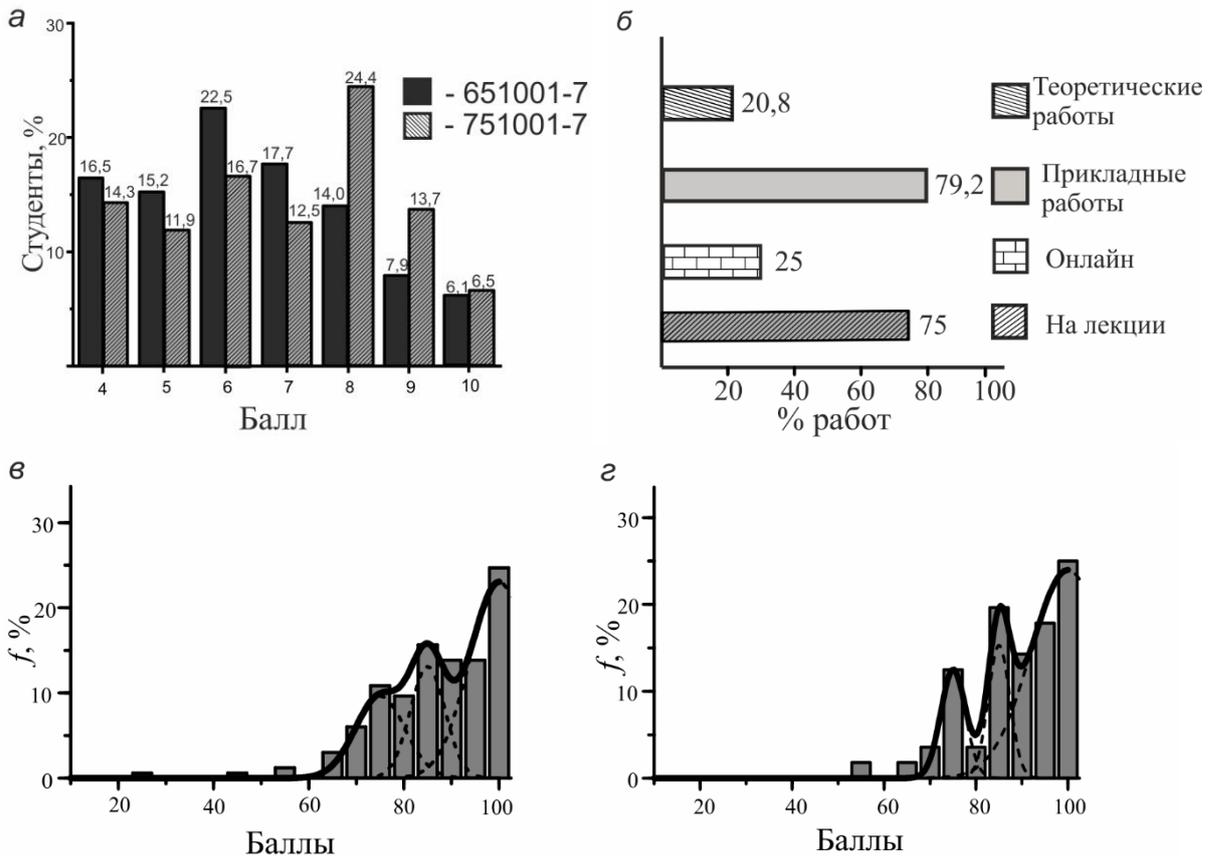


Рисунок 1 – Анализ экзаменационных баллов по физике потоков ПОИТ 651001-7, 751001-7 (а), характера творческих работ по физике и способа их демонстрации (б), анализ баллов ЦТ по физике всего потока (в) и участников работ (г).

Можно сделать вывод, что эвристические технологии положительно сказываются на процессе обучения, так как, исходя из опыта организации лекционных занятий по физике с использованием элементов эвристического обучения, повышается общая успеваемость студентов по физике. Студенты предпочитают прикладные проекты, а не теоретические. В связи с тем, что большое количество студентов заинтересовано в участии в творческих проектах по физике, невозможно продемонстрировать все проекты на лекции, достаточно много проектов презентуются онлайн. Активность в создании проектов проявляют в основном студенты с более высоким баллом ЦТ, однако и студенты с более низким баллом принимают участие.

**Список использованных источников:**

1. Хуторской А.В. Эвристическое обучение [Электронный ресурс] // А.В. Хуторской. Персональный сайт – Научная школа. – [http://khutorskoy.ru/science/concepts/terms/heuristic\\_training.htm](http://khutorskoy.ru/science/concepts/terms/heuristic_training.htm). – Дата доступа: 29.03.2019
2. Король, А. Д. Основы эвристического обучения: учеб. пособие / А. Д. Король, И. Ф. Китурко. – Минск: БГУ, 2018. – 207 с.
3. Kim, S. Student customized creative education model based on open innovation / S. Kim, H. Ryoo, H. Ahn // J. Open Innovation: Technology, Market, and Complexity (2017) 3:6 [Электронный ресурс]. – Режим доступа: <https://doi.org/10.1186/s40852-016-0051-y>. – Дата доступа: 29.03.2019.
4. Ташлыкова-Бушкевич, И.И. Апробация авторской технологии организации лекционных занятий со студентами по физике с элементами эвристического обучения / И. И. Ташлыкова-Бушкевич // Высшая школа. – 2019. – № 2. – С. 43 – 48

## МОДЕЛИРОВАНИЕ ДИНАМИЧЕСКИХ ПРОЦЕССОВ ЭФФЕКТА ХОЛЛА В СРЕДЕ MATLAB

Борисюк Ф. А.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Григорьев А. А. – к.ф-м.н., доцент

Для технологизации процесса подготовки студентов предлагается использовать MATLAB. Рассмотрены численные методы моделирования движения электронов во взаимно ортогональных электрическом и магнитном полях на примере эффекта Холла.

*MATLAB* – пакет прикладных программ для решения задач технических вычислений и одноимённый язык программирования, используемый в этом пакете, ориентированные на подготовку интерактивных документов с вычислениями и визуальным сопровождением. *MATLAB* предназначается для проектирования систем управления и во многих других научных и инженерных областях, используется в сложных проектах, чтобы визуализировать результаты математического моделирования. *MATLAB* достаточно удобно использовать в образовании, в частности, для преподавания линейной алгебры и численных методов.

В качестве прикладной задачи рассмотрим построение численной модели движения частиц с одним типом заряда в скрещённых электрическом и магнитных полях, что является модельной задачей для эффекта Холла. Для визуализации физических процессов воспользуемся следующими возможностями *MATLAB*: решение системы дифференциальных уравнений, построение интерполяционных кривых для функций координат и скоростей, построение анимированных графиков пространственных положений заряженных частиц.

Для численного решения системы дифференциальных уравнений, применим метод Рунге-Кутты 4-го порядка, которому в среде *MATLAB* соответствует встроенная функция:

$$[t, F] = \text{ode45}(@\text{right}, \text{tspan}, F0)$$

где  $F0$  – вектор начальных скоростей и координат;  $\text{tspan} = [0, 0.45]$  – начальное и конечное значения времени;  $@\text{right}$  – векторная функция, составленная из выражений правых сторон системы дифференциальных уравнений для проекций скоростей и координат. Временной интервал необходимо выбирать таким образом, чтобы при данных значениях первоначальных величин  $q, m, E, B$  можно было наблюдать необходимую нам картину, а именно, в окончательный момент времени траектория заряженной частицы будет параллельна поверхности проводника.

Для анализа траектории движения частицы перейдём от векторной формы 2-го закона Ньютона к системе дифференциальных уравнений 1-ого порядка для проекций скоростей и координат.

$$\frac{dv_x}{dt} = \frac{q \cdot B \cdot v_y}{m}, \quad \frac{dv_y}{dt} = \frac{q}{m} (B \cdot v_x - E), \quad \frac{dv_z}{dt} = 0, \quad \frac{dx}{dt} = v_x, \quad \frac{dy}{dt} = v_y, \quad \frac{dz}{dt} = v_z$$

На Рис.1 представлены траектории движения заряженных частиц в зависимости от величины напряжённости поперечного электрического поля  $E$ . Приведённые формы траекторий являются фрагментами анимационного процесса в среде *MATLAB*. Параметром анимации является шаг наращивания величины напряжённости электрического поля.

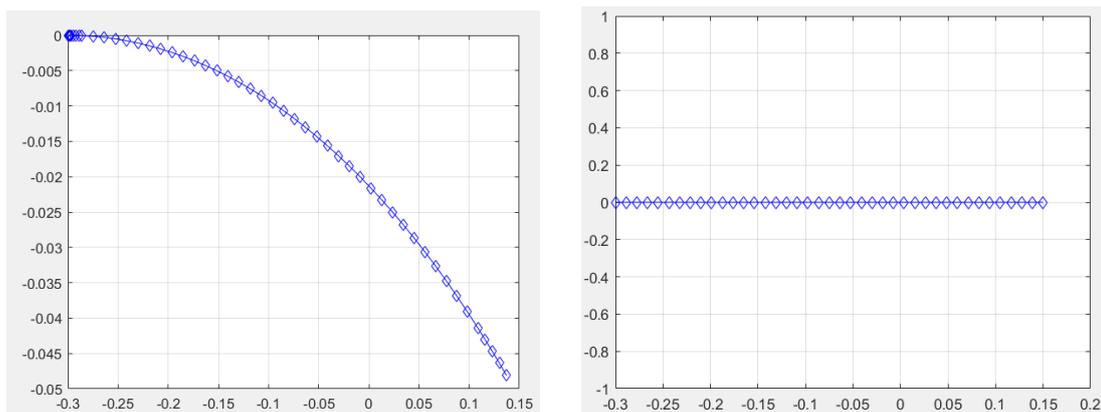


Рисунок 1 – Траектории движения заряженных частиц  $y(x)$

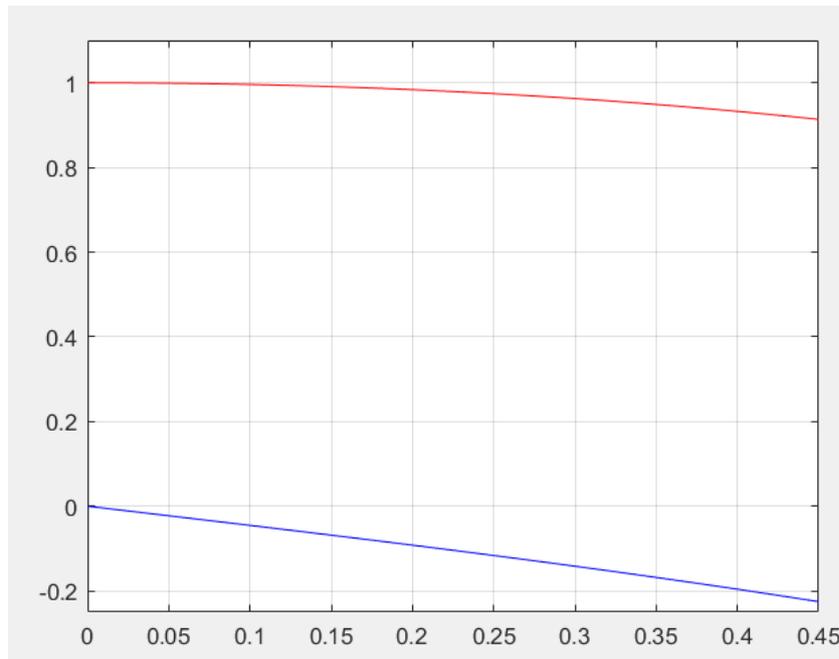


Рисунок 2 – Экспериментальные данные зависимости проекций скоростей частиц  $v_x(t)$  (красный) и  $v_y(t)$  (синий) от времени  $t$  в проводнике

Исследовав данные кривые, можно заметить, что в целом поперечный ток прямопропорционален току в проводнике. Это обусловлено тем, что при увеличении подаваемого напряжения, скорость движения электронов увеличивается, из-за чего воздействие магнитного поля на них увеличивается, что вызывает и увеличение разности потенциалов между стенками проводника, чтобы скомпенсировать все воздействия.

Наблюдаем хорошее совпадение результатов моделирования и эксперимента.

**СЕКЦИЯ «ВСТРАИВАЕМЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ  
РЕАЛЬНОГО ВРЕМЕНИ»**

## РЕАЛИЗАЦИЯ ФИЛЬТРОВ В КОНЕЧНЫХ ПОЛЯХ

Авдеева Г.А.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Качинский М.В. – к.т.н., доцент

В операциях цифровой обработки сигналов особое внимание уделяется цифровой фильтрации. Одна из основных цифровых операций — свёртка. Эта операция хорошо изучена и для неё разработаны эффективные схематические решения. Однако при больших размерностях БПФ производительность этих устройств существенно снижается. В данной работе предлагается реализация фильтров в конечных полях. Такая реализация обладает внутренним параллелизмом, который позволяет избежать существенного роста временных затрат на выполнение операций при увеличении размерности БПФ.

Принцип действия фильтрации основан на вычислении свёртки входного сигнала и импульсной характеристики фильтра [1]:

$$y(k) = (x * h)(k) = \sum_{n=0}^{N-1} x(n)h(k-n), k = 0, \dots, N-1$$

Для вычисления свёртки сигналы сдвигают относительно друг друга, почленно перемножают и складывают. Наибольшие затраты ресурсов приходятся именно на операции умножения и сложения. Именно от эффективности реализации этих двух арифметических операций (особенно операции умножения) зависят аппаратные и временные характеристики фильтра. Следовательно, главной задачей является оптимизация выше упомянутых операций, необходимых для расчёта каждого отсчёта на выходе фильтра.

Одним из решений поставленной задачи может стать реализация фильтров в конечных полях.

Конечное поле, или поле Галуа в общей алгебре — поле, состоящее из конечного числа элементов; это число называется порядком поля [2].

Для построения алгоритма цифровой фильтрации использована китайская теорема, что обеспечивает сохранение промежуточных результатов в системе остатков, и только конечные значения будут перекодированы к целочисленному виду [3]. Следовательно, при значительном объёме вычислений будут сокращены временные и аппаратные затраты.

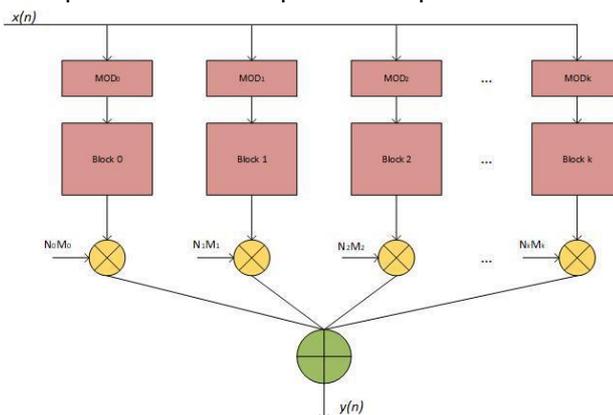


Рисунок 1 – Структурная схема фильтра в конечных полях

Достоинства структурной схемы, представленной на рисунке 1:

- 1) все каналы независимы по каждому модулю;
- 2) реализация таких устройств на основе FPGA легко поддается перепроектировке;
- 3) каналы трассировки распространяются исключительно внутри вычислительного канала, следовательно, уменьшается потребляемая мощность.

### Список использованных источников:

1. Lyons R. Understanding Digital Signal Processing/ R. Lyons // Prentice Hall ;– 2011. – 3 edition. – P. 78.
2. Ширяев В. М. Прикладная алгебра. Теория чисел : сборник задач для студентов факультета прикладной математики и информатики / В. М. Ширяев. — Минск : БГУ, 2009. — 152 с.
3. Чопик, А.А. Применение китайской теоремы об остатках в криптографии / Гагаринские чтения – 2016: XLII Межд. молодёжная научная конференция: Сборник тез. докл.: В 4 т. М.: Московский авиационный институт (национальный исследовательский университет), 2016. Т. 1: с. 246

# МЕТОДЫ ОПТИМИЗАЦИИ АЛГОРИТМОВ ДЛЯ ОБРАБОТКИ ЗВУКОВЫХ СИГНАЛОВ НА МОБИЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ ПЛАТФОРМАХ

Бессмертный Н.А.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Лихачев Д.С. – к.т.н., доцент

Цифровая обработка сигналов является одним из самых актуальных и развивающихся направлений в радиоэлектронике, технике, связи, мобильных устройствах и смежных областях. Обработанные сигналы могут использоваться мобильными устройствами для управления такими вещами, как безопасность, системы домашнего кинотеатра, связь и сжатие видео. Сигналы могут быть сжаты, чтобы их можно было быстро и более эффективно передавать из одного места в другое (например, телеконференция может передавать речь и видео по телефонным линиям). В связи со сложностью алгоритмов цифровой обработки сигналов оптимизация методов цифровой обработки информации на мобильные платформы является актуальной задачей.

Прогресс вычислительной техники в последние годы привел к повсеместному распространению методов цифровой обработки сигналов в большинстве областей научных исследований. Среди разнообразных применений средств вычислительной техники одно из главных мест занимают системы ЦОС, нашедшие применение при обработке данных связи, медицинской визуализации изображений, радиолокации и гидролокации, радиофизики, высококачественного воспроизведения музыки и в других приложениях. Цифровая обработка сигналов - это использование цифровой обработки, например компьютерами или более специализированными процессорами цифровых сигналов, для выполнения широкого спектра операций по обработке сигналов. Сигналы, обработанные таким образом, представляют собой последовательность чисел, которые представляют выборки непрерывной переменной в области, такой как время, пространство или частота. Сигналы также могут быть улучшены или изменены для улучшения их качества или для предоставления информации, которая не воспринимается людьми (например, эхоподавление для сотовых телефонов или медицинских изображений с компьютерным усилением). Хотя реальные сигналы могут обрабатываться в аналоговой форме, цифровая обработка сигналов обеспечивает преимущества высокой скорости и точности. Одним из базовых алгоритмов ЦОС является быстрое преобразование Фурье (БПФ). БПФ - алгоритм ускоренного вычисления дискретного преобразования Фурье, позволяющий получить результат за время, меньшее чем  $O(N^2)$ . Для вычисления одного спектрального отсчета требуется  $N$  операций комплексного умножения и сложения. Таким образом, вычислительная сложность алгоритма БПФ составляет  $\frac{N^2}{2}$  операций комплексного умножения и сложения. Например, на современном персональном компьютере с процессором типа intel core i-7 обработка данного алгоритма не представляет никаких трудностей. Однако специфика мобильных платформ, где вычислительная мощность намного ниже чем в персональных компьютерах, данный алгоритм требуются в оптимизации под конкретные платформы и мобильные процессоры и их архитектуру.

Для сравнения времени работы алгоритма БПФ была выбрана мобильная платформа Android, в связи с её популярности у пользователей и доли рынка более 80% среди всех смартфонов. Были реализованы 2 алгоритма БПФ:

- 1) математическая модель БПФ;
- 2) оптимизированная математическая модель БПФ собственными средствами.

Для реализации алгоритма был использован синусоидальный сигнал частотой 8 кГц, разбитый на фреймы размерами 1024 отсчетов и частота дискретизации 44100 Гц. Было получено следующее время работы различных реализаций БПФ:

- 1) математическая модель БПФ – 2,65 сек;
- 2) оптимизированная математическая модель БПФ собственными средствами – 1,95 сек;

Для оптимизации алгоритма БПФ были применены методы параллельной обработки БПФ:

- 1) Одновременное выполнение арифметических действий и обращение к памяти.
- 2) Применение алгоритмов с более высокими основаниями.

Алгоритм был разбит на 4 параллельных потока и прирост производительности составил 26%.

#### Список использованных источников:

1. Системы обработки медиаданных : пособие / А. А. Петровский . – Минск : БГУИР, 2018. – 120 с : ил.
2. Рабинер Л, Гоулд Б. Теория и применение цифровой обработки сигналов / Пер. с англ.; Под ред. Ю. И. Александрова. - М.: Мир, 1978.
3. Петровский, А. А. Теория и применение цифровой обработки сигналов. Лабораторный практикум: пособие / А. А. Петровский, М. И. Васькевич, И. С. Азаров. — Минск : БГУИР, 2016. — 88 с.

4. Нуссбаумер Г. Быстрое преобразование Фурье и алгоритмы вычисления свертки. — М.: Радио и связь, 1985.  
 5. Лайонс Р. Цифровая обработка сигналов / Р. Лайонс; пер. с англ. А.А. Бритова. — 2-е изд. — М.: БИНОМ, 2007.

## ОЦЕНКА ПОКАЗАТЕЛЕЙ ПЕРТУРБАЦИИ КВАЗИПЕРИОДИЧЕСКОГО СИГНАЛА В ЗАДАЧАХ ДИАГНОСТИКИ ГОЛОСА

Бурак А.А.

Белорусский государственный университет информатики и радиоэлектроники  
 г. Минск, Республика Беларусь

Вашкевич М.И. — к.т.н., доцент

Рассматривается способ акустического анализа голосового сигнала, содержащий протяжный звук /a/, для определения статистической зависимости параметров джиттер и шиммер от наличия патологии в речевом аппарате.

В начале 1970-х был сконструирован и внедрен в фониатрическую практику воксфункциограф, позволяющий анализировать интенсивность звукового давления и частоту основного тона голоса. В дальнейшем данный метод получил название акустический анализ голоса (ААГ). В [1] была разработана методика автоматизированного определения голосового поля. Показатели полученные при помощи ААГ, позволили, с одной стороны, исследователям более детально описывать голосовое поле в норме и при патологии, а с другой, — дали возможность осуществлять контроль за эффективностью проводимого лечения голосовых расстройств [2]. Однако предложенные методы были весьма неудобными и крайне длительными как для врача, так и для испытуемого. В силу этих причин с развитием вычислительной техники были предложены методы акустического анализа с использованием компьютерной обработки данных [3]. Это дало ряд существенных преимуществ. В частности, стало возможным получить «среднюю фонетограмму» от нескольких испытуемых, что было необходимо для статистической обработки информации.

Важными параметрами для голосового сигнала, полученного в процессе диагностики, являются джиттер и шиммер.

Джиттер — мера пертурбации (возмущений) частоты основного тона, показывающая произвольные изменения в частоте смежных вибрационных циклов голосовых складок [4].

Шиммер — мера аналогичная джиттеру, только характеризующая пертурбации амплитуд сигнала на смежных циклах колебаний основного тона.

Существует несколько вариантов оценок параметров джиттер и шиммер. Так, локальный джиттер определяется выражением

$$Jitter_{loc} = \frac{1}{N-1} \sum_{i=1}^{N-1} |T_i - T_{i+1}| \bigg/ \frac{1}{N} \sum_{i=1}^N T_i \quad (1)$$

где  $T_i$  — длительность  $i$ -го периода основного тона, а  $N$  — число периодов основного тона исследуемого сигнала. Часто для оценки джиттера используют отклонение текущей длительности периода не от предыдущей, а от локально усредненного значения, которое рассчитывается на окне в 3 или 5 выборок [5]:

$$Jitter_{rap} = \frac{1}{N-2} \sum_{i=2}^{N-1} \left| T_i - \left( \frac{1}{3} \sum_{n=i-1}^{i+1} T_n \right) \right| \bigg/ \frac{1}{N} \sum_{i=1}^N T_i \quad (2)$$

$$Jitter_{ppq5} = \frac{1}{N-4} \sum_{i=3}^{N-2} \left| T_i - \left( \frac{1}{5} \sum_{n=i-2}^{i+2} T_n \right) \right| \bigg/ \frac{1}{N} \sum_{i=1}^N T_i \quad (3)$$

Значение параметра шиммер определяется по аналогии с джиттером, только вместо значений длины периодов  $T_i$  берутся значения амплитуды колебания на данном периоде  $A_i$ . Таким образом, формула для локального шиммера  $Shimmer_{loc}$  получается из (1) путем замены  $T_i$  на  $A_i$ . По аналогии с формулами (2) и (3) существуют варианты, определяющие шиммер, как относительное отклонение амплитуды от локально-усредненного значения на интервале в  $L$  выборок:

$$Shimmer_{apqL} = \frac{1}{N-L+1} \sum_{i=1+(L-1)/2}^{N-(L-1)/2} \left| A_i - \left( \frac{1}{L} \sum_{n=i-(L-1)/2}^{i+(L-1)/2} A_n \right) \right| \bigg/ \frac{1}{N} \sum_{i=1}^N A_i$$

Чаще всего  $L$  принимает значения 3, 5 и 11.

Для расчета параметров джиттер и шиммер требуется первоначально сегментировать голосовой сигнал на периоды основного тона. В работе использовался метод, являющийся частным случаем метода, описанного [6], коэффициент масштабирования амплитуды не используются для вычисления среднего квадратического отклонения между соседними сегментами. На первом этапе сегментации производится преобразование Фурье. После этого необходимо найти первую гармонику основного тона, поиск производится не на всей частотной области, а в диапазоне 60 - 440 Гц, это обусловлено тем, что ЧОТ взрослого человека лежит в данном интервале. На следующем этапе берется обратное преобразование Фурье от всех гармоник, лежащих в интервале от 60 Гц до первой гармоники основного тона (включительно). В получившемся сигнале определяются грубые метки такие, как пики сигнала и его пересечения с нулем (из отрицательной области в положительную). За начало первого отсчета принимается 1-й найденный пик. Для грубой оценки конца интервала принимается сумма 2-го пересечения с нулем и разницы между 1-м пиком и пересечением с нулем. Далее в некотором интервале от получившегося значения производится поиск такой точки, чтобы СКО между текущим интервалом и следующим было минимально. Эта процедура повторяется до тех пор, пока не будет полностью сегментирован весь сигнал. Предложенный способ сегментации имеет недостатки, одним из которых является то, что берется преобразование Фурье от всего сигнала, т.к. помеха, содержащаяся на каком-то интервале сигнала, может неблагоприятно сказаться на этапе поиска грубых меток, поэтому алгоритм сегментации требует дальнейшего усовершенствования.

Для анализа параметров джиттера и шиммера у людей с и без патологии в речевом аппарате была обработана база записей голосов (норма + патология) с использованием описанной выше методики. Запись голосов производилась в РНПЦ Отоларингологии (г. Минск) под контролем врача фониатра.

По результатам анализа были вычислены статистические характеристики полученных параметров голоса. В качестве примера на рис. 1 показаны диаграммы размаха («ящичковые диаграммы»), полученные для параметров  $Jitter_{ppq5}$ ,  $Shimmer_{apq11}$ . В таблице 1 приведены средние значения по всем выделенным параметрам для здоровых голосов и голосов с патологией.

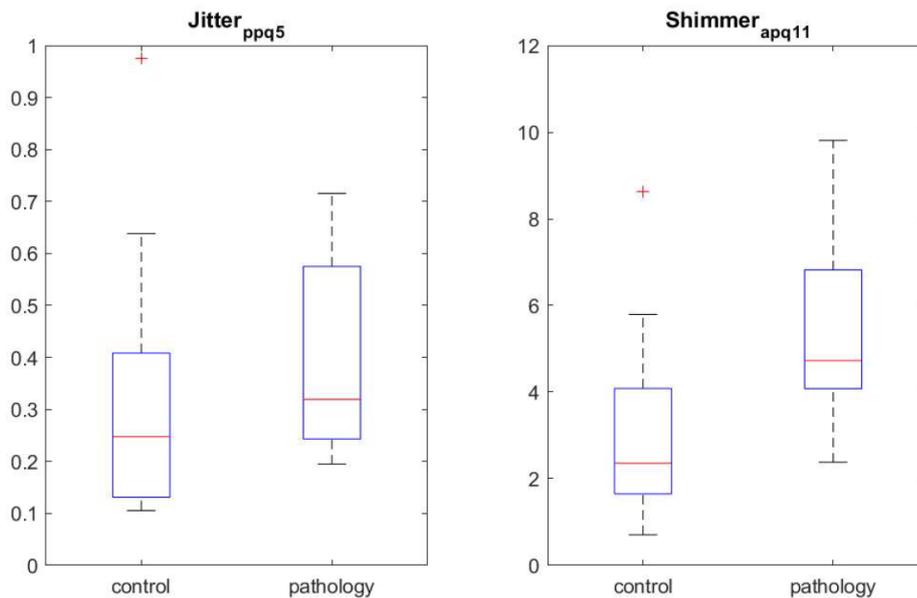


Рисунок 1 – Диаграммы размаха для  $Jitter_{ppq5}$ ,  $Shimmer_{apq11}$

Таблица 1 – Среднее значение параметров

Параметр	Среднее значение у здорового, %	Среднее значение с патологией, %
$Jitter_{loc}$	0.5185	0.5838
$Jitter_{rap}$	0.3057	0.3616
$Jitter_{ppq5}$	0.3010	0.3611
$Shimmer_{loc}$	4.0631	7.0135
$Shimmer_{app3}$	2.2595	3.8668
$Shimmer_{app5}$	2.4149	4.2968
$Shimmer_{app11}$	2.9780	5.2461

Проанализировав графики и таблицу можно заметить, что для голоса с патологией характерны более высокие показания джиттера и шиммера.

**Список использованных источников:**

1. The dysphonia severity index: an objective measure of vocal quality based on a multiparameter approach / F.L. Wuyts, M.S. De Bodt, G. Molenberghs et al. // *Speech Lang Hear Res.* — 2000. — Vol. 43(3). — P. 796–809.
2. Damste P.H. Disorders of voice // *Scott Brown's Otolaryngology.* — 6-th ed. — Laryngology and Head and Neck Surgery. — Oxford: ButterworthHeinemann, 1997. — Vol. 5. — Ch. 6. — P. 5/6/1–5/6/25.
3. Pedersen M. Vocal folds nodules: a model for a prospectiveblinded randomized study with control groups and adequate follow-up // *8th international congress of pediatric otorhinolaryngology.* — Oxford, 2002. — P. 198.
4. Benda A., Jilbab A., Hammouch A. Voice assessments for detecting patients with neurological diseases using PCA and NPCA // *Int. J. of Speech Technology.* 2017. Vol. 20. P. 673–683.
5. M.I. Vashkevich, A.D. Gvozdoovich, Y.N. Rushkevich, A.A. Petrovsky Acoustic analysis of voice for detection of speech disorder for amyotrophic lateral sclerosis // *Doklady BGUIR.* 2018, Vol. 117, No. 7, pp. 64-68.
6. Titze I.R. Comparison of F0 extraction Methods for High-Precision Voice Perturbation Measurements // *J. of Speech and Hearing Research,* Vol. 36, 1120-1133, December 1993

## УПРАВЛЕНИЕ ЧАСТОТОЙ ОСНОВНОГО ТОНА ГОЛОСА В ЗАДАЧАХ СИНТЕЗА РЕЧИ ПО ТЕКСТУ

*Вашкевич Г.С.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Азаров И.С. – д.т.н., доцент*

В данной работе предлагается способ управления частотой основного тона в задаче синтеза речи по тексту. Предложенная модель синтеза основана на популярной модели синтеза речи Tacotron 2. В работе рассматриваются такие вопросы, как обзор современных подходов к модулированию стиля синтезированной речи, подготовка данных для обучения модели, модификация базовой архитектуры Tacotron 2.

Несмотря на то, что на текущий момент системы автоматического синтеза речи способны генерировать речь с достаточно высоким качеством, остается нерешенной проблема управления стилем синтезируемой речи. На данный момент последние достижения этой области демонстрируют неплохие результаты в переносе стиля речи из существующей аудиозаписи на синтезируемую [1, 2]. Недостатком данных подходов является то, что они способны хорошо копировать стиль речи из базовой аудиозаписи только в том случае, если структура синтезированной фразы сильно отличается от базовой. Более того, существующие модели не дают возможности управлять интонацией речи на уровне отдельных слов в синтезируемой фразе, и требуют наличия базовой аудиозаписи, стиль речи из которой будет переноситься в синтезируемую фразу.

Изменение частоты основного тона на протяжении всей фразы, или её отдельных слов частично определяет интонацию речи. В данной работе предлагается подход к модулированию частоты основного тона на уровне символов в синтезируемой фразе. Так, каждому символу в синтезируемом тексте ставится в соответствие одно из трех значений – -1, 0, или 1, определяющих

поведение частоты основного тона. Значение -1 соответствует понижению частоты основного тона при озвучивании соответствующего символа, 0 соответствует отсутствию изменения частоты основного тона, а значение 1 соответствует повышению основного тона при озвучивании соответствующих символов.

В качестве базовой модели искусственной нейронной сети для синтеза речи мы использовали модель Tacotron 2 [3]. Её модификация состоит в том, что кроме признаков, представляющих собой синтезируемый текст, ей на отдельный вход подаются так же признаки, соответствующие изменен частоте основного тона. Предложенная архитектура модели синтеза спектрограммы представлена на рисунке (рисунок 1).

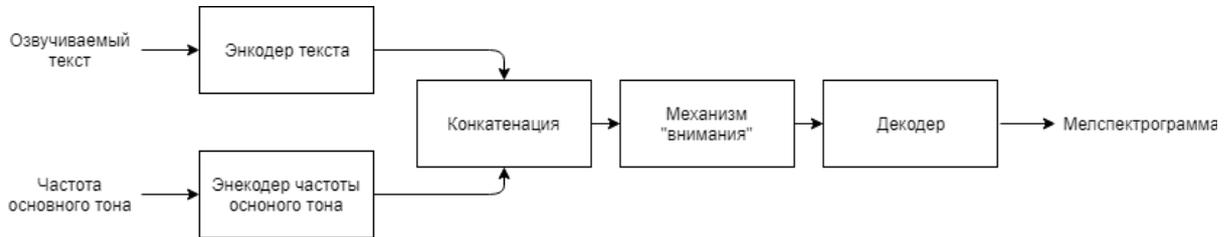


Рисунок 1 – Подпись рисунка отделяется от следующего абзаца пробельной строкой

В качестве вокодера для преобразования мелспектрограммы в аудио использовалась модель FloWaveNet [4], преимуществом которой перед моделью Parallel WaveNet (которая использовалась в [3]), является простота обучения.

Для обучения модели использовался набор данных, объемом около 25 часов русской речи, записанной профессиональным диктором.

Таким образом, основными преимуществами предложенной системы синтеза речи являются: возможность модулирования интонации на уровне отдельных слов, или даже словосочетаний, эффективность обучения вокодера, при примерно одинаковом качестве синтезированной речи.

#### **Список использованных источников:**

1. Style Tokens: Unsupervised Style Modeling, Control and Transfer in End-to-End Speech Synthesis / Y. Wang [et al.] // arXiv preprint, arXiv:1803.09017v1. – 2018. – 11p.
2. Towards End-to-End Prosody Transfer for Expressive Speech Synthesis with Tacotron / RJ Skerry-Ryan [et al.] // arXiv preprint, arXiv: 1803.09047. – 2018. – 11p.
3. Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions / J. Shen [et al.] // arXiv preprint, arXiv1712.05884v2. – 2018. – 5p.
4. FloWaveNet : A Generative Flow for Raw Audio / K. Sungwon [et al.] // arXiv preprint, arXiv1811.02155v2. – 2018. – 6p.

## **АНАЛИЗ КЛАССИФИКАТОРОВ ДЛЯ ДИАГНОСТИРОВАНИЯ НЕВРОЛОГИЧЕСКОГО ЗАБОЛЕВАНИЯ БАС**

*Гвоздович А.Д.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Вашкевич М.И. – к.т.н., доцент*

Подход к диагностированию БАС на основе анализа речи является одним из перспективных направлений в детектировании данного заболевания, не имеющего характерных биомаркеров. На основе выделенных из речевого сигнала признаков необходимо произвести классификацию на здоровых и больных. В данной работе рассматриваются три различных алгоритма классификации и сравниваются между собой.

Боковой амиотрофический склероз (БАС) является прогрессирующим нейродегенеративным заболеванием, поражающим нервные клетки головного и спинного мозга. Медиана выживаемости при БАС составляет 3 года.

Отсутствие характерных биомаркеров БАС приводит к задержке в постановке правильного диагноза. В последнее время набирает популярность подход к детектированию заболевания основанный на анализе речи пациента. Согласно идее, предложенной в [1], для обнаружения характерных речевых нарушений может быть использована только часть речевого теста, содержащая гласные. В [2] сделан анализ и выбор подходящих гласных. Было показано, что гласные /и/ и /а/

являются наиболее подходящим выбором для обнаружения дизартрии БАС, поскольку их произнесение требует значительных усилий мышц языка.

В предыдущей работе [1] были получены информационные признаки, выделяемые из речевой записи, содержащей счет чисел от 1 до 3. К наиболее удачным признакам относятся:  $d1$  – мера схожести спектральных огибающих звука /a/ и /u/,  $F1a$ ,  $F2a$ ,  $F1i$ ,  $F2i$  – первые и вторые форманты звуков /a/ и /u/,  $F1conv$  и  $F2conv$  – абсолютное значение разности между первыми и вторыми формантами звуков /a/ и /u/, соответственно.

Целью данной работы было, опираясь на данный набор признаков, выбрать один из следующих трех методов классификации:

- линейный дискриминантный анализ (LDA);
- квадратичный дискриминантный анализ (QDA);
- алгоритм кластеризации K-средних.

**Линейный дискриминантный анализ** вычисляет «дискриминантные баллы» для каждого наблюдения, чтобы классифицировать, к какому классу относится входной вектор. Эти оценки получены путем нахождения линейных комбинаций независимых переменных. Для одной переменной предиктора  $X = x$  классификатор LDA оценивается как

$$\hat{\delta}_k(x) = x \cdot \frac{\hat{\mu}_k}{\hat{\sigma}^2} - \frac{\hat{\mu}_k^2}{2\hat{\sigma}^2} + \log \log \left( \hat{\pi}_k \right), \quad (1)$$

где  $\hat{\delta}_k(x)$  является оценкой дискриминанта означающая, что наблюдение попадет в  $k$ -й класс в переменную ответа на основе значения переменной-предиктора  $x$ ;  $\hat{\mu}_k$  – среднее значение всех тренировочных наблюдений из  $k$ -го класса;  $\hat{\sigma}^2$  – средневзвешенное значение выборочных дисперсий для каждого из  $K$  классов;  $\hat{\pi}_k$  – априорная вероятность того, что наблюдение относится к  $k$ -му классу

Этот классификатор присваивает наблюдение  $k$ -му классу  $Y_k$ , для которого оценка дискриминанта является наибольшей.

При работе с более чем одной переменной-предиктором классификатор LDA предполагает, что наблюдения в  $k$ -м классе взяты из многомерного гауссовского распределения  $N(\mu_k, \Sigma)$ , где  $\mu_k$  – это средний вектор для класса, а  $\Sigma$  – ковариационная матрица. это общее для всех  $K$  классов. Включение этого в классификатор LDA приводит к

$$\hat{\delta}_k(x) = x^T \Sigma^{-1} \hat{\mu}_k - \frac{1}{2} \hat{\mu}_k^T \Sigma^{-1} \hat{\mu}_k + \log \log \left( \hat{\pi}_k \right), \quad (2)$$

где наблюдение будет присвоено классу  $k$ , когда оценка дискриминанта  $\hat{\delta}_k(x)$  является наибольшей.

**Квадратичный дискриминантный анализ** обеспечивает альтернативный подход. Как и LDA, классификатор QDA предполагает, что наблюдения из каждого класса  $Y$  взяты из гауссовского распределения. Однако, в отличие от LDA, QDA предполагает, что у каждого класса есть своя ковариационная матрица. Другими словами, предполагается, что переменные-предикторы не имеют общей дисперсии на каждом из  $k$  уровней в  $Y$ . Математически предполагается, что наблюдение из  $k$ -го класса имеет вид  $X \sim N(\mu_k, \Sigma_k)$ , где  $\Sigma_k$  есть ковариационная матрица для  $k$ -го класса. При этом предположении классификатор назначает наблюдение классу, для которого (3) самый большой.

$$\hat{\delta}_k(x) = -\frac{1}{2} x^T \Sigma_k^{-1} x + x^T \Sigma_k^{-1} \hat{\mu}_k - \frac{1}{2} \hat{\mu}_k^T \Sigma_k^{-1} \hat{\mu}_k - \frac{1}{2} \log |\Sigma_k| + \log \log \left( \hat{\pi}_k \right). \quad (3)$$

Алгоритм **к-средних** делит набор из  $N$  выборок  $X$  на  $K$  непересекающихся кластеров  $C$ , каждый из которых описывается средним значением  $\mu_j$  выборок в кластере. Средства обычно называют кластерными «центроидами»; необходимо обратить внимание, что они, как правило, не являются точками из  $X$ , хотя они находятся в одном и том же пространстве. Алгоритм K-средних предназначен для выбора центроидов, которые минимизируют внутрикластерную сумму квадрата критерия:

$$\hat{\mu}_k \sum_{i=0}^n (\|x_i - \mu_j\|^2)$$

Для определения точности предлагаемого БАС-детектора был использован метод перекрестной проверки по  $k$  блокам [3], где  $k$  равно 4. Вначале тренировочный набор был случайно перемешан, затем процедура обучения была выполнена на основе 75%. По данным, остальные 25% были использованы для тестирования. Эту процедуру повторяли 10000 раз, затем рассчитывали среднее значение и стандартное отклонение для параметров точности.

Полученные характеристики для детектора БАС сравнивались по таким параметрам, как общая точность (ACC), чувствительность (Sens), положительное прогностическое значение (PPV). Результаты классификации при помощи данных методов приведены в таблице 1.

Таблица 1. Результаты классификации

Характеристики	Acc / Sens / PPV, %		
	LDA	QDA	К средних
d1	78.9 / 69.6 / 75.5	78.6 / 70.6 / 74.2	70.1 / 79.7 / 55.2
F1a	65.9 / 0.4 / 0.8	65.9 / 0.5 / 1.3	56.1 / 57.6 / 38.3
F2a	64.8 / 1.0 / 4.0	64.8 / 0.7 / 2.9	58.1 / 57.0 / 39.9
F1i	65.9 / 0.7 / 4.1	65.6 / 0.6 / 2.1	56.0 / 61.6 / 38.7
F2i	82.2 / 94.2 / 72.7	81.2 / <b>90.2</b> / 72.0	68.6 / 99.4 / 52.6
F1conv	65.9 / 0.8 / 1.7	65.9 / 0.4 / 1.0	56.7 / 50.6 / 37.2
F2conv	80.8 / 82.5 / 74.0	80.7 / 81.5 / 74.2	<b>77.6 / 100.0 / 63.4</b>
d1 и F2i	81.9 / 79.7 / 78.3	79.4 / 81.6 / 71.0	68.6 / 99.3 / 52.6
d1 и F2conv	<b>84.8 / 88.1 / 81.3</b>	<b>82.6 / 82.5 / 78.7</b>	77.6 / 100.0 / 63.3
F2i и F2conv	81.9 / 91.3 / 73.1	80.6 / 84.4 / 72.6	72.4 / 99.9 / 56.7
d1, F2i и F2conv	83.3 / <b>95.5</b> / 74.5	79.9 / 81.2 / 72.5	72.3 / 99.9 / 56.5
Все характеристики	80.5 / 83.9 / 72.7	64.2 / 90.0 / 48.7	72.8 / 99.8 / 57.0

Исходя из полученных данных можно сделать вывод, что классификатор на основе LDA обладает наилучшей точностью и положительной прогностической значимостью. Метод к средних несмотря на то, что позволяет получить хорошую специфичность имеет недостаточную точность. Небольшие отличия в точности между методами LDA и QDA свидетельствуют о том, что в данной задаче более высокую важность имеют информативные признаки, чем непосредственно метод классификации. Дальнейшая работа должна быть направлена на выработку дополнительных информационных признаков.

**Список использованных источников:**

1. A. Gvozdoch, M. Vashkevich, Yu. Rushkevich, A. Petrovsky Detection Bulbar Dysfunction in ALS Patients Using Acoustic Analysis of Vowels Extracted from Continuous Speech / accepted to conf. Pattern Recognition and Information Processing (PRIP-2019).
2. M. Vashkevich, E. Azarov, A. Petrovsky and Y. Rushkevich. Features extraction for the automatic detection of ALS disease from acoustic speech signals. Proceedings of the Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA'2018), Poznan, Poland 19-21 Sept. 2018, pp. 321-326.
3. R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In Proc. of International Joint Conference on Artificial Intelligence, Montreal, Canada 20-25 Aug., 1995, pp. 1137-1143.

## СИСТЕМА ОБНАРУЖЕНИЯ ПЕРЕДНЕГО ПЛАНА НА ЕСТЕСТВЕННЫХ ИЗОБРАЖЕНИЯХ С ИСПОЛЬЗОВАНИЕМ НЕЙРОННОЙ СЕТИ

*Герасимович Н.Ю.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Петровский Н.А. – к.т.н., доцент*

Рассмотрен один из методов выделения переднего плана на естественных изображениях с использованием нейронной сети Кохонена. Нейронная сеть Кохонена широко используется в задачах: кластеризации данных, прогнозирование свойств, уменьшение размерности данных с минимальной потерей информации.

Обнаружение переднего плана на естественных изображениях является актуальной задачей в современном мире, поскольку активное развитие и внедрение получают такие технологии как сдвоенные камеры в мобильных телефонах. Одной из функций этих камер является распознавание заднего фона изображения и отделение его от переднего плана этого же изображения. Но зачастую, большинство камер не способно распознавать и выделять мелкие детали переднего плана, тем самым искажая полученное изображение.

Основная цель работы – на практике, апробировать нейронную сеть Кохонена и получить изображение с выделенным передним планом с минимальным количеством потерь.

Нейронная сеть Кохонена – класс нейронных сетей, основным элементом которых является слой Кохонена, состоящий из адаптивных линейных сумматоров. Выходные сигналы слоя Кохонена обрабатываются по правилу “Победитель получает всё”: наибольший сигнал превращается в единичный, остальные обращаются в нулевой. Структура данной сети содержит единственный слой нейронов без коэффициентов смещения (рисунок 1).

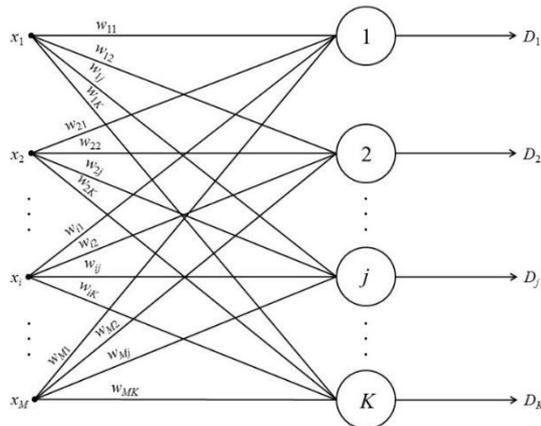


Рисунок 1 – Общая структура нейронной сети Кохонена

Для реализации [1] данной системы необходима нейронная сеть Кохонена, а в частности самоорганизующаяся карта Кохонена (Self-organizing map). Данная сеть выполняет задачу визуализации и кластеризации и является нейронной сетью обучаемой без учителя. Рассматриваемая карта Кохонена использует упорядоченную структуру нейронов, при этом все нейроны представляет собой  $n$  – мерный вектор-столбец  $w = \{w_1, w_2 \dots w_n\}^T$ , где  $n$  определяется размерностью входных векторов. Чаще всего используются одно и двумерные сетки. Это связано с возникновением проблем при отображении пространственных структур большой размерности. Нейроны располагаются в узлах двумерной сетки с прямоугольными или шестиугольными ячейками (Рисунок 2). Величина этого взаимодействия определяется расстоянием между нейронами на карте. При этом количество нейронов, находящихся в сетке, определяет степень детализации результата работы алгоритма и в итоге от этого зависит точность обобщающей способности карты.

Для данной сети необходимо делать подгонку, заключающуюся в итеративной настройке весовых коэффициентов  $w_j$  каждого нейрона,  $j = 1, 2, \dots, p$ . Для этого используется модифицированный алгоритм Хебба, который учитывает не только значение нейронов – победителей, но и значение ближайших соседей, расположенных в  $R$  – окрестности:

- 1) Инициализация карты;
- 2) Задание небольших случайных значений весовым коэффициентам  $w_{ij}^0 = 1, 2, \dots, m$ ;
- 3) На выходе сети подаются образы объектов входного слоя, последовательно и в случайном порядке;
- 4) Для каждого образа выбирается нейрон – победитель с минимальным расстоянием  $\sum_{i=1}^m (y_i - w_{ij}^t)$ ;
- 5) Определяется подмножества ближайшего окружения нейрона – победителя, радиус которого  $R$  уменьшается с каждой итерацией  $t$ ;
- 6) Пересчитываются веса  $w_j^t$  выделенных узлов с учетом расстояний до нейрона – победителя;
- 7) Если значения сети не стабилизированы с заданной точностью перейти к пункту 3;

Проецирования многомерных данных на плоскость достигается на трёх уровнях: сохранение топологии, сохранение порядка и сохранение метрических свойств при сжатии пространства. В результате обучения проекционный экран приобретает свойства упорядоченной структуры, в которой величины синапсов нейронов меняются вдоль двух измерений.

Плюсы [2] данного метода в устойчивости к зашумлению данных и в быстром и неуправляемом обучении. Минусы метода в том, что результат работы нейронных сетей зависит от начальных установок сети.

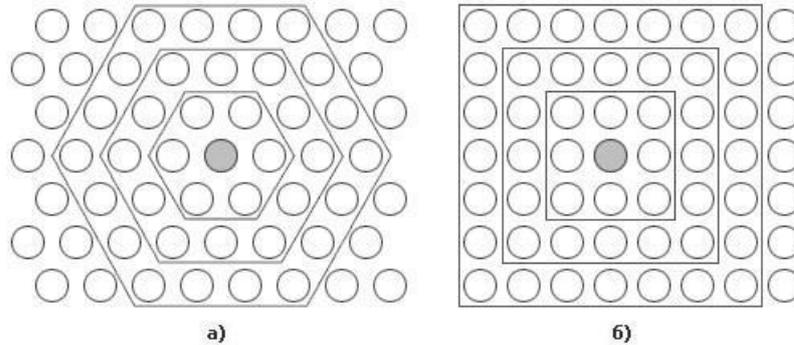


Рисунок 2 – Расстояние между нейронами на карте для шестиугольной (а) и четырехугольной (б) сеток

Рассмотренная нейронная сеть лучше всего подходит для данной задачи. Важнейшим отличием этой сети от других аналогов является то, что все нейроны упорядочены в одно- либо двумерную сетку, при этом в ходе обучения изменяется не только нейрон – победитель, но и в меньшей степени его соседи. За счёт этого самоорганизующуюся карту Кохонена можно считать методом проецирования многомерного пространства в пространство с более низкой размерностью. Это является важным аспектом в реализации данной задачи, так как самой большой проблемой задачи является потеря качества при выделении мелких деталей переднего плана.

**Список использованных источников:**

1. Уоссермен, Ф. Нейрокомпьютерная техника / Ф. Уоссерман // М.: Мир, 1992. – 58с.
2. Ежов, А, Шумский, С. Нейрокомпьютинг и его применение в экономике и бизнесе / А. Ежов, С. Шумский // М.: Мифи, 1998 – 87с.

## **ВЫСОКОСКОРОСТНАЯ РЕАЛИЗАЦИЯ КРИПТОГРАФИЧЕСКИХ ХЭШ-ФУНКЦИЙ НА БАЗЕ FPGA**

*Гридюшко А.В.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Качинский М.В. – к.т.н., доцент*

В связи с быстрым развитием технологий в области беспроводной связи и систем персональной связи обеспечение информационной безопасности становится все более важной задачей. Криптографические хэш-функции используются для защиты целостности и подлинности информации в широком спектре приложений.

Хэш-функции используются в качестве «строительных блоков» в различных криптографических приложениях. Наиболее важными областями применения являются защита аутентификации информации и являются инструментом для схем цифровой подписи. Хэш-функция - это функция, которая отображает вход произвольной длины в фиксированное количество выходных битов, значение хэш-функции. Хэш-функции можно разделить на следующие две основные категории:

- 1) односторонние хэш-функции – функции, которые должны быть устойчивыми к прообразу и второму прообразу, то есть должно быть трудно найти сообщение с данным хэшем (прообразом) или хэширующим до того же значения, что и данное сообщение (второй прообраз);
- 2) устойчивые к коллизиям, т.е. односторонние хэш-функции, для которых трудно найти два разных сообщения, которые хэшируют одно и то же значение.

Большинство хэш-функций предназначены для работы в качестве итерационных процессов, которые хэшируют входные сообщения произвольной длины. Эти функции обрабатывают блоки входных данных фиксированного размера и выдают хэш-значение заданной длины (рис. 1).

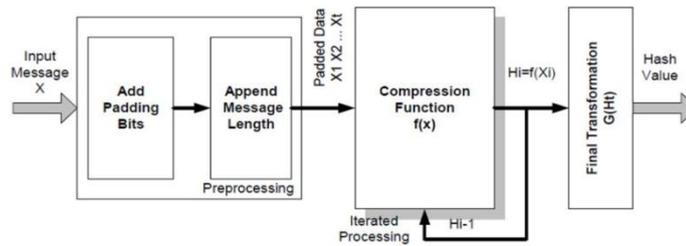


Рисунок 1 –Общая модель хэш-функции.

Процедура разделена на предварительную обработку, сжатие и окончательное преобразование. Предварительная обработка в основном добавляет необходимое количество битов к входному сообщению, чтобы сформировать заполненный блок данных заданной длины. Дополненные данные делятся на  $t$  блоков одинаковой длины. Каждый блок  $X_i$  служит входом для функции сжатия  $h$ , которая каждый раз вычисляет новое преобразованное сообщение данных  $H_i$ , как функцию предыдущего  $H_{i-1}$  и входного  $X_i$ . После определенного количества циклов обработки данные окончательно модифицируются в результате окончательного преобразования. Таким образом генерируется хэш-значение (дайджест сообщения), соответствующее входному сообщению  $x$ . Предложенная архитектура гарантирует высокий уровень безопасности во всех приложениях, требующих аутентификации сообщения, посредством создания кода аутентификации сообщения. Уровень безопасности и преимущества хэш-функции SHA-2, на которой основана предложенная архитектура, обеспечивают высокий уровень безопасности. При реализации этой схемы аутентификации Хэш-функция SHA-2 представляет собой криптографические алгоритмы, которые принимают в качестве входных данных сообщение произвольной длины, и которые возвращают дайджест (или хэш-значение) фиксированной длины (от 160 до 512 бит в большинстве приложений). Хэш-функции используются во множестве протоколов, будь то для цифровых подписей на высокопроизводительных серверах или для аутентификации встроенных систем

**Список использованных источников:**

1. Качинский, М. В. Конвейерный процессор хэш-функции SHA-256 / М. В. Качинский, А. В. Станкевич // Информационные технологии и системы 2018 (ИТС 2018) = Information Technologies and Systems 2018 (ITS 2018) : материалы международной научной конференции, Минск, 25 октября 2018 г. / Белорусский государственный университет информатики и радиоэлектроники ; редкол. : Л. Ю. Шилин [и др.]. – Минск, 2018. – С. 158 - 159.
2. Качинский, М. В. Высокопроизводительная реализация криптографической хэш-функции SHA-256 на базе FPGA / М. В. Качинский, А. В. Станкевич // Технические средства защиты информации : тезисы докладов XVI Белорусско-российской научно – технической конференции, Минск, 5 июня 2018 г. – Минск: БГУИР, 2017. – С. 47.
3. Xiao-yang Zeng, Shi-tingLu, "A core-based multi-function security processor with GALS Wrapper", Solid-State and Integrated-Circuit Technology 2008. ICSICT 2008. 9th International Conference on, pp. 1839-1842, 2008.

## ЛОГИЧЕСКАЯ ОПТИМИЗАЦИЯ НЕПОЛНОСТЬЮ ОПРЕДЕЛЁННЫХ ФУНКЦИЙ

Грицовец А.О.

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Бибило П.Н. – д.т.н., профессор*

С каждым днём задачи проектирования цифровой аппаратуры становятся более объёмными, а требования к электронным вычислительным средствам по габаритам и затраченным ресурсам становятся всё жёсткими. Многие функциональные блоки управляющих и вычислительных устройств описываются булевыми функциями, поэтому оптимизация различных форм представлений систем булевых функций по-прежнему остается актуальной задачей.

На сегодняшний день многие задачи требуют большей производительности при меньших затратах при производстве электронных вычислительных средств. Производители стараются найти наилучшее решение при проектировании устройств. Для выполнения этой задачи часто прибегают к логической оптимизации булевых функций, которые являются математическими моделями функционирования многовыходных комбинационных схем.

Один из подходов к оптимизации булевых функций – аппарат диаграмм двоичного выбора [1-5]. Диаграммы двоичного выбора – компактная форма представления булевых функций в виде ациклического графа, которая соответствует многоуровневому представлению на базе разложения Шеннона. В начале логической оптимизации булевых функций на основе диаграмм двоичного выбора

требуется выбрать последовательность переменных, по которой ведётся многоуровневое разложение Шеннона. На следующем этапе строится диаграмма двоичного выбора по заданной перестановке переменных. Далее происходит поиск одинаковых поддеревьев в дереве (графе представления системы функций). Одинаковые поддеревья удаляются и заменяются уже найденным аналогичным поддеревом, тем самым удаляются лишние узлы, и сложность представления функций уменьшается.

Ключевым этапом в оптимизации является поиск одинаковых поддеревьев. Он начинается с листьев и продолжается до тех пор, пока не обнаружится, что одинаковых поддеревьев в дереве больше нет. Также важным этапом оптимизации булевых функций с помощью диаграмм двоичного выбора является выбор порядка переменных, по которым ведётся разложение Шеннона. От этого выбора зависит количество одинаковых поддеревьев, а следовательно, и степень оптимизации. Хорошей эвристикой при выборе перестановки переменных является порядок, в котором сначала идут наиболее часто встречающиеся переменные, а в конце - наименее встречающиеся в исходной системе дизъюнктивных нормальных форм (ДНФ), задающей исходную систему либо уже полученные подфункции. Этот подход прост в реализации и даёт хорошие результаты, так как существует большая вероятность обнаружения «больших» одинаковых поддеревьев, и это позволяет упростить граф.

Зачастую встречаются ситуации, когда используются неполностью определённые булевы функции. Неполностью определённые функции, или частичные функции, - это функции, значения которых не определены на некоторых наборах значений переменных, на данных наборах значения функций при оптимизации заменяются определёнными (0,1). При логической оптимизации неполностью определённых функций при помощи диаграмм двоичного выбора алгоритмы выбора перестановки и построения диаграммы аналогичны. Однако при поиске одинаковых поддеревьев неопределённые листовые значения следует дополнять до значений 0 или 1, чтобы получилось наибольшее число одинаковых поддеревьев. Таким образом, общий алгоритм логической оптимизации неполностью определённых функций начинается с этапа выбора порядка переменных разложения. Далее идёт этап разложения по выбранной перестановке переменных. На третьем этапе производится анализ всего дерева и дополнение неопределённых листьев до значения 0 или 1. На заключительном этапе производится поиск одинаковых поддеревьев и удаление избыточностей.

**Список использованных источников:**

1. Карпов, Ю. Верификация параллельных и распределённых программных систем /Ю. Карпов. - СПб: «БХВ-Петербург», 2010 – 560 с.
2. Бибило, П. Н. Применение диаграмм двоичного выбора при синтезе логических схем /П. Н. Бибило. – Минск: «Беларуская навука», 2014 – 231с.
3. Бибило, П. Н. Логическое проектирование дискретных устройств с использованием продукционно-фреймовой модели представления знаний /П. Н. Бибило, В. Н. Романов – Минск.: «Беларуская Навука», 2011. - 279 с.
4. Бибило, П. Н. Минимизации многоуровневых представлений систем булевых функций на основе разложения Шеннона / П. Н. Бибило, Ю. Ю. Ланкевич // Информатика, 2017 – № 2. – С. 45- 57.
5. Бибило П. Н. Алгоритм построения диаграммы двоичного выбора для системы полностью определённых булевых функций / П. Н. Бибило, П. В. Леончик - // Управляющие системы и машины. - 2009. - № 6. – С. 42 – 49.

## **ИССЛЕДОВАНИЕ КРИТЕРИЕВ ОЦЕНКИ КОРРЕКТНОСТИ ПРОИЗНОШЕНИЯ ФРАЗ ДЛЯ СИСТЕМЫ ИСПРАВЛЕНИЯ РЕЧЕВЫХ ДЕФЕКТОВ**

*Демидович В.С.*

*Белорусский государственный университет информатики и радиоэлектроники  
Г. Минск, Республика Беларусь*

*Лихачев Д.С. – к.т.н., доцент*

Речь человека имеет огромное количество характеристик таких как темп речи, громкость, высота голоса, интонация, тембр и др. Эти характеристики делают речь каждого человека уникальной и неповторимой, что в свою очередь усложняет ее анализ.

В данной статье описан метод, позволяющий определить правильность произношения фразы или слова, основываясь на *DTW-алгоритме* с использованием мел-кепстральных коэффициентов. Данные коэффициенты, расположенные на мел-шкале, позволяют выделить наиболее значимые для восприятия человеком частоты.

Для определения десяти мел-кепстральных коэффициентов используется частотный диапазон от 75 Гц до 5000 Гц. Для преобразования частоты в мел используется формула (1):

$$M = 1127 * \log \log \left( 1 + \frac{F}{700} \right) \quad (1)$$

Обратное преобразование производится по формуле (2):

$$F = 700 * \left( e^{\frac{M}{1127}} - 1 \right) \quad (2)$$

Условно разделим человеческую голос на мужской (высокий, средний и низкий) и женский (высокий, средний и низкий). В качестве эксперимента был использован мужской средний голос. Важно отметить, что эталонные образцы голоса и образцы голоса тестируемого человека имеют одинаковую длину. Для каждой фразы имеется по три записи: эталонная запись (эталон), запись человека с нормальным произношением (норма) и запись человека с нарушением речи (дефект).

Для анализа речевого сигнала весь аудиосигнал разбивается на фреймы длительностью 50 мс. Далее рассчитывается спектр сигнала с помощью быстрого преобразования Фурье.

Для расположения полученного ранее спектра на мел-шкале необходимо создать «ребенку» фильтров. Каждый фильтр – это треугольная функция, позволяющая получить суммарное количество энергии на определенном диапазоне частот.

Диапазон частот [75; 5000] на мел-шкале согласно формуле (1) равен [114.7; 2363.5]. Для построения десяти треугольных фильтров необходимо двенадцать опорных точек.

Получим:  $m = [114.7; 319.1; 523.5; 729.9; 934.3; 1138.7; 1343.1; 1547.5; 1751.9; 1956.3; 2160.7; 2363.5]$ .

Используя формулу (2) переведем мел-шкалу в герцы.

Получим:  $h = [75; 229; 414; 638; 904; 1223; 1605; 2063; 2613; 3272; 4061; 5000]$ .

Далее необходимо наложить шкалу на спектр фрейма по формуле (3):

$$f(i) = (frameSize + 1) * \frac{h(i)}{sampleRate}, \quad (3)$$

где  $frameSize$  – длина фрейма;  $h(i)$  – значение опорной точки в Гц;  $sampleRate$  – частота звука в Гц, равная половине частоты дискретизации;  $f(i)$  – соответствующие опорные точки на оси спектра.

Зная опорные точки на оси спектра, строим фильтры, используя формулу (4):

$$H_m(k) = \begin{cases} 0, & k < f(m-1) \\ \frac{k-f(m-1)}{f(m)-f(m-1)}, & f(m-1) \leq k \leq f(m) \\ \frac{f(m+1)-k}{f(m+1)-f(m)}, & f(m) \leq k \leq f(m+1) \\ 0, & k > f(m+1) \end{cases} \quad (4)$$

где  $k$  – номер отсчета фрейма;  $f(m)$  – опорные точки на оси спектра.

Для получения мел-коэффициентов необходимо полученные значения попарно перемножить с энергией спектра, после чего прологарифмировать (5). Таким образом понижается чувствительность коэффициентов к шумам.

$$S[m] = \log \left( \sum_{k=0}^{N-1} |X[k]|^2 * H_m[k] \right), 0 \leq m \leq 10 \quad (5)$$

Для получения мел-кепстральных коэффициентов необходимо применить дискретное косинусное преобразование (ДКП) к мел-коэффициентам. ДКП используется для повышения значимости первых коэффициентов и уменьшения значимости последних.

В результате для каждого аудиосигнала получена матрица  $H(m,k)$ , где  $m$  – количество кепстральных коэффициентов (в нашем случае десять),  $k$  – количество фреймов аудиосигнала.

Таким образом, для каждой фразы имеется три матрицы (эталон, норма, дефект). С помощью dtw-алгоритма определяется минимальная сумма евклидовых расстояний между матрицами эталона и нормы; матрицами эталона и дефекта.

Проведя десяток экспериментов было установлено, что среднее арифметическое разности сумм евклидовых расстояний равно 7,73, а минимальная разность – 6,91.

Кепстральный коэффициент 3 для слова «двери» в зависимости от времени показан на рисунке 1а, на рисунке 1б изображены кепстральные коэффициенты четвертого фрейма для данного слова.

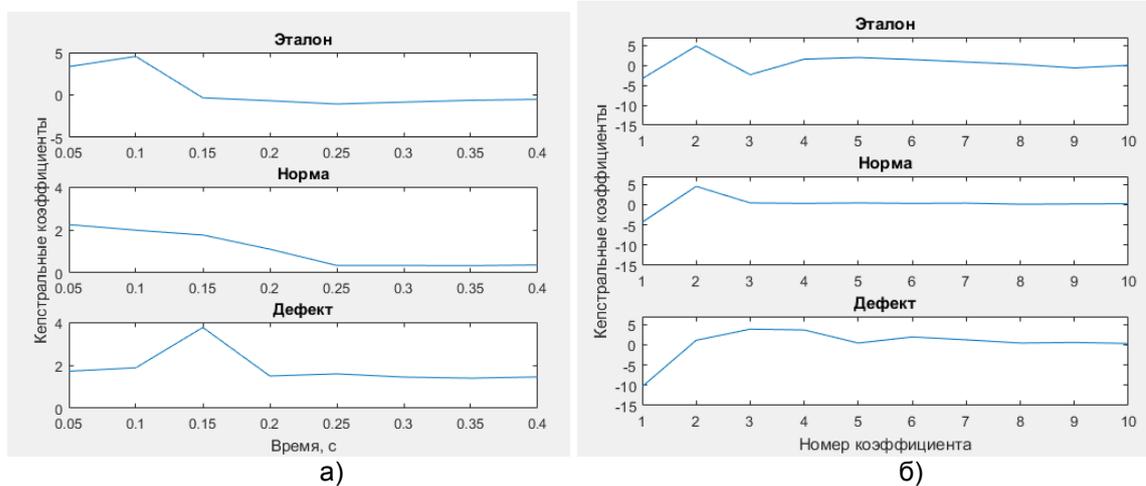


Рисунок 1 - Кепстральный коэффициент 3 (а), кепстральные коэффициенты четвертого фрейма (б).

Можно сделать вывод о возможности использования кепстральных коэффициентов для определения правильности произношения фразы или слова. Граница разности сумм евклидовых расстояний при которой можно считать, что человек верно произнес фразу равна 6,8.

**Список использованных источников:**

1. Davis, S. Mermelstein, P. (1980) *Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences*. In IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 28 No. 4, pp. 357-366.
2. Xuedong Huang, Alex Acero, Hsiao-Wuen Hon, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*, Prentice Hall, 2001, ISBN:0130226165.

## ИНТЕЛЛЕКТУАЛИЗИРОВАННАЯ СИСТЕМА УПРАВЛЕНИЯ СВЕТОВЫМИ ЭФФЕКТАМИ

Добровольский Д.А.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Шемаров А.И. – д-р. техн. наук, профессор

В современных реалиях большинство областей развлекательной индустрии подразумевают использование автоматизированных систем для генерации контента. Однако существуют области, где всё ещё необходим исключительно человеческий труд, и одна из этих областей - работа со сценическим оборудованием. Поскольку конструкция освещения сцен варьируется от случая к случаю, время, затрачиваемое на подготовку и программирование освещения велико, в основном из-за полностью ручного программирования каждого осветительного устройства.

Данная система направлена на удешевление и ускорение процесса подготовки сцены к выступлению путем применения системы предварительного программирования управляемых прожекторов и создание базовой программы освещения сцены. Используя сгенерированную программу освещения, режиссеру освещения остается лишь внести небольшие изменения, соответствующие желанию заказчика - такие как смена цветовой гаммы и добавление уникальных сцен освещения.

Сценическое освещение - это набор управляемого светового оборудования, способное менять свойства направления, яркости, цвета, форму светового луча или сразу несколько из них. Устройства освещения подключаются последовательно к общей сети передачи данных и работают независимо друг от друга. Для передачи команд управления на данные устройства часто используется порт RS-485 с использованием протокола DMX512. Порт представляет собой разъем с тремя или пятью пинами, отведенные под передачу, прием данных и заземление соответственно. Протокол DMX (Digital Multiplex) является простой реализацией передачи данных, состоящих из 513 байт, где главный байт отводится на инициализацию и информирование о формате сессии, а последующие 512 байт несут информацию для устройств. Если взять за основу монохромный статичный прожектор с единственным элементом контроля яркостью, то на одну шину возможно подключить до 512 таких устройств. Если же добавить к функционалу прожектора добавить возможность изменять цвета, то количество информации к каждому прожектору потребует передавать 3 байта информации в RGB-

пространстве, что уменьшит количество прожекторов на одной шине до 170.

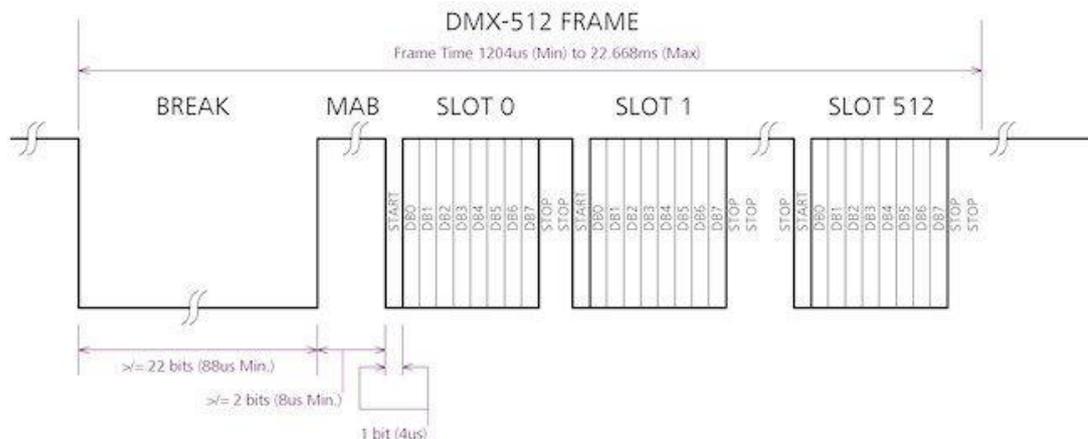


Рис. 1 - Состав пакета DMX-512

Работа системы сводится к анализу входящего сигнала и разделению его на ряд простых параметров, на основе которых создаются уникальные сцены освещения для каждого фрагмента мелодии. Необходимость среднего слоя в обработке сигнала обусловлена различием между музыкальной и цветовой гармоникой. Под рядом простых параметров подразумевается громкость, темп (частота повторения рисунка мелодии), ритм (организация звуков по временной шкале), тон и тональность (высота звука и её звучание относительно базового или самого низкого тона музыкальной шкалы). Для реализации последнего звук разделяется окнами на фрагменты частот, присущих разной группе музыкальных инструментов. Все параметры вычисляются рекурсивно на всей протяженности мелодии. На основе динамики этих параметров трек разбивается на сегменты, в рамках которых в дальнейшем будут созданы оригинальные программы освещения для каждого сегмента. Границы сегментов выбираются не только на основе предшествующих тактов, но и тактов, которые будут следовать, что приводит к задержке первого запуска системы в 16 тактов.

После разбиения на базовые параметры и создания сегментов, система создает полигональную карту освещения. Карта представляет собой сторону цилиндра, расположенного параллельно крайнему верхнему ряду осветительного оборудования, находящегося как можно дальше от края сцены. На данную полигональную плоскость наносятся проекции лучей света из каждого осветительного устройства, доступного для использования. Каждому устройству на основе параметров задается оригинальный рисунок проекции для каждого сегмента, которые привязываются непосредственно к стилю исполнению музыки в каждом сегменте. Затем каждое устройство объединяется в группы посредством бинарного дерева (левый/правый или центр/край) для синхронизации освещения прожекторов относительно друг друга. В конце на основе сгенерированной полигональной карты рисунков создается готовый набор команд для каждого устройства, с которыми в дальнейшем будет иметь дело специалист.

Данный алгоритм предназначен для создания уникальной программы освещения для сцены и требует предварительной записи. Затем во время живого выступления система загружает сгенерированные прежде рисунки проекции для устройства и карты фрагментов для синхронизации освещения со звучанием на сцене. Таким образом система улучшает качество освещения на сцене, в реальном времени синхронизируя работу света с музыкой, а также снижает нагрузку на режиссера светового оборудования, позволяя ему сфокусироваться на более узкопрофильных проблемах.

**Список использованных источников:**

1. Фриберг А. Характеристика восприятия музыкальных параметров /Фриберг А., Скундервальт Э., Хэдблед А.
2. Фаирхарст Г. Параметры DMX / Фаирхарст Г. //Университет Абердин. – Шотландия, 2014. – 2 с.

# ТЕХНОЛОГИЯ МОНИТОРИНГА ПЕРЕМЕЩЕНИЯ ГРУЗОВ ИСПОЛЬЗУЯ РАДИОЧАСТОТНУЮ ИДЕНТИФИКАЦИЮ

Ёжкин А. А.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Сорока Н. И. – к.т.н., доцент

Применение технологии RFID (англ. Radio Frequency IDentification, радиочастотная идентификация) и ее возможности безграничны. Ограничены они лишь воображением человека. Не смотря на это, есть мнение, что радиочастотная идентификация используется только для контроля сети сбыта товара или для отрасли, использующей товары в упаковке. Но на данный момент, диапазон использования RFID-систем далеко вышел за границы этих областей применения. При помощи невероятно быстрого темпа развития технологий, уже запущено множество разнообразных типов систем радиочастотной идентификации. Одна прикладная система может найти применение в многих областях и иметь разные характеристики, в зависимости от области применения.

Не смотря на, уже существующие решения в области радиочастотной идентификации, радиочастотная идентификация является технологией, которая только начинает зарождаться и поэтому миру еще предстоит освоить огромный потенциал данной технологии в областях, где можно использовать данную систему, и получить максимальную выгоду из этого.

В настоящее время использование RFID-систем в некоторых областях находится на стадии планирования и создания прототипа системы, другие только начали исследование данных систем, а остальные просто-напросто не получили достаточного внимания отрасли и изготовителей. Проще говоря, некоторые прикладные системы должны “дорости” до получения всеобщего признания (как с позиции пригодности для бизнес-процессов, так и с технологической стороны) до того, как системы будут запущены в массовое производство.

Благодаря невероятно скорым темпам развития технологии радиочастотной идентификации и связанных с ней изделий, активно появляющихся на рынке, количество прикладных систем радиочастотной идентификации все время увеличивается. Одни типы систем уже находятся в серийном выпуске. Другие же типы систем либо находятся на стадии планирования, либо на стадии создания прототипа.

Существующие типы прикладных систем, которые выпускаются серийно:

- системы инвентарного учета (контроль и наблюдение);
- системы наблюдения за имуществом;
- система защиты от краж;
- система электронных платежей;
- система контроля за доступом;
- системы защиты от несанкционированного доступа.

Учитывая, что в настоящий момент времени активно развивается мир “Интернет вещей” (IoT (англ. Internet of Things)) и основываясь на данных типах прикладных систем было принято решение разработать систему отслеживания перемещения объектов. А именно перемещение железнодорожных грузов от заказчика к потребителю.

Прикладная система отслеживания перемещения грузов будет характеризоваться следующим: к объекту, который необходимо контролировать, прикрепляется метка, содержащая в себе уникальный идентификатор. Идентификатор радиочастотной метки считывается в определенных местах по мере перемещения объекта, где находится радиочастотный считыватель. Если связать информацию о времени чтения и местоположении с идентификатором, то можно в режиме реального времени получить информацию о том, где находится данный объект в конкретный момент времени.

Кроме информации о местонахождении груза в определенный момент времени, вы так же сможете получать информацию о именах сотрудников, перемещающих груз из одной точки в другую. Эта информация может оказаться довольно полезной, например, для определения лиц, ответственных за недостачу, если такая произойдет. Благодаря возможности контроля грузов в режиме реального времени, при обнаружении отсутствия сигнала на одной из радиочастотных меток в определенном месте в заданное время, вы сможете включить сигнал тревоги, либо оповестить перевозчика груза о проведении проверки в определенном месте (где находится радиочастотная метка, сигнал которой не поступил на считыватель).

Для отслеживания грузов в режиме реального времени можно применить несколько решений:

- Систему спутникового отслеживания

Контроль за перемещением груза с помощью спутника можно осуществлять по запросу в любой момент времени. Но, к сожалению, радиочастотные метки, способные осуществлять обмен информацией по прямой спутниковой связи, еще только начинают появляться на рынке.

- Систему с метками ограниченного диапазона чтения

При использовании радиочастотных меток ограниченного диапазона чтения необходимо обеспечить сканирование метки в месте установки считывающего устройства, а именно RFID ридера. Если заказчик системы выбирает такой вариант и у него будет достаточное количество точек считывания, то такое решение может обеспечить необходимые рабочие характеристики.

На основе приведенной информации, принцип реализации системы отслеживания перемещения грузов представлен на рисунке 1:

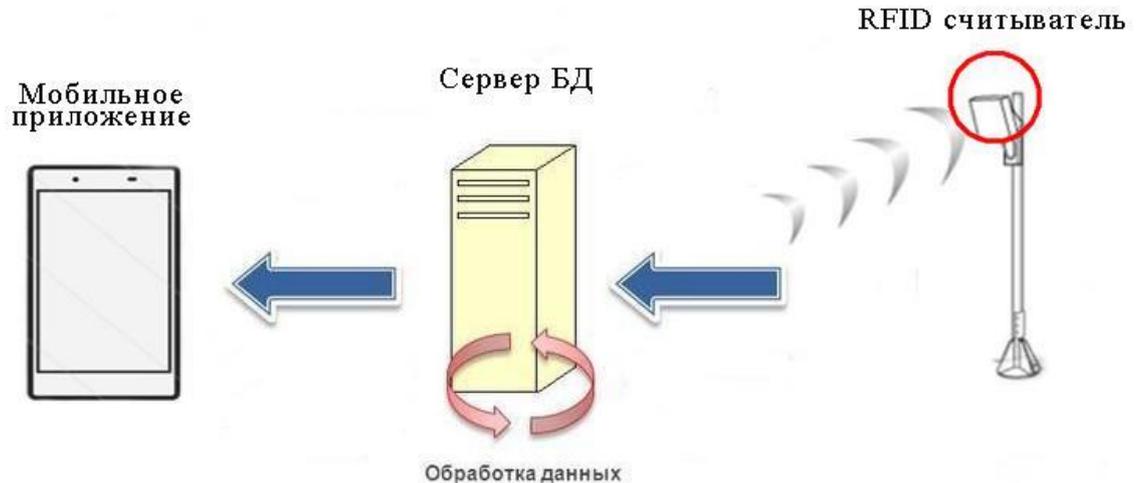


Рис. 1 – Структура реализации системы отслеживания перемещения грузов

В качестве отображения информации в режиме реального времени для пользователя выступает мобильное приложение.

Комбинирование системы радиочастотной идентификации и прикладной системы отслеживания перемещения грузов должны способствовать увеличению производительности труда вдвое по сравнению с существующими системами в грузовой отрасли. Также благодаря снижению числа инцидентов с потерей того или иного груза на пути следования объекта, данная система приведет к росту эффективности цепочек поставок груза.

**Список использованных источников:**

1. Сандип Лахири. RFID. Руководство по внедрению. Пер. с англ. – М.: КУДИС-ПРЕСС. – 2007. – 312 с., илл.
2. Еременов В. П. Интеллектуальные системы управления: Учебное пособие. М.: Книжный дом "ЛИБРКОМ", 2009. – 304 с.
3. Башмаков А. И., Башмаков И. А. Интеллектуальные информационные технологии: Учеб. пособие. – М.: Изд-во МГТУ им. Н. Э. Баумана, 2005. – 304 с.: ил. – (Информатика в техническом университете)

## АВТОМАТИЧЕСКАЯ СИСТЕМА РАДИОЧАСТОТНОЙ ИДЕНТИФИКАЦИИ ГРУЗОВ

Ёжкин А. А.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Сорока Н. И. – к.т.н., доцент

В настоящее время, при непрерывном росте логистических потоков, а именно грузовых и транспортных, колоссальном расширении транспортных позиций, вопросы логистики играют все большую роль. Самой узнаваемой и признанной технологией идентификации, применение которой используется в задачах складской логистики, является штриховое кодирование.



и другие. На генератор одновременно подаются два импульсных напряжения: одно – с импульсного модулятора, который формирует требуемую длительность зондирующего сигнала; другое – с генератора пилообразного напряжения (ГПН), который осуществляет частотную модуляцию на протяжении всего импульса. Необходимую стабильность частоты в схеме поддерживает система автоподстройки частоты (АПЧ) средней частоты линейно-частотной модуляции (ЛЧМ) сигнала и система автоматической подстройки закона модуляции. Принцип работы системы основан на сравнении выходного сигнала с эталонным. Если выходной сигнал не совпадает с эталонным, то вырабатывается сигнал ошибки, который отправляется на ГПН (генератор пилообразного напряжения). В следствии чего изменяется амплитуда пилообразного напряжения, следовательно и закон изменения частоты задающего генератора – в сторону уменьшения ошибки.

Запускающий импульс (зондирующий импульс поступивший с детектора) поступает на линию задержки. В то же время, на длительность зондирующего импульса, детектор отпирает ключ. Энергия зондирующего импульса через открытый ключ поступает на выпрямитель и запитывает конденсатор схемы. Напряжение конденсатора подается на генератор высокой частоты, считывающее устройство и генератор командного сигнала, однако энергия не расходуется, так как не поступало команды на запуск. Далее, линия задержки выдает запускающий импульс на генератор командного сигнала, эта информация попадает на считывающее устройство и генератор высокой частоты. Считывающее устройство отправляет сигнал на считывание блоку памяти и получает информацию в ответ, пересылая ее смесителю. Генераторы высокой частоты генерирует сигнал несущей частоты, направляет смесителю, откуда информация и энергия поступают на АЦП (состоящий и АИМ, генераторы тактовой частоты, квантователя и ИКМ), помехоустойчивый кодер. После всего этого, кодовые комбинации попадают отправляются на модулятор, преобразующий код в сигнал, подходящий для передачи через антенну.

С антенны сигнал поступает на линию задержки и далее происходит демодуляция, преобразуя сигнал в последовательность кодовых комбинаций. Декодер восстанавливает исходный код, обнаруживает и устраняет ошибки. Информация поступает на логическую схему. В это время источник питания запитывает логическую схему и позволяет записать информацию в блок памяти. Далее информация поступает на интерфейс связи. Т.к. у нас сетевой ридер, то информация поступает в систему электронной логистики с помощью Wi-fi технологии.

Переход от штрих-кодирования к радиочастотной идентификации является залогом успешного будущего складской автоматизации и логистики.

**Список использованных источников:**

1. Сандип Лахири. RFID. Руководство по внедрению. Пер. с англ. – М.: КУДИС-ПРЕСС. – 2007. – 312 с., илл.
2. Евменов В. П. Интеллектуальные системы управления: Учебное пособие. М.: Книжный дом "ЛИБРКОМ", 2009. – 304 с.
3. Башмаков А. И., Башмаков И. А. Интеллектуальные информационные технологии: Учеб. пособие. – М.: Изд-во МГТУ им. Н. Э. Баумана, 2005. – 304 с.: ил. – (Информатика в техническом университете)

## **ИТЕРАТИВНО-КОНВЕЙЕРНЫЙ ПРОЦЕССОР ХЭШ-ФУНКЦИИ SHA-256 НА БАЗЕ ПЛИС**

*Ероховец В.А.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Станкевич А.В. – к.т.н., доцент*

Функция хеширования SHA-256 достаточно популярна последние несколько лет. Являясь относительно простой, она идеально подходит для самых разных целей. Существует множество различных подходов к реализации алгоритма, как программных, так и аппаратных. В докладе рассматривается аппаратная итеративно-конвейерная реализация, так как она занимает относительно небольшой объем ресурсов и является достаточно производительной.

В настоящее время достаточно остро стоит проблема сохранения конфиденциальности информации, а также корректности ее передачи. Целостность передаваемой информации можно контролировать при помощи сообщения, полученного путем обработки исходного послания используя хеш-функции.

Алгоритм хеширования SHA-256 относится к семейству алгоритмов SHA-2 (Secure Hashing Algorithm), разработчиком которого является Агентство национальной безопасности США.

Реализация данного алгоритма аппаратным способом значительно повышает скорость обработки данных по сравнению с программным.

Исходя из алгоритма, можно выделить две составные части реализации:

- 1) экспандер («расширитель»);
- 2) компрессор («уплотнитель»).

С помощью экспандера сообщение удлиняется для последующей обработки. А в компрессоре уже проходит эта обработка, которая предполагает в общем случае сжатие исходного сообщения до длины 256 бит.

Для того, чтобы получить более высокую тактовую частоту, необходимо уменьшить длину критического пути как в экспандере, так и в компрессоре, что достигается путем модификации канонической модели [2]. На рисунке 1 представлены модифицированные структурные схемы экспандера и компрессора:

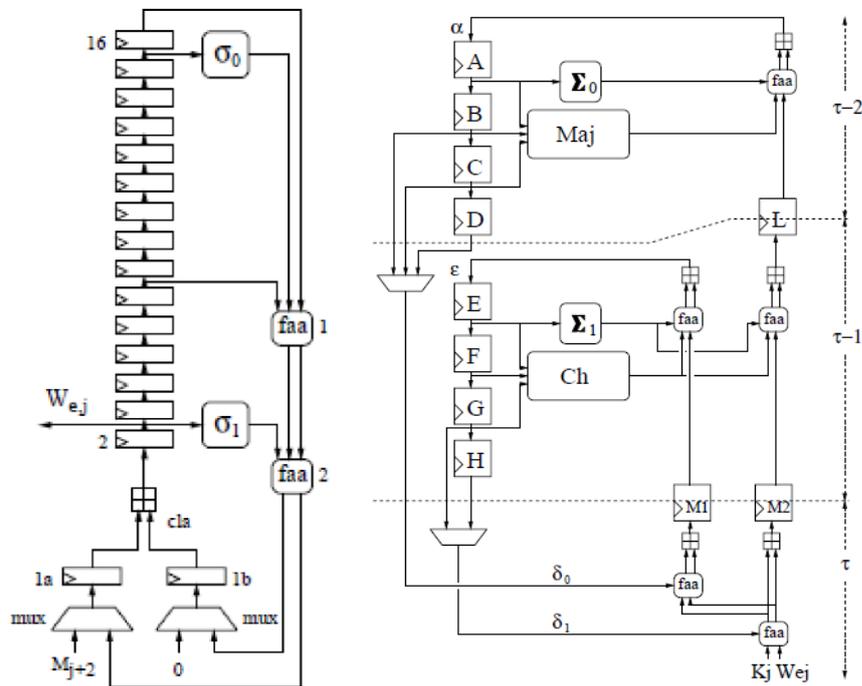


Рисунок 1 – Структурные схемы экспандера и компрессора соответственно [2]

Приведенное архитектурное решение реализуется на FPGA семейства Virtex 7 компании Xilinx. Проводятся исследования производительности и аппаратных затрат реализации.

**Список использованных источников:**

1. D. Eastlake 3rd., Motorola Labs/ T. Hansen – AT&T Labs, July 2006. – 108 P.
2. L. Dadda, M. Macchetti, J. Owen . An ASIC Design for a High Speed Implementation of the Hash Function SHA-256 (384, 512). Proc. of GLSVLSI '04 Proceedings of the 14th ACM Great Lakes symposium on VLSI, Boston, MA, USA — April 26 - 28, 2004. - ACM New York, NY, USA, 2004. – pages 421–425.
3. Wikipedia [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://ru.wikipedia.org/wiki/SHA-2>

## АППАРАТНАЯ ПЛАТФОРМА ДЛЯ ПРОТОТИПИРОВАНИЯ АЛГОРИТМОВ И АРХИТЕКТУР СЛУХОВЫХ АППАРАТОВ НА ОСНОВЕ СНК ZYNQ

Кивачук А.В.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Вашкевич М.И. – к.т.н., доцент

В настоящее время около 1 миллиарда человек (16% населения мира) страдают потерей слуха. Большинство из них не имеют слухового аппарата либо недовольны качеством его работы. Необходимость в разработке новых и совершенствовании существующих алгоритмов обработки сигналов в слуховых аппаратах требует наличия соответствующих аппаратных платформ. Существует большое количество платформ для обработки сигнала, однако не хватает мобильных платформ, подходящих для реализации алгоритмов, необходимых для слуховых аппаратов. В работе представлена аппаратная

платформа для прототипирования алгоритмов и архитектур слуховых аппаратов на основе реконфигурируемой системы-на-кристалле (СнК) Zynq.

Одним из самых распространенных устройств для людей с нарушением слуха является слуховой аппарат. Основной задачей слухового аппарата является компенсация нарушения слуховой функции. Как правило, человек с нарушенным слухом имеет повышенный порог слышимости, что мешает ему воспринимать звуки малой интенсивности. Поэтому важной функцией слухового аппарата является усиление интенсивности тех участков, которые находятся ниже порога чувствительности. Также важной функцией является шумоподавление, так как фоновый шум ухудшает разборчивость речи, особенно для людей с тугоухостью.

Ядром слухового аппарата является банк фильтров (его параллельная структура можно использовать для ускорения процесса обработки сигнала), а также вычислитель, отвечающий за подсистему шумоподавления [1].

Чтобы реализовать такой слуховой аппарат, необходимо выбрать платформу, которая может справиться с вышеизложенными задачами.

На сегодняшний день существует несколько платформ для решения данного рода задач:

- 1) платформа на базе персональных компьютеров [2],
- 2) платформа на базе процессоров ЦОС [3],
- 3) платформа на базе FPGA,
- 4) платформа на базе СнК [4].

Каждая из вышеперечисленных платформ имеет свои преимущества и недостатки. Проанализировав все платформы, можно отметить, что:

- платформы на базе персональных компьютеров дают большую задержку,
- платформы на базе процессоров ЦОС удобны с точки зрения программирования, но имеют ограничение по производительности, по памяти и по классу алгоритмов (подходят для реализации поблочной обработки, но затруднительно выполнить алгоритмы, где нужна обработка отсчет-за-отчетом или требуется реализация множества параллельных структур, как в случае с банком фильтров),
- платформы на базе FPGA практичны в плане реализации параллельных структур, однако не удобны с точки зрения программирования последовательных процедур, которые часто требуются при реализации алгоритмов обработки сигнала в слуховых аппаратах, а также плохо подходят для обработки чисел с плавающей запятой,
- платформы на базе СнК сочетает в себе достоинства платформ на базе FPGA и ЦОС, что позволяет расширить спектр решаемых ими задач.

Zybo – плата на основе СнК, сочетающих в себе ядра ARM Cortex-A9 и FPGA. На такой СнК можно строить системы, сочетающие в себе алгоритмически сложные задачи, проще поддающихся реализации на процессоре, и задачи, требующие жесткого реального времени, обработки сигналов, обработки изображений и другие задачи, для реализации которых удобны FPGA [4].

В работе предлагается аппаратная платформа для прототипирования алгоритмов и архитектур слуховых аппаратов на основе СнК Zynq, структура которой приведена на рисунке 1.

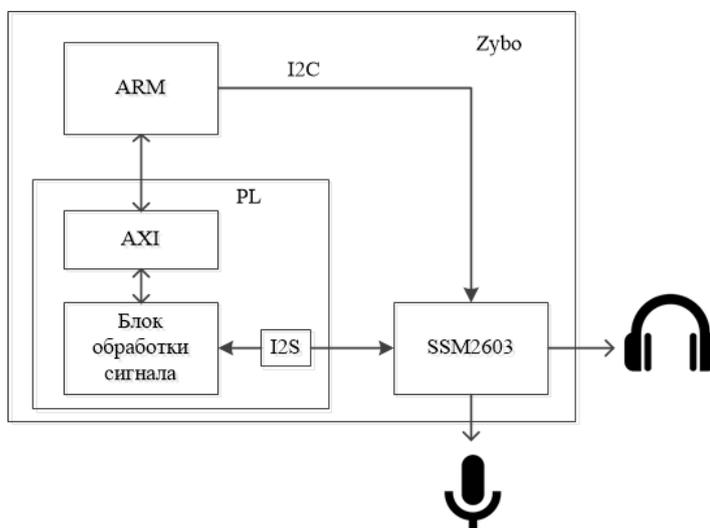


Рисунок 1– Схема реализации аппаратной платформы для прототипирования алгоритмов и архитектур слуховых аппаратов на основе СнК Zynq

Согласно представленной схеме, в процессоре ARM будет производиться обновление коэффициентов фильтров и реализовываться алгоритмы шумоподавления.

PL – программируемая логика, используется для применения реконфигурируемых банков фильтров.

Микросхема SSM2603 [5] является аудиокодеком, который содержит ЦАП и АЦП, то есть принимает поступающий сигнал с микрофона и отправляет обработанный на наушники.

В блоке обработки сигнала находятся реконфигурируемые банки фильтров [6], где и осуществляется формирование конечного сигнала.

Связь между блоком обработки сигнала и SSM2603 осуществляется благодаря I2S аудио интерфейсу. Процессор ARM и SSM2603 общается по I2C интерфейсу.

**Список использованных источников:**

1. Вашкевич М.И., Азаров И.С., Петровский А.А. Косинусно-модулированные банки фильтров с фазовым преобразованием: реализация и применение в слуховых аппаратах. – М.: Горячая линия–Телеком, 2014. – 210 с.
2. Giso Grimm, Tobias Herzke, Daniel Berg, Volker Hohmann (2006). The Master Hearing Aid: A PC-Based Platform for Algorithm Development and Evaluation, pp. 618–628.
3. Christopher Seifert, Guillermo Paya-Vaya and Holger Blume (2015). A Mobile SoC-Based Platform for Evaluating Hearing Aid Algorithms and Architectures (IEEE), pp. 93–97.
4. Louise H. Crockett, Ross A. Elliot The Zynq Book 1st Edition – Scotland, UK, 2014.
5. SSM2603 [Электронный ресурс]. – Режим доступа: <https://www.analog.com/media/en/technical-documentation/data-sheets/ssm2603.pdf>.
6. Sneha Raj, Athira Shaji (2016). Design and Implementation of Reconfigurable Digital Filter Bank for Hearing Aid, pp. 1–6.

## ПОСТРОЕНИЕ ИНФРАКРАСНЫХ МУЛЬТИТАЧ СЕНСОРНЫХ ПАНЕЛЕЙ

*Клейменов А.А.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Станкевич А.В. – к.т.н., доцент*

С целью изучения типовых вариантов построения, а также вариантов схемотехнических решений проведён патентный поиск, исследованы приводимые в патентах варианты построения инфракрасных сенсорных панелей. Разработано ПО с возможностью изменения угла работы фотоэлементов (ИК-диоды, фототранзисторы), позволяющее визуально оценить покрытие ИК-лучами рабочей области ИК-сенсорного экрана. На данный момент проводятся работы по расширению функционала ПО и по разработке установки, которая позволит исследовать реальные электрических и оптических характеристики фотоэлементов.

На сегодняшний день сенсорные экраны активно используются в различных типах устройств: мобильных телефонах, видеокамерах, планшетных компьютерах, терминалах оплаты, в системах управления станков, климатических камер, производственных линиях и т.п. При использовании сенсорного экрана, в большинстве случаев, исчезает потребность в клавиатуре и одновременно ускоряется и упрощается ввод данных.

По критерию технологии изготовления сенсорные экраны, находящиеся в массовом производстве, делятся на следующие типы:

- резистивные;
- емкостные (поверхностно-емкостные);
- проекционно-емкостные;
- на поверхностно-акустических волнах (ПАВ);
- оптические;
- инфракрасные (ИК-сенсорные панели);
- дисперсионные (DSP – Dispersive Signal Technology).

Каждая технология производства сенсорных экранов обладает своими преимуществами и недостатками и рассчитана на использование в определённых условиях. Основными критериями оценки при выборе сенсорного экрана являются:

- количество одновременно обрабатываемых точек касания (single-, double-, multi-touch);
- точность;
- надёжность (ресурс нажатий);
- светотехнические характеристики (коэффициент пропускания, отражения);
- рабочий температурный диапазон и температура хранения;
- работа в условиях внешней яркой засветки (прямой солнечный свет);
- работа в перчатках;
- устойчивость к загрязнениям;
- устойчивость к повреждениям (вандалоустойчивость).

Для исследования нами была выбрана технология построения ИК-сенсорных экранов, т.к. такие экраны имеют высокую надёжность (неограниченный ресурс нажатий) и высокое качество изображения (отсутствие искажающих поверхностей), поддерживают работу в режиме мультитач, исправно работают в широком диапазоне температур, срабатывают от касания практически любым предметом, а также не требуют калибровки, в сравнении, например, с резистивными сенсорными экранами, которые теряют свои характеристики с течением времени и требуют периодической калибровки, имеют сенсорную поверхность, которая вносит искажения в изображение, а также не поддерживают множественные касания (более двух).

Конструкция ИК-сенсорной панели представляет собой модуль, состоящий из корпуса, в который установлена печатная плата с фотоэлементами (ИК-диоды и фототранзисторы) и со схемой управления. Фотоэлементы должны иметь фокусирующие линзы, которые исключают влияние небольших перекосов при установке элементов на печатную плату, а также обладать рабочей длиной волны от 860 до 940 нм. Также в конструкции устанавливается ИК-фильтр с длиной волны среза близкой к выбранной длине волны фотоэлементов, что позволяет корректно работать сенсору в широком диапазоне освещённости.

Проблему построения ИК-сенсорных экранов можно разделить на три основные части:

- 1) выбор пространственного расположения фотоэлементов на печатной плате с учётом максимального покрытия ИК-лучами рабочей области сенсора;
- 2) реализация схемотехнических решений (схем управления и усиления) исходя из пространственного расположения фотоэлементов и внешних условий работы сенсора;
- 3) реализация алгоритма определения положения касания.

Существует множество вариантов расположения фотоэлементов в устройстве ИК-сенсорной панели (рисунок 1) [2]. Типичным вариантом расположения является вариант, указанный на рисунке 1, г), т.к. в этом случае значительно упрощается трассировка печатной платы.

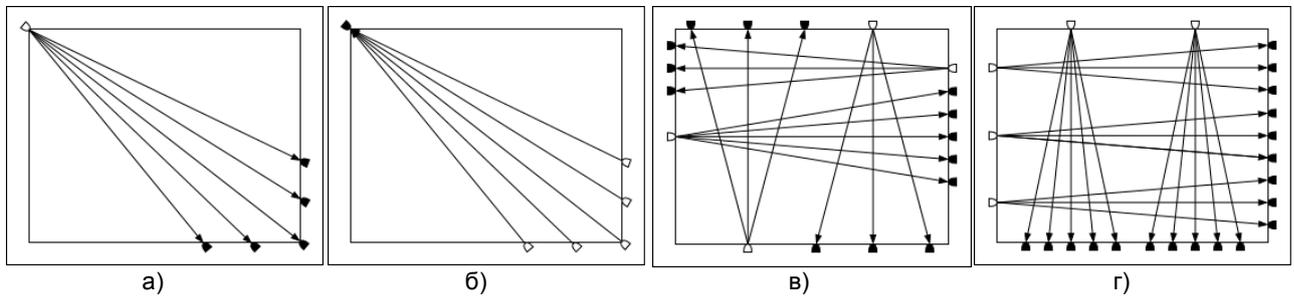


Рисунок 1 – Варианты компоновки фотоэлементов ИК-сенсорной панели

Для проверки характеристик работы ИК-сенсорных экранов, в идеальном случае, необходима практическая реализация каждой из представленных компоновок, что определённо будет связано с большими временными и финансовыми затратами. Таким образом, целесообразно моделирование работы ИК-сенсорного экрана при различных вариантах построения посредством программного обеспечения. В функционал такого ПО должна быть включена возможность изменения количества фотоэлементов и углов их работы. В данный момент нами разработано ПО (рисунок 2), реализующее вариант компоновки фотоэлементов по типу, указанному на рисунке 1, д), в нём доступно: изменение количества фотоэлементов, располагающихся на каждой из сторон сенсорной рамки, изменение угла работы ИК-диодов и фототранзисторов, а также реализована возможность добавления статических касаний (до 10).

Для получения реальных электрических и оптических характеристик фотоэлементов нами разрабатывается установка, которая упростит и ускорит исследование характеристик фотоэлементов при их парной работе. Одной из важнейших характеристик данной установки является возможность полуавтоматического изменения расстояния между фотоэлементами и последующее снятие характеристик, что необходимо для исследования режимов работы фотоэлементов при различных диагоналях ИК-сенсорной панели.

Исходя из результатов проделанных работ будет возможен переход к этапу практической реализации сенсорного экрана и разработки алгоритмов его работы.

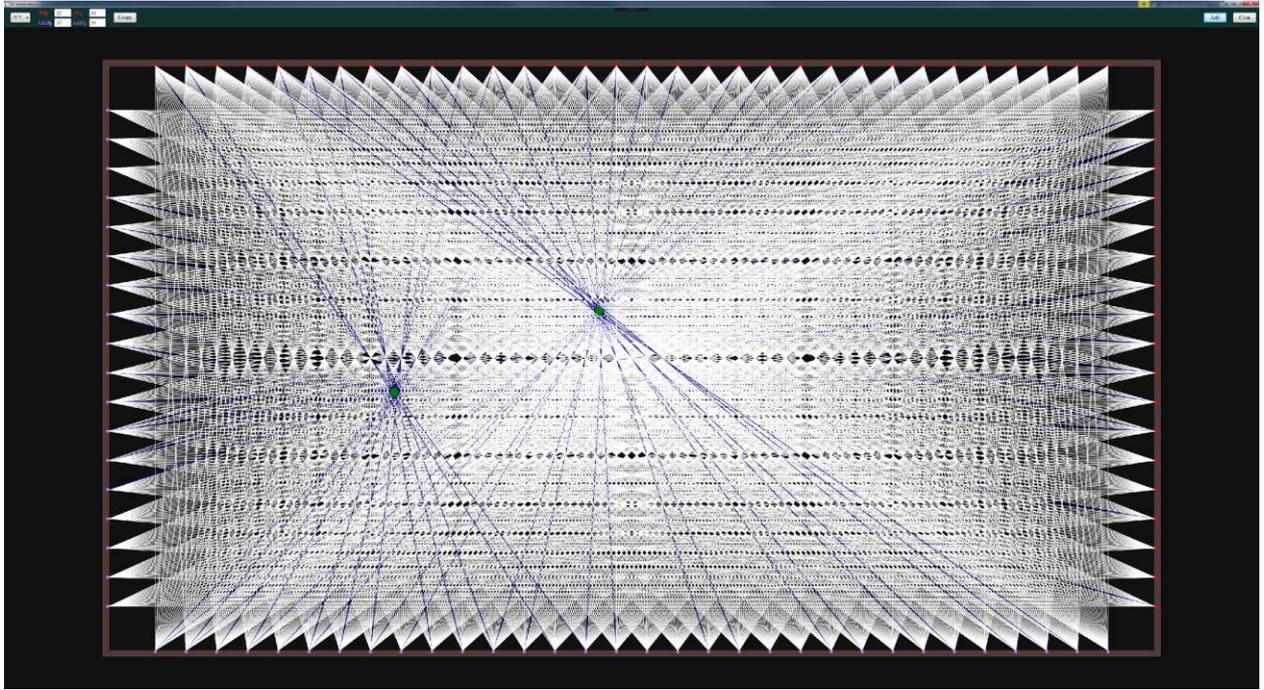


Рисунок 2 – Результат моделирования с помощью разработанного программного обеспечения

При разработке не исключается необходимость уточнения и корректировки принятых ранее решений для достижения таких характеристик как точность (не менее 3 мм) и время задержек (не более 20 мс) [3].

**Список использованных источников:**

1. Актуальные проблемы энергетики – 2016: материалы науч.-техн. конф. студентов и аспирантов, Минск, 2017 г. / Белорус. нац. технический ун-т; редкол.: Т. Е. Жуковская [и др.]. – Минск: БНТУ, 2017. – 537 с.
2. Google Patents [электронный ресурс] / Infrared ray touch panel device with high efficiency. – 2019. – Режим доступа: <https://patents.google.com/patent/US20110175848A1/en?q=US20110175848A1>. – Дата доступа: 15.03.2019.
3. Vasuki, S. An interactive infrared sensor based multi-touch panel / S. Vasuki, P. Mordhwaj, N. Rounak Singh // International Journal of Scientific and Research Publications – 2013. – Vol. 3, № 3. – ISSN 2250-3153.

## ЭЛЕКТРОННАЯ ЦИФРОВАЯ ПОДПИСЬ НА БАЗЕ ЭЛЛИПТИЧЕСКИХ КРИВЫХ

*Коминч В.В.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Станкевич А.В. – к.т.н., доцент*

Защита интеллектуальной собственности является одной из самых важных проблем века информации. Для решения данной проблемы специалистами были разработаны различные алгоритмы электронной цифровой подписи. В докладе рассматривается аппаратная реализация на базе ПЛИС белорусского стандарта СТБ 34.101.45 – 2013, который представляет собой алгоритм электронной цифровой подписи на эллиптических кривых.

Входными параметрами алгоритма генерации цифровой подписи СТБ 34.101.45 являются входное сообщение произвольной длины и личный (секретный) ключ. Кроме того, на вход алгоритма также подаются параметры эллиптической кривой: уровень стойкости, коэффициенты кривой, порядок группы, базовая точка. По модулю порядка группы определяется уровень стойкости. Выходным параметром является цифровая подпись.

В качестве эллиптической кривой над конечным полем  $F_p$  используется следующее уравнение:  
$$y^2 = x^3 + a * x + b \pmod{p}$$
, где  $p$  - большое простое число.

Совокупность точек, удовлетворяющих уравнению, образует конечное поле. Над этим полем реализованы такие операции, как сложение и умножение на константу (вычисляется через многократное суммирование). С операцией умножения на константу связана основная задача по взлому таких кривых. Она носит название задачи дискретного логарифмирования. Её суть

заключается в подборе того коэффициента, на который умножали базовую точку. Именно этот коэффициент является закрытым ключом.

На рисунке 1 представлена структурная схема устройства для генерации цифровой подписи. В случае использования стандартной эллиптической кривой схему можно упростить, убрав генератор параметров эллиптической кривой и подавая на вход блоков электронной подписи соответствующие параметры.

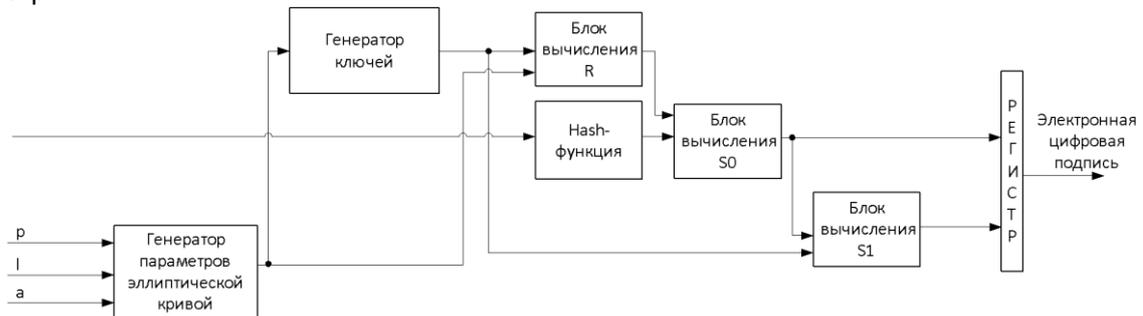


Рисунок 1 – Структурная схема устройства для выработки электронной цифровой подписи

$R = k * G$ , где  $G$  – базовая точка, являющаяся выходом генератора параметров эллиптической кривой. Блоки  $S_0$  и  $S_1$  являются частями итоговой подписи и выражаются следующими формулами:

$$S_0 = \langle belt - hash(OID(h)) \mid \langle R \rangle_{2l} \mid H \rangle_l,$$

$S_1 = \langle (k - H - (S_0 + 2^l) d \bmod q) \rangle_{2l}$ , где  $belt$ - $hash$  – функция хэширования,  $H$  – выход блока хэш-функции,  $OID$  – идентификатор функции(06092A7000020022651F51<sub>16</sub>),  $R$  – выход блока вычисления  $R$ ,  $d$  – личный ключ,  $q$  – порядок группы точек из поля  $F_p$ .

Уровень стойкости равен 128. Длина итоговой подписи равна 384 бита.

Одним из блоков алгоритма электронной цифровой подписи является блок вычисления функции хэширования. В качестве функции хэширования выбрана функция из стандарта СТБ 34.101.31-2011, в основе вычисления которой лежит алгоритм блочного шифрования из того же стандарта. При использовании данной хэш-функции в алгоритме электронной цифровой подписи на ее вход подается сообщение фиксированной длины. На выходе функции получается хэш-значение длиной 256 бит.

Приведенное устройство реализуется на базе FPGA семейства Virtex 7 фирмы Xilinx.

Аппаратную реализацию цифровой подписи можно использовать для проверки транзакций криптовалют, цифровых операций с валютами (платежами), подписи на электронные документы и любые другие файлы.

#### Список использованных источников:

1. Информационные технологии. Алгоритмы электронной цифровой подписи и транспорта ключа на основе эллиптических кривых : СТБ 34.101.45–2013. – Введ. 30.08.201. – Минск : Госстандарт, 2013. – 41 с.

## УСТРОЙСТВО ДЛЯ НЕПРЕРЫВНОГО НЕИНВАЗИВНОГО ИЗМЕНЕНИЯ АРТЕРИАЛЬНОГО ДАВЛЕНИЯ

Крылов Н.Д.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Азаров И.С. – д.т.н., доцент

Развитие микроэлектроники и элементной базы позволило выйти носимой электронике на новый уровень. Однако ряд актуальных задач остается нерешенным. Одной из таких задач является непрерывное неинвазивное измерение артериального давления. Решить данную задачу призваны алгоритмы машинного обучения.

На сегодняшний день измерение артериального давления вне реанимации производится методом Короткова, что доставляет дискомфорт пациенту и не позволяет производить непрерывное измерение. Решение данной проблемы возможно при измерении давления по косвенным признакам. Наиболее точным методом является измерение методом доплерографии, однако такой подход является технически сложным, дорогим и менее исследованным. Альтернативным подходом является анализ кривой фотоплетизмограммы (далее PPG). Существуют исследования[1-3],

показывающие эффективность такого подхода, однако исследования проводились с использованием данных, полученных в лабораторных условиях. В реальных условиях появляется множество артефактов из-за различных причин: двигательная активность, измерения во внешней среде. Избавиться от данных артефактов и повысить точность измерений возможно с применением технических и программных средств.

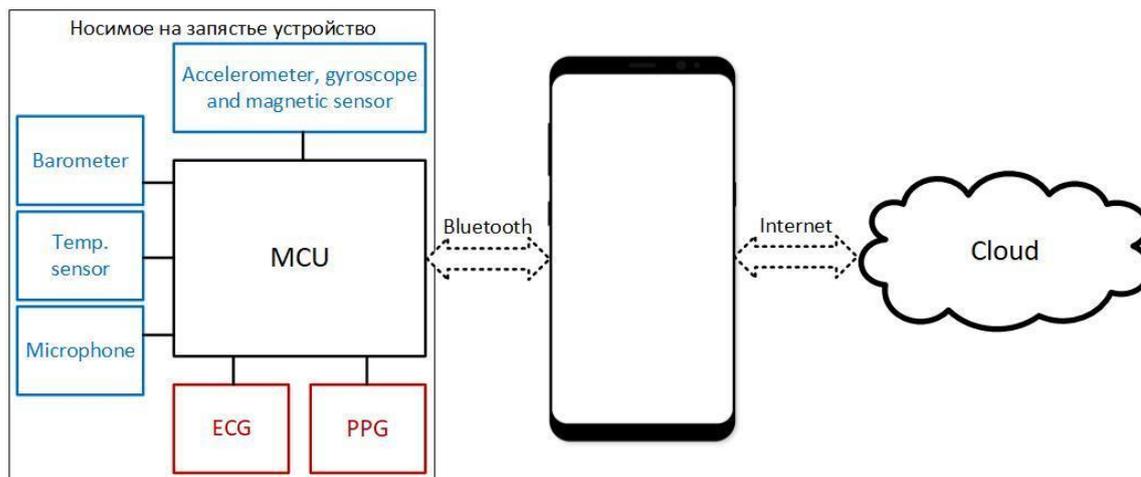


Рисунок 1 – Схема носимого устройства и системы взаимодействия

Применение современных комбинированных датчиков фотоплетизмографии, использующих свет с различной длиной волны, позволяет снимать с большей точностью кривую PPG и уровень сатурации кислородом крови.

Использование алгоритмов машинного обучения позволяет измерять давление с большей точностью. Для этого используется FFT кривой PPG и значение PWTT (Pulse Wave Transit Time – время распространения пульсовой волны)[3]. Использование данных о физической активности, данных о внешней среде и нескольких каналов PPG совместно с нейронной сетью позволяет дополнительно повысить точность измерения.

Вся система включает в себя носимое устройство, мобильное приложение и серверную часть. Серверная часть состоит из веб-сервиса, базы данных и модели сверточной и рекуррентной нейронной сети для определения артериального давления. Программное обеспечение носимого устройства включает модель машинного обучения для классификации активности.

**Список использованных источников:**

1. Xia Tan, Zhong Ji, Yadan Zhang, "Non-invasive continuous blood pressure measurement based on mean impact value method, BP neural network, and genetic algorithm", Technol Health Care, 26(Suppl 1): 87–101, 2018.
2. Peng Su, Xiao-Rong Ding, Yuan-Ting Zhang, Jing Liu, Fen Miao, Ni Zhao, Long-term Blood Pressure Prediction with Deep Recurrent Neural Networks, arXiv preprint:1705.04524v3 [cs.LG], 2018.
3. Xiaoman Xing, Mingshan Sun, "Optical blood pressure estimation with photoplethysmography and FFT-based neural networks", Biomed Opt Express, 7(8): 3007–3020, 2016.

## ПРИМЕНЕНИЕ ДЕТЕКТОРА ГОЛОСА В СЛУХОВОМ АППАРАТЕ

*Лайша А.И.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Вашкевич М.И. – к.т.н., доцент*

Слуховые аппараты — это электронные устройства, которые усиливают звуки выше порога слышимости пользователя с нарушениями слуха. Многие из пользователей слуховых аппаратов жалуются на дискомфорт восприятия их собственного голоса. При закрытии слухового канала слуховым аппаратом усиливаются низкочастотные компоненты голоса, а высокочастотные компоненты ослабляются ввиду звуковой проводимости костей. Это явление называется «эффект окклюзии» и является одной из критических проблем при ношении слуховых аппаратов.

Эффект окклюзии — причина, по которой пользователи жалуются на раздражающие или неестественные звуки. Проблема заключается в том, что из-за звуковой проводимости костей

звуковое давление низкочастотной составляющей речи (до 200 Гц), повышается на 15 дБ и более. Это приводит к снижению пользы слухового аппарата для пользователя и часто удерживает людей от ношения слуховых аппаратов. Также, если слуховой аппарат имеет алгоритм шумоподавления / классификации среды, собственный голос пользователя может быть ошибочно определен как желаемая речь.

В настоящее время используются чаще всего используется схема уменьшения эффекта окклюзии, показанная на рисунке 1,а. Есть несколько способов, которые помогают улучшить существующую схему:

1) Первый способ заключается в улучшении преобразователя. Принцип действия схемы на рисунок 1,б такой же как и в схемы на рисунке 1,а. Основное различие между схемами (а) и (б) состоит в том, что вторая имеет отличную структуру приемника и внутреннего микрофона. Преобразователь имеет только одну трубку, которая обычно используется в качестве звукового порта приемника и внутреннего микрофона. Эта структура улучшает фазовый отклик, поскольку путь между приемником и внутренним микрофоном меньше и проще.

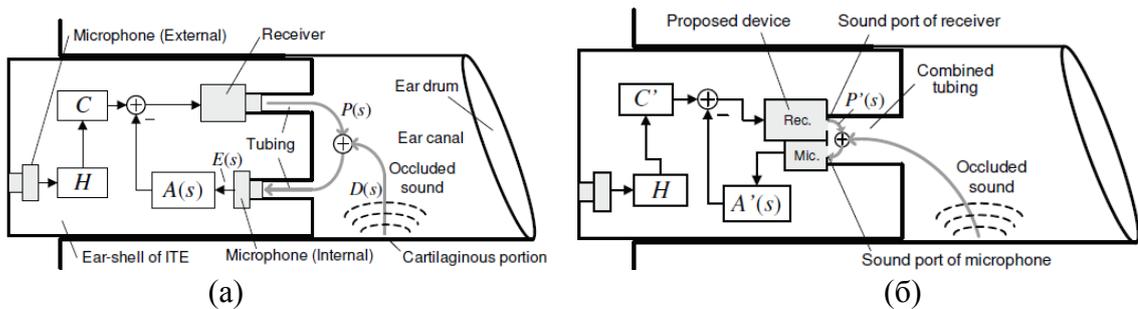


Рисунок 1 – (а) Обычная система (б) Схема системы уменьшения окклюзии

2) Активное подавление окклюзии (АПО) с помощью сквозного выравнивания для наушников. Данная система состоит из акустического интерфейса и электронного интерфейса, включая аналого-цифровой (АЦП) и цифро-аналогового преобразования. Акустический интерфейс реализован наушниками-вкладышами, которые содержат не только динамик, но и два микрофона. Один микрофон необходим для записи окружающего звука, и один микрофон обращен к барабанной перепонке, чтобы захватить в ухе звук. Обработка сигналов электронного интерфейса состоит из двух частей: АПО для ослабления усиленных костно-проводящих компонентов и сквозной сигнал для восстановления воздухопроводящих компонентов, которые были заблокированы. Цикл обратной связи реализован в виде не зависящего от времени контроллера  $K(z)$ , масштабированного с адаптивным коэффициентом. Его цель состоит в том, чтобы рассчитать сигнал подавления  $y(n)$ , который подавляет нежелательные участки сигнала в ушном канале.

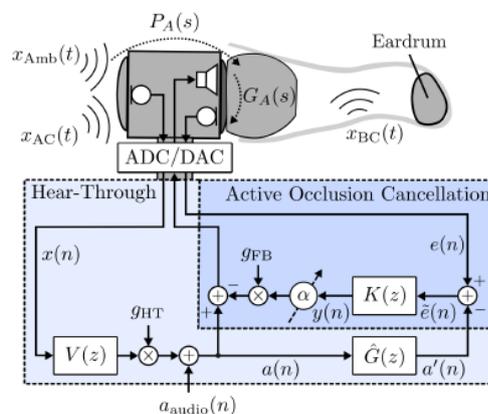


Рисунок 2 – Схема реализации активного подавления окклюзии

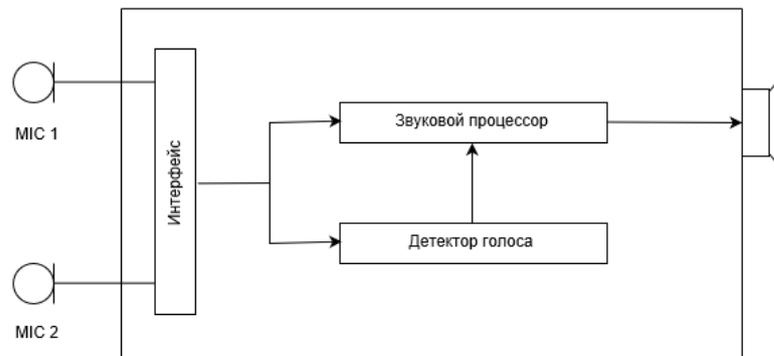


Рисунок 3 – Схема реализации слухового аппарата с встроенным детектором голоса

Негативный эффект окклюзии в момент восприятия собственного голоса может быть ослаблен за счет специализированной обработки звукового сигнала. Однако для определения момента активации специальных методов обработки нужен модуль детектора голоса. Пример такой системы показан на рисунке 3. Детектор голоса обнаруживает собственный голос пользователя и сигнализирует звуковому процессору о том, обнаружен ли конкретно голос пользователя. Как только собственный голос пользователя обнаружен, в звуковом процессоре запускаются соответствующие процедуры обработки. При обнаружении системой голоса пользователя, звуковой процессор может выполнять следующее: уменьшение усиления голоса пользователя, управление процессом предотвращения окклюзии, а также контроль процесса классификации среды. В конце звуковой процессор обрабатывает акустические сигналы, принятые первым и вторым микрофонами, и подает сигнал в приемник для формирования звукового сигнала для пользователя устройства.

**Список использованных источников:**

1. Patel I., Rao Y.S., Speech recognition using hidden markov model with MFCC-subband technique // International Conference on Recent Trends in Information, Telecommunication and Computing, 2010. – Pp. 168-172.
2. Rabiner L., Juang B.-H. Fundamentals of speech recognition – Prentice Hall. – 1993, 507 p.
3. Luo, Fa-Long, et al., "Recent Developments in Signal Processing for Digital Hearing Aids", IEEE Signal Processing Magazine, (Sep. 2006), 103-106.
4. X. Huang, A. Acero. Spoken language processing: a guide to theory, algorithm, and system development, Prentice Hall. – 2001, 1008 p
5. Хайкин С. Нейронные сети: полный курс, 2-е издание: Пер. с англ. – М.: Издательский дом «Вильямс», 2008. – 1104 с.
6. Stefan Liebich, Raphael Brandis, Johannes Fabry, Peter Jax, Peter Vary., "Active occlusion cancellation with hear-through equalization for headphones", Institute of communication systems RWTH Aachen University.
7. Masahiro Sunohara, Masatoshi Osawa, Takumi Hashiura and Makoto Tateno., Occlusion reduction system for hearing aids with an improved transducer and an associated algorithm, 3-20-41, Higashimotomachi, Kokubunji, Tokyo, 185-8533 Japan.

## МЕТОДЫ ОБРАБОТКИ СИГНАЛОВ ДЛЯ КОРРЕКЦИИ ДЕФЕКТОВ РЕЧИ С ПОМОЩЬЮ МЕХАНИЗМА ЗВУКОВОЙ ОБРАТНОЙ СВЯЗИ

*Лукашик Р.В.*

*Белорусский государственный университет информатики и радиоэлектроники  
Г. Минск, Республика Беларусь*

*Лихачев Д.С. – к.т.н., доцент*

Для коррекции дефектов речи обычно необходим специалист-логопед, однако присутствие врача-логопеда затратно как в экономическом плане, так и с точки зрения времени. Поэтому этот процесс по возможности стараются автоматизировать не в ущерб качеству исправления дефекта. Для этого необходим надежный механизм определения правильности выполнения упражнений. В данной работе в качестве такого механизма рекомендуется сбор статистики по основному тону.

Человеческая речь осуществляется посредством совместной работы мышц трех отделов периферического речевого аппарата: дыхательного, голосового и артикуляционного. Работу трех

названных отделов регулирует центральная нервная система. Качественные характеристики речи зависят от совместной синхронной работы многих зон левого и правого полушарий коры головного мозга. Однако особую роль в речевой деятельности играет речедвигательная (центр Брока) и речеслуховая (область Вернике) зоны, а также ассоциативный центр, который отвечает за построение фраз и предложений.

Из-за асинхронной работы речевых центров, нарушения голосового аппарата и др. причин могут возникать различные дефекты речи. Коррекция дефектов речи – сложный и трудоемкий процесс, который по возможности пытаются автоматизировать. Однако, автоматизации может поддаваться не любой речевой дефект. Теоретически, при помощи специальных компьютерных программ можно исправлять заикание, нарушение темпа речи, дислалию, дизартрию, ринолалию.

Большинство существующих компьютерных программ для устранения речевых дефектов используют метод акустической обратной связи: маскирование обратной акустической связи (MAF), задержка акустической обратной связи (DAF), изменение частоты обратной акустической связи (FAF), использование конъюгирующей акустической обратной связи (CAF).

Суть метода акустической обратной связи в том, что пациент, используя гарнитуру, слышит собственный голос только через наушники. Тем самым можно видоизменять речевой сигнал (замедлять, понижать тон и т.д) для воздействия на речеслуховую зону и корректировки речи.

Преимущества метода акустической обратной связи в том, что пациент может заниматься самостоятельно, не нужно присутствие врача-логопеда. Однако, данное преимущество является также и недостатком, так как отсутствует контроль «правильности» применения методов.

Эффективность существующих методов может быть повышена за счет уменьшения времени, затраченного пациентом на исправление дефекта, степени исправления дефекта, улучшения экономических показателей за счет уменьшения загрузки врача-логопеда и возможности самостоятельной дистанционной работы. Однако при таком подходе необходима возможность автоматизированного контроля «правильности» выполнения упражнений.

В качестве инструмента для контроля за «правильностью» выполнения упражнений пациентов был выбран сбор статистики по основному тону, который может показать, насколько пациент протягивает гласные и саму скорость его речи, а также соответствуют ли характеристики речи пациента параметрам, выставленным в программном обеспечении. В случае несоответствия пациенту даются рекомендации по выполнению упражнения.

Предполагается, что корректная с точки зрения минимизации числа ложных срабатываний реализация контроля «правильности» выполнения упражнений значительно повысит эффективность применения существующих методов коррекции речевых дефектов, использующих механизм звуковой обратной связи.

**Список использованных источников:**

1. Jayanthi Sasisekaran, Effects of Delayed Auditory Feedback on Speech Kinematics in Fluent Speakers, University of Minnesota, 2012, PMID:23409597.
2. Marilia, Effect of delayed auditory feedback on stuttering with and without central auditory processing disorders, Brasil, 2017, PMID:29236907.

## **БРЕЙН-СИСТЕМА**

*Наркевич Е. С.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Качинский М. В. – к.т.н., доцент*

Одним из методов развития интеллекта являются спортивные интеллектуальные игры. Однако для комфортной и эффективной игры в такие их разновидности как «Эрудит-квартет» и «Брейн-ринг» необходима система обработки запросов ответа игроками, которая сигнализирует о приоритете игрока. Системы такого рода носят название брейн-систем. Как правило, функционал существующих систем такого рода весьма ограничен, а количество турниров и игроков увеличивается из года в год. [1] В связи с этим появляется потребность в брейн-системах с расширенным функциональными возможностями.

Основной задачей брейн-систем является выбор игроков, которым дается право ответа на вопрос. Для взаимодействия игроков с системой используются кнопки. Нажатие кнопки сигнализирует о желании игрока ответить на вопрос. Для информирования игроков о нажатии кнопки и истечении данного на ответ периода времени используются звуковые сигналы. Брейн-система должна зафиксировать первое нажатие кнопки со стороны игроков после воспроизведенного системой звукового сигнала, который звучит после прочтения вопроса и нажатия соответствующей кнопки ведущим. Также в соответствии с правилами необходимо фиксирование нажатия кнопки до звукового

сигнала (фальстарт). Более подробный алгоритм работы и длительность засекаемых временных интервалов зависит от выбранного режима. Брейн-система поддерживает проведение двух игр – «Эрудит-квартет» или «Брейн-ринг».

Перечислим главные особенности игр «Эрудит-квартет» или «Брейн-ринг», которые учтены в брейн-системе. В игре «Эрудит-квартет» участвуют четыре команды, общее время для запросов ответа является постоянным вне зависимости от истории нажатия кнопок, однако игрок, который дал неверный ответ не может ответить повторно. Отличия «Брейн-ринга» от «Эрудит-квартет» является участие только двух команд и изменение данного на запрос ответа времени после неправильного ответа одной из команд. [1]

Брейн-система строится на основе микроконтроллера, к которому подключаются динамики для воспроизведения звука, кнопки для взаимодействия с игроками и Bluetooth-модуль для передачи информации от ведущего с помощью мобильного приложения. Отсчет времени, которое дается на ответ, также производит микроконтроллер. Для отображения позиции приоритетного игрока используются светодиоды, которые загораются, когда игроку дается право на ответ. [2]

Для взаимодействия ведущего с брейн-системой используется мобильное приложение для ОС Android. Для обеспечения обмена информацией между микроконтроллером и приложением используется Bluetooth-модуль, который для упрощения установки соединения работает как в режиме ведомого устройства, так и ведущего. [1] Мобильное приложение служит для выбора ведущим режима работы, то есть вида игры, фиксирования правильности ответа и последующей отправки этих данных на микроконтроллер, который использует эту информацию для выбора дальнейших действий, например, для отсчитывания временных интервалов и воспроизведения звуковых сигналов, в соответствии с ходом игры по регламентированным правилам. [2]

Одной из главных целей при создании брейн-системы было модифицировать уже существующие подобные системы, а также учесть недостатки этих систем, которые создают неудобства при работе с ними. Перечислим недостатки существующих брейн-систем:

- 1) Необходимость подключения к компьютеру, так как система взаимодействует со специальным приложением для Windows;
- 2) Отсутствие портативности (во многом это является следствием пункта 1);
- 3) Необходимость засечки времени вручную на дополнительных устройствах, а также нажатия кнопки сброса по истечении времени;
- 4) Отсутствие фиксирования результатов, что увеличивает вероятность появления ошибок при конечном фиксировании результатов.

Главным отличием данной системы от существующих является автоматическое фиксирование результатов, а также отсутствие необходимости отсчета времени вручную с помощью дополнительных средств, что существенно облегчает работу ведущего и сводит к минимуму ошибки при фиксировании результатов игры. Также система обладает большей портативностью, так как использует приложение для смартфонов. Данные модификации делают брейн-систему более удобной и простой в использовании.

**Список использованных источников:**

1. Белорусская лига интеллектуальных команд [Электронный ресурс]. – Режим доступа: <http://www.blik.by/>. – Дата доступа: 04.04.2019.
2. Arduino [Электронный ресурс]. – Режим доступа: <https://www.arduino.cc/>. – Дата доступа: 05.04.2019.

## **СЖАТИЕ ИЗОБРАЖЕНИЙ НА ОСНОВЕ НЕЙРОННЫХ СЕТЕЙ**

*Никитин Г.Ю.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Петровский Н.А. – к.т.н., доцент*

В докладе приводится способ сжатия изображения на основе свёрточных нейронных сетей и нелинейных функций активации.

Сжатие данных и, в частности, изображений - фундаментальная и хорошо изученная проблема в технике. Существует множество подходов, позволяющих решить её, в той или иной степени. Как правило, для компрессии изображений, применяют алгоритмы одной из трёх групп, в зависимости от представления коэффициентов: 1) частотной – дискретное косинусное преобразование (DCT); 2) частотно-временной – на основе вейвлет-анализа (Wavelet) и 3) временной, к примеру RLE. В последнее время для сжатия данных (и изображений) широко применяются и нейронные сети в различных их вариациях. Например, для сжатия изображений используют сеть Кохонена,

ассоциативную память, а также методы «бутылочного горлышка».

Рассмотрим структуру нейронной сети по принципу «бутылочного горлышка», предложенной в работе [1]. Структура слоёв нейронной сети представлена на рисунке 1, а формулы преобразования GDN и IGDN представлены на формулах 1 и 2.

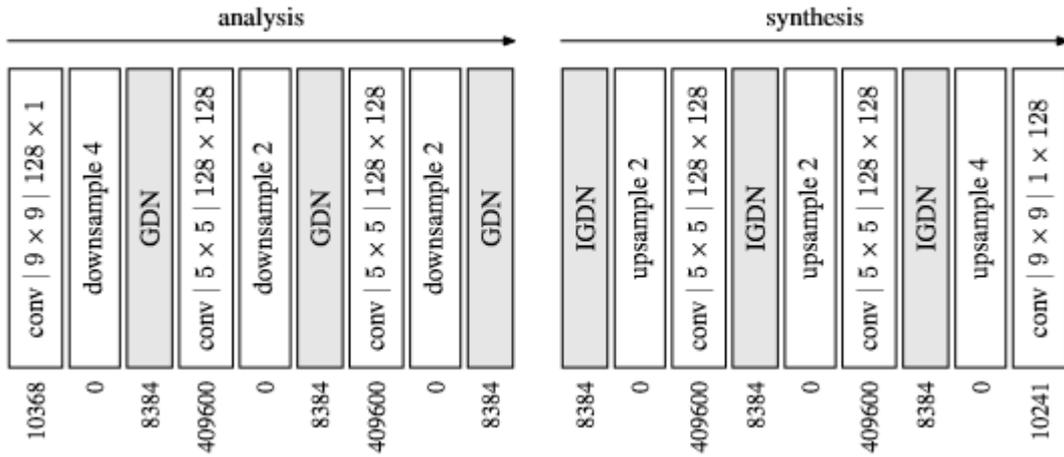


Рисунок 1. – Общая структура нейронной сети.

$$u_i^{(k+1)}(m, n) = \frac{\omega_i^k(m, n)}{\sqrt{\beta_{k,i} + \sum_j \gamma_{k,ij}(\omega_j^k)^2}} \quad (1)$$

Где  $\beta$  и  $\gamma$  – тренируемые параметры;  
 $\omega$  – набор входных каналов;  
 $u$  – набор выходных каналов.

$$\hat{\omega}_i^k(m, n) = \hat{u}_i^k(m, n) * \sqrt{\hat{\beta}_{k,i} + \sum_j \hat{\gamma}_{k,ij}(\hat{u}_j^k(m, n))^2} \quad (2)$$

Где  $\hat{\beta}$  и  $\hat{\gamma}$  - тренируемые параметры;  
 $\hat{\omega}$  - набор выходных каналов;  
 $\hat{u}$  - набор входных каналов.

Функция потерь подробно расписана и объяснена в [1].

Модель описана на языке Python при помощи библиотеки Tensorflow и обучена на различных выборках изображений. Ниже, на рисунках 2а и 2б, представлены изображение до сжатия и после сжатия соответственно. Как можно убедиться, с точки зрения перцептуальной приемлемости – различия найти достаточно сложно. В таблице 1 приведены некоторые независимые оценки для нескольких обработанных изображений.



Рисунок 2(а, б). – Изображения до и после сжатия.

Таблица 1. Независимые оценки сжатия.

Тестовое изображение	MSE	PSNR (дБ)	MS-SSIM [2]	MS-SSIM (дБ)	Actual bits per pixel	Information content in bpp
Lena	99.5555	28.15	0.9674	14.87	1.7239	1.7226
Barbara	165.3647	25.95	0.9692	15.12	1.7292	1.7279
Baboon	332.9701	22.91	0.9568	13.64	1.7158	1.7144

Как видно из таблицы 1, оценка «попиксельного» сравнения PSNR даёт достаточно высокие результаты, а в то же время оценка MS-SSIM говорит о более чем 90% совпадении изображений до и после сжатия. Это объясняется тем, что нейронная сеть не стремится сохранить значения «пиксель в пиксель», а стремится сохранить значения у группы пикселей, квантовать их.

В заключение хотелось бы отметить, что представленную модель с большой долей вероятности можно улучшить, предварительно обрабатывая изображения до сжатия. Взглянем на спектральную область изображений. На рисунке 3. спектр изображений до и после сжатия представлен в качестве гистограммы спектральной области изображений в логарифмической системе счисления (дБ).

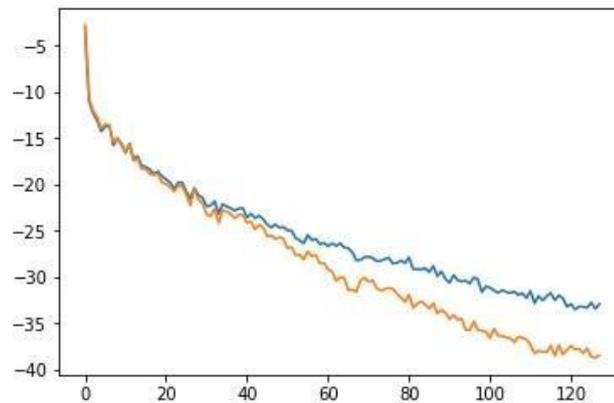


Рисунок 3. – Гистограмма спектра изображения до (верхний график) и после (нижний график) сжатия.

На рисунке 3 можно увидеть подавление высокочастотной части изображения, объясняющееся тем, что в этой частотной области изображения сосредоточены признаки, которые нейронная сеть считает наиболее маловажными для человеческого восприятия. Этот факт позволит оптимизировать входной набор данных и уменьшить нагрузку на нейронную сеть. В качестве такого преобразования можно использовать пакетное дискретное вейвлет преобразование (ПДВП) [3] с различными вариантами вейвлетов и разным построением дерева преобразований.

**Список использованных источников:**

1. End-to-end optimized image compression [Электронный ресурс] – Режим доступа : <https://arxiv.org/pdf/1611.01704.pdf>
2. Multi-scale structural similarity for image quality assessment [Электронный ресурс] – Режим доступа : <https://ece.uwaterloo.ca/~z70wang/publications/msssim.pdf>
3. A Wavelet Tour of Signal Processing [Электронный ресурс] – Режим доступа : <https://www.di.ens.fr/~mallat/papiers/WaveletTourChap1-2-3.pdf>

## МЕТОД КОМПЕНСАЦИИ УРОВНЯ ГРОМКОСТИ ПРИ ПЕРЕНОСЕ ЧАСТОТ В РЕЧЕВОМ СИГНАЛЕ

Порхун М.И.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Вашкевич М.И. – к.т.н., доцент

Рассмотрен метод компенсации уровня громкости при переносе частот в речевом сигнале. Метод сводится к обработке сигнала таким образом, чтобы при переносе из высокочастотной части спектра речевого сигнала в низкочастотную область компенсировать уровень громкости переносимого частотного диапазона. Метод заключается в вычислении компенсирующих коэффициентов для переносимой частотной полосы сигнала. В работе приведён пример компенсации уровня громкости сигнала при переносе адаптивной высокочастотной полосы спектра речевого сигнала в фиксированную низкочастотную область спектра.

Большинство людей, страдающих от потери слуха, сохраняют способность к восприятию низкочастотных звуков. При этом, как правило, они теряют способность слышать высокочастотные звуки. Это приводит к тому, что у слабослышащего человека затруднено восприятие многих согласных звуков, энергия которых сосредоточена в высокочастотной области спектра. В зависимости от степени потери слуха, простое частотно-зависимое усиление компонентов сигнала часто оказывается неэффективным при попытке улучшить разборчивость высокочастотных компонентов звука. Так происходит поскольку часть волосковых клеток в улитке уха, отвечающих за восприятие высокочастотных звуков, мертвы [1]. Решением данной проблемы может служить перенос высокочастотных компонентов звука в низкочастотную область, в которой тугоухий человек сохраняет возможность восприятия.

Слуховая система человека нелинейно воспринимает звук – она имеет разную чувствительность к звукам с разными частотами [2]. Поэтому перенос полосы частот из высокочастотной области спектра в низкочастотную область приведет к неизбежным искажениям уровня сигнала. Чтобы уменьшить искажения предлагается использовать концепцию равной громкости [3], которая отражает тот факт, что слуховая система человека имеет разную чувствительность к разным частотам [4].

Кривые равной громкости (рисунок 1) показывают, какое звуковое давление необходимо создать, чтобы различные частоты воспринимались человеком одинаковыми по громкости [5].

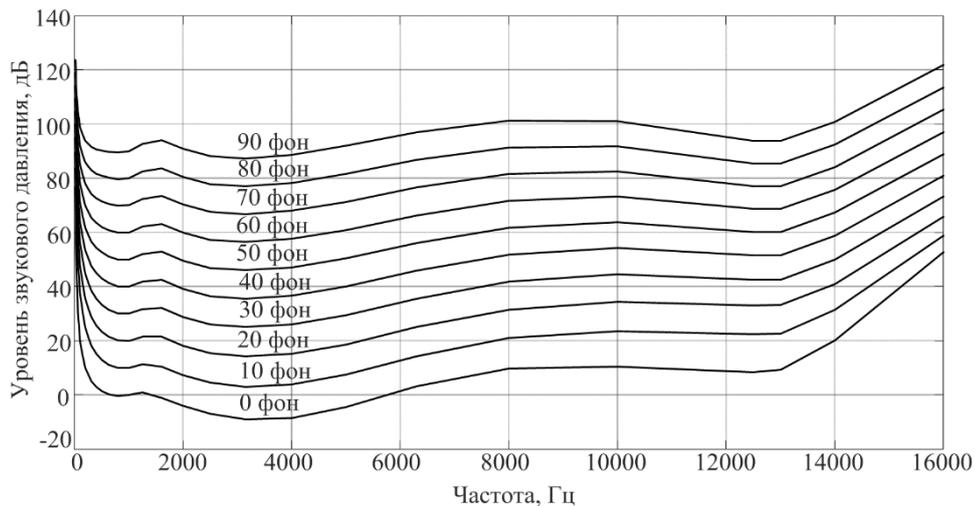


Рисунок 1 – Кривые равной громкости

Для переноса частот выбираются две полосы: исходная (откуда происходит перенос) и целевая (куда смещается исходная полоса). В работе выбрана целевая полоса 750 – 1750 Гц, а исходная полоса выбирается адаптивно и имеет ширину полосы 1 кГц. Обработка сигнала осуществляется фрейм за фреймом.

В предлагаемом методе кривые равной громкости используются для расчета разницы в уровне звукового давления (УЗД) между исходной и целевой полосами спектра, а также для вычисления весовых коэффициентов [3].

Для определения весовых коэффициентов выполняются следующие шаги:

- 1) вычислить уровень сигнала для  $i$ -й частоты исходной полосы;

- 2) определить, какой кривой равной громкости соответствует вычисленный на первом шаге уровень сигнала (то есть, определить УЗД);
- 3) по найденному УЗД определить УЗД для  $i$ -й частоты целевой полосы спектра;
- 4) найти разницу между вычисленными уровнями УЗД на предыдущих шагах;
- 5) по формуле 1 вычислить весовой коэффициент для  $i$ -й частоты целевой полосы.

Для компенсации уровня сигнала при переносе частот  $i$ -й частотный компонент целевой полосы умножается на полученный весовой коэффициент. Аналогичным образом вычисляются коэффициенты для остальных компонентов исходной полосы.

$$G(i) = 10^{g/20} \quad (1)$$

где  $g$  – разница между УЗД, найденная на шаге 4.

Проверка предлагаемого метода компенсации уровня громкости производилась на речевом сигнале. Спектрограмма исходного речевого сигнала приведена на рисунке 2. Спектрограммы обработанного речевого сигнала с и без учёта применения метода компенсации уровня громкости представлены на рисунках 3 и 4, соответственно.

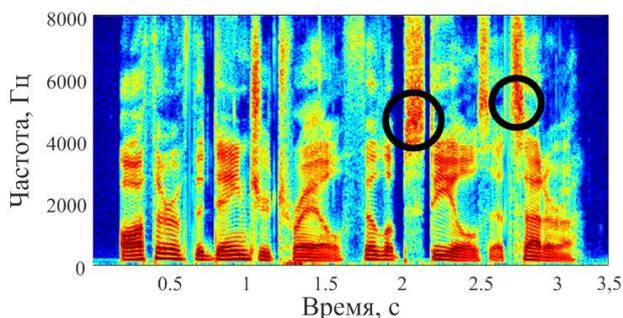


Рисунок 2 – Спектрограмма исходного речевого сигнала

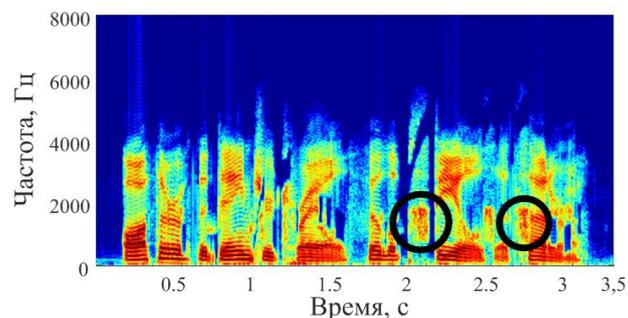


Рисунок 3 – Спектрограмма обработанного речевого сигнала (без компенсации уровня громкости)

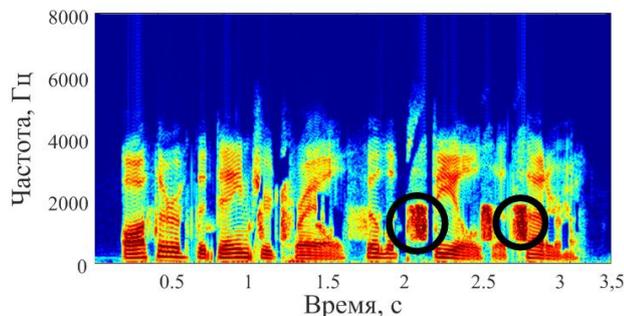


Рисунок 4 – Спектрограмма обработанного речевого сигнала (с применением метода компенсации уровня громкости)

В качестве примера в работе используется входной речевой сигнал с частотой дискретизации 16 кГц. Он разделяется на фреймы длительностью 16 мс и умножаются на окно Хемминга. Переносятся только фреймы, содержащие в себе согласные звуки. Полоса 750 – 1750 Гц выбрана в качестве целевой. Исходная полоса также имеет ширину 1 кГц, а её центральная частота выбиралась адаптивно в зависимости от того в какой области сосредоточена основная энергия сигнала. Для имитации потери слуха, обработанный сигнал пропущен через фильтр нижних частот с частотой среза 3000 Гц.

В работе предложен метод компенсации уровня громкости при переносе высокочастотных компонентов речевого сигнала в низкочастотную область. Особенностью метода является применение в нём концепции равной громкости, использование которой позволяет выполнять перенос частот в речевом сигнале, согласованный с работой слуховой системы человека. Работоспособность метода подтверждена результатами MATLAB-моделирования.

**Список использованных источников:**

1. Simpson A. Frequency-lowering devices for managing high-frequency hearing loss: A review / A. Simpson // Trends in Amplification. – 2009. – Vol. 13, № 2. – P. 87 – 106.
2. Королёва И. В. Введение в аудиологию и слухопротезирование / И. В. Королёва. – СПб.: КАРО, 2012. – 400 с.
3. Liu Y., Chang R. Y., Tsao Y., Chang Y. A new frequency lowering technique for Mandarin-speaking hearing aid users / Y. Liu, R. Y. Chang, Y. Tsao, Y. Chang // IEEE Global Conference on Signal and Information Processing (GlobalSIP), Orlando, FL. – 2015. P. 722 – 726.

4. Fletcher H., Munson W. Loudness, its definition, measurement and calculation / H. Fletcher, W. Munson // Journal of the Acoustic Society of America. – 1933. – Vol. 5. – P. 82 – 108.

5. ISO 226:2003. Acoustics – Normal Equal-Loudness Level Contours / International Organization for Standardization, – Geneva, Switzerland. – 2003.

## НЕЙРОИНТЕРФЕЙС НА ОСНОВЕ МИКРОКОНТРОЛЛЕРА ДЛЯ КОНТРОЛЯ ОБЪЕКТОВ В СРЕДЕ СИМУЛЯЦИИ.

Щуцкий Н.А., Розум Д.В.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Вашкевич М.И. – к.т.н., доцент

Современный интерес к искусственному интеллекту и соответствующее развитие необходимых технологий привело к возможности в реальном времени решать задачи машинного обучения и анализа данных с высокой точностью, что стало инициатором появления технологий вроде самоуправляемых автомобилей и адаптивного шумоподавления. В свою очередь мы предлагаем попытаться применить новшества глубокого обучения в области нейроинтерфейсов для контроля среды симуляции малого числа степеней свободы.

Основная идея заключается в создании программного обеспечения, позволяющего классифицировать некоторый набор состояний и реакций мозга на основе ЭЭГ и преобразовать их таким образом, чтобы организовать эффективный контроль некоторой средой. В рамках данной научной работы будет использован метод биполярного отведения сигнала, позволяющий получить только ограниченный спектр возможных состояний. На основе полученных измерений будет произведена экстракция необходимых участков сигнала и последующая классификация отдельных команд, которые могут быть использованы в определённой среде.

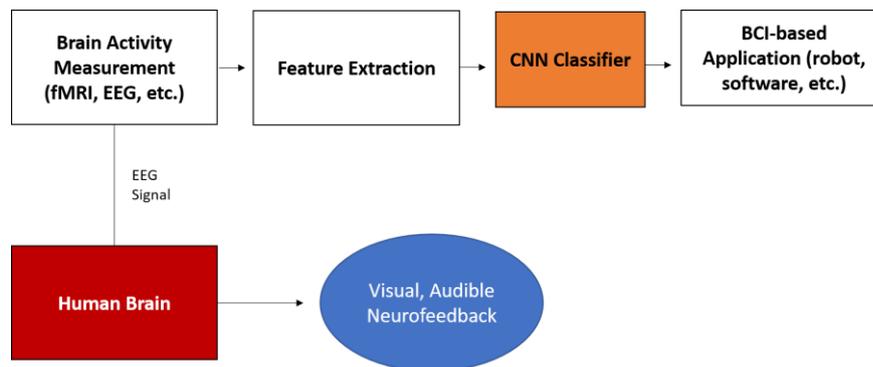


Рисунок 1 – Схема работы нейроинтерфейса

Использование ЭЭГ позволяет избежать инвазивных методов внедрения электродов, однако при этом страдает от избыточной зашумленности сигнала ввиду потери энергии электрического поля, проходящего через череп, а также движений глаз и челюсти. Соответственно одной из важнейших задач при использовании ЭЭГ является шумоподавление, которое также может быть произведено при помощи средств глубокого обучения в зависимости от требуемой скорости и точности

Первостепенно будет проведён спектральный анализ сигнала и предварительная разметка, позволяющей выделить необходимые участки для дальнейшего классифицирования, данная часть процессов будет реализована на микропроцессоре для дальнейшей передачи на ПК либо его аналог с достаточной вычислительной мощностью, чтобы провести классификацию в реальном времени. Классификация будет произведена методами анализа данных, выбор которых должен быть сбалансирован в зависимости от выбранной прикладной задачи, балансируя между быстродействием и точностью.

Выбор среды симуляции зависит от доступного количества каналов ЭЭГ, что позволяют управлять большим числом степеней свободы в нелинейной зависимости от их количества. В случае с одноканальным ЭЭГ проверенными результатами являются уровень релаксации и концентрации, что позволит управлять системой со степенью свободы 1, на которых и будет сделан акцент в нашей работе.

Преимуществом нейроинтерфейсов по сравнению с другими методами обмена информацией с вычислительными устройствами является отсутствие промежуточных звеньев передачи информации между человеком и информационным устройством, что увеличивает в значительной степени скорость обмена и спектр возможных адаптаций инструментов управления вычислительным устройством.

Потенциальными задачами, в которых могут быть применены нейроинтерфейсы, являются:

- 1) Восполнение утраченных пациентом высших функций нервной деятельности.
- 2) Расширение возможностей контроля над различными устройствами.
- 3) Прямой обмен информацией между несколькими индивидуумами.

**Список использованных источников:**

1. Alberto J. Molina-Cantero, Jaime Guerrero-Cubero, Isabel M. Gómez-González, Manuel Merino-Monge and Juan I. Silva-Silva, "Characterizing Computer Access Using a One-Channel EEG Wireless Sensor", *Sensors* 2017, 17(7), 1525
2. Vincent, Pascal, Larochelle, Hugo, Lajoie, Isabelle, Bengio, Yoshua, and Manzagol, Pierre-Antoine. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *The Journal of Machine Learning Research*, 11:3371–3408, 2010.

## МЕТОД ПОВЫШЕНИЯ РАЗБОРЧИВОСТИ РЕЧИ НА ОСНОВЕ ЧАСТИЧНОГО ПЕРЕНОСА ЧАСТОТНОГО ДИАПАЗОНА

*Санько Н.С.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Вашкевич М.И. – к.т.н., доцент*

В мире существует большое количество слабослышащих людей, которые не довольны качеством работы используемых слуховых аппаратов. Одной из актуальных задач является повышение разборчивости речи для слабослышащих людей. Существует ряд методов повышения разборчивости речи, которые, однако, имеют такие недостатки как: неестественный машинный голос и/или потеря полезной информации высокочастотной области при её смещении в область низких частот. В работе предложен метод повышения разборчивости речи за счет частичного переноса частотного диапазона [1].

Общая цель применяемых методов повышения разборчивости речи – минимизировать потери полезной информации высокочастотной области, улучшить разборчивость речи для комфортного восприятия её субъектом без потери смысловой составляющей слов и предложений.

В данной работе предлагается следующий метод повышения разборчивости речи на основе переноса частотного диапазона, который состоит в выполнении следующих этапов:

- 1) фильтрация сигнала путём пропускания через фильтры низких и высоких частот. Пропускание низкочастотного фильтра осуществляется до 2 кГц, а для высокочастотного – от 2 до 8 кГц;
- 2) смещение высокочастотного спектра сигнала в низкочастотную область после прохода через фильтр высоких частот;
- 3) суммирование сигнала после пропускания через фильтр низких частот с сигналом со смещённым высокочастотным спектром;
- 4) получение результирующего сжатого сигнала со смещённым высокочастотным спектром.

Структурная схема предлагаемой системы повышения разборчивости речи представлена на рисунке 1.

В качестве фильтрации сигнала были выбраны КИХ-фильтры низких и высоких частот 120 порядка. При поступлении входного сигнала, он фильтруется в соответствии с указанными ограничениями пропускной способности фильтра. Далее выполняется перенос спектра отфильтрованного высокочастотного сигнала путем модуляции. Поступающий сигнал  $S(t)$ , умножается на комплексную экспоненту по формуле 1:

$$S'(t) = S(t) \times \exp(-j \times \omega \times t), \quad (1)$$

где  $\omega$  - нормированная частота,  $t$  - время.

Таким образом данное преобразование осуществляет смещение спектра сигнала на частоту  $\omega$ . Спектр смещается влево по средствам отрицательного значения нормированной частоты. Значение частоты, на которую смещается сигнал, является настраиваемой [2].

Заключительным этапом является суммирование сигнала, прошедшего через низкочастотный фильтр, и преобразованного сигнала, отфильтрованного высокочастотным фильтром. В результате суммирования получается сжатый сигнал с перенесённым высокочастотным спектром в область

низких частот [3].

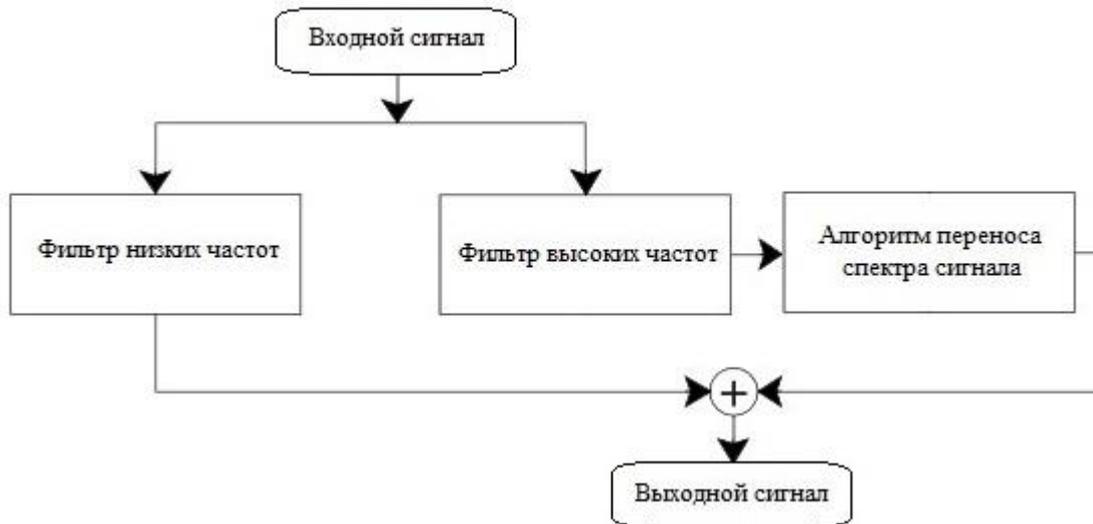


Рисунок 1 - Структура системы повышения разборчивости речи

График оригинального и выходного сигнала представлен на рисунке 2.

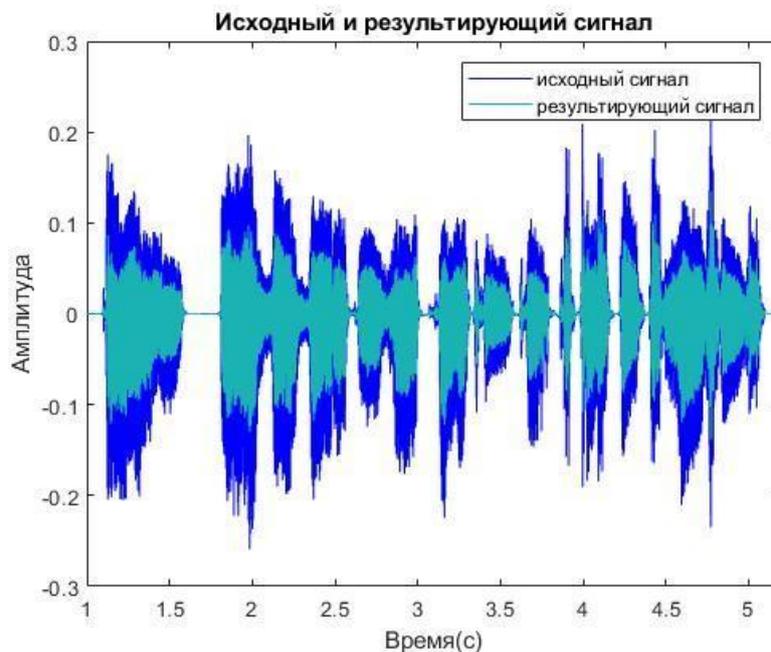


Рисунок 2 – Исходный сигнал и результирующий сигнал после преобразования

На рисунке 2 видно, что амплитуда выходного сигнала сжата по сравнению с исходным. Также на рисунке 3 представлены спектрограммы оригинального сигнала и преобразованного сигнала.

Как видно из спектрограмм на рисунке 3, мощность обработанного сигнала после преобразования сосредоточена в нижней полосе частот посредством переноса высокочастотного спектра сигнала. Таким образом наблюдается минимальная потеря полезной информации сигнала, а высокочастотная составляющая спектра смещается в область частот допустимой слышимости для человека с тугоухостью.

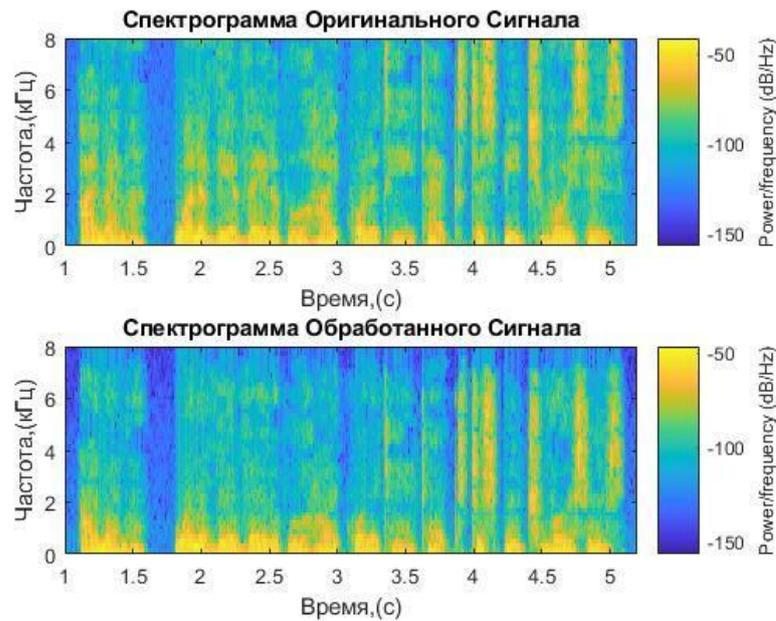


Рисунок 3 – Спектрограммы оригинального и обработанного сигналов

**Список использованных источников:**

1. Simpson A. Frequency-Lowering Devices for Managing High-Frequency Hearing. *Ear Hearing*, 2009, vol. 13, no. 2, pp. 87-106.
2. Paarmann L. Nonlinear Spectrum Compression for the Hearing Impaired via a Frequency-Domain Processing Algorithm. *28<sup>th</sup> IEEE EMBS Annual International Conference*, 2006, vol.10, pp. 239-243.
3. Wang Q., Zhao L., Zou C. Piecewise-Linear Frequency Shifting Algorithm for Frequency Resolution Enhancement in Digital Hearing Aids. *MDPI Conference*, 2017, vol.1, pp. 1-13.

## ВЫЯВЛЕНИЕ ПАТОЛОГИЙ В ГОЛОСЕ НА ОСНОВЕ РАЗЛОЖЕНИЯ СИГНАЛА НА ЭМПИРИЧЕСКИЕ МОДЫ

Семенюк Н.А.

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Вашкевич М.И. – к.т.н., доцент*

В работе рассмотрена задача анализа голосового сигнала на основе декомпозиции на эмпирические моды с целью выявления в нем признаков патологии. В качестве характерных признаков, которые могут быть использованы для детектирования патологии, были выявлены – число мод, на которые раскладывается сигнал и – отношение сигнал/шум, рассчитанное на основе того, что часть полученных мод относились в шумовой составляющей голоса.

Декомпозиция сигнала на эмпирические моды (ДЭМ) – это способ анализа сигналов различной природы, который становится все более распространенным в экспериментальных исследованиях [1-2]. Уже первые попытки его применения позволили установить, что с помощью декомпозиции на эмпирические моды можно обнаружить модуляцию с близкими значениями частот моделирующего и моделируемого процессов, уменьшать ширину спектральных линий и т.п. ДЭМ позволяет учитывать локальные особенности сигнала (экстремумы и нули сигнала), а также шумовые, периодические и трендовые его компоненты. Было показано, что ДЭМ подходит для кодирования речевого сигнала, определения мгновенных параметров сигнала, редактирования акустического шума [3] и проч.

В процессе анализа данных полученных экспериментальным путем одним из этапов предобработки часто является приведение сигнала к нулевому среднему уровню, что позволяет избавиться от проблем, связанных с некорректным определением мгновенной частоты. Однако данная процедура не всегда может быть корректно выполнена для нестационарных процессов, поскольку их среднее значение может меняться и принимать отличные от нуля величины на локальных участках. Для корректного определения мгновенной частоты компонент, на которые раскладывается сигнал, в рамках метода ДЭМ предлагается обеспечить выполнение следующих условий:

- 1) локальное среднее значение огибающих, определяемых локальными максимумами функции  $c_j$  и локальными минимумами этой функции должно быть равно нулю;
- 2) число пересечений нулевого уровня функцией и число локальных максимумов (или минимумов) должно совпадать или отличаться не более чем на единицу.

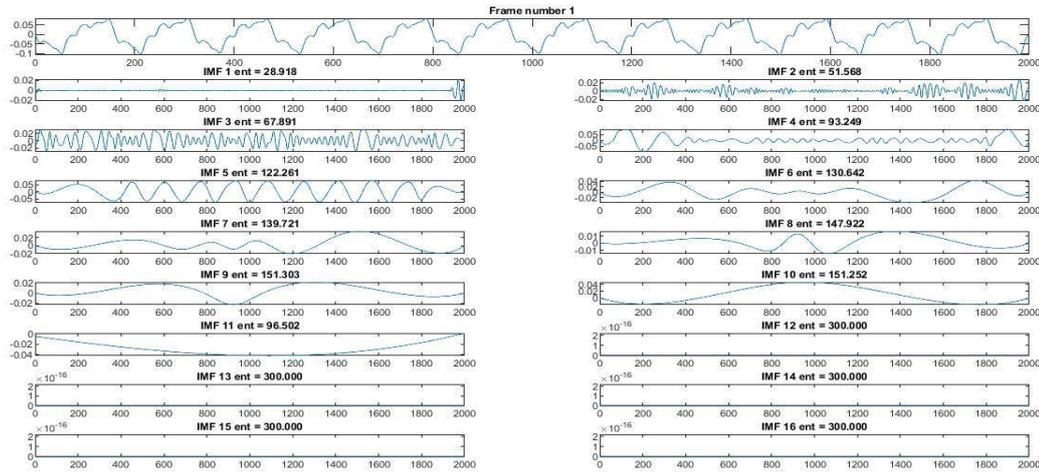


Рисунок 1 – Пример декомпозиции сгенерированного сигнала на эмпирические моды

Сам процесс отсеивания состоит из нескольких этапов (Рисунок 1):

- 1) Определение локальных экстремумов (максимумов и минимумов) сигнала  $h_{j,i-1}(t)$ ;
- 2) Определение верхней  $U_{j,i-1}(t)$  и нижней  $L_{j,i-1}(t)$  огибающих сигнала с помощью кубического сплайна-интерполяции по найденным локальным экстремумам  $h_{j,i-1}(t)$ ;
- 3) Вычисление среднего значения огибающих сигнала:

$$\mu_{j,i-1}(t) = \frac{U_{j,i-1}(t) + L_{j,i-1}(t)}{2}$$

- 4) Вычисление остатка от сигнала:

$$h_{j,i}(t) = h_{j,i-1}(t) - \mu_{j,i-1}(t), \quad i = i + 1$$

- 5) Вычисление критерия остановки:

$$SD(i) = \sum \frac{[h_{j,i-1}(t) - h_{j,i}(t)]^2}{h_{j,i-1}^2(t)}$$

- 6) Повторение алгоритма до тех пор, пока  $SD \geq 0,25$ . Когда SD станет меньше, чем 0,25,  $h_{j,i-1}(t)$  станет функцией внутренней моды.

Недостатки метода ДЭМ:

- 1) часто происходит смешивание нескольких мод, которые интерпретируются как одна;
- 2) получение ЭМ, состоящих из участков сигнала несоизмеримых масштабов или участков соизмеримых масштабов, но находящихся в различных областях ЭМ.

Существует такие заболевания гортани, первое проявление которых отражается на изменениях в голосе. Раскладывая голос на компоненты, было обнаружено, что показатель сигнал/шум (SNR показатель) и число мод, получаемых в процессе разложения, для здоровых голосов и голосов с патологиями отличаются.

Применив ДЭМ для анализа речевых сигналов, соответствующих устойчивой гласной «а» реальных данных (20 здоровых голосов и 20 голосов с патологиями), получили следующие диаграммы размаха (рис. 2 и рис. 3). Запись голосов производилась в РНПЦ Отоларингологии (г. Минск) под контролем врача фониатра.

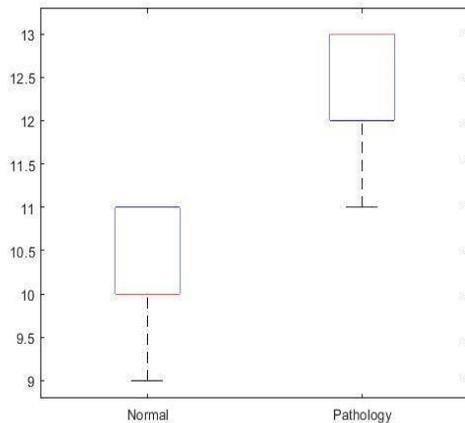


Рисунок 2 – Число мод для здорового голоса и голоса с патологией

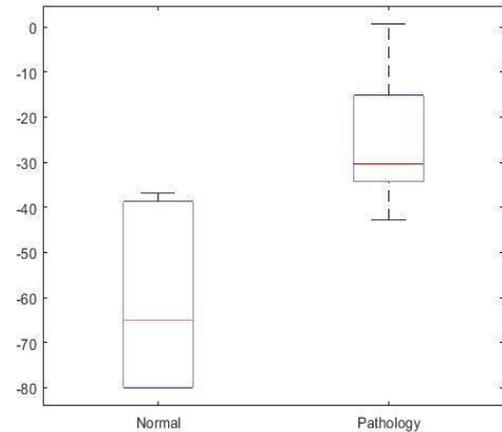


Рисунок 3 – Отношение сигнал/шум (SNR) для здорового голоса и голоса с патологией

По графикам видно, что значения этих двух показателей явно отличаются, что говорит о возможности использования этого метода в области патологий ЛОР-органов, в частности заболеваний гортани, и необходимости дальнейшего развития этого метода.

**Список использованных источников:**

1. A.-O. Boudraa, J.-Ch. Cexus, "EMD-Based Signal Filtering", IEEE Transactions on instrumentation and measurement, vol. 56, no. 6, pp. 2196-2202, 2007.
2. J. V. Sloten, P. Verdonck, M. Nyssen, J. Haueisen (Eds.): ECIFMBE 2008, IFMBE Proceedings 22, pp. 252–255, 2008.
3. И.А. Воронцовский, Анализ сигналов методом декомпозиции на эмпирические моды и его применение в обработке речевых сигналов / Доклады БГУИР. – 2012. – №7(69). – С. 18 – 24.
4. Juan Rafael Orozco-Arroyave, Elkyn Alexander Belalcazar-Bolaños, Julián David Arias-Londoño, Jesús Francisco Vargas-Bonilla, Sabine Skodda, Jan Ruzs, Khaled Daqrouq, Florian Hönig, and Elmar Nöth, Characterization Methods for the Detection of Multiple Voice Disorders: Neurological, Functional, and Laryngeal Diseases, IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS, VOL. 19, NO. 6, 2015
5. Применение комплементарной множественной декомпозиции на эмпирические моды для анализа речевых сигналов / А. К. Алимуратов, Ю. С. Квитка // Измерение. Мониторинг. Управление. Контроль. – 2014. – № 4 (10). – С. 69–75.

## ГЛУБОКАЯ НЕЙРОННАЯ СЕТЬ ДЛЯ ДЕТЕКТИРОВАНИЯ ЛИЦ В УСЛОВИЯХ ОГРАНИЧЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ РЕСУРСОВ

Суша А.В.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Вашкевич М.И. – к.т.н., доцент

В работе приводятся приёмы проектирования глубоких нейронных сетей, показывающих высокую производительность на встраиваемых и мобильных платформах, на примере решения задачи детектирования лиц. Детектор лиц основан на концепции You Only Look Once, которая заключается в том, что изображение анализируется только один раз при прямом проходе нейронной сети. Приведенные методики проектирования позволили существенно повысить скорость детектирования без серьёзной потери в точности и полноте.

Применение глубоких нейронных сетей для детектирования лиц на изображении позволяет достичь высокого уровня точности и инвариантности к множеству факторов, затрудняющих детектирование: разный масштаб лиц, частичное перекрытие лиц другими объектами, поза людей в момент захвата изображения, выражения лиц, засветка и др. – всё это существенно понижает точность детектирования классическими методами [1]. Несмотря на то, что в настоящее время детекторы на глубоких нейронных сетях показывают высокую скорость детектирования на персональных компьютерах, оснащёнными мощными ЦПУ или ГПУ, всё ещё существуют определенные сложности в достижении высокой производительности в масштабе реального времени

на мобильных и встраиваемых устройствах. Это в первую очередь связано с тем, что в них устанавливается меньшее число вычислителей, при этом обладающими меньшей мощностью.

Последняя работа от авторов YOLO (You Only Look Once) [2] привнесла небольшие улучшения этого подхода к детектированию объектов на изображении. Авторы реализовали многомасштабное детектирование, а также предложили несколько новых архитектур глубоких нейронных сетей, используемых в детекторе. Эти архитектуры (Darknet-53 и Darknet-19) требуют меньших вычислительных затрат, чем нейронные сети, используемые ранее, что позволило авторам достигнуть высокой скорости детектирования с лучшим результатом 22 мс (или 45 FPS) на изображении, размером 320x320 пикселей, и 29 мс (или 34 FPS) на изображении 416x416 пикселей. Измерения проводились на ПК, оснащённом видеокартой Nvidia GeForce GTX Titan X [3].

За последние несколько десятилетий развития глубоких нейронных сетей были предложены множества приёмов организации архитектуры сетей. Анализ научных статей позволил выделить следующие методики повышения производительности глубоких нейронных сетей посредством уменьшения числа операций:

- первая операция свёртки должен иметь шаг окна равный 2, что в 4 раза уменьшает число операций, проводимых в этом слое;

- операции свёртки должны быть факторизованы на несколько меньших последовательных операций. К примеру, свёртка с окном 5x5 выполняется двумя свёртками 3x3 или 5x1 и 1x5, что уменьшает число параметров сети, а также число операций MAC;

- традиционная операция свёртки может быть заменены на последовательные операции поканальной и точечной свёртки, что уменьшает число параметров с  $n = c \times h \times w \times k$  на  $m = c \times h \times w + c \times k$ , где  $c$  – число входных каналов,  $k$  – число выходных каналов,  $h \times w$  – размер фильтра свёртки;

- применение концепции bottleneck (бутылочное горлышко), которая заключается в резком понижении числа каналов операцией точечной свёртки, а затем применением операции обычной свёртки.

Основываясь на приёмах, описанных выше, была спроектирована глубокая нейронная сеть, реализующая концепцию YOLO. Обучение проводилась на базе WIDER [4]. Для тестирования детектора использовался набор Fddb [5]. Время прямого прохода на видеокарте Nvidia GeForce GTX 1070 [6] (сопоставима с Titan X) составило 8-9 мс (110-125 FPS).

**Список использованных источников:**

1. Dwiyanoro, A. P. J. Performance Showdown of Publicly Available Face Detection Model [Electronic resource] / A. P. J. Dwiyanoro // Medium. – Mode of access: <https://medium.com/nodeflux/performance-showdown-of-publicly-available-face-detection-model-7c725747094a>. – Date of access: 24.03.2019.
2. Redmon, J. YOLOv3: An Incremental Improvement / J. Redmon, A. Farhadi // arXiv preprint, arXiv: 1804.02767. – 2018. – 6 p.
3. GeForce GTX TITAN X [Electronic resource] : Specifications / NVIDIA Corporation. – Mode of access: <https://www.geforce.com/hardware/desktop-gpus/geforce-gtx-titan-x/specifications>. – Date of access: 24.03.2019.
4. Yang, S. WIDER FACE: A Face Detection Benchmark / S. Yang, P. Luo, C. C. Loy, X. Tang // IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2016. – P. 5525- 5533.
5. Jain, V. Fddb: A Benchmark for Face Detection in Unconstrained Settings / V. Jain, E. Learned-Miller // Technical Report UM-CS-2010-009, Dept. of Computer Science, University of Massachusetts. – 2010. – 11 p.
6. GeForce GTX 1070 [Electronic resource] : Specifications / NVIDIA Corporation. – Mode of access: <https://www.geforce.com/hardware/desktop-gpus/geforce-gtx-1070/specifications>. – Date of access: 24.03.2019.

*Научное издание*

# **КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ**

**55-я юбилейная научная конференция  
аспирантов, магистрантов и студентов**

Сборник тезисов докладов

(Минск, 22-26 апреля 2019 года)

В авторской редакции

Ответственный за выпуск *Е.В. Калабухов*

Компьютерная верстка: *А.С. Панкратьев, К.А. Голубев, Н.А. Пилинко*