

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

КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ

**МАТЕРИАЛЫ 54-Й НАУЧНОЙ КОНФЕРЕНЦИИ АСПИРАНТОВ,
МАГИСТРАНТОВ И СТУДЕНТОВ**

(Минск, 23-27 апреля 2018 года)

УДК 004.9+519.681/.688

Редакционная коллегия:

В. А. Прытков (главный редактор),
Е. В. Калабухов, Е. А. Сасин, Н. В. Лапицкая, С. Е. Карпович,
Г. Ф. Смирнова, Г. И. Малыхина, Н. А. Волорова, Д. И. Самаль,
М. В. Качинский, В. В. Цегельник

К63 Компьютерные системы и сети: материалы 54-й науч-
ной конференции аспирантов, магистрантов и студентов
(Минск, 23-27 апреля 2018 г.). – Минск: БГУИР, 2018. – 257 с.

В сборник включены доклады, которые были представлены на 54-й научной конференции аспирантов, магистрантов и студентов БГУИР, отобранные по следующим направлениям: электронные вычислительные машины, программное обеспечение информационных технологий, высшая математика, физика, информатика, философия и жизнь, встраиваемые вычислительные системы реального времени.

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

УДК 004.9+519.681/.688

© УО «Белорусский государственный
университет информатики
и радиоэлектроники», 2018

СОДЕРЖАНИЕ

СЕКЦИЯ «ЭЛЕКТРОННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ»	13
1. ПРОГРАММНОЕ СРЕДСТВО ИДЕНТИФИКАЦИИ ЧЕЛОВЕКА ПО ЦИФРОВЫМ ИЗОБРАЖЕНИЯМ ОТПЕЧАТКОВ ПАЛЬЦЕВ	14
2. МУЗЫКАЛЬНАЯ ТРАНСКРИПЦИЯ ПРИ ПОМОЩИ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ	15
3. БЕСПИЛОТНЫЕ АВТОМОБИЛИ. ПРИНЦИПЫ РАБОТЫ И ТЕХНОЛОГИИ....	16
4. КЛАССИФИКАЦИЯ ГАЛАКТИК.....	17
5. ОСОБЕННОСТИ АРХИТЕКТУРЫ МНОГОПОЛЬЗОВАТЕЛЬСКИХ ИГР	18
6. ОБНАРУЖЕНИЕ DNS-ТУННЕЛЕЙ С ПОМОЩЬЮ FEEDFORWARD НЕЙРОННОЙ СЕТИ.....	19
7. ЭЛЕКТРОННАЯ СИСТЕМА ГОЛОСОВАНИЯ НА ОСНОВЕ ТЕХНОЛОГИИ БЛОКЧЕЙН.....	21
8. ПЕРЕНОС ОБУЧЕНИЯ ДЛЯ СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ	22
9. БЛОКЧЕЙН КАК ОСНОВА ЗАЩИЩЁННЫХ РАСПРЕДЕЛЕННЫХ БАЗ ДАННЫХ	23
10. ИЗВЛЕЧЕНИЕ ХАРАКТЕРНЫХ ПРИЗНАКОВ В ИСПОЛНЯЕМЫХ ФАЙЛАХ ДЛЯ ОБУЧЕНИЯ НЕЙРОННЫХ СЕТЕЙ ДЛЯ ОБНАРУЖЕНИЯ ВРЕДОНОСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ.....	23
11. МЕТОД МНОГОСТРУЙНОГО МОДЕЛИРОВАНИЯ В 3D ПЕЧАТИ.....	26
12. ПРОГРАММНОЕ СРЕДСТВО ДЛЯ КЛАССИФИКАЦИИ ОБИТАТЕЛЕЙ МОРСКИХ ГЛУБИН	28
13. РАСПОЗНАВАНИЕ КАТЕГОРИИ ПОМЕЩЕНИЯ НА ОСНОВЕ ОНТОЛОГИЧЕСКОГО АНАЛИЗА	28
14. ОБЗОР ОСНОВНЫХ АРХИТЕКТУР СЕМАНТИЧЕСКОГО СЕГМЕНТИРОВАНИЯ.....	29
15. ОБЗОР СИСТЕМ ДЛЯ ТЕЛЕМЕДИЦИНЫ	31
16. VERT.X КАК НОВОЕ СРЕДСТВО РАЗРАБОТКИ МИКРОСЕРВИСОВ.....	32
17. МАТЕМАТИЧЕСКИЙ КАЛЬКУЛЯТОР.....	32
18. «УМНАЯ» ТЕПЛИЦА НА БАЗЕ КОНТРОЛЛЕРА ARDUINO UNO	33
19. ВИЗУАЛИЗАЦИЯ АЛГОРИТМОВ УМНОЖЕНИЯ НА ARDUINO.....	34

20.	ОРГАНИЗАЦИЯ УПРАВЛЕНИЯ ДОКУМЕНТАЦИОННЫМИ ПРОЦЕССАМИ КАФЕДРЫ В СИСТЕМЕ ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА «ЭТЛАС»	35
21.	СТАТИСТИЧЕСКИЕ МЕТОДЫ БИOMETРИЧЕСКОЙ ИДЕНТИФИКАЦИИ IRIS	35
22.	АВТОМАТИЗАЦИЯ РЕДАКТИРОВАНИЯ ИЗОБРАЖЕНИЙ.....	37
23.	SCRUM: ОБУЧЕНИЕ И РЕАЛЬНОСТЬ	38
24.	СИСТЕМА КОНТРОЛЯ СОСТОЯНИЯ КОМПОНЕНТОВ ЛОКАЛЬНОЙ СЕТИ	39
25.	ОПТИМИЗАЦИЯ ЗАПРОСОВ К БАЗЕ ДАННЫХ С ПОМОЩЬЮ АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ	40
26.	СЕРВИСЫ РАСШИРЕНИЯ ФУНКЦИОНАЛА В ОБЛАЧНЫХ ТЕХНОЛОГИЯХ	41
27.	АНАЛИЗ ОСНОВНЫХ АРХИТЕКТУР НЕЙРОННЫХ СЕТЕЙ ГЛУБОКОГО ОБУЧЕНИЯ	44
28.	АЛГОРИТМ АНАЛИЗА ЭМОЦИОНАЛЬНОГО СОДЕРЖАНИЯ КОММЕНТАРИЯ ПОЛЬЗОВАТЕЛЯ.....	44
СЕКЦИЯ «ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»		47
29.	ТАРГЕТИРОВАННЫЕ СОЦИАЛЬНЫЕ СЕТИ.....	48
30.	СПОСОБЫ ОЦЕНКИ УРОВНЯ БЕЗОПАСНОСТИ СООРУЖЕНИЙ	49
31.	ОПТИМИЗАЦИЯ ПОИСКА ПУТЕЙ ТРАНСПОРТНЫХ СРЕДСТВ В ПЕРЕВОЗКАХ	51
32.	РАСПОЗНАВАНИЕ ЭМОЦИЙ ПО ГОЛОСУ НА ОСНОВЕ МАШИННОГО ОБУЧЕНИЯ	52
33.	СРАВНИТЕЛЬНАЯ ХАРАКТЕРИСТИКА КАРТОГРАФИЧЕСКИХ СЕРВИСОВ	53
34.	ДИАГНОСТИКА НЕВРОЛОГИЧЕСКИХ ЗАБОЛЕВАНИЙ НА ОСНОВЕ ОБРАЗЦОВ ПОЧЕРКА	54
35.	МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ОЦЕНКИ НАВЫКОВ ПОЛЬЗОВАТЕЛЯ	55
36.	МЕТОДЫ И АЛГОРИТМЫ ВЕДЕНИЯ ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА.....	55
37.	АНАЛИЗ АРХИТЕКТУРЫ ДЕЦЕНТРАЛИЗОВАННОЙ СИСТЕМЫ БЕЗ ПОСРЕДНИКОВ	57
38.	НЕОДНОРОДНАЯ РАСПРЕДЕЛЕННАЯ ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА.....	58

39.	ПРИМЕНЕНИЕ НЕЙРОННОЙ СЕТИ ХОПФИЛДА ДЛЯ РЕШЕНИЯ ЗАДАЧИ ОПТИМАЛЬНОГО РАССЕЛЕНИЯ СТУДЕНТОВ ОБЩЕЖИТИЯ	59
40.	ПРОГРАММНОЕ СРЕДСТВО “ПЛАНИРОВАНИЕ ЛИЧНЫХ ФИНАНСОВ”	60
41.	ВОССТАНОВЛЕНИЕ ДАННЫХ МЕТОДОМ ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ	61
42.	ОСОБЕННОСТИ КОМПЬЮТЕРНОГО РАСПОЗНАВАНИЯ НОМЕРОВ ВАГОНОВ ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА.....	61
43.	ЗАЩИТЫ ЛИЧНЫХ СООБЩЕНИЙ ДЛЯ СОЦИАЛЬНЫХ СЕТЕЙ.....	63
44.	ОБУЧАЮЩИЙ МОДУЛЬ ‘ШАХМАТЫ’	64
45.	ОСОБЕННОСТИ ИНТЕГРАЦИИ СИСТЕМЫ ДОКУМЕНТООБОРОТА EDI С MICROSOFT DYNAMICSAX	65
46.	ОБНАРУЖЕНИЕ АНОМАЛИЙ В ТЕЛЕМЕТРИЧЕСКОМ СИГНАЛЕ МАЛОГО КОСМИЧЕСКОГО АППАРАТА НА ОСНОВЕ ЭВРИСТИЧЕСКИХ АЛГОРИТМОВ ВОЗМОЖНОСТНОЙ КЛАСТЕРИЗАЦИИ.....	66
47.	АНАЛИЗ СУЩЕСТВУЮЩИХ МОДЕЛЕЙ И МЕТОДОВ РАСПОЗНАВАНИЯ ЛИЦ	68
48.	АНАЛИЗ МОДИФИКАЦИЙ ГЕНЕТИЧЕСКОГО АЛГОРИТМА	69
49.	РЕАЛИЗАЦИЯ ГОЛОВОЛОМКИ «КУБИК РУБИКА».....	70
50.	АЛГОРИТМЫ И СИСТЕМЫ РАСЧЕТА БУКМЕКЕРСКИХ КОЭФФИЦИЕНТОВ НА СПОРТИВНЫЕ СОБЫТИЯ	71
51.	ГОЛОСОВОЕ УПРАВЛЕНИЕ ПЕРСОНАЛЬНЫМ КОМПЬЮТЕРОМ.....	73
52.	ПРОГНОЗИРОВАНИЕ КЛИМАТИЧЕСКИХ УСЛОВИЙ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИЙ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ.....	74
53.	МЕТОДЫ ВОЗВЕДЕНИЯ ЧИСЛА В СТЕПЕНЬ ПО МОДУЛЮ	75
54.	ВНЕДРЕНИЕ МОБИЛЬНЫХ ТЕХНОЛОГИЙ В ПРОЦЕСС ОБУЧЕНИЯ.....	77
55.	СИСТЕМА КОНТРОЛЯ И ОЦЕНКИ СОСТОЯНИЯ ПРОЦЕССА РАЗРАБОТКИ КОМПЛЕКСНЫХ ПРОГРАММНЫХ СРЕДСТВ ДЛЯ ТЕЛЕКОММУНИКАЦИОННОЙ ОТРАСЛИ	78
56.	АЛГОРИТМ DESX КАК СРЕДСТВО ПАРАЛЛЕЛЬНОГО ПОТОКОВОГО ШИФРОВАНИЯ БОЛЬШИХ ОБЪЁМОВ ДАННЫХ.....	79
57.	АЛГОРИТМ РАБОТЫ СИСТЕМЫ УЧЕТА ЗАДОЛЖЕННОСТИ ПО БАНКОВСКИМ КРЕДИТНЫМ КАРТАМ	80
58.	ИСПОЛЬЗОВАНИЕ МНОГОКРАТНЫХ ТЕСТОВ ДЛЯ ПСЕВДОИСЧЕРПЫВАЮЩЕГО ТЕСТИРОВАНИЯ ОЗУ	81

59.	ПРОГРАММНО-АЛГОРИТМИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ОБРАБОТКИ ИНФОРМАЦИИ В ТОРГОВОЙ ДЕЯТЕЛЬНОСТИ	83
60.	ИДЕНТИФИКАЦИЯ РУКОПИСНЫХ ПОДПИСЕЙ С ИСПОЛЬЗОВАНИЕМ НЕЙРОСЕТЕЙ	85
61.	МОДЕЛИ И РЕАЛИЗАЦИЯ ДИСТАНЦИОННОГО УПРАВЛЕНИЯ СКЛАДСКОЙ ДЕЯТЕЛЬНОСТЬЮ	86
62.	УНИФИЦИРОВАННЫЙ ИНТЕРФЕЙС ДЛЯ СОЗДАНИЯ КРОССПЛАТФОРМЕННЫХ ВЕБ-ОРИЕНТИРОВАННЫХ SMARTTV ПРИЛОЖЕНИЙ	87
63.	БАЛАНСИРОВАНИЕ НАГРУЗКИ В WEB ПРИЛОЖЕНИЯХ	88
64.	ЭМПИРИЧЕСКАЯ МОДОВАЯ ДЕКОМПОЗИЦИЯ С ИСПОЛЬЗОВАНИЕМ ГРАФИЧЕСКИХ ПРОЦЕССОРОВ NVIDIA, ПОДДЕРЖИВАЮЩИХ ПРОГРАММНО-АППАРАТНУЮ АРХИТЕКТУРУ CUDA	88
65.	МОДЕЛЬ ПОСТРОЕНИЯ ОБЩИХ ТРЕБОВАНИЙ ПО ОБРАБОТКЕ БАНКОВ ДАННЫХ В ЗАДАЧАХ СОПРОВОЖДЕНИЯ СИСТЕМ	90
66.	ПОСТРОЕНИЕ ИСКУССТВЕННОЙ НЕЙРОННОЙ СЕТИ ДЛЯ РАСПОЗНАВАНИЯ ОБРАЗОВ	91
67.	АНАЛИЗ ПОБОЧНЫХ ДАННЫХ СОСТОЯНИЯ СИСТЕМЫ И ДАННЫХ РЕЗУЛЬТИРУЮЩИХ ТОЧЕК СИСТЕМЫ.....	93
68.	КОНТРОЛЬ КАЧЕСТВА РАЗРАБОТКИ ПРОГРАММНЫХ СРЕДСТВ.....	94
69.	ОПТИМИЗАЦИЯ АВТОМАТИЗИРОВАННОГО ОБНАРУЖЕНИЯ УЯЗВИМОСТЕЙ К SQL-ИНЪЕКЦИЯМ В WEB-ПРИЛОЖЕНИЯХ.....	95
70.	ВИЗУАЛИЗАЦИЯ БОЛЬШИХ ДАННЫХ (BIGDATA)	96
71.	РАСПОЗНАВАНИЕ ВИДОВ ДЕЯТЕЛЬНОСТИ ЧЕЛОВЕКА С ПОМОЩЬЮ СМАРТФОНА.....	97
72.	ПРОГРАММНОЕ СРЕДСТВО ДЛЯ ОРГАНИЗАЦИИ ЗАЩИЩЕННОГО ХРАНЕНИЯ ДАННЫХ НА ВНЕШНИХ НОСИТЕЛЯХ	99
73.	МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ИЗУЧЕНИЯ ИНОСТРАННЫХ СЛОВ НА ОСНОВЕ ИНТЕРВАЛЬНОГО МЕТОДА.....	100
74.	АЛГОРИТМ ПРИМЕНЕНИЯ СЕМАНТИЧЕСКОЙ СЕТИ ДЛЯ ПОИСКА ОТВЕТА НА ВОПРОС.....	101
75.	ОЦЕНКА ЭФФЕКТИВНОСТИ ВНЕДРЕНИЯ АВТОМАТИЗАЦИИ ТЕСТИРОВАНИЯ	103
76.	AWS СЕРВИСЫ ПРИ РАЗРАБОТКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА.....	104

77.	МИКРОПРОЦЕССОРНАЯ СИСТЕМА ДЛЯ ФОРМИРОВАНИЯ ИСПЫТАТЕЛЬНОГО МАГНИТНОГО ПОЛЯ ПРОМЫШЛЕННОЙ ЧАСТОТЫ	106
78.	СРЕДСТВО КОММУНИКАЦИИ ПРЕПОДАВАТЕЛЯ И СТУДЕНТА.....	108
79.	МЕТОДЫ МАШИННОГО ОБУЧЕНИЯ В РЕКОМЕНДАТЕЛЬНЫХ СИСТЕМАХ БАНКОВСКОЙ СФЕРЫ И ОБЛАСТИ БИЗНЕС-АНАЛИЗА.....	109
80.	ЗАЩИТА ДАННЫХ В ОБЛАЧНОМ ХРАНИЛИЩЕ.....	110
81.	СТАТИСТИЧЕСКИЙ ПОДХОД К УПРАВЛЕНИЮ РЕСУРСАМИ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ	111
82.	ДЕНОРМАЛИЗАЦИЯ КАК СРЕДСТВО УЛУЧШЕНИЯ ПРОИЗВОДИТЕЛЬНОСТИ БАЗ ДАННЫХ	112
83.	АВТОМАТИЗАЦИЯ ТЕСТИРОВАНИЯ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ.....	113
84.	ВЕБ-ПРИЛОЖЕНИЕ ОКАЗАНИЯ УСЛУГ СТРАХОВОЙ КОМПАНИИ.....	114
85.	ОБЛАСТИ ПРИМЕНЕНИЯ АЛГОРИТМОВ РАСПОЗНАВАНИЯ ОБЪЕКТОВ.	115
86.	СИСТЕМА ПРЕДОСТАВЛЕНИЯ ТУРИСТИЧЕСКИХ УСЛУГ С ИСПОЛЬЗОВАНИЕМ НЕЙРОСЕТЕВЫХ ТЕХНОЛОГИЙ.....	116
87.	РАСПОЗНАВАНИЕ ДОРОЖНЫХ ЗНАКОВ С ПРЕДВАРИТЕЛЬНОЙ ЛОКАЛИЗАЦИЕЙ НА ИЗОБРАЖЕНИИ.....	117
88.	ГЕОИНФОРМАЦИОННАЯ СИСТЕМА ДЛЯ ОЦЕНКИ УРОВНЯ РАДИОАКТИВНОГО ЗАГРЯЗНЕНИЯ.....	118
89.	СХЕМА ВЗАИМОДЕЙСТВИЯ КЛИЕНТА И WEB-СЕРВЕРА	119
90.	ВЕБ-ПРИЛОЖЕНИЕ ПО СОЗДАНИЮ И УПРАВЛЕНИЮ ПЛАНОМ ХИМИЧЕСКИХ ЭКСПЕРИМЕНТОВ ДЛЯ РОБОТИЗИРОВАННЫХ СТАНЦИЙ	120
91.	ИСПОЛЬЗОВАНИЕ ШТРИХ-КОДОВ ДЛЯ АВТОМАТИЗАЦИИ РАБОТЫ С ТЕСТОВЫМИ ЗАДАНИЯМИ	122
92.	ПРОГРАММНОЕ СРЕДСТВО «РАСПИСАНИЕ ЗАНЯТИЙ БГУИР» ДЛЯ ПЛАТФОРМЫ ANDROID	123
93.	ПРИМЕНЕНИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ В УЧЕБНОМ ПРОЦЕССЕ ВЫСШИХ УЧЕБНЫХ ЗАВЕДЕНИЙ	124
94.	WEB-СЕРВИС КАК СРЕДСТВО УПРОЩЕНИЯ ОБМЕНА ИНФОРМАЦИЕЙ МЕЖДУ РАЗНЫМИ ПЛОЩАДКАМИ.....	125
	СЕКЦИЯ «ВЫСШАЯ МАТЕМАТИКА».....	129
95.	КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ ОБЪЁМНЫХ ТЕЛ И ПОВЕРХНОСТЕЙ	130

96. О ЗАДАЧЕ ИОСИФА ФЛАВИЯ.....	131
97. ГИПОТЕЗА РИМАНА.....	134
98. МАТЕМАТИКО-СИНТАКСИЧЕСКАЯ ТЕОРИЯ КАК СРЕДСТВО ОПТИМИЗАЦИИ АЛГОРИТМОВ РАБОТЫ С ЕСТЕСТВЕННЫМИ ЯЗЫКАМИ	136
99. ФРАКТАЛЫ В ПРИРОДЕ.....	136
СЕКЦИЯ «ФИЗИКА».....	138
100. АНАЛИЗ МЕТОДОМ СКАНИРУЮЩЕЙ ЗОНДОВОЙ МИКРОСКОПИИ МИКРОРЕЛЬЕФА ПОКРЫТИЙ НА СТЕКЛЕ, НАНЕСЕННЫХ ИОННО- АССИСТИРОВАННЫМ ОСАЖДЕНИЕМ	139
101. ВЛИЯНИЕ СВЧ-ИЗЛУЧЕНИЯ НА ЖИВЫЕ ОРГАНИЗМЫ.....	141
102. АНАЛИТИЧЕСКАЯ ОЦЕНКА СТРУКТУРНЫХ ОСОБЕННОСТЕЙ ТОНКИХ МЕТАЛЛИЧЕСКИХ ПЛЕНОК, ФОРМИРУЕМЫХ МЕТОДОМ ИОННО- ПЛАЗМЕННОГО НАПЫЛЕНИЯ	142
103. ОПТИМИЗАЦИЯ СОЛНЕЧНЫХ БАТАРЕЙ С ПОМОЩЬЮ РОБОТИЗИРОВАННОЙ СИСТЕМЫ, УЧИТЫВАЮЩЕЙ ТРАЕКТОРИЮ ДВИЖЕНИЯ СОЛНЦА	144
104. МОДЕЛИРОВАНИЕ ДИФРАКЦИИ ФРЕНЕЛЯ	146
СЕКЦИЯ «ИНФОРМАТИКА».....	148
105. МЕТОДЫ ОПТИМИЗАЦИИ ДОСТАВКИ БОЛЬШИХ ОБЪЕМОВ ТЕЛЕМЕТРИИ В УСЛОВИЯХ УЗКИХ КАНАЛОВ СВЯЗИ	149
106. КЛАССИФИКАЦИЯ СОСТОЯНИЙ РАСТЕНИЙ С ПОМОЩЬЮ СВЁРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ	150
107. МЕССЕНДЖЕР С ВАРИАТИВНОСТЬЮ ИСПОЛЬЗОВАНИЯ КРИПТОСИСТЕМ	151
108. ПРИМЕНЕНИЕ РЕКУРСИВНЫХ НЕЙРОННЫХ СЕТЕЙ ДЛЯ АНАЛИЗА ТОНАЛЬНОСТИ ТЕКСТА.....	152
109. SQRT-ДЕРЕВО КАК НЕЗАСЛУЖЕННО ЗАБЫТАЯ СТРУКТУРА ДАННЫХ...	153
110. ГРАФОВЫЕ БАЗЫ ДАННЫХ	154
111. КЛАСТЕРИЗАЦИЯ В НАDOOP	155
112. МЕТОДЫ УЛУЧШЕНИЯ АЛГОРИТМА Q-ОБУЧЕНИЯ.....	156
113. АСПЕКТЫ СОЗДАНИЯ УНИВЕРСАЛЬНОЙ СИСТЕМЫ ОРГАНИЗАЦИИ УЧЁТА И ОТЧЁТНОСТИ ПРЕДПРИЯТИЯ В СФЕРЕ УСЛУГ	157

114. ИССЛЕДОВАНИЕ ПОГРЕШНОСТЕЙ ЧИСЛЕННЫХ МЕТОДОВ РЕШЕНИЙ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ В ЗАВИСИМОСТИ ОТ БАЗИСА РАЗЛОЖЕНИЯ ФУНКЦИИ	158
115. ГРАФИЧЕСКОЕ ПОСТРОЕНИЕ ФРАКТАЛОВ ЖЮЛИА И МАНДЕЛЬБРОТА. ПРИМЕНЕНИЕ ФРАКТАЛОВ ДЛЯ ВЫЧИСЛЕНИЯ ФУНКЦИИ ВЕЙЕРШТРАССА	160
116. ОНЛАЙН-МАГАЗИН ПРОИЗВОДСТВЕННО-ТОРГОВОГО УНИТАРНОГО ПРЕДПРИЯТИЯ.....	161
117. РАСПОЗНАВАНИЕ ЖАНРА МУЗЫКАЛЬНЫХ КОМПОЗИЦИЙ С ПОМОЩЬЮ МАШИННОГО ОБУЧЕНИЯ	162
118. ИНДЕКСАЦИЯ БАЗ ДАДЗЕННЫХ	164
119. СИЛЬНЫЕ СТОРОНЫ ФРЕЙМВОРКА ANGULAR	165
120. ПРИМЕНЕНИЕ СКРЫТЫХ МАРКОВСКИХ МОДЕЛЕЙ ДЛЯ СОПОСТАВЛЕНИЯ КАРТ	166
121. ПОСТРОЕНИЕ РЕКОМЕНДАТЕЛЬНОЙ СИСТЕМЫ НА ОСНОВЕ АЛГОРИТМОВ ОБРАБОТКИ ЕСТЕСТВЕННОГО ЯЗЫКА И ГРАФОВЫХ БАЗ ДАННЫХ	168
122. КОПУЛЫ КАК ИНСТРУМЕНТ МАТЕМАТИЧЕСКОЙ СТАТИСТИКИ	170
123. ПРОГРАММНАЯ СИСТЕМА СБОРА НОВОСТЕЙ ДЛЯ ГОРОДСКИХ ИНФОРМАЦИОННЫХ ПОРТАЛОВ	171
124. АРХИТЕКТУРА СОБЫТИЙНОГО ПРЕДСТАВЛЕНИЯ ДАННЫХ	172
125. ПРОГРАММНОЕ СРЕДСТВО ОБФУСКАЦИИ VHDL-КОДА.....	173
126. ДОСТАТОЧНЫЕ УСЛОВИЯ ЭКСТРЕМУМА ФНП	175
127. МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ КОММУНИКАЦИИ МЕЖДУ СТУДЕНТАМИ И САМОКОНТРОЛЯ УЧЕБНОЙ ДЕЯТЕЛЬНОСТИ	175
128. ПРИЛОЖЕНИЕ UVENT ДЛЯ РАЦИОНАЛЬНОГО ПОИСКА МЕСТ ОТДЫХА И ОРГАНИЗАЦИИ МЕРОПРИЯТИЙ	176
129. СЕРВИС FOLLOW THE ART ДЛЯ ПРОДАЖИ И ЗАКАЗА КАРТИН.....	178
130. АППРОКСИМАЦИЯ ФУНКЦИЙ В ВИДЕ ЭКСПОНЕНЦИАЛЬНОЙ СУММЫ ..	179
131. ПРИЛОЖЕНИЕ ДЛЯ ПОМОЩИ ПЕРВОКУРСНИКАМ В УЧЁБЕ	180
132. ИСКУССТВЕННЫЕ НЕЙРОННЫЕ СЕТИ В МЕДИЦИНСКОЙ ДИАГНОСТИКЕ	182
133. ПРОГРАММНОЕ СРЕДСТВО МОДЕЛИРОВАНИЯ СТРУКТУРНО-СЛОЖНЫХ СИСТЕМ.....	183

134. АЛГОРИТМЫ И МЕТОДЫ ОПТИМАЛЬНОГО КОДИРОВАНИЯ АППАРАТУРЫ САМОТЕСТИРОВАНИЯ ДЛЯ ОПЕРАТИВНЫХ ЗАПОМИНАЮЩИХ УСТРОЙСТВ	185
135. УПРАВЛЕНИЕ ДОСТУПОМ В ПРИЛОЖЕНИЯХ НА БАЗЕ ФРЕЙМВОРКА RUBY ON RAILS	187
136. РЕКУРРЕНТНАЯ НЕЙРОННАЯ СЕТЬ LSTM ДЛЯ ПРОГНОЗИРОВАНИЯ РЕЗУЛЬТАТОВ СОРЕВНОВАНИЙ ПО КИБЕРСПОРТУ	187
137. ПОСТРОЕНИЕ КРИПТОСТОЙКОГО ГЕНЕРАТОРА ПСЕВДОСЛУЧАЙНЫХ ЧИСЕЛ.....	189
138. МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ ДИСТАНЦИОННОГО УПРАВЛЕНИЯ ПЕРСОНАЛЬНЫМ КОМПЬЮТЕРОМ.....	191
139. СЕРВИС ДЛЯ КОНТРОЛЯ ВЫПОЛНЕНИЯ ЗАДАЧ	192
140. БАЗИСЫ ГРЁБНЕРА. АЛГОРИТМ БУХБЕРГЕРА	193
141. ПРОГРАММА СБОРА ДАННЫХ О СТРУКТУРЕ ВЕБ-САЙТОВ.....	195
142. КРИПТОГРАФИЧЕСКИЕ АЛГОРИТМЫ ШИФРОВ ЗАМЕНЫ.....	196
143. ДИАГНОСТИКА НА РАК ПРИ ПОМОЩИ НЕЙРОННЫХ СЕТЕЙ.....	198
144. ПРИМЕНЕНИЕ РЕЖИМА ПОНИЖЕННОГО ЭНЕРГОПОТРЕБЛЕНИЯ ДОЗУ В ИДЕНТИФИКАЦИИ ЦИФРОВЫХ СИСТЕМ	201
145. KUBERNETES КЛАСТЕР ДЛЯ ПОЛНОТЕКСТОВОГО ПОИСКА.....	203
146. УЛУЧШЕНИЕ КАЧЕСТВА ОЦЕНКИ АРХИТЕКТУРЫ ПО С ПОМОЩЬЮ ИСПОЛЬЗОВАНИЯ SAAM МЕТОДОЛОГИИ.....	204
147. ГРАФИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ ДЕЙСТВИТЕЛЬНЫХ И КОМПЛЕКСНЫХ КОРНЕЙ ТРИГОНОМЕТРИЧЕСКИХ И ГИПЕРБОЛИЧЕСКИХ УРАВНЕНИЙ	205
148. ОБРАБОТКА БОЛЬШИХ ОБЪЕМОВ ИНФОРМАЦИИ С ИСПОЛЬЗОВАНИЕМ ПЛАТФОРМЫ HADOOP И СЛУЖБЫ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ MICROSOFT AZURE	208
149. ОБРАБОТКА БОЛЬШИХ ОБЪЕМОВ ДАННЫХ В АВТОМАТИЗИРОВАННЫХ СИСТЕМАХ УПРАВЛЕНИЯ АВТОМОБИЛЬНЫМ ПОТОКОМ.....	209
150. ПОСТРОЕНИЕ РЕКОМЕНДАТЕЛЬНОЙ СИСТЕМЫ НА ОСНОВЕ АЛГОРИТМОВ ОБРАБОТКИ ЕСТЕСТВЕННОГО ЯЗЫКА	210
151. ПРИНЯТИЕ РЕШЕНИЯ О ДЕНОРМАЛИЗАЦИИ БАЗЫ ДАННЫХ.....	211
152. CHLORINE ENGINE И МИНИ-ИГРА НА НЕМ	212
СЕКЦИЯ «ФИЛОСОФИЯ И ЖИЗНЬ»	214

153. ПРОБЛЕМЫ ГЛОБАЛИЗАЦИИ ИНФОРМАЦИОННОГО ОБЩЕСТВА.....	215
154. СВОБОДА ВОЛИ КАК ПРЕДМЕТ ФИЛОСОФСКОЙ РЕФЛЕКСИИ.....	216
155. ПОНЯТИЕ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА И ПРОБЛЕМА ЕГО ОСМЫСЛЕНИЯ	217
156. ЭТИЧЕСКАЯ ПРОБЛЕМАТИКА НЕЙРОМАРКЕТИНГА	218
157. ПРАБЛЕМА ІНТЭРПРЭТАЦЫІ ТЭРМІНА “УСТОЙЛІВАЕ РАЗВІЦЦЁ” У СУЧАСНЫМ БЕЛАРУСКІМ МЫСЛЕННІ.....	219
158. СОЦИАЛЬНО-ЭКОЛОГИЧЕСКАЯ КУЛЬТУРА БЕЛАРУСИ ПОСЛЕ АВАРИИ НА ЧЕРНОБЫЛЬСКОЙ АЭС.....	220
СЕКЦИЯ «ВСТРАИВАЕМЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ».....	221
159. АЛГОРИТМ ОБУЧЕНИЯ МНОГОСЛОЙНОГО НЕЙРОСЕТЕВОГО КВАНТОВАТЕЛЯ АУДИОКОДЕРА.....	222
160. ПЕРСОНИФИКАЦИЯ ЗВУЧАНИЯ ПРИ ПОМОЩИ ЦИФРОВОЙ ТОНКОМПЕНСАЦИИ	224
161. МЕТОД ШУМООЧИСТКИ РЕЧИ НА ОСНОВЕ ВЫЧИСЛЕНИЯ АВТОКОРРЕЛЯЦИОННОЙ ФУНКЦИИ, СОГЛАСОВАННОЙ С ПЕРИОДОМ ОСНОВНОГО ТОНА.....	226
162. ДЕТЕКТОР РЕЧЕВОЙ АКТИВНОСТИ НА ОСНОВЕ СВЁРТОЧНОЙ НЕЙРОННОЙ СЕТИ С РЕКУРРЕНТНЫМИ СВЯЗЯМИ	228
163. СИСТЕМА ДИАГНОСТИКИ И ВЫЯВЛЕНИЯ НЕВРОЛОГИЧЕСКИХ ЗАБОЛЕВАНИЙ НА ОСНОВЕ АНАЛИЗА РЕЧЕВОГО СИГНАЛА.....	230
164. ЗАЩИЩЕННЫЙ РАДИОКАНАЛ ДЛЯ ДИСТАНЦИОННОГО УПРАВЛЕНИЯ АВТОМАТИЗИРОВАННЫМИ ОБЪЕКТАМИ	232
165. СИСТЕМА КОРРЕКЦИИ РЕЧИ.....	233
166. ОПТИКО-ЭЛЕКТРОННАЯ СИСТЕМА ВИДЕОНАБЛЮДЕНИЯ.....	234
167. КЛАССИФИКАЦИЯ ЗВУКОВ С ПОМОЩЬЮ SEMI-SUPERVISED ПОДХОДА.....	235
168. СИСТЕМА ТЕХНИЧЕСКОГО ЗРЕНИЯ МОБИЛЬНОГО КОМПЛЕКСА ОБНАРУЖЕНИЯ БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ	237
169. IP – ЯДРО АЛГОРИТМА SHA – 1	237
170. ОЦЕНКА РЕЧЕВОЙ МАСКИ ДЛЯ ИДЕНТИФИКАЦИИ ДИКТОРА.....	239
171. СТИЛИЗАЦИЯ ГОЛОСА С ИСПОЛЬЗОВАНИЕМ ГЛУБОКОГО ОБУЧЕНИЯ.....	241

172. ПОВЫШЕНИЕ КАЧЕСТВА ИЗОБРАЖЕНИЯ НА БАЗЕ АЛГОРИТМОВ НЕЙРОННЫХ СЕТЕЙ.....	242
173. СИСТЕМА МОДЕЛИРОВАНИЯ ЭФФЕКТА ПОТЕРИ СЛУХА ПО АУДИОГРАММЕ.....	244
174. СИСТЕМА ОХРАННОЙ СИГНАЛИЗАЦИИ.....	246
175. СИСТЕМА КОМПЕНСАЦИИ ГРОМКОСТИ ДЛЯ СЛУХОВОГО АППАРАТА НА ОСНОВЕ КОМПРЕССИИ И ПЕРЕНОСА ВЫСОКОЧАСТОТНЫХ КОМПОНЕНТ РЕЧЕВОГО СИГНАЛА.....	248
176. ПРИМЕНЕНИЕ ЛОГАРИФМИЧЕСКОЙ СИСТЕМЫ СЧИСЛЕНИЯ ДЛЯ АЛГЕБРЫ КВАТЕРНИОНОВ.....	250
177. СИСТЕМА ОПРЕДЕЛЕНИЯ ГЕОМЕТРИЧЕСКИХ ПАРАМЕТРОВ И РАЗМЕРОВ КРИСТАЛЛОВ СЛОЖНОЙ ФОРМЫ	251
178. ДЕТЕКТИРОВАНИЕ ЛИЦ НА ИЗОБРАЖЕНИИ С ОДНОВРЕМЕННОЙ КЛАССИФИКАЦИЕЙ ЛИЦ С ПОМОЩЬЮ СВЁРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ.....	252
179. СИСТЕМЫ ФАЗОВОЙ АВТОПОДСТРОЙКИ ЧАСТОТЫ.....	254
180. ВЕРИФИКАЦИЯ VHDL-МОДЕЛЕЙ КОНЕЧНЫХ АВТОМАТОВ	255

СЕКЦИЯ «ЭЛЕКТРОННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ»

ПРОГРАММНОЕ СРЕДСТВО ИДЕНТИФИКАЦИИ ЧЕЛОВЕКА ПО ЦИФРОВЫМ ИЗОБРАЖЕНИЯМ ОТПЕЧАТКОВ ПАЛЬЦЕВ

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

Абрамович Н.А.

Насуро Е.В. – к.т.н., доцент

Высокая популярность биометрических систем идентификации человека в современном мире обусловлена их высокой скоростью работы, надежностью и отсутствием необходимости для пользователя носить или запоминать что-либо. Одним из самых дешёвых и при этом надёжных признаков, по которым может проводиться идентификация, являются отпечатки пальцев. Простота интеграции и низкая стоимость выгодно отличает данный метод от идентификации по геометрии лица и сетчатке глаза. По сравнению с распознаванием по радужной оболочке глаза и рисунку вен, идентификация данным методом в среднем выполняется быстрее (около одной секунды) и доставляя меньше дискомфорта пользователю. На данный момент, благодаря своим преимуществам, системы идентификации по отпечатку пальца занимают более половины рынка систем биометрической идентификации.

Системы биометрической идентификации состоят из программного и аппаратного компонентов. Аппаратный компонент представляет собой сканер, предназначенный для считывания узора отпечатка пальцев пользователя и перевод его в цифровую форму. Полученное изображение часто бывает искажено из-за мелких повреждений кожи и её физического состояния.

Программный компонент служит для сравнения полученного отпечатка с отпечатками в базе и выделения наилучшего совпадения. Среди основных способов реализации программного компонента можно выделить статистические методы и идентификацию с использованием машинного обучения[1][2].

Статистические методы распознавания неплохо работают при небольшом наборе данных, но на достаточно больших наборах их точность снижается. Также такие методы требуют четкой алгоритмизации и плохо выявляют скрытые зависимости.

Идентификация на основе средств машинного обучения же практически лишена вышеперечисленных недостатков. Её преимуществами, безусловно, являются:

- универсальность
- возможность работать с большими наборами данных

Среди недостатков же можно выделить низкую степень устойчивости к трансформациям и искажениям.

Комбинирование обоих методов (устранение помех и искажений линейными алгоритмами и распознавание с помощью систем машинного обучения) позволяет компенсировать недостатки обоих методов и имеет высокий потенциал для использования в системах идентификации. Недостатком комбинирования является некоторое замедление работы. Таким образом, данный метод не следует применять в тех случаях, когда время работы критично.

В настоящее время одним из самых популярных алгоритмов сравнения отпечатков пальцев является сравнение по особым точкам. В качестве особых точек выбираются точки ветвления и конечные точки. Эти точки выбираются на двух изображениях, а затем сравниваются и по количеству совпавших точек принимается решение по идентичности отпечатков.

Схема работы алгоритма:

- 1) адаптивная фильтрация, выделение зоны интереса;
- 2) бинаризация, выделение однородных областей;
- 3) морфологическая обработка;
- 4) скелетизация;
- 5) векторизация;
- 6) векторная постобработка;
- 7) сравнение двух наборов особых точек.

Скелетизация - морфологическая операция утоньшения приводит бинарное изображение к варианту, в котором толщина всех линий – 1 пиксел. Операция стягивает линии в центр, не делая при этом разрывов.

Волновой метод. Его задачей является векторное представление изображения в виде нагруженного графа – то есть определение концевых точек, точек пересечения (вершины графа), а также линий и дуг, составляющих фигуры (рёбра графа) [3].

Метод заключается в анализе пути прохождения сферической волны по изображению. На каждом этапе анализируется смещение центра масс точек, образующих новый шаг волны, относительно его предыдущих положений [4]. После завершения построения скелета с помощью сферической волны, полученный результат оптимизируется и анализируется, отыскиваются особые точки.

Метод состоит из следующих шагов:

- построение скелета изображения с помощью сферической волны;
- оптимизация полученного скелета.

Список использованных источников:

- 1) Lin Hong - Automatic personal identification using fingerprints (1998, ISBN:0-599-07596-1)
- 2) D.K.Isenor, S.G.Zaky - Fingerprint identification using graph matching (2003)
- 3) Fingerprint identification using Delaunay triangulation (2002, ISBN:0-7695-0446-9)
- 4) Andrew K.HrechakJames A.McHugh - Automated fingerprint recognition using structural matching (1990)

МУЗЫКАЛЬНАЯ ТРАНСКРИПЦИЯ ПРИ ПОМОЩИ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ

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

Андрэа А.И.

Насуро Е. В. - к.т.н., доцент

Транскрибирование – распознавание нот в цифровых музыкальных аудиозаписях. Исследуется полифоническая запись фортепиано, т.е. запись, на которой воспроизводится множество нот одновременно. Необходимо определить время воспроизведения каждой отдельно сыгранной ноты и ее длительность. На данный момент нету единого оптимального подхода для решения этой задачи, а существует некоторое их кол-во, каждый из которых имеет как свои индивидуальные сильные и слабые стороны.

Звук — распространение в виде упругих волн механических колебаний в твёрдой, жидкой или газообразной среде. Обычный человек способен слышать звуковые колебания в диапазоне частот от 16 Гц до 20 кГц. В музыке звук обычно получается колебаниями частей инструмента, исключение — духовые инструменты. Звукам, извлекаемых из фортепиано, в итоге соответствуют частоты от приблизительно 16 Гц до 15800 Гц. Фундаментальная частота - самая низкая частота ноты, это также частота, соответствующая высоте, которая идентифицируется человеком слушатель. Все гармоники кратны фундаментальной частоте.

Чтобы получить представление о частотах необходимо перевести временное представление композиции в частотно-временное.

Рассмотренные существующие решения осуществляют переход к частотно-временному представлению при помощи STFT (Short-time Fourier transform), получают спектрограмму звукозаписи. Это позволяет работать с отдельными частотными компонентами звука. Ось x спектрограммы – время, y – частота, цвет – интенсивность определенной частоты в данный момент времени. Для получения спектрограммы сигнал разделяется на временные сегменты с перекрытием, выполняется на каждом сегменте FFT и далее совмещаются все FFT сегментов на одном графике. Недостаток STFT в сложности нахождения компромисса между временным и частотным разрешениями. Вместе с тем, при использовании алгоритмов быстрого преобразования Фурье невозможно произвольным образом выбирать частоты его компонент. Это создает неудобства при дальнейшей обработке, поскольку невозможно точно определить количество звуковой энергии, приходящейся на частоты, соответствующие ступеням звукоряда. При фиксированном размере окна FFT, информация, используемая для расчета FFT ограничена. Если выбрать большое окно, то получим очень хорошее частотное разрешение, но это приведет к перекрытию нот. С другой стороны, выбирая малое окно, достигаем хорошего разделения нот, но частотное представление нот хуже, что, возможно, уменьшит способность разделять ноты.

В преобразовании постоянного качества (constant-Q преобразование) в отличие от преобразования Фурье, размер анализируемого фрагмента и размер оконной функции зависят от номера соответствующей частотной компоненты. Достоинством этого преобразования является легкость дальнейшей работы со спектром, поскольку его компоненты напрямую соответствуют ступеням звукоряда. Недостатками являются большая сложность вычислений и зависимость от правильного определения частоты настройки.

Гребёнка фильтров (filter bank). В цифровой обработке сигналов любое преобразование сигнала называют фильтром. В данном случае под фильтром понимается полосовой фильтр – преобразование, сохраняющее в звуке только частоты, находящиеся в некоторой полосе частот. Быстрое преобразование Фурье эквивалентно вполне определенной гребёнке достаточно грубых фильтров. Вместо них можно использовать любые другие фильтры, у каждого из которых центр полосы пропускания соответствует частоте одной из ступеней звукоряда, а ширина полосы пропускания достаточно мала, чтобы не охватывать частоты соседних ступеней. Эти фильтры можно подобрать так, чтобы они были менее грубыми, то есть более точно определяли количество звуковой энергии, приходящейся на их полосы пропускания. Недостатком данного метода является большая вычислительная сложность в сравнении с алгоритмом быстрого преобразования Фурье.

Предлагаемое улучшение – использовать вейвлет-преобразование.

Вейвлет-преобразование Вейвлет-преобразование (WT) – еще одна техника для перевода сигнала из временного представления в частотно-временное. WT является альтернативой STFT и позволяет преодолеть проблемы связанные с неопределенностью частотного и временного разрешений. STFT представляет одинаковое временное разрешение для всех частот, в то время как DWT предоставляет высокое временное разрешение и низкое частотное для высоких частот и высокое частотное разрешение и низкое временное для низких частот. Человеческое ухо демонстрирует аналогичные частотно-временные характеристики разрешения.

Список использованных источников:

1. George Tzanetakis, Georg Essl, Perry Cook, Audio Analysis using the Discrete Wavelet Transform
2. Ross Kidson, Musical Note Identification Violin Score Reproduction from Audio Recordings
3. E. Poliner, P. W. Ellis — A Discriminative Model for Polyphonic Piano Transcription (2006)
4. M. Marolt — A Connectionist Approach to Automatic Transcription of Polyphonic Piano Music (2004)

БЕСПИЛОТНЫЕ АВТОМОБИЛИ. ПРИНЦИПЫ РАБОТЫ И ТЕХНОЛОГИИ

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

Белко К.А.

Одинец Д. Н. - к.т.н., доцент

Беспилотный автомобиль – транспортное средство, оборудованное системой автоматического управления, которое может передвигаться без вмешательства человека. Устройство, ведущее автомобиль по заданной траектории, называется автопилотом. Наиболее часто автопилоты используются для управления летательными аппаратами (т.к. в большинстве случаев полёт происходит в пространстве без большого количества препятствий) и транспортными средствами, движущимися по рельсовым путям.

Беспилотные автомобили имеют встроенные системы навигации и датчики, позволяющие точно определить местоположение автомобиля и проложить оптимальный маршрут до пункта назначения. Высокочувствительные лазерные датчики устанавливаются на переднем и заднем бампере для распознавания препятствий перед движущимся автомобилем и более точной ориентации в пространстве. Датчики передают данные на бортовой компьютер, который производит анализ полученной информации. С помощью этой системы автомобиль быстро реагирует на резкие изменения на дороге. В беспилотных автомобилях также устанавливаются внутренние камеры, с помощью которых бортовой компьютер распознаёт приближающиеся объекты, сигналы светофора, дорожные знаки и т.д. Также может использоваться LIDAR (LightIdentificationDetectionandRanging) – вращающийся датчик, который устанавливается на крыше автомобиля и сканирующий пространство вокруг автомобиля в радиусе порядка сотни метров с помощью света, отражённого от объектов.

а)

б

Рисунок 1. LIDAR:

а – внешний вид, б – трёхмерная карта, сгенерированная с помощью LIDAR'a

В настоящее время множество компаний, таких как Tesla, Google, Yandex, General Motors, Audi и другие занимается разработкой беспилотных автомобилей.

Преимущества беспилотных автомобилей:

- применение беспилотных автомобилей для перевозки грузов в опасных зонах;
- снижение стоимости транспортных услуг (за счёт экономии на заработной плате водителей);
- экономия времени;
- возможность перемещаться в автомобиле людям с физическими недостатками, препятствующими управлять автомобилем;
- сведение к минимуму количества ДТП (бортовой компьютер отреагирует на дорожную ситуацию намного быстрее человека).

Недостатки беспилотных автомобилей:

- надёжность программного обеспечения;
- водители теряют рабочие места;
- этическая проблема (проблема вагонетки);
- потеря приватности (соединение беспилотных автомобилей со сторонними службами, которые будут отслеживать их местоположение).

В рамках диссертационного исследования будет реализована система детектирования препятствий перед движущимся автомобилем. Разрабатываемая система необходима для обеспечения помощи водителю в управлении транспортным средством, а также для применения в беспилотных автомобилях. Данная система будет с открытым исходным кодом.

Список использованных источников:

1. Русанов, А.Д. Обзор принципов работы и алгоритмов распознавания объектов окружающей среды в беспилотных автомобилях / А.Д. Русанов, Д.К. Некрасов. – НИУ Высшая школа экономики, 2016. – 7 с.
2. Guizzo E. How google's self-driving car works / E. Guizzo. – IEEE Spectrum Online, 2011.
3. Broggi A. The VisLab Intercontinental Autonomous Challenge: an extensive test for a platoon of intelligent vehicles / A. Broggi, P. Cerri, M. Felisa, M. Chiara Laghi, L. Mazzei, P. Paolo Porta. – Int. J. Vehicle Autonomous Systems, 2012.

КЛАССИФИКАЦИЯ ГАЛАКТИК

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

Береснев Д.В.

Самаль Д.И. – к.т.н., доцент

Крайне высокие темпы роста объемов информации, повышение качественных характеристик собранных данных открывают перед астрономами и астрофизиками новые направления для исследований. Однако очень большие объемы собранных данных требуют создания и применения новых подходов к их обработке и новых методов анализа, которые стали доступны благодаря высокотехнологичному астрофизическому оборудованию, и которые реализуются с помощью последних достижений физико-математических наук, которые в свою очередь позволяют реализовать эффективные алгоритмы, составляющие основу машинного обучения и интеллектуального анализа данных.

Галактика – это массивная гравитационно связанная система, состоящая из звезд, звездных остатков, межзвездного вещества и большого количества неизвестной субстанции, условно называемой темной материей.

Морфологическая классификация галактик — система разделения галактик на группы по визуальным признакам, используемая в астрономии. В современной астрономии есть несколько схем разделения галактик на морфологические типы. Одна из наиболее известных схем была составлена Эдвином Хабблом, и впоследствии развита Полом Ходжем, Уильямом Морганом, Жераром де Вокулером, Сидни ван ден Бергом и Аланом Сендиджем. Каждый из способов морфологической классификации галактик имеет свои особенности и соответственно имеет свои преимущества и недостатки. Однако на практике, в современной астрономии и астрофизике в ходе проведения исследований используется иной подход к морфологической классификации галактик.

Практико-ориентированный метод (см. рис. 1), на базе которого производилось построение программного средства для классификации, подробно описан в [2]. Данный подход дополняет и адаптирует последовательность Хаббла (см. рис. 2) для решения практических задач на реальных астрономических данных. При проектировании программного обеспечения для морфологической классификации галактик использовались некоторые теоретические и прикладные аспекты из [3]. В ходе реализации программного средства была построена нейронная сеть, архитектура которой представлена на рис. 3.

Рис. 2 – Последовательность Хаббла

Рис. 1 – Методика принятия решения, предлож. в проекте GalaxyZoo

Рис. 3 – Архитектура нейросети

Таким образом, разработанное программное средство позволяет производить классификацию галактик на три основных класса: эллиптические, спиральные и иррегулярные в соответствии с упомянутым выше практико-ориентированным подходом с достаточно высоким уровнем точности. Построение систем обработки астрономических данных, а также разработка эффективных методов анализа являются крайне актуальными задачами в современных исследованиях. Разработка интеллектуальных систем позволит автоматизировать многие процессы, что в свою очередь способно ускорить прогресс в астрономии и астрофизике.

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: 09.04.2018.
3. Banerji, M. Galaxy Zoo: Reproducing Galaxy Morphologies Via Machine Learning / Manda Banerji, Ofer Lahav, Chris J. Lintott. – USA : Coornell University, 2010. – 13 p. – Mode of access: <https://arxiv.org/pdf/0908.2033.pdf>. – Date of access: 09.04.2018.

ОСОБЕННОСТИ АРХИТЕКТУРЫ МНОГОПОЛЬЗОВАТЕЛЬСКИХ ИГР

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

Жешко А.А., Близнюк Н.М.

Искра Н.А. - старший преподаватель

Обзор многопользовательских игр, топологий и типичных проблем.

При проектировании многопользовательской игры наибольший акцент ставится на пропускные способности сети, а также на ее топологию.

Для обеспечения высокой пропускной способности предлагается ряд мер как аппаратных, так и программных [1]: минимальное расстояние от сервера до клиента, размер пакета должен стремиться к MTU, сжатие информации в пакетах и т.д.

На сегодняшний день, существует несколько топологий сетей в многопользовательских играх:

1. "клиент-сервер", самая популярная. Обусловлено тем, что только сервер имеет представление о полной картине игры. Это делает невозможным читерство в игре. Код игры делится на клиентский и серверный.
2. "точка-точка". Каждый клиент одновременно является и сервером с точки зрения геймплея. Существует мастер-клиент для избежания коллизий внутренних id для игроков. Требуется повышенной пропускной способности, однако растет она линейно.
3. "точка-точка через ретранслятор". Проблема топологии "точка-точка" заключается в том, что каждому клиенту сообщается ip-адрес других клиентов. При подключении этим клиентам может возникнуть проблема с технологией NAT.

Для многопользовательских игр очень большую роль играет выбранный сетевой протокол. На данный момент выбор осуществляется из двух: TCP и UDP. TCP обладает огромными возможностями по повторной отправке пакетов, сохранению очередности и т.д. Однако в большинстве случаев для игр с высоким и средним требованием к пропускной способности не все данные нужно передавать с такой точностью (данные о местоположении). UDP же не обладает этими возможностями, но это "чистый лист", на котором можно программно создать свои алгоритмы повтора, сохранения очередности пакетов и т.д.

Пример архитектуры карточной игры.

В игре используется протокол TCP, т.к. игра пошаговая и нетребовательна к пропускной способности сети.

По аналогии с сетевыми протоколами, удобно разделить сетевой код на слои и дать вышестоящему слою удобное API для работы [3].

Каждый слой отвечает за свое взаимодействие с сетью и игрой:

1. Application - сама игра, с сетью взаимодействует исключительно через методы NetworkManager.
2. Network Manager - предоставляет API для взаимодействия с сервером/клиентом (для каждой игры это API различается). Уведомляет игру о новых пакетах. Формирует высокоуровневые пакеты-классы и передает их HLAPI.
3. HLAPI - принимает высокоуровневые пакеты от Network Manager, формирует очередь высокоуровневых пакетов, трансформирует низкоуровневые пакеты из LLAPI в высокоуровневые пакеты при приеме и высокоуровневые в низкоуровневые при передаче LLAPI. Постоянно опрашивает LLAPI на наличие новых пакетов.
4. LLAPI - работает через сокеты Беркли, формирует поток байтов (низкоуровневые пакеты) и передает их HLAPI. Передает низкоуровневые пакеты по сети.

Для передачи пакетов используется собственная реализация протокола прикладного уровня. Структура пакета состоит следующих компонент [4]:

- стартовые байты (фиксируют начало пакета);
- название пакета (уникальный идентификатор для пакета и его обработки);
- тег сообщения (для формирования программных очередей пакетов);
- данные пакета (информация, которая состоит преимущественно из набора информации простейших типов данных).

Список использованных источников:

1. Глейзер Дж., Мадхав С., Многопользовательские игры. Разработка сетевых приложений. - СПб.: Питер, 2017. - 368 с.
2. Искра Н.А., Близнюк Н.М., Жешко А.А. Подход к обучению объектно-ориентированному проектированию и программированию через реализацию игрового приложения. / Объектные системы. 2016. № 13. С. 25-31.
3. Game programming patterns [Электронный ресурс]. - Режим доступа: <https://gameprogrammingpatterns.com/>
4. Networking. Gaffer on Games [Электронный ресурс]. - Режим доступа: <https://gafferongames.com/tags/networking/>

ОБНАРУЖЕНИЕ DNS-ТУННЕЛЕЙ С ПОМОЩЬЮ FEEDFORWARD НЕЙРОННОЙ СЕТИ

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

Бубнов Я.В.

Иванов Н.Н. - канд. физ-мат. наук., доцент

Методика DNS-туннелирования эксплуатируется при разработке вредоносных программ с целью сокрытия их сетевой активности. Современные механизмы обеспечения защиты сетевой инфраструктуры не предоставляют необходимых инструментов для решения данной проблемы. В статье предлагается способ обнаружения DNS-туннелирования с помощью feedforward нейронной сети.

DNS-туннель (далее туннель) представляет собой способ предоставления сетевых сервисов с помощью инкапсуляции протоколов различных уровней модели OSI. Гибкость протокола DNS позволяет использовать разнообразные техники для создания туннелей. Самый распространенный способ подразумевает добавление передаваемой информации через субдомены делегированных зон [1].

В течение предыдущих лет создано большое количество вредоносных программ, таких как *Morto* [2], *Feederbot* [3], а также программ для кражи информации из платежных систем, как *BernhardPOS* и *FrameworkPOS* [4]. Упомянутые программы используют технику DNS-туннелирования для передачи данных из скомпрометированной системы.

В статье [5] Надлер, Аминов, Шабтай описывают два способа обнаружения туннелей: с помощью *Isolation Forest*, а также с помощью *One-Class SVM*. Исследование содержит позитивные результаты использования описанных техник, в частности авторы добились 99-процентной чувствительности разработанной модели. Однако авторы работы решают задачу бинарной классификации, то есть выявляют вредоносный трафик среди регулярного DNS трафика.

В данной статье описывается способ классификации сетевого трафика для обнаружения туннелей, а также программ, с помощью которых созданы туннели. Таким образом задача - классифицировать DNS трафик по типам программ для создания туннелей.

Исходные данные для решения задачи собраны путем создания туннелей с помощью наиболее распространенных программ, находящихся в открытом доступе: *dns2tcp*, *dnscapy*, *iodine* а также *tuns*. Исходя из этого можно выделить 6 классов: 4 вида туннелирования, регулярный трафик и весь остальной трафик.

Экспериментальная установка состоит из клиентской и серверной станций с установленным между ними туннелем. На серверной станции запущен SSH сервер, а клиентская станция копирует случайным образом сгенерированный файл размером 2MiB по инкапсулированному протоколу SSH. В таблице 1 указано количество выборок (DNS пакетов) для каждого вида программы. Помимо вредоносного трафика, на установке собран регулярный трафик путем разрешения списка доменных имен, представленных в репозитории *OpenDNS* [6].

Таблица 1. - Исходные данные задачи

Тип трафика	Размер выборки
Dns2tcp туннель	9070
Dnscapy туннель	14424
Iodine туннель	12228
Tuns туннель	12543
Регулярный трафик	17280

Для каждой секции DNS пакета из полученной выборки выделены атрибуты, указанные в таблице 2. Атрибуты состоят из атрибутов ресурсных записей (RR) каждой из секций: запроса, ответа, дополнительного ответа, а также из авторитативной секции.

Таблица 2. - Выбранные атрибуты DNS пакетов

Атрибут	Описание
RR count	Количество пакетов в секции сообщения.
RR name length	Длина имени ресурсной записи.
RR name shannon entropy	Информационная энтропия Шеннона имени ресурсной записи.
RR type	Тип ресурсной записи.
RR data length	Длина значения ресурсной записи.
RR data shannon entropy	Информационная энтропия Шеннона значения ресурсной записи.

Полученная совокупность атрибутов представлена на рисунке 1а с помощью пространственного уплотнения *t-SNE* [7].

Для решения задачи используется трехслойная нейронная сеть со следующей конфигурацией:

где *ReLU* - функция активации первого слоя нейронной сети $z^{(1)}$, состоящая из 128 нейронов; *D_{0.1} dropout* слой для предотвращения переобучения сети [8] $z^{(1)}$; *SoftMax* - функция активации выходного слоя сети $z^{(3)}$ количеством 128 нейронов. В качестве оптимизатора исследуемой функции классификации используется функция Нестерова-Адама [9], а в качестве функции потерь - перекрестная энтропия.

Обучение нейронной сети осуществляется поэтапно выборками фиксированного размера, где каждая

выборка случайным образом переупорядочивается с целью минимизации переобучения сети.

Результаты классификации с помощью описанной нейронной сети представлены в таблице 3. Как видно, точность распознавания трафика описанной модели составляет 83%, значение чувствительности составляет 81%, а специфичности - 84%.

Таблица 3. - Оценки качества классификации

Метрика	Значение
Accuracy	0,835156
Recall	0,818359
Precision	0,841027

Для оценки качества классификации построены ROC-кривые, представленные на рисунке 1б, для каждого из рассматриваемого класса, а также усредненное значение по всей совокупности классов. Собранные оценки характеризует высокую степень распознавания различных механизмов DNS туннелирования предложенной методикой.

Рис. 1а - Исходные данные в двухмерном t-SNE представлении.

Рис. 1б – ROC-кривые для каждого класса, а также их среднее значение.

Большую точность в распознавании программы туннелирования dns2tcp можно объяснить тем, что это единственная программа, которая использует исключительно ресурсные записи типа TXT. Причем, в отличие от туннеля dnscару, который периодически меняет тип записей с TXT на CNAME, программа dns2tcp не полностью утилизирует доступное пространство пакета: его длина в среднем меньше длины пакетов других программ.

В работе представлены результаты использования feedforward нейронной сети для решения задачи классификации DNS трафика с целью выявления туннелирования. Результаты подтверждают эффективность использования предложенной методики, дальнейшая работа будет направлена на улучшение качества распознавания, а также на возможность распознавания новых методик туннелирования.

Список использованных источников:

1. Farham, G. Detecting DNS Tunneling / G. Farham, A. Atlasis. - Boston : SANS Institute, 2013.
2. Zhong, X. Stealthy Malware Traffic - Not as Innocent as It Looks / X. Zhong, Y. Fu, L. Yu, R. Brooks, K. Venayagamoorthy - Clemson : Clemson University Press, 2017.
3. Deitrich, C. On Botnets that use DNS for Command and Control / C. Deitrich, C. Rossow, F. Freiling, H. Bos, M. van Steen, N. Pohlman - Gelsenkirchen : Institute for Internet Security, 2011.
4. Valenzuela, I. Game Changer: Identifying and Defending Against Data Exfiltration Attempts / I. Valenzuela. - Boston : Sans Institute, 2015.
5. 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.
6. OpenDNS Top Domains List [Электронный ресурс]. - Режим доступа: <http://github.com/opensns/public-domain-lists>. - Дата доступа: 03.04.2018.
7. Van der Maaten, L. Visualizing Data using t-SNE / L. van der Maaten, G. Hinton. - Cambridge : Journal of Machine Learning Research, 2008.
8. Srivastana, N. Dropout: A Simple Way to Prevent Neural Networks from Overfitting / N. Srivastana, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov. - Cambridge : Journal of Machine Learning Research, 2014.
9. Dozat, T. Incorporating Nesterov Momentum into Adam / T. Dozat. - Stanford University Press, 2017.

ЭЛЕКТРОННАЯ СИСТЕМА ГОЛОСОВАНИЯ НА ОСНОВЕ ТЕХНОЛОГИИ БЛОКЧЕЙН

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

Волков М.А.

Насуро Е.В. – к.т.н., доцент

Демократическое голосование является одним из решающих и, безусловно, серьезным политическим событием в любой стране. В настоящее время самый распространенный способ голосования - через бумажную систему. Однако, на сегодняшний день, такая система безнадежно устарела и требует внедрения более современных технологий. Системы электронного голосования позволяют ускорить процесс подсчета голосов, а также упростить голосование людям с ограниченными возможностями.

Цифровое голосование - это использование электронных устройств, таких как машины для голосования или интернет-браузер, для подачи голосов. Голосование с использованием специальных устройств получило название «электронное голосование» (e-voting), а голосование с помощью интернет-браузера называют «интернет голосованием» (i-voting).

Системы электронного подсчета голосов применяются на выборах с 1960-х годов, с тех пор, как появились перфокарты. Первые перфокарты и компьютеризированные машины для подсчета голосов использовались в Грузии в 1964 году, вскоре примеру последовало и США (в Орегоне и Калифорнии). Системы прямой записи голосов (DRE - Direct Recording Electronic), накапливающие голоса на одном устройстве, используются повсеместно в Бразилии, также достаточно широко распространены в Индии, Нидерландах, Венесуэле и США. Система голосования с прямой записью осуществляет сбор голосов путём предоставления механических или электрооптических компонентов (как правило, кнопки или сенсорные экраны), которые могут быть использованы избирателем. Информация о голосах накапливается на специальных носителях; после голосования она сводится в таблицы, хранимые на съёмных носителях, а также может быть распечатана. Также система может передавать итоги в центр голосования для сверки и подсчёта. Как и все машины для голосования, системы DRE увеличивают скорость подсчета голосов. Они также могут включать широкий спектр вспомогательных технологий для самых больших классов людей с ограниченными возможностями, позволяя им голосовать без потери анонимности своего голоса. Эти машины могут использовать наушники и другие адаптивные технологии для обеспечения необходимой доступности голосования.

Системы Интернет-голосования завоевали популярность и используются в правительственных выборах и референдумах в Великобритании, Эстонии и Швейцарии, а также муниципальных выборах в Канаде и партийных выборах в США и Франции. Интернет-выборы — один из способов электронного голосования, представляющий собой проведение части или полностью голосования на выборах и референдуме с использованием сети Интернет. В Эстонии электронное голосование проводится с 2005 года, а в 2007 она стала первой страной в мире, разрешившей голосование в режиме онлайн. На парламентских выборах 2015 года 30,5% голосов было подано в рамках интернет системы голосования (i-voting). Основой этой системы является национальная идентификационная карточка, которая предоставляется абсолютно всем гражданам Эстонии. Эти карты содержат зашифрованные файлы, которые идентифицируют владельца, и позволяют выполнять ряд онлайн и электронных мероприятий. Когда избиратель подает свой голос, он передается через общедоступный сервер подачи голосов в базу, где он будет храниться в зашифрованном виде, пока не закончится онлайн период для голосования. Затем из поданного голоса стирается вся личная информация избирателя, и все голоса переносят на сервер подсчета голосов, который не имеет доступа ни к одной сети. Сервер расшифровывает и подсчитывает голоса, а затем выводит результаты.

Основным минусом традиционных систем остается открытость и безопасность. В последнее время данные общественного мнения часто фальсифицируются для достижения различных политических, социальных и корпоративных целей. В связи с этим вопрос честного и открытого голосования стоит очень остро и требует современных и надежных способов решения проблемы. Электронное голосование подвержено хакерским атакам со многих сторон. Взломанное ПО может быть установлено как на машинах для голосования (личных устройствах либо специальных средствах), так и на центральных серверах подсчета голосов. Небезопасным является и процесс передачи данных по сети. Наличие единого сервера открывает возможность для осуществления DDoS и другого рода атак. Из-за сложности протоколов электронного голосования, потенциальных компьютерных ошибок и хакерских атак избирательные комиссии в Казахстане (2011) и Нидерландах (2008, 2017) возвращались к бумажным бюллетеням, урнам и ручному подсчету голосов.

Технология блокчейн решает многие проблемы безопасности, связанные с другими более традиционными системами голосования на основе общего единого сервера. Такая система позволяет создать прозрачные и надежные инструменты удаленного волеизъявления и сможет обеспечить защиту от внешнего воздействия на результаты голосования. Это форма распределенной базы данных, где записи принимают форму транзакций. Основная ценность блокчейн-цепи заключается в том, что она позволяет напрямую делиться базой данных без центрального администратора, вместо того, чтобы иметь некоторую централизованную логику приложения. Поскольку независимые участники системы, которые не могут доверять друг другу, должны подтвердить точность каждой транзакции и договориться о том, попадет ли

очередная запись в регистр или нет, она обеспечивает уровень прозрачности и постоянства, недоступный для традиционных способов голосования.

Разработка электронной системы голосования на основе технологии блокчейн позволило бы обеспечить прозрачность выборов, конфиденциальность информации избирателей, а также полную неприкосновенность и безопасность данных. Технология блокчейн обладает всеми характеристиками, которые общественность ожидает получить от платформы, которая, возможно, является самой важной частью демократического общества.

Список использованных источников:

1. Andrew Barnes, Digital Voting with the use of Blockchain Technology
2. Roger Wattenhofer, The Science of the Blockchain : 2016
3. Springall, D., Security Analysis of the Estonian Internet Voting System : <https://jhalderm.com/pub/papers/ivoting-ccs14.pdf> : 25 сент. 2016

ПЕРЕНОС ОБУЧЕНИЯ ДЛЯ СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ

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

Гаврилович Ю.А.

Кобяк И.П. – к.т.н., доцент

Современные глубокие нейронные сети обладают интересным феноменом: будучи натренированными на большом количестве изображений, они все на первых слоях «выучивают» одинаковые шаблоны-фильтры или по-другому одинаковые приметы/признаки – простые линии, градиенты, углы, цветные пятна округлой формы. Присутствуют эти фильтры настолько часто, что отсутствие их при обучении нейронных сетей на не синтезированных наборах изображений у опытных разработчиков вызывает подозрения в программных ошибках при разработке или обучении сети. Феномен проявляется на задачах обучения как с учителем, так и без.

Т.к. похоже, что нахождение/изучение таких одинаковых примет на начальных слоях нейронной сети не зависит от типа функции потерь (lossfunction) или от типа натурального (не синтетического) набора данных, то можно называть такие приметы *общими*. С другой стороны, мы знаем, что приметы, выученные на последних слоях натренированной сети, очень сильно зависят от используемого набора данных. Например, в сети с N-мерным выходным softmax слоем, которая была натренирована на задаче классификации с учителем, каждый выходной узел будет специфичным для соответственного класса. Таким образом, приметы, выученные на последних слоях, можно называть *специфическими*.

Наличие *общих* и *специфических* примет так же хорошо согласуется с идеей иерархического обучения представлению, которая отражается в принципе работы глубоких нейронных сетей.

Рис. 1 – градиация выученных фильтров от общих до специфических [1]

Почему нас интересует наличие *общих* или *специфических* примет? Потому, что если в процессе обучения нейронной сети можно выделить набор общих признаков, то далее мы можем использовать эти признаки для так называемого *переноса обучения* или *transfer learning* [2]. Для переноса обучения мы сначала тренируем *базовую* сеть на базовом наборе данных, затем мы переиспользуем выученные признаки, перенося их на вторую *целевую* сеть, после чего дообучиваем ее на целевых данных и целевой задаче. Такой процесс будет работать, если признаки для переноса являются *общими*, а не *специфическими*, т.е. подходят для решения как базовой, так и целевой задачи.

Если целевой набор данных значительно меньше базового, перенос обучения может быть мощным инструментом для обучения большой целевой сети без эффекта переобучения. Данная техника позволяет добиваться действительно впечатляющих результатов [3].

Типичный подход применения переноса обучения заключается в том, чтобы обучить базовую сеть, затем скопировать n первых слоев базовой сети и использовать их как n первых слоев целевой сети. Оставшиеся слои целевой сети инициализируются случайным образом и дотренировываются на целевой задаче. Так же в зависимости от ситуации можно либо «заморозить» n первых слоев в процессе дальнейшего дообучения, либо распространять ошибки обучения и на них. Такой процесс называется *fine-tuning*.

Список использованных источников:

1. Visualizing and Understanding Convolutional Networks [Электронный ресурс] : Cornell University - Электронные данные. – Режим доступа: <https://arxiv.org/abs/1311.2901>
2. Deep Learning of Representations for Unsupervised and Transfer Learning [Электронный ресурс] : Bengio et al./Workshop on Unsupervised and Transfer Learning - Электронные данные. – Режим доступа: <http://proceedings.mlr.press/v27/bengio12a/bengio12a.pdf>
3. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition Networks [Электронный ресурс] : UC

БЛОКЧЕЙН КАК ОСНОВА ЗАЩИЩЁННЫХ РАСПРЕДЕЛЕННЫХ БАЗ ДАННЫХ

*УО Белорусский Государственный Университет Информатики и Радиоэлектроники
г. Минск, Республика Беларусь*

Глоба А.А.

Ганжа В.А. – к.физ.-мат.н., доцент

В настоящее время в связи с ростом числа пользователей Интернет-сервисов растёт и потребность в принципиально новом способе хранения данных, защищённом от атак и легко масштабируемом по горизонтали. Отличным кандидатом на эту роль может стать технология 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. Для записи нового блока, необходимо последовательное считывание информации о старых блоках, при этом новый блок добавляется, если большая часть узлов в сети подтвердит, что в нем не нарушаются какие-либо заранее определенные правила. Все данные в блокчейн накапливаются и формируют постоянно дополняемую базу данных. С этой базы данных невозможно ничего удалить или провести замену/подмену блока, доступно только чтение.

Рис. 1 – Схема блоков в БД

Главные особенности технологии блокчейн:

1. Распределенность данных: нет одного сервера, хранящего информацию. При традиционном подходе в случае какой-либо атаки на сервер или отказа оборудования по какой-либо причине теряется доступ к данным и к сервису. В случае технологии блокчейн данные распределены в виде локальных копий на разных узлах – система будет сохранять работоспособность до тех пор, пока хотя бы один узел остается в сети.
2. Невозможность изменения уже существующих данных: в случае с обычными БД всегда есть возможность преднамеренно или случайно изменить существующие записи. Сфальсифицировать новую запись можно только в том случае, если взять под контроль более 50% (если не все) узлы сети, что физически практически невозможно.

Существуют и отрицательные черты БД на основе блокчейн:

1. Низкая скорость добавления новой записи в цепочку: механизм установления консенсуса предполагает участие всех (или большинства) узлов в сети для установления соответствия нового блока правилам системы, что вызывает определенную задержку;
2. Необходимость хранения локальных копий БД: в некоторых случаях БД может разрастаться до больших размеров. Например, цепочка Биткойна уже занимает размер в 100Gb, что является внушительным числом для среднестатистического ПК.

При этом нет никаких ограничений на формат хранимых данных. К примеру, Ethereum позволяет хранить в блокчейне не только транзакции, но и полноценные Тьюринг-полные программы, называемые смарт-контрактами, которые позволяют очень тонко настроить блокчейн на прикладную задачу, например, распределенный DNS-сервер.

Использованные источники:

1. Melanie Swan, Blockchain: Blueprint for a New Economy.
2. Jacob William, Blockchain: The Simple Guide To Everything You Need To Know.

ИЗВЛЕЧЕНИЕ ХАРАКТЕРНЫХ ПРИЗНАКОВ В ИСПОЛНЯЕМЫХ ФАЙЛАХ ДЛЯ ОБУЧЕНИЯ НЕЙРОННЫХ СЕТЕЙ ДЛЯ ОБНАРУЖЕНИЯ ВРЕДОНОСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

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

Гороховик Я.В

Воронов А.А. – к.т.н., доцент

Вредоносное программное обеспечение в общем случае представляется бинарными исполняемыми файлами, которые, как правило, регистрируются в системе, распространяются по ней, копируя себя, и оказывают на систему вредоносное воздействие. Современные антивирусные системы определяют вредоносное ПО, обладая знаниями о различных шаблонах поведения вирусов, однако определение новых, ранее не выявленных угроз, представляет определённую сложность для них. Огромное количество эвристических методов, используемых антивирусными решениями, потребляют значительное количество памяти и других ресурсов процессора. Эта нагрузка может быть преодолена путём обучения искусственных нейронных сетей, обученных на характерных признаках вредоносного ПО, содержащихся в самих исполняемых файлах. Исполняемые файлы формата PortableExecutable (PE) содержат множество полей, которые могут использоваться для предсказания поведения программы. Подход к сбору этих признаков из PE-файлов описан в данной работе.

В настоящее время одной из главных проблем информационной безопасности является выявление нового вредоносного ПО и новых угроз. Известные вирусы не представляют особой угрозы, так как могут быть легко детектированы сигнатурным анализом. Однако, выявление новых угроз требует намного более сложных эвристических подходов. Имеются следующие методы выявления таких угроз:

1. нахождение сходств между семействами вирусов. Как правило, данный метод основывается на различных алгоритмах машинного обучения, таких как Байесовская сеть или генетических алгоритмы;
2. реализация алгоритмов, эмулирующих методы принятия решений аналитика-человека – фактически, создание экспертной системы;
3. анализ файла в «песочнице». Для этого необходимо реализовать перехваты важных функций режимов пользователя и ядра. Этот метод предполагает изучения реального поведения файла в виртуальной среде[1].

Однако, каждый из перечисленных методов имеет свои ограничения:

1. первый метод может стоить значительного количества ошибок первого рода и занимать ресурсы ЭВМ, что приемлемо для тестовой среды, но не для домашнего компьютера;
2. второй метод наиболее удачен, так как предполагает изучение действительного поведения, как бы его проводил аналитик, однако, данный метод сложен и требует много памяти и ресурсов ЦП;
3. третий метод в значительной степени зависит от качества реализации соответствующего эмулятора ЦП и перехватов системных вызовов в драйверах и библиотеках. Он эффективен, однако стоит много ресурсов и денег.

В данной работе предлагается подход к извлечению характерных особенностей из исполняемых файлов для создания и обучения системы, которая принимает большое количество признаков PE-файла для определения его легитимности.

Предлагаемый метод включает сбор характерных признаков исполняемого файла следующими способами:

1. просмотр таблицы импорта для нахождения системных вызовов, наиболее часто используемых вредоносным ПО (AdjustTokenPrivileges, CreateRemoteThread, GetProcAddress, VirtualProtectEx, WriteProcessMemory и т.д.);
2. просмотр структуры PE в целом для нахождения наиболее важных полей.

Для получения особенностей таблицы импорта исполняемого файла создаётся список наиболее часто используемых вредоносным ПО функций в алфавитном порядке. Если в таблице импорта встречается такая функция, она кодируется единицей, если нет, то нулём. Полученная строка из единиц и нулей принимается за бинарное представление входного вектора нейронной сети. Длина входного вектора зависит от количества выбранных функций.

Выбор наиболее значимых полей PE структуры осуществлялся с помощью Критерия Фишера.

Критерий Фишера является статистическим критерием и используется для сравнения дисперсий двух вариационных рядов, то есть для определения значимых различий между групповыми средними в установке дисперсионного анализа[2]. Критерий Фишера вычисляется по следующей формуле:

$$R_i = \frac{|\mu_{i,p} - \mu_{i,n}|}{\sigma_{i,p} + \sigma_{i,n}}$$

В нашем случае, Критерий Фишера используется для сравнения дисперсий характеристик положительных («легитимных») и отрицательных (вредоносных) примеров.

Данные для обучения представлены вредоносными и обычными файлами. Каждый файл производит IAT-сигнатуру – характеристику таблицы импорта и характеристики, представленные полями PE-структуры.

К примеру, при выборе 96 наиболее популярных Win32 API функций, для вредоносного файла Trojan-Spy.Win32.Zbot.dsba сигнатура таблицы импорта представляет собой следующую строку: 01000000 00000101 00000000 00001000 00010001 00000100 00000100 00000001 00000100 00000000 11000000 11001110.

Данная строка кодируется в 12 байт, которые подаются на вход нейронной сети: 0x40, 0x05, 0x00, 0x08, 0x11, 0x04, 0x04, 0x01, 0x04, 0x00, 0xC0, 0xCE.

Для проверки, какие характеристики PE-заголовка наиболее значимы, были взяты 10 вредоносных и 10 обычных исполняемых файлов.

В таблицах 1 и 2 представлены значения полей PE-заголовков для «легитимных» файлов. Таблицы 3 и 4 содержат значения полей вредоносных файлов.

Таблица 1 – Значения полей PE-структуры первых пяти «легитимных» файлов

Feature Name	File 1	File 2	File 3	File 4	File 5
MajorLinkerVersion	2	8	2	2	10
MinorLinkerVersion	22	0	25	56	0
SizeOfInitializedData	472576	745472	18608128	107008	79872
SizeOfUnInitializedData	19456	0	0	3584	0
MajorOSVersion	4	4	5	4	5
MinorOSVersion	0	0	0	0	2
MajorImageVersion	1	1	0	1	0
MinorImageVersion	0	0	0	0	0
Checksum	695126	0	19630716	141070	225362
DLLCharacteristics	0	0	33088	320	33088

Таблица 2 – Значения полей PE-структуры следующих пяти «легитимных» файлов

Feature Name	File 1	File 2	File 3	File 4	File 5
MajorLinkerVersion	10	10	10	8	8
MinorLinkerVersion	10	10	0	0	0
SizeOfInitializedData	30208	1536	10752	2048	733184
SizeOfUnInitializedData	0	0	0	0	0
MajorOSVersion	6	6	5	4	4
MinorOSVersion	2	2	2	0	0
MajorImageVersion	6	6	0	0	0
MinorImageVersion	2	2	0	0	0
Checksum	161757	36371	38736	49763	76922
DLLCharacteristics	320	1344	320	34112	320

Таблица 3 - Значения полей PE-структуры первых пяти вредоносных файлов

Feature Name	File 1	File 2	File 3	File 4	File 5
MajorLinkerVersion	5	6	3	2	7
MinorLinkerVersion	2	0	0	25	10
SizeOfInitializedData	64000	121344	4608	944640	65536
SizeOfUnInitializedData	0	1024	0	0	0
MajorOSVersion	4	4	4	4	4
MinorOSVersion	0	0	0	0	0
MajorImageVersion	0	0	0	0	0
MinorImageVersion	0	0	0	0	0
Checksum	0	0	0	0	0
DLLCharacteristics	0	0	0	0	0

Таблица 4 - Значения полей PE-структуры следующих пяти вредоносных файлов

Feature Name	File 1	File 2	File 3	File 4	File 5
MajorLinkerVersion	2	5	8	2	6
MinorLinkerVersion	25	0	0	25	0
SizeOfInitializedData	12288	95232	135168	61440	212992
SizeOfUninitializedData	274432	0	0	651264	0
MajorOSVersion	1	4	4	4	4
MinorOSVersion	0	0	0	0	0
MajorImageVersion	0	0	0	0	0
MinorImageVersion	0	0	0	0	0
Checksum	0	0	716819	0	0
DLLCharacteristics	0	0	0	0	0

После извлечения признаков из двадцати исполняемых файлов были высчитаны среднеквадратичные и стандартные отклонения «легитимных» и вредоносных файлов. С помощью критерия фишера были вычислены ранги полей PE-заголовков исполняемых файлов.

В качестве входных данных нейронной сети были выбраны 7 полей с наибольшими рангами. Выбранные характерные признаки PE-заголовка исполняемых файлов и значения их рангов представлены в таблице 5.

Таблица 5 – Результаты применения Критерия Фишера (l – «легитимные файлы», m – вредоносные файлы)

Feature Name	Mean(l)	Mean(m)	Standard(l)	Standard(m)	Rank
MajorLinkerVersion	7	4.6	3.376	2.107	0.438
MinorLinkerVersion	12.3	8.7	17.123	11.055	0.128
SizeOfInitializedData	2078078.4	171724.8	2216994.091	264048.899	0.329
SizeOfUninitializedData	2304	92672	5816.329	203366.116	0.432
MajorOSVersion	4.7	3.7	0.781	0.9	0.595
MinorOSVersion	0.8	0	0.978	0	0.816
MajorImageVersion	1.5	0	2.291	0	0.655
MinorImageVersion	0.4	0	0.8	0	0.500
Checksum	2174862.3	71681.9	5824453.419	215045.7	0.348
DLLCharacteristics	10291.2	0	15153.838	0	0.678

Представленные признаки могут быть поданы на вход нейронной сети для обучения.

В данной работе были представлены два алгоритма выбора характерных признаков исполняемых файлов. Первый алгоритм используется для извлечения из исполняемого файла сигнатуры таблиц импорта. Второй алгоритм использует Критерий Фишера для нахождения наиболее важных полей PE-заголовка исполняемого файла.

Список использованных источников:

1. Koret, J. The Antivirus Hacker's Handbook / J. Koret, E. Bachaalany – John Wiley & Sons, Inc., Indianapolis, Indiana. – 2015. – P. 165-175.
2. Stopel D. Improving Worm Detection with Artificial Neural Networks through Feature Selection and Temporal Analysis Techniques / D. Stopel, Z. Boger, R. Moskovitch, Y. Shahar, Y. Elovici // Deutsche Telekom Laboratories at Ben-Gurion University, Be'er Sheva, Israel. – 2006.

МЕТОД МНОГОСТРУЙНОГО МОДЕЛИРОВАНИЯ В 3D ПЕЧАТИ

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

Дубовский А.Л.

Селезнёв И. Л. – к.т.н., доцент

Технологии объемной печати перестали быть только средством прототипирования и перешли в область промышленного производства. Трёхмерная печать, которая является процессом создания трёхмерных объектов практически любой геометрической формы на основе цифровой модели, быстро развивается в последние годы. Современные аддитивные технологии позволяют реализовать ресурсосберегающий, инновационный подход к проектированию и изготовлению деталей по сравнению с традиционными методами, используя различные материалы, такие как: акрил, нейлон, бетон, гидрогель, бумага, гипс, деревянное волокно, лед, металлический порошок.

Аддитивные технологии (AF — Additive Manufacturing) — это технологии послойного синтеза, обеспечивающими практически безотходное материало- и энергоэффективное производство многих видов изделий из металлических, полимерных и композитных материалов. С момента появления в середине 80-ых годов стереолитографии и технологии послойного наплавления, техника аддитивного производства непрерывно совершенствовалась [1]. По данным Wohlers Associates, современный мировой рынок аддитивных

технологий с 2016-го по 2017-й года увеличился на 17,4%, превысив в 2017 году 6 млрд. долларов. Согласно их анализу, среди ведущих производителей 3D принтеров, 51% печатают полимерами, 29,2% — металлами и полимерами, 19,8% — металлами.

Технология многоструйного моделирования (MJM — MultiJet Modeling) заключается в построении объекта с помощью печатной головки, оснащенной массивом сопел, из которых выделяется жидкий расходный материал. Печать осуществляется термопластиками, восками и фотополимерными смолами. В первых двух случаях материалы затвердевают за счет постепенного охлаждения. Для печати фотополимерной смолой используется ультрафиолетовый излучатель, способствующий быстрой полимеризации. Так как печать выполняется послойно, то для нависающих частей требуется поддерживающий материал, для этого используется вспомогательный материал, например, воск, который впоследствии удаляется.

Рис. 1 — Процесс печати

Печатающая головка формирует слои материала, постепенно выстраивая из них объект, выдавливая жидкий материал на платформу. Материал сразу же затвердевает благодаря УФ лампам. Одновременно можно подавать несколько материалов. Для выступов при использовании данной технологии печати необходимы несущие конструкции, которые обычно изготавливаются из другого легкоудаляемого материала.

Основным преимуществом данной технологии является возможность использования нескольких материалов, позволяя создавать разноцветные изделия с сочетанием материалов с различными свойствами. Технология может обеспечить высокую точность, малую толщину наносимого слоя, высокое разрешение печати.

Недостатки:

- ограниченное количество восковых материалов, доступных для струйной печати;
- хрупкость деталей из-за типа материала;
- процесс создания медленнее в сравнении с другими технологиями.

Данная технология обладает значительным преимуществом, возможностью создавать изделия с гетерогенной структурой. Технология MJM используется в различных отраслях, требующих создания высокоточных прототипов и готовых изделий для стоматологии, ювелирного дела, разработки электронных компонентов. Таким образом, данный метод многоструйного моделирования обязательно занимает свою нишу на рынке.

Список использованных источников:

1. Аддитивные технологии, материалы и конструкции : материалы науч.-техн. конф. (Гродно, 5-6 окт. 2016 г.) / Нац. акад. наук Беларуси [и др.] ; редкол.: А.И. Свириденок (гл. ред.) [и др.]. – Гродно : ГрГУ, 2016. – 274 с.
2. US 5204055 A - Three-dimensional Printing Techniques - The Lens - Free Patent Search & Scholarly Journal Articles [Электронный ресурс]. — Режим доступа: https://www.lens.org/lens/patent/US_5204055_A
3. Wohlers Report 2017 Shows Vibrant New Business Activity in 3D Printing [Электронный ресурс]. — Режим доступа: <http://wohlersassociates.com/press72.html>
4. Additively [Электронный ресурс]. — Режим доступа: <https://www.additively.com>

ПРОГРАММНОЕ СРЕДСТВО ДЛЯ КЛАССИФИКАЦИИ ОБИТАТЕЛЕЙ МОРСКИХ ГЛУБИН

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

Евсаев П.В.

Фролов И.И. – к.т.н., доцент

Автоматизация методов поиска новых видов морских обитателей является одним из приоритетных направлений океанологии и представляется своевременным и целесообразным. Поиск новых видов - трудоемкий процесс, требующий погружения исследователей на морское дно, что связано с риском для жизни. Создание программного средства, способного по фотографии определить, является ли представитель фауны ранее неизученным видом - актуальная проблема, требующая решения.

На первом этапе алгоритма выполняется предобработка входного изображения. Исследователи фотографируют ОМГ крупным планом в высоком разрешении, что существенно упрощает процесс обработки изображения. С учетом упомянутых особенностей съемки, на этапе отделения ОМГ от фона хороший результат показывает алгоритм адаптивной бинаризации по пороговому значению, после которого последовательно применяются операции математической морфологии «Замыкание» и «Размыкание». Операция «Замыкание» объединяет мелкие детали ОМГ и заполняет в нем дыры, а операция «Размыкание» убирает небольшие группы белых пикселей, которые не относятся к объекту.

На полученном бинарном изображении необходимо выделить область интереса для последующего анализа. Для поиска связанных областей выполняется свертка бинарного изображения с 3-компонентной маской (центральный, верхний + левый пиксели), после чего строится граф, к которому применяется алгоритм «Поиск в ширину» для определения эквивалентных областей. Предполагается, что представитель фауны на изображении один, поэтому интересующей нас областью будет являться область с наибольшей площадью. После удаления всех остальных областей с бинарного изображения, оно маской накладывается на исходный снимок, оставляя цвет только пикселям, принадлежащим ОМГ.

Задача выбора параметров для классификации ОМГ является нетривиальной, поскольку глубоководные обитатели постоянно двигаются, меняя форму. В связи с этим, желательно определить признаки, которые будут инвариантны к изменению формы ОМГ, а также ракурса съемки.

Визитной карточкой ОМГ является его уникальный окрас. Задача выбора доминирующих оттенков сводится к разбиению всего набора цветов на несколько кластеров. Для решения задачи кластеризации используется метод К-средних. Центры кластеров и будут являться основной палитрой изображения. После определения палитры, необходимо преобразовать ее в инвариантное к уровню освещенности представление, вычислив, например, разности цветовых компонент составляющих ее цветов, либо осуществив их перевод в цветовую систему LAB.

Вектор, по которому будет производится классификация, получается путем объединения инвариантного представления окраса ОМГ с его статистически-значимыми геометрическими признаками - удлинненностью и компактностью.

Реализованное программное средство позволяет определить факт присутствия ОМГ Северного моря на изображении и классифицировать его как представителя одного из следующих видов: *aequorea*, *aureliaLimbata*, *seaAngel*, *pegeaConfoederata*, *other*

Список использованных источников:

1. Старовойтов В.В. Цифровые изображения: от получения до обработки / В.В. Старовойтов, Ю.И. Голуб – Минск: ОИПИ НАН Беларуси, 2014, ISBN 978-985-6744-80-1.
2. David A. Forsyth, Jean Ponce. Computer Vision: A Modern Approach, Prentice Hall: 2002, ISBN 978-013-0851-98-7.

РАСПОЗНАВАНИЕ КАТЕГОРИИ ПОМЕЩЕНИЯ НА ОСНОВЕ ОНТОЛОГИЧЕСКОГО АНАЛИЗА

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

Игнатъев Е.Д., Иванов Н.Н.

Иванов Н.Н – к.ф-м.н., доцент

Для семантического анализа сцен, представленных в виде изображения, текста и другими средствами в настоящее время широко используются онтологии. В сообщении предлагается применить семантические сети для распознавания назначения помещения (кабинет, столовая, учебная аудитория, и пр.). Рассматривается ситуация, когда робот, передвигаясь в заданных помещениях, выполняет фото- или видео-съемку статичных сцен, передавая изображения на компьютер. Используя библиотеку OpenC, приложение сегментирует сцены. Затем по заданным семантическим сетям выполняется распознавание объектов, по совокупности которых выполняется классификация помещения.

Ставится задача распознавания вида помещения. Применяются аппаратные средства: робот с видеокамерой, свободно передвигающийся по заданным помещениям. Он оснащен видеокамерой, изображение с которой передается в память компьютера. Предварительно вручную создаются семантические сети предметов мебели и других объектов, которые характеризуют назначение комнаты. Кроме того, строятся семантические сети помещения различных категорий (коридор, гостиная, кухня и т.п.).

Подобную задачу ставил создатель теории фреймов М. Минский [1], он начинал с изучения процессов узнавания предмета человеком. Фрейм – это аналог семантической сети. Вначале М. Минский построил фреймы для классификации простых геометрических трехмерных предметов: куба, пирамиды, параллелепипеда. После этого он усложнил задачу, добавив перемещение предметов по комнате и проверял гипотезу: тали эта комната. Кроме того, при фиксации сцены фотокамерой возможно искажение их линзами объектива. Съёмка сцены при движении наблюдателя также усложняла задачу.

Для решения нашей задачи на компьютере установлено приложение, которое сегментирует полученные с камеры изображения. Оно использует библиотеку модулей компьютерного зрения OpenCV, в которой, в частности, представлены модули обработки изображений на языке Python. Так раздел Image Processing этой библиотеки содержит полезные для сегментации модули, такие как Geometric Transformations of Images, CannyEdgeDetection и многие другие [2].

Семантическая сеть предметов мебели и других объектов, характеризующих помещение, строится в виде ориентированного графа, узлы которого представляют объекты, это, например, конструктивные элементы мебели (сиденье, ножка, спинка и пр.) и их атрибуты (расположение относительно других элементов, форма, допустимые размеры и пр.) [3]. Семантическая сеть помещения также является ориентированным графом, описывающим объекты, присущие данной категории комнаты. Все построенные семантические сети хранятся в памяти компьютера.

Компьютерное приложение сегментирует изображение и методом полного перебора проверяет выделенные объекты на их соответствие семантическим сетям предметов. После проверки всех предметов, находящихся в комнате, приложение выполняет полный перебор семантических сетей категорий комнат. Если найдено соответствие сети с набором предметов, то распознавание комнаты заканчивается и приложение переходит к другой комнате.

Возможное практическое использование созданной системы классификации – это ревизия складских помещений с распознаванием предметов и вычислением их количества [4]. Можно фиксировать номера госрегистрации автомобилей, использующих стоянку и записывать моменты их прибытия и убытия. В помещениях, неблагоприятных для человека, возможна как проверка воздуха (температура, влажность, химический состав) так и составление списка предметов в помещении и фиксация изменения их положения. В помещениях с недостатками освещенности можно для распознавания животных использовать инфракрасные видеокамеры.

В развитии предлагаемого алгоритма предполагается изучить возможность автоматического построения семантических сетей для помещений. Для этого будет использоваться нейронная сеть с обучающей выборкой изображений помещений. Автоматическое построение семантических сетей даже таких простых объектов, как предметы мебели, представляется проблематичным ввиду того, что трудно автоматически разделить конструкцию на составляющие ее элементы. Динамические сцены с наличием людей и животных, которые затевают предметы в комнате еще более усложняют задачу.

Требование о решении задачи в реальном времени вызывает необходимость оптимизировать программное обеспечение и применять более мощные вычислительные средства. Кроме того, в этом случае может возникнуть проблема корректировки модулей, заимствованных из библиотеки OpenCV, то есть потребуются создавать собственные модули сегментации сцены.

Список использованных источников:

1. Минский М. Фреймы для представления знаний // М.: Энергия, 1979, С.338.
2. Pronobis A., Martinez M.O., Caputo B., Jensfelt P. Multi-modal semantic place classification // Int. J. Robot. Res. 2010, 29, 298–320.
3. Astua C., Barber R., Crespo J., Jardon A. Object detection techniques applied on mobile robot semantic navigation // Sensors 2014, 14(4), 6734–6757.
4. Ekvall S., Kragic D., Jensfelt P. Object detection and mapping for service robot tasks // Robotica, 2007, 25(2), 175-187.

ОБЗОР ОСНОВНЫХ АРХИТЕКТУР СЕМАНТИЧЕСКОГО СЕГМЕНТИРОВАНИЯ

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

Куница Е.Ю.

Лукашевич М.М. – к.т.н., доцент

Сегментация, в обработке изображений — это разделение изображения на множество пикселей (суперпикселей). Обычно сегментация используется для выделения объектов и их границ.

При семантической сегментации каждому множеству пикселей ставится в соответствие определённый

класс объектов. Семантическая сегментация является высокоуровневой задачей обработки изображений. Она является даже более сложной, чем задача классификации изображений и поиска объектов, что обусловлено не только необходимостью определения классов объектов, но и выявления их структуры, правильного выделения частей объектов на изображении.

Одним из популярных начальных подходов к глубокому обучению была классификация patch-based, где каждый пиксель отдельно классифицировался на классы с использованием патча изображения вокруг него. Основная причина использования патчей заключалась в том, что в классификационных сетях обычно есть полностью связанные слои и поэтому требуются изображения фиксированного размера.

В 2014 началась популяризация использования FCN нейронных сетей. Так как в данной архитектуре отсутствуют полносвязные слои, это позволило создавать карты сегментации для изображения любого размера. Так же FCN выполняется намного быстрее по сравнению с подходом к классификации патчей. Основная проблема данного подхода является низкое разрешение на выходе.

Однако семантическая сегментация требует точного выравнивания карт классов. Для решения этой проблемы используются архитектуры типа кодер-декодер. Наиболее часто используемой нейронной сетью данной архитектуры является сеть U-Net. Архитектура U-Net сети представляет собой последовательность слоёв свёртка+пулинг, которые сначала уменьшают пространственное разрешение картинки, а потом увеличивают его, предварительно объединив с данными картинки и пропустив через другие слои свёртки. Таким образом, сеть выполняет роль своеобразного фильтра.

Наиболее важными наборами данных для семантической сегментации.

— VOC2012;

— COCO.

Они предоставляют наборы изображений с эталонными масками.

Однако полученные результаты необходимо обработать так, чтобы получить метрики, по которым можно сказать, насколько хорошо выполнена сегментация, определить какая из архитектур подходит для преследуемых целей.

— – точность пикселей;

— – средняя точность;

— – средняя IoU (Intersection over Union);

— – частотно-взвешенное IoU.

– количество классов включённые в карту сегментации, – количество пикселей класса i определённых как класс j , – общее количество пикселей класса i .

Наиболее часто используемые фреймворки и библиотеки для разработки приложений машинного обучения, а в частности семантической сегментации являются:

1. Scikit-learn использует Python для математической и научной работы. scikit-learn включает инструменты для многих стандартных задач машинного обучения (таких как кластеризация, классификация, регрессия и т. д.). И так как scikit-learn разрабатывается большим сообществом разработчиков и экспертов по машинному обучению, перспективные новые методы, как правило, включаются в довольно короткий срок.
2. TensorFlow применяется для создания графов потоков данных. Узлы графа представляют собой математические операции, а ребра графа представляют собой многомерные массивы данных (тензоры), которые связаны между ними. Эта гибкая архитектура позволяет развернуть вычисления на один или несколько CPU или GPU на десктопе, сервере или мобильном устройстве без перезаписи кода. TensorFlow также включает в себя TensorBoard – набор инструментов для визуализации данных. Графы можно создавать при помощи Python и C++.
3. Keras - это высокоуровневый API для нейронных сетей, написанный на Python и способный работать поверх TensorFlow, CNTK или Theano. Он был разработан с упором на возможность быстрого экспериментирования. Keras позволяет легко и быстро создавать прототипы (благодаря удобству, модульности и расширяемости). Поддерживает как сверточные сети, так и повторяющиеся сети, а также комбинации этих двух.

Список использованных источников:

1. Jonathan Long, Evan Shelhamer, Trevor Darrell, Fully Convolutional Networks for Semantic Segmentation, 2015.
2. Fei-Fei Li, Andrej Karpathy, Justin Johnson, Segmentation and Attention, Lecture 13, 2016.
3. Sasank Chilamkurthy, Guide to Semantic Segmentation with Deep Learning, 2017.

ОБЗОР СИСТЕМ ДЛЯ ТЕЛЕМЕДИЦИНЫ

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

Макоед В.Н.

Лукашевич М.М. – к.т.н., доцент

В 2017 году Министерство здравоохранения Республики Беларуси анонсировало планы по объединению всех учреждений здравоохранения в единую систему телемедицинского консультирования. По факту единое программное средство для осуществления этих планов ещё не разработано, но все поликлиники и больницы оснащены необходимым оборудованием и имеют выход в интернет. Вместе с тем, в других странах такие системы успешно применяются среди большого числа специалистов и пациентов[1].

Телемедицинские сервисы предназначены для удалённого оказания медицинских услуг. Их использование решает следующие задачи:

1. онлайн-консультации между врачами и пациентами;
2. оказание услуг для пациентов с ограниченными возможностями передвижения либо проживающих на большом расстоянии от медицинских учреждений;
3. диагностика специалистом в чрезвычайных ситуациях (например, в машине скорой помощи);
4. проведение клинических исследований научными работниками и фармацевтическими компаниями [2].

Система для телемедицины реализуется на следующих уровнях:

1. транспортный уровень;
2. уровень медицинского оборудования;
3. уровень программного обеспечения.

Одна из задач программных средств для телемедицины – передача текста, визуальной информации и звуков. К текстовой информации относятся, к примеру, результаты ЭКГ, лабораторные измерения, истории болезни пациентов. К визуальной информации относятся, например, рентгенограммы или видеозаписи операций. Примеры аудиосигналов – голосовые сообщения, звукозапись работы лёгких.

Передача такой информации ведётся двумя методами:

1. передача в реальном времени;
2. отложенная передача информации.

На транспортном уровне подобных программных средств используются такие протоколы, как P2P, HTTPS, DICOM, e-mail.

Для предоставления услуг удалённой медицины необходимо оснащение персонала и пациента устройствами для телекоммуникации, а также организация сервера, предназначенного для размещения базы данных и приложения. Для более стабильного соединения применяется спутниковая связь. Ниже приведён вариант организации инфраструктуры на примере системы TeLeCare [3].

Рис. 1 – Инфраструктура системы TeLeCare

Как видно из рисунка, организация телемедицинской системы сводится к разработке и развёртыванию приложения на клиент-серверной архитектуре. Однако к такой системе выдвигаются повышенные требования совместимости оборудования и стабильности соединения.

Список использованных источников:

1. Врач всегда готов [Электронный ресурс]. – Режим доступа: <http://www.belta.by/society/view/vse-organizatsii-zdravoohranenija-belarusi-v-2017-godu-podkljuchatsja-k-sisteme-telemeditsinskogo-245731-2017/>
2. The Best Telemedicine Software For 2018 [Электронный ресурс] – Режим доступа: <https://www.business.com/categories/best-telemedicine-software/>
3. Use of telemedicine-based [Электронный ресурс]. – Режим доступа: <https://www.dovepress.com/use-of-telemedicine-based-care-for-the-aging-and-elderly-promises-and-peer-reviewed-fulltext-article-SHTT>

VERT.X КАК НОВОЕ СРЕДСТВО РАЗРАБОТКИ МИКРОСЕРВИСОВ

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

Садовников И.В.

Искра Н.А. – старший преподаватель

Vert.x - это асинхронный, событийно-ориентированный фреймворк работающий на JVM. Данный фреймворк показывает высокую производительность, легко конфигурируется и поддерживает асинхронную модель программирования. Первый релиз Vert.x состоялся в 2012 году [1].

Основными преимуществами Vert.x являются:

1. мультиязычность - компоненты приложения могут быть разработаны на Java, JavaScript, Groovy, Ruby, Ceylon, Scala или Kotlin;
2. асинхронность - очень простая схема асинхронного взаимодействия без блокировок;
3. параллелизм - простая модель параллельного выполнения задач, основанная на event-loops [2];
4. распределенная шина событий - данная шина позволяет распределять нагрузку между всеми вертикалями [3] автоматически;
5. модульная структура - модули позволяют использовать только те компоненты, что необходимы для проекта;
6. Open Source - полностью открытый исходный код продукта, активно развиваемого сообществом.

Vert.x поддерживает механизмы кластеризации прямо из коробки (out of the box). По умолчанию для кластеризации используется Hazelcast, но также есть возможность использовать такие инструменты как Infinispan, Apache Ignite, Apache Zookeeper [4].

Пример команды запускающей вертикаль в кластере: `vertx run MyVerticle.java -cluster`

Рис 1. – Схема работы вертикалей Vert.x в кластере[3]

Vert.x поддерживает механизм отказоустойчивости своих компонентов, работающих в одном процессе vertx или в одном кластере, таким образом если часть компонентов выходит из строя, другая часть берет их функциональность на себя, пока решается проблема с остановившимися компонентами [5]. Для того чтобы воспользоваться данной функцией, в команду строку надо всего лишь добавить параметр запуска -ha. При этом запуск будет автоматически подниматься в кластере, поэтому необходимости добавлять параметр -cluster нет.

Пример команды запускающей вертикаль с механизмом отказоустойчивости:

`vertxrunMyVerticle.java-ha`

Список использованных источников:

1. Официальный сайт Vert.x [Электронный ресурс]. - Режим доступа: <https://vertx.io/>
2. Официальная документация основного модуля Vert.x [Электронный ресурс]. - Режим доступа: <https://vertx.io/docs/vertx-core/java/>
3. Официальная документация о Verticle [Электронный ресурс]. - Режим доступа: https://vertx.io/docs/vertx-core/java/#_verticles
4. Julien Ponge, Thomas Sigismont, Julien Viet, A gentle guide to asynchronous programming with Eclipse Vertx for Java developers.
5. Clement Escoffier. Building Reactive Microservices in Java.

МАТЕМАТИЧЕСКИЙ КАЛЬКУЛЯТОР

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

Ковбаса Г.А.

Ковальчук А.М. – старший преподаватель

Проводя обзор аналогов, наиболее широко распространенными программными продуктами, занимающимися символическими и численными вычислениями, являются СКА – системы компьютерной алгебры, такие как AXIOM, Maxima, Reduce и более простые Derive, MathCAD, Maple, Mathematica-2.1, WolframAlpha.

Преимущество СКА – в полноте представленных операций, возможности составления пользователем своих собственных алгоритмов большой вычислительной мощности, подходящей для выполнения расчетов в высокоточных исследованиях и широкой представленности на рынке.

С другой стороны, недостатками таких программ являются их высокая стоимость, необходимость соединения с интернетом, повышенные требования к системе компьютера (процессор: Intel Pentium Dual-Core или эквивалентный; дисковое пространство: 14 GB; оперативная память (RAM): рекомендовано 2 GB+ - для Mathematica, процессор: частота 1 GHz или выше; дисковое пространство: 4 GB; оперативная память (RAM): 4

GB+ - для Maple 2018), длительное усвоение навыков работы и т.п.

Системы компьютерной алгебры, в большинстве своем, базируются на древовидных структурах данных, традиционно характерных для языков, реализующих парадигму функционального программирования, каким, к примеру, является Lisp. В моей программе на основе введенного пользователем выражения, путем последовательного считывания элементов строки, создается объект класса дерева выражений. Данное дерево представляет собой усовершенствованное В-дерево (сбалансированное, сильноветвистое дерево), узлы которого могут быть двух видов: узлы суммы и узлы произведения (рисунок 1).

Корень дерева представляет собой сумму всех последующих операций. Данное представление обосновано тем, что любую математическую операцию можно представить в виде суперпозиции этих двух операций над некоторыми величинами. К примеру, операция вычитания является операцией сложения двух величин, одна из которых имеет отрицательный коэффициент.

Поиск в дереве и его анализ на зависимость функций от некоторой другой функции или переменной проводятся за меньшее время, чем в стандартном бинарном дереве поиска, т.к. доступ к любому элементу суперпозиции выражения можно получить из корневого узла, и у каждого узла-потомка может быть несколько узлов-предков. Таким образом, алгоритмы интегрирования, дифференцирования и вычисления значения будут производиться быстрее.

Рисунок 1 – Схема дерева выражения

Поэтому, в сравнении с рассмотренными аналогами, достоинствами моей разработки являются низкие системные требования (процессор: 0.233 GHz или выше; свободное пространство на диске: 100 MB; оперативная память (RAM): 64 МБ RAM или выше), портативность, отсутствие необходимости подключения к сети интернет, наличие базовых возможностей символьных вычислений, достаточных для использования в качестве вспомогательного средства для учебы и домашнего пользования.

Программа имеет удобный пользовательский, с необходимыми пунктами меню, интерфейс на базе Windowsforms. В программе выполняются следующие действия: работа с дифференциалами, основными пределами и интегралами, а также построение графиков функций одной переменной. Предусмотрена операция отмены последних действий и сохранение промежуточных пользовательских вычислений.

Для реализации приложения используется объектно-ориентированный язык программирования C++, среда разработки MicrosoftVisualStudio 2015, ОС Windows7.

Список использованных источников:

1. Страуструп, Б. Программирование: принципы и практика использования C++ / Б. Страуструп : пер. с англ. — М. : ООО "И.Д. Вильямс", 2011. — 1248 с.: ил.
2. Тан, К.Ш. Символьный C++: Введение в компьютерную алгебру с использованием объектно-ориентированного программирования / К. Тан, В. Стиб, И. Харди : пер. со 2-го англ. изд. — М.: Мир, 2001. —622с, ил.
3. Дэвенпорт, Дж. Компьютерная алгебра / Дж. Дэвенпорт, И. Сирэ, Э. Турнье. - М.: Мир, 1991. - 352 с.
4. Wikipedia[электронный ресурс]– Режим доступа https://ru.wikipedia.org/wiki/К-мерное_дерево – Дата доступа 05.11.2017
5. Wikipedia[электронный ресурс]– Режим доступа [https://msdn.microsoft.com/ru-ru/library/system.string\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/system.string(v=vs.110).aspx)– Дата доступа 05.11.2017
6. Wikipedia[электронный ресурс].– Режим доступа [https://msdn.microsoft.com/ru-ru/library/6sh2ey19\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/6sh2ey19(v=vs.110).aspx)– Дата доступа 05.11.2017

«УМНАЯ» ТЕПЛИЦА НА БАЗЕ КОНТРОЛЛЕРА ARDUINO UNO

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

Гук В.А.

Луцук Ю.А. – к.т.н., доцент

Использование микроконтроллерных устройств в повседневной жизни позволяет автоматизировать многие повседневные задачи. К числу таких задач относится контроль за состоянием параметров внутреннего микроклимата теплицы.

Известно, что рост растений и качество плодов зависит от таких показателей, как влажность и температура воздуха и почвы. Для более точного контроля за факторами, влияющими на условия окружающей среды (например, в теплице) в работе была предпринята попытка создать устройство, которое без вмешательства человека следит за климатическим состоянием в теплице. К таким факторам относятся: температура и влажность воздуха и почвы.

В ходе выполнения данной работы был собран прототип теплицы. Контроль за указанными выше факторами в ней осуществляется программно с помощью микроконтроллера семейства Atmega на базе которого реализована плата Arduino UNO, а также датчиков DHT-11 (DHT-22, BMP180), LM35 (DS18B20) для определения влажности и температуры воздуха и почвы. Используемые датчики представлены на рисунке 1:

Рис. 1 - Датчики DHT-11 и LM35

Для визуального контроля регистрируемых датчиками показаний используется, семи сегментный индикатор. Модификация режима работы устройства осуществляется с помощью кнопок и отображается на LCD-дисплее.

В программном модуле предусмотрена возможность подключения различных датчиков температуры (влажности). Это расширяет возможности устройства и позволяет получать более точную информацию. Ниже приведен фрагмент кода для получения информации с 3 видов датчиков.

```
switch (sens_temp) { // Анализ с какого датчика получена температура
  case 1: tft.print(temp,1); break; // Вывод значение temp из датчика DHT**
  case 2: tft.print(Temperature/10.,1); break; // Вывод значение Temperature из датчика BMP180
  case 3: tft.print(Celsius/16.0,1); // Вывод значения температуры из датчика DS18B20
}
```

Реализация данного проекта была выполнена в среде программирования Arduino IDE на языке Си. Управляющая программа занимает 27% оперативной памяти контроллера, в связи с этим могут быть расширены функциональные возможности разработанного устройства (например, добавлены датчики позволяющие контролировать уровень освещенности и др.). А также планируется подключить к устройству накопитель на SD флеш-карте для регистрации и хранения информации о климатических показателях.

Рис. 2 – Макет резервуара

В разработанном устройстве предусмотрен режим изменение температуры почвы путем подачи нагретой жидкости в резервуар, расположенный под почвой, показан на рисунке 2.

А также возможна изменение температуры воздуха путем подачи потока теплого воздуха. Устройство является актуальным из-за своей дешевизны и простоты сборки и использования.

ВИЗУАЛИЗАЦИЯ АЛГОРИТМОВ УМНОЖЕНИЯ НА ARDUINO

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

Рапичук А.С., Хурсов П.С.

Луцки Ю. А. – канд. техн. наук, доцент

В рамках дисциплины «АИЛО ВТ» одним из вопросов является изучение машинных алгоритмов выполнения арифметических операций. При изучении алгоритмов умножения выполняются действия в двоичной системе. Это достаточно трудоемкий процесс, и визуализатор алгоритмов умножения позволяет упростить эту задачу, сделав процесс более наглядным.

Целью данной работы явилась разработка устройства, предназначенного для визуализации каждого такта умножения положительных целых чисел в двоичной системе. В разработанном устройстве предусмотрена возможность изменения входных данных (сомножителей) и вывод результатов (промежуточных и конечного) работы устройства.

При выполнении данного проекта в качестве основных частей были выбраны элементы, изображенные на рисунке 1:

Рис. 1 - Компоненты, составляющие основу устройства

Первоначально устройство включало плату Arduino Uno с подключенными к ней дисплеем и кнопкой (смена такта умножения). Первым был реализован один алгоритм умножения (школьный). Устройство показало себя с лучшей стороны. В процессе работы над устройством был дописана программа для микроконтроллера, которая позволила реализовать все 4 известных алгоритма умножения. Для развития интерфейса были добавлены вспомогательные кнопки (перебор алгоритмов, выбор сомножителя для его увеличения или уменьшения). В ходе работы над проектом было выявлено что устройство может быть уменьшено за счет использования более компактной Arduino Nano, ресурсов которой (аппаратных и программных) достаточно для выполнения поставленной перед устройством задачи.

Для реализации программного обеспечения на языке Си использована среда программирования Arduino IDE. Программа занимает 25% памяти устройства, что позволяет в дальнейшем расширять функциональные возможности разработанного устройства.

Для придания разработанному устройству законченного вида был спроектирован корпус в программе AutoCad. Созданная модель корпуса была распечатана на 3D принтере.

Работа данного устройства была опробована на дисциплине «АИЛО ВТ».

В качестве достоинств разработанного устройства можно выделить следующие:

- наглядный принцип работы каждого из алгоритмов;
- программный код устройства является открытым, что позволяет модифицировать и модернизировать

его.

ОРГАНИЗАЦИЯ УПРАВЛЕНИЯ ДОКУМЕНТАЦИОННЫМИ ПРОЦЕССАМИ КАФЕДРЫ В СИСТЕМЕ ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА «ЭТЛАС»

*Белорусская государственная академия связи
г. Минск, Республика Беларусь*

Ханько А.В.

Воронов А.А. – к.т.н., доцент

В государственных учреждениях включая учреждения образования, документы играют важную роль: являются инструментом управления, обеспечивают взаимодействие между структурными подразделениями учебного заведения, кафедрами и преподавателями. Создание, движение, обработка, распространение и хранение документов выполняется ежедневно, поэтому от эффективности документооборота зависит грамотно выстроенный процесс обмена информацией, что благоприятно отразится на учреждении образования, и улучшит его работу.

Результатом стремительного развития технологий документооборота стало появление новой концепции государственного управления – концепции электронного правительства, ориентированной на повышение эффективности и комфорта работы в условиях глобального Информационного общества [1]. Белорусские учреждения образования внедряют системы электронного документооборота, оценив преимущества новых современных технологий работы с документами.

Ежедневно при работе кафедр формируется значительный массив нормативной документации, сопровождающий учебный процесс. Наиболее рациональным решением для организации процесса документооборота является система электронного документооборота (СЭД).

Для управления документационными процессами на кафедре ИКТ «Белорусской государственной академии связи» выбрана система «ЭТЛАС» (рис. 1).

СЭД «ЭТЛАС» обеспечивает процесс создания, управления доступом и распространения электронных документов в компьютерных сетях, обеспечивает контроль над потоками документов. Предоставляет возможность хранения писем, приказов, положений, электронных журналов по группам, курсовых работ (проектов) и т.д.

Рисунок 1 – Диалоговое окно системы электронного документооборота кафедры ИКТ

Система «ЭТЛАС» не требует высокой производительности клиентских компьютеров. Отсутствует необходимость в приобретении дополнительного дорогостоящего программного обеспечения и серверного оборудования. Использование СЭД повышает эффективность производительности труда и снижает затраты времени на обработку документа.

Список использованных источников:

1. Развитие электронного правительства в Республике Беларусь. [Электронный ресурс]. – Режим доступа: <http://elibr.bsu.by/handle/123456789/29424>.
2. Система электронного документооборота «ЭТЛАС». [Электронный ресурс]. – Ресурс просмотра: <http://www.atlas-soft.ru/>.

СТАТИСТИЧЕСКИЕ МЕТОДЫ БИОМЕТРИЧЕСКОЙ ИДЕНТИФИКАЦИИ IRIS

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

Харкевич И.С.

Сечко Г.В. – к.т.н., доцент

В данной статье рассмотрены статические методы биометрической идентификации по радужной оболочке глаза FAR и FRR, сравнительные характеристики скорости обработки изображений, а также приведен анализ оценки вероятностей ложного совпадения работы iris-системы пункта контроля доступа.

Биометрические данные — это физиологические и биологические особенности человека, на основании которых можно установить его личность. В качестве двух основных характеристик любой биометрической системы для проверки статистических гипотез используются ошибки первого и второго рода. В задачах

радиолокационного обнаружения, как правило, их называют «ложная тревога» и «пропуск цели», а в биометрии — *FAR* (False Acceptance Rate) и *FRR* (False Rejection Rate).

Ложная скорость приема или *FAR* является мерой вероятности того, что биометрическая система безопасности будет неправильно принимать попытку доступа неавторизованным пользователем. Системный *FAR* обычно указывается как отношение количества ложных приемников, деленное на количество попыток идентификации.

Ложная скорость распознавания или *FRR* является мерой вероятности того, что биометрическая система безопасности будет неправильно отклонять попытку доступа авторизованного пользователя. *FRR* системы обычно указывается как отношение количества ложных распознаваний, деленное на количество попыток идентификации.

При биометрической идентификации и аутентификации личности человека используются уникальные признаки и особенности радужной оболочки человеческого глаза. Радужная оболочка (радужка, РОГ, iris) – это тонкая, подвижная, светонепроницаемая диафрагма глаза со зрачковым отверстием в центре, которая разделяет пространство между роговицей и хрусталиком. Формируется к восьмому месяцу эмбрионального развития и окончательно стабилизируется в возрасте около двух лет. Узор трабекулярной сети практически не изменяется в течение жизни, кроме как в результате серьезных травм или хирургического вмешательства.

Метод распознавания по РОГ является одним из наиболее точных среди всех биометрических методов. Благодаря сложности рисунка iris удается отобрать порядка 250 точек, которые обеспечивают высокую степень надежности аутентификации (для сравнения при распознавании по лицу используют около 80 точек).

Задача биометрической системы идентификации личности по iris состоит в следующем: получение детального изображения человеческого глаза с устройства, первичная обработка рисунка и передача алгоритму, производящему сравнение записей (изображения) с записями в базе данных, с дальнейшей передачей команды о решении допуска к исполнительному устройству.

В современных системах время первичной обработки составляет примерно 300-500 мс, при скорости сравнения полученного изображения с базой в 50000-150000 в секунду на обычном компьютере. Следовательно, оптимизация поиска по iris при использовании специализированных вычислителей и алгоритмов, не накладывают ограничений как в больших организациях, так и при идентификации человека среди жителей целой страны.

Для анализа вероятностей статистических характеристик *FAR* и *FRR* оцениваются количество ложных совпадений, при использовании системы идентификации на проходной организации с численностью персонала равную N человек. Допустим, что вероятность ложных совпадений для записей из базы данных из Miris-сканов равна $FAR \times N$, и каждый рабочий день через пункт контроля доступа проходит также около N человек. Тогда вероятность появления ошибки за день будет равняться $FAR \times (N \times N)$. Очевидно, что количество ложных совпадений за единицу времени может сильно варьироваться в зависимости от алгоритма биометрической системы идентификации, но если принять допустимым одну ошибку в течение рабочего дня, то [1]:

Тогда получим, что стабильная работа iris-системы при характерном значении $FAR=0.00001\%$ возможна при численности персонала $N \approx 3000$. В статье приведены характеристики библиотеки распознавания радужной оболочки алгоритма *EyeR SDK*, которые соответствуют проверенному по тем же базам алгоритму *VeriEye*. Использовались базы фирмы *CASIA*, полученные их сканером (рис 1) [2].

FAR	FRR (Casia1)	FRR(Casia3)
0,10%	0,05%	0,08%
0,01%	0,05%	0,09%
0,00%	0,13%	0,10%
0,00%	0,13%	0,17%
0,00%	0,13%	0,19%

Рисунок 1 – Характеристики FAR и FRR алгоритма EyeR SDK

Следует отметить, что для захвата двух глаз на одном кадре достаточное минимальное разрешение камеры составляет 1.3Мп. Поскольку, FAR и FRR статистически-независимые вероятности, то при идентификации по двум глазам значение FAR приблизительно равняется квадрату значения FAR для одного глаза, следовательно, FRR будет в два раза выше для двух глаз, чем соответствующее значение FRR для одного глаза.

При проектировании выбор статической системы прежде всего зависит от требований безопасности к ней. Радужная оболочка глаз считается самой статистически-надежной и устойчивой к подделке среди всех биометрических систем контроля доступа. Но, самыми надежными, достигая крайне больших точностей, будут комбинированные системы. В свою очередь, точность iris-системы всегда можно увеличить практически в два раза, без потерь, сделав ее на два глаза. Обобщив результаты для методов, можно сказать, что для средних и больших объектов, а также для объектов с максимальным требованием в безопасности следует использовать радужную оболочку глаз в качестве биометрического доступа.

Список использованных источников:

- Сукиасян А.А., Современная наука: теоретический и практический взгляд: сборник статей Международной научно-практической конференции [О надежности метода дактилоскопии], в 2 ч. Ч.2/ - Уфа: АЭТЕРНА, 2015. – 326 с.: ISBN 978-5-906808-22-6
- Аварханов М., Биометрия в сфере физической культуры и спорта: Учебное пособие / Аварханов М.А. - М.:МПГУ, 2015. – 120 с.: ISBN 978-5-4263-0207-5.

АВТОМАТИЗАЦИЯ РЕДАКТИРОВАНИЯ ИЗОБРАЖЕНИЙ

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

Черноокий Р.С.

Насуро Е.В. – к.т.н., доцент

Редактирование изображений является сложным и творческим процессом. Однако, некоторые элементы процесса редактирования хоть и являются затратными с точки зрения времени, но при этом они остаются монотонными и однообразными. Решение проблем нормализации гистограммы, приведения в норму уровня яркости и четкости, исправления некоторых распространенных дефектов изображения поддается автоматизации, что позволит как и подготовить изображение для редактирования профессионалом, так и помочь пользователю, не обладающему навыками редактирования изображений.

Существующие способы автоматизации редактирования изображений, используемые профессиональными художниками и любителями, часто включают в себя использование таких редакторов изображений как Photoshop или GIMP. Эти способы очень ограничены и представляют собой повторение записанных действий пользователя в случае редактора Photoshop[1] или исполнение написанного на языке Python скрипта в случае редактора GIMP[2]. В обоих случаях автоматизация редактирования лишена интеллектуальности и пригодна только для массового выполнения однотипного редактирования множества однотипных изображений.

Эти ограничения не позволяют автоматизировать однотипные действия, которые должны выполняться не над изображением в целом, а над отдельными объектами на изображении, которые могут располагаться в разных местах на разных изображениях. К тому же, такие способы не могут быть использованы, когда есть необходимость редактирования разных типов изображений разными способами. Поэтому целесообразно использовать интеллектуальные методы редактирования.

Интеллектуальное редактирование можно разбить на две части:

- анализ изображения;
- редактирование изображения согласно с результатами анализа.

Важно не начинать применение алгоритмов редактирования до выполнения анализа. Это не только позволит сэкономить время и ресурсы компьютера, но и применить те алгоритмы, которые подойдут наилучшим образом в этом конкретном случае.

Предлагается анализировать изображение следующими способами:

- анализ гистограммы;
- анализ с помощью двумерных математических операторов;

- выделение контуров и распознавания объектов.

После анализа гистограммы можно будет получить информацию об уровне яркости и выявить необходимость его корректировки. Это позволит улучшить качество изображений, полученных в результате фотографирования в условиях недостаточного освещения, и изображений, полученных в условиях избыточно экспозиции.

Различные двумерные операторы помогут идентифицировать проблемы размытия изображения и подготовить изображение для распознавания объектов. В дальнейшем размытие изображения может быть исправлено фильтром Винера, фильтром Люси-Ричардсона и другими способами.

Распознавание объектов в свою очередь поможет выделить лица, пейзаж, силуэты людей и другие объекты, часто встречающиеся на изображениях. После этого будет проведено выравнивание горизонта, исправление «красных глаз», улучшение цвета кожи.

Преимущества предложенного способа автоматизации редактирования изображений:

- выполняет монотонные и типовые действия за художника, что позволяет экономить время;
- детектирование бракованных изображений, не подлежащих восстановлению, из большого множества изображений;
- выполнение трудоёмких действий.

Недостатком является то, что интеллектуальные алгоритмы ничего не знают о замысле художника. Некоторые дефекты изображения: красные глаза, нестандартный наклон горизонта, размытие – могут быть внесены художником сознательно, но алгоритм всё равно попытается их исправить. При высоких разрешениях изображения процесс интеллектуальной обработки может занимать большие промежутки времени и является малоприменимым для обработки больших массивов изображений. Таким образом, интеллектуальное редактирование изображений подходит только для типовых изображений небольших разрешений и не подходит для редактирования художественных снимков.

Список использованных источников:

1. Andrew Faulkner, Conrad Chavez Adobe Photoshop CC Classroom in a Book / Adobe Press – 2018, - 559с.
2. Olivier Lacarme, Karine Delvare The Book of GIMP / No Scratch Press – San Francisco 2013, - 679с.

SCRUM: ОБУЧЕНИЕ И РЕАЛЬНОСТЬ

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

Чехович Е.С.

Искра Н.А. – ст. преподаватель

На сегодняшний день большинство ведущих IT-компаний используют в своей работе Agile-методологию, популярнейшим фреймворком в рамках которой является Scrum [1]. Белорусские компании также внедряют данный фреймворк в рамках разработки проектов.

Согласно отчёту о популярности Agile [2], 28% европейских компаний используют данную методологию, а 58% из них работают с фреймворком Scrum. Однако, даже те, кто работают в этих компаниях, не всегда в полной степени владеют данным фреймворком.

Такая ситуация приводит к увеличению количества вакансий, где одним из требований к соискателю является владение основами Scrum. К сожалению, мало в каких университетах в программе предусмотрено изучение фреймворка, а если его и изучают, то только “на бумаге”, т.е. практических навыков специалист не получает.

Проблема. Соответствие специалистов требованиям работодателей.

Пути решения. Обучать. Но подходов к обучению много.

На практике в IT-компаниях фреймворк Scrum преподаётся в виде 1-2 дневного семинара, в рамках которого читается краткий курс теории и проводится несколько “игр в лего”. Но самое важное в изучении Scrum “адаптировать” процесс и “адаптироваться” к нему, т.е. попробовать.

Вот уже 3 года на кафедре ЭВМ в рамках дисциплины “Технологии разработки и тестирования ПО” [3] практикуется изучение фреймворка Scrum в “реальном” процессе разработки проектов.

Однако полностью воссоздать рабочий процесс невозможно. Есть как положительные, так и отрицательные отличия обучающего процесса от рабочего.

Положительные отличия:

1. Если в процессе обучения команда допустит ошибки, влияющие на качество или скорость разработки проекта, ничего страшного не произойдёт. Но в реальном процессе такие ошибки недопустимы и могут привести к провалу проекта.
2. Во время обучения команда может выбрать новую или интересную для себя технологию и потратить время на её изучение. В реальности такой подход к выбору технологий недопустим и, чаще всего, команды годами работают в рамках тех технологий, которые им хорошо известны.
3. В процессе обучения студент, как правило, получает оценку своей работы и удовлетворительной оценкой может считаться довольно низкой.

Отрицательные отличия:

1. Вознаграждение в обучающем процессе или не предусматривается или предполагается после завершения проекта. В реальном процессе команда работает не на перспективу, а получает быстрый отклик на свою работу.
2. Невозможность сэмулировать процесс полностью. Ежедневный митинг невозможен в принципе, поскольку занятия проходят раз в неделю. Также невозможно воссоздать полный рабочий день, поскольку учебной программой предусмотрено времени гораздо меньше.

В общем случае можно уменьшить влияние отрицательных отличий. Способы уменьшения были выявлены опытным путём в процессе разработки проектов.

Одним из способов уменьшения влияния является привязка графика работы к расписанию занятий. Таким образом нивелируется сложность воссоздания рабочего процесса, а у команды вырабатывается ритм работы и нет соблазна отложить разработку проекта на конец спринта.

Снизить влияние отрицательных отличий позволяет использование средств удалённого общения и выбор проектов небольшого размера или проектов, с которыми команды уже работали.

Таким образом, используя все положительные отличия и уменьшая влияние отрицательных, возможно изучать Scrum в “реальном” процессе разработки проектов. В итоге студенты получают практические навыки работы в рамках данного фреймворка, что в будущем поможет им быстрее влиться в рабочий процесс.

Список использованных источников:

1. ScrumAlliance: The Scrum Guide [Электронный ресурс]. – Режим доступа: <https://www.scrumalliance.org/learn-about-scrum/the-scrum-guide> – Заглавие с экрана. – (Дата обращения: 10.04.2018).
2. 11th Annual State of Agile Report [Электронный ресурс]. – Режим доступа: <https://explore.versionone.com/state-of-agile/versionone-11th-annual-state-of-agile-report-2> – Заглавие с экрана. – (Дата обращения: 12.04.2018).
3. Искра, Н. А. Технологии разработки и тестирования программного обеспечения [Электронный ресурс] / Н. А. Искра // Учебная программа учреждения высшего образования по учебной дисциплине для специальности 1-40 02 01 Вычислительные машины, системы и сети – 2015. – Режим доступа: <https://libeloc.bsuir.by/handle/123456789/30530>. – (Дата обращения: 12.04.2018).

СИСТЕМА КОНТРОЛЯ СОСТОЯНИЯ КОМПОНЕНТОВ ЛОКАЛЬНОЙ СЕТИ

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

Шантарович В.Д.

Ганжа В.А. – к. ф.-мат. н, доцент

Концепция Internet of Things, как множество физических объектов (вещей), взаимодействующих с друг другом и средой, является нашим будущим и от него не убежать. Вещи вокруг нас становятся умнее с каждым днём и многие используемые задачи должны быть автоматизированы. Так как наша зависимость от IoT растёт с каждым днём, мы должны быть уверены, что такие устройства работают надёжно, безопасно и выполняют задачи, возложенные на них. Реализация такой системы контроля требует безопасную операционную систему для коммуникации с вещами, так как они могут пересылать персональную информацию, и, одновременно, эффективную, так как аппаратные ресурсы IoT скудны. Вот где Zabbix вступает в игру. Zabbix – высоко интегрированное решение мониторинга сети, которое предлагает множество функций в одном пакете.

Zabbix – это программное обеспечение для мониторинга многочисленных параметров сети, жизнеспособности и целостности серверов. Zabbix использует гибкий механизм оповещений, что позволяет пользователям конфигурировать уведомления основанные на e-mail практически для любого события. Это позволяет быстро реагировать на проблемы с серверами. Zabbix предлагает отличные функции отчетности и визуализации данных основанные на данных истории. Это делает Zabbix идеальным для планирования мощности[1].

Zabbix предлагает крошечного, скромного к ресурсам, агента для таких устройств: датчики в доме, торговый аппарат, электронное устройство и так далее. Данный агент собирает данные о производительности, доступности, статусе данных и других, различных метриках, передавая их другим устройствам или в облако. Создаётся истинная синергия между вещами IoT.

Zabbix состоит из нескольких основных программных компонентов.

Zabbix-сервер является основным компонентом, которому агенты сообщают информацию и статистику о доступности и целостности. Сервер является главным хранилищем, в котором хранятся все данные конфигурации, статистики, а также оперативные данные.

Как таковая вся информация о конфигурации, а также данные собранные Zabbix, хранятся в базе данных.

Для легкого доступа к Zabbix из любого места и с любой платформы, поставляется интерфейс на основе Веб. Интерфейс является частью Zabbix сервера и обычно (но не обязательно) работает на том же самой физической машине, что и сервер.

Zabbix прокси может собирать данные о производительности и доступности от имени Zabbix сервера. Прокси является опциональной частью Zabbix; однако он может быть полезен, чтобы распределить нагрузку одного Zabbix сервера.

Рис. 1 –инфраструктура Zabbix

Zabbix агенты разворачиваются на наблюдаемых системах для активного мониторинга за локальными ресурсами и приложениями, и для отправки собранных данных Zabbix серверу или прокси.

Кроме того, важно сделать шаг назад и взглянуть на весь поток данных в Zabbix. Для того чтобы создать элемент данных, который будет собирать данные, вы должны сначала создать узел сети. Перемещаясь в другой конец спектра Zabbix, у вас должен быть элемент данных, чтобы создать триггер. У вас должен быть триггер, чтобы создать действие.

Таким образом, если вы хотите получать оповещения о слишком высокой загрузке CPU на Сервере X, вы сначала должны создать запись о узле сети для Сервера X, затем элемент данных для наблюдения за CPU, затем триггер, который сработает, если загрузка CPU будет слишком высокой, а затем действие, которое отправит вам email.

Zabbix можно определить как распределённую систему мониторинга с централизованным веб-интерфейсом (в котором мы можем управлять практически всем). Наряду с основными функциями, мы выделим следующие:

- Zabbix имеет централизованный веб-интерфейс;
- его сервер может выполняться на большинстве Unix-подобных операционных систем;
- система мониторинга имеет внутренне присутствующих агентов для большинства операционных систем Unix, Unix-подобных и Microsoft Windows;
- данная система легко интегрируется в другие системы благодаря своему API, доступному во многих различных языках программирования и параметрах, поддерживаемых Zabbix самой по себе;
- Zabbix может осуществлять мониторинг через SNMP (v1, v2 и v3), IPMI, JMX, ODBC, SSH, HTTP(s), TCP/UDP, а также Telnet;
- Эта система мониторинга даёт нам возможность создания индивидуальных элементов и графиков, а также интерпретации данных;
- Система просто настраивается под персональные требования[2].

Одно из наиболее важных свойств Zabbix состоит в его ёмкости для хранения данных истории. Это свойство является жизненно важным в процессе предсказания тенденций. Предсказание наших тенденций не является простой задачей и является важным для рассмотрения того бизнеса, который мы обслуживаем, а когда мы рассматриваем историю данных, нам следует увидеть будут ли присутствовать повторяющиеся периоды, либо существует некоторая формула, которая может описать нашу тенденцию [3].

Список использованных источников:

1. zabbix [Электронный ресурс]. –Режим доступа:<https://www.zabbix.com>. – Дата доступа: 04.04.2018.
2. ООО "Модуль-Проекты" [Электронный ресурс]. –Режим доступа:<http://onreader.mdl.ru>. – Дата доступа: 03.04.2018
3. Zabbix и UbuntuSnappyCore[Электронный ресурс]. –Режим доступа:<http://vasilisc.com/zabbix-ubuntu-snappy-core>. –

Дата доступа: 25.03.2018

ОПТИМИЗАЦИЯ ЗАПРОСОВ К БАЗЕ ДАННЫХ С ПОМОЩЬЮ АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ

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

Шараев Е.В.

Луцик Ю.А. – к.т.н., доцент

В наше время существует большое число решений в области систем управления базами данных. Несмотря на растущую сложность реализации и большое обилие алгоритмов, обеспечивающих бóльшую производительность, по-прежнему существует некоторое число проблем, связанных с производительностью запросов. В то время как одна часть проблем с производительностью решается с помощью увеличения технических характеристик машин, на которых развернуты базы данных, создания реплик, тонкой настройки конфигураций СУБД либо других оптимизаций, существует так же возможность использования алгоритмов машинного обучения для выбора наиболее оптимальных стратегий для взаимодействия с хранящимися в базе данными.

Идея применения алгоритмов машинного обучения не является новой и уже довольно давно применяется для оптимизации работы различных действий.

Одной из наиболее продвинутых РСУБД в наши дни является OracleDatabase, которая имеет внушительное количество конфигураций, а также реализованных алгоритмов оптимизации производительности, включая алгоритмы машинного обучения. Для оптимизации запросов в Oracle (как и в

большинстве других РСУБД) применяется стоимостной оптимизатор (CBO, CostBasedOptimizer) [1]. Стоимость запроса рассчитывается как совокупная стоимость каждой операции в запросе и определяется в зависимости от планируемого числа записей, которые будут извлечены из базы (значение *cardinality*). Приблизительное значение числа записей извлекается из статистики, которую собирает база. При этом отдельно учитывается стоимость для различных операций: чтения и записи на диск (IO_COST), стоимость действий на процессоре (CPU_COST) и так далее. Для решения проблемы производительности при взаимодействии с данными, которые имеют зависимость между собой (например, атрибуты *военнообязанный* и *возраст* у таблицы *пользователи*) применяются алгоритмы, позволяющие подбирать наилучший план. Данная реализация имеет название AdaptiveQueryOptimization [2]. Она позволяет выбирать наиболее производительный план при повторном выполнении запросов одного типа, обучаясь после каждого запроса. Несмотря на все преимущества, существует так же и недостаток – платная и довольно дорогостоящая лицензия на использование OracleDatabase.

Так же существует аналогичная экспериментальная реализация адаптивных запросов для PostgreSQL [3], которая, в свою очередь, имеет бесплатную лицензию. Как и в OracleDatabase здесь используется метод подбора значения *cardinality* для схожих запросов. Однако данная реализация так же имеет несколько недостатков. В качестве признаков здесь используются значения селективности (процент кортежей, удовлетворяемых конкретному условию), извлекаемые из статистики, что может работать не совсем корректно, так как неэквивалентные условия в запросе могут превращаться в эквивалентные по значению признаки. Например, если положить, что условия *age < 15* и *age > 25* имеют одинаковое значение селективности 0.05, то условия *age < 15 AND reservist IS TRUE* и *age > 25 AND reservist IS TRUE* будут иметь эквивалентное предсказание значения *cardinality*, что не является верным, так как в действительности первый запрос, в отличие от второго, не вернет кортежей (предполагается, что не существует военнообязанных моложе 15 лет). Кроме того, существуют так же и технические недочеты, такие как отсутствие оптимизации на slave-репликах и так далее. Хотя упомянутая наработка не является наиболее оптимальной, важно отметить, что выбранное направление в действительности имеет потенциал, однако нуждается в пересмотре некоторых решений.

Таким образом, на высоком уровне абстракции планируемый подход заключается в следующем: для множества однообразных запросов создать множество соответствующих им наборов параметров модели машинного обучения, при поступлении нового запроса использовать соответствующий набор параметров модели для предсказания, а затем, после выполнения самого запроса, с учетом предсказания сравнить с полученным результатом и, таким образом, настроить параметры модели. Однообразность запросов определяется посредством схожести их условий (например, *col1 > 1 AND col2 > 1* и *col1 > 2 AND col2 > 2* будем считать однообразными). Модель представляет собой реализацию определённого алгоритма машинного обучения (например, перцептрон, набором параметров которого будет массив весов). В качестве признаков используются операнды условий. Предсказанием является число, отражающее ожидаемое количество записей, которые удовлетворяют соответствующему условию. Используя число записей, которое было предсказано, а также число записей, которое было получено после выполнения запроса, мы можем осуществить оптимизацию нашей модели и иметь более точное предсказание при последующих запросах.

Список использованных источников:

1. Jonathan Lewis, Cost-Based Oracle Fundamentals;
2. Tariq Farooq, Charles Kim, Nitin Vengurlekar, Sridhar Avantsa, Guy Harrison, Syed Jaffar Hussain, Oracle Exadata Expert's Handbook;
3. Oleg Ivanov, Sergey Bartunov, Adaptive Cardinality Estimation.

СЕРВИСЫ РАСШИРЕНИЯ ФУНКЦИОНАЛА В ОБЛАЧНЫХ ТЕХНОЛОГИЯХ

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

Янчевский В.И.

Селезнёв И.Л. – к. техн. н, доцент

В докладе рассматривается построение локального облака (private cloud) с обеспечением высокого уровня безопасности данных, предоставляющего функционал веб-сервиса, которое используются в системах с SOA (сервис-ориентированной архитектурой); локальное облако предоставляет услуги через программные интерфейсы SOAP и REST другим веб-сервисам или приложениям-клиентам. Предлагается принцип построения многосервисной облачной структуры, как совокупности набора элементов, представляющих функционально-независимые многопользовательские облачные сервисы, отвечающие требованиям архитектуры SOA.

В процессе развития облачных технологий стало ясно, что концепция облака гораздо шире, чем просто использование виртуальных вычислительных ресурсов. Следуя за спросом, требуется разрабатывать более широкий функционал решений для облачных технологий, включая инструменты для полноценного управления виртуальными машинами и базовыми сценариями.

SOA (сервис-ориентированная архитектура) рассматривается в виде совокупности веб-сервисов, часто

строящихся как распределённые системы и работающие на разных платформах. Веб-сервисы могут взаимодействовать как друг с другом, так и с приложениями, созданными на основе SOA, посредством сообщений. Эти сообщения передаются стандартными протоколами, которые получили наибольшее распространение – SOAP (протокол обмена xml-данными) и REST (стиль архитектуры программного обеспечения для распределённых систем), в формате (языка разметки) XML и/или JSON.

В качестве транспорта сообщений используют протокол HTTP. В SOAP применяется специальный язык описания веб-сервисов и доступа к ним (WSDL) в формате XML. В REST нет такого описания типа, так как используется фиксированный набор методов доступа к веб-сервисам: GET, POST, PUT, DELETE. Указанные веб-сервисы основаны на открытых международных стандартах, которые поддерживаются современными операционными системами.

Программное обеспечение как сервис (Software software as a service, SaaS) является перспективным направлением в облачных технологиях стимулирующим развитие архитектуры SOA.

SaaS предполагает, что пользователь получает в распоряжение ПО, функции которого доступны через веб-интерфейс, в то время, как основная программная часть приложения находится на сервере разработчика. (Рис.1)

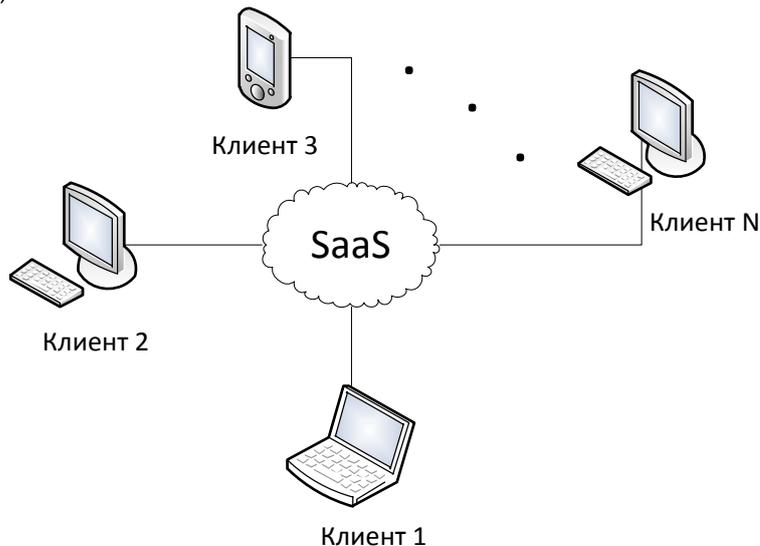


Рисунок 1 – Однооблачная структура сервиса SaaS

Наиболее перспективными средствами интеграции приложений SaaS в системы с архитектурой SOA являются специализированные средства интеграции приложений — платформы интеграции SaaS на базе PaaS (Platform as a Service), что предполагает использование аппаратных компонентов на стороне поставщика услуг — например, дисковых накопителей или процессора. В PaaS для взаимодействия сервисов используется Middleware (ПО среднего слоя) как связующее ПО.

Платформа PaaS, на базе которой функционируют приложения SaaS может обеспечить средства для интегрирования веб-сервисов, предоставлять сервер приложений или веб-сервер для размещения интегрированной веб-службы (SaaS mashup), а также может организовать систему управления SOA для совместного использования ресурсов SaaS.

На рис. 2 представлен пример сервис-ориентированной архитектуры (SOA) интегрированной службы или SaaS mashup, которая использует в качестве источников информации сторонние (SaaS-1...SaaS-3) и собственные веб-сервисы (SaaS-4, SaaS-5).

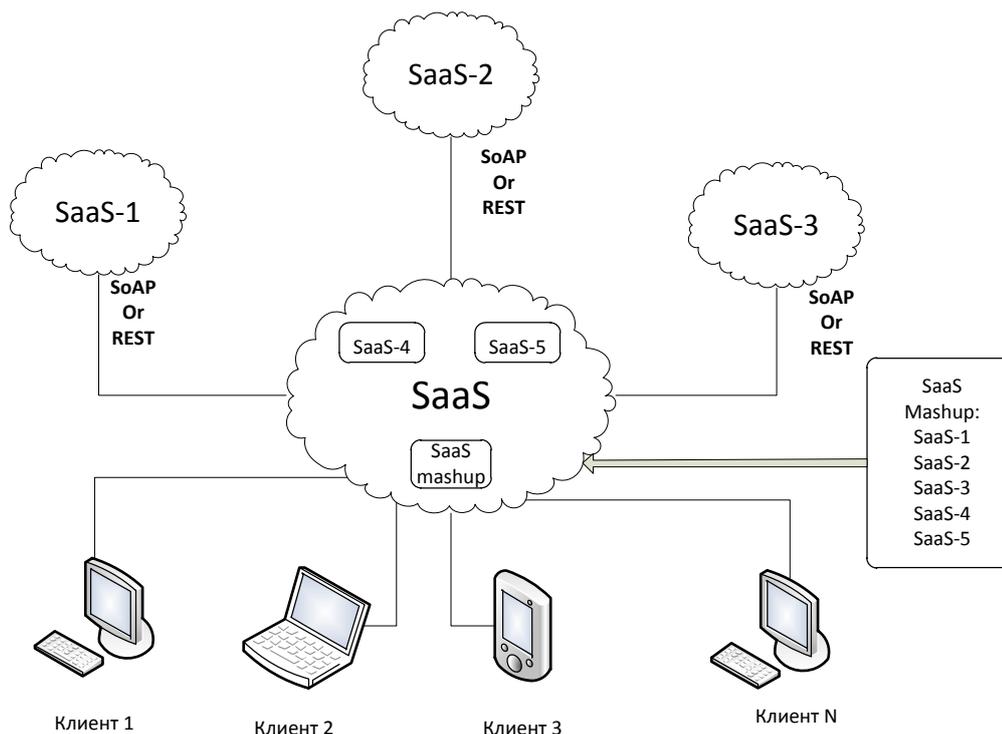


Рис.2 – Многооблачная структура системы SaaS

Веб-сервисы (SaaS-1...SaaS-5) взаимодействуют с интегрированным веб-сервисом SaaS mashup через Middleware. SaaS mashup или связанное распределенное приложение размещается на сервере приложений (App server) платформы PaaS, доступ к этому сервису осуществляется через пользовательский интерфейс. Таким образом, сервис-ориентированное распределенное приложение SaaS mashup представляет собой результат интегрирования веб-служб (SaaS-1...SaaS-5) в одно, логически завершенное, связанное приложение.

К примеру, можно разработать собственный SaaS-сервис, взаимодействующий с github репозиторием.

Современная парадигма построения IT-инфраструктуры для управления бизнес-процессами ориентирована на динамическую (гибкую), а не на статическую (жесткую) инфраструктуру систем, которые могут своевременно реагировать на изменение рыночных условий или особенностей ведения бизнеса. Отсюда следует, что IT-инфраструктура должна проектироваться как приложение с сервис-ориентированной архитектурой в виде набора веб-сервисов, так как именно архитектура SOA обеспечивает требуемую адаптацию к изменяющимся условиям ведения бизнеса.

Сервис - ориентированная архитектура приложений обеспечивает возможность переориентации их конфигурации на решение новых задач бизнеса. Таким образом, принципы сервис-ориентированных архитектур являются наиболее современным способом интеграции крупномодульных, слабосвязанных и общедоступных по сети удаленных веб-сервисов. Помимо этого, системы на основе SOA способствуют развитию BPM (Business Process Management) как класспрограммного обеспечения для управления бизнес-процессами и административными регламентами.

Список использованных источников:

1. Веб-сервисы и облачные вычисления [Электронный ресурс]. – Режим доступа: <http://www.lessons-tva.info/archive/nov032.html>. – Дата доступа: 01.04.2018.
2. Янчевский В.И., Стасилевич М.Н. Эффективное описание границ объектов постоянной яркости: 51-я научная конференция аспирантов, магистрантов и студентов по направлению 8: Информационные системы и технологии: 18.04.15. Минск, 2015. С. 19.
3. REST [Электронный ресурс]. – Режим доступа: <http://blogger.sapronov.me/2014/02/rest.html>. – Дата доступа: 05.04.2018.

АНАЛИЗ ОСНОВНЫХ АРХИТЕКТУР НЕЙРОННЫХ СЕТЕЙ ГЛУБОКОГО ОБУЧЕНИЯ

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

Козлова О.В.

Лукашевич М.М. – к.т.н., доцент

Как часть концепции искусственного интеллекта, глубокое обучение лежит в основе различных инноваций: беспилотные автомобили, распознавание голоса, изображения и многое другое. С 2014 года рынок глубокого обучения демонстрирует непрерывный рост. Самой популярной задачей для нейронных сетей глубинного обучения является детекция объектов на изображениях. Исходя из этого формулируются следующие задачи:

- реализовать программный модуль глубинного обучения нейронных сетей;
- обучить его распознавать объекты разных классов на изображениях.

Для детекции объектов используются сверточные нейронные сети (CNN). Суть сверточных нейронных сетей заключается в получении иерархических признаков изображений. Сначала выделяются простые признаки, из них комбинируются сложные признаки, из них еще более сложные и в конце можно скомбинировать очень сложный признак — конкретный человек, конкретная машина. Затем на основе этих признаков определяется класс объекта.

Одним из популярных подходов детекции объектов на изображениях является семейство алгоритмов R-CNN (R-CNN, Fast R-CNN, Faster R-CNN). В их основе лежат сверточные нейронные сети.

Суть алгоритма заключается в поиске на изображении регионов, которые потенциально содержат объект. С помощью CNN строится карта признаков изображения и производится классификация. Затем точно выделяются границы объектов.

Каждый из подходов подразумевает использование любых сверточных нейронных сетей, классификаторов и различных алгоритмов для поиска регионов. Данная особенность позволяет разработать алгоритм для решения разных задач.

Список использованных источников:

1. Ross Girshick, Fast R-CNN.
2. Ross Girshick, Jeff Donahue, Trevor Darrell, Rich feature hierarchies for accurate object detection and semantic segmentation.
3. Shaoqing Ren, Kaiming He, Ross Girshick, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.

АЛГОРИТМ АНАЛИЗА ЭМОЦИОНАЛЬНОГО СОДЕРЖАНИЯ КОММЕНТАРИЯ ПОЛЬЗОВАТЕЛЯ

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

Курашкевич В.В.

Калабухов Е.В. - старший преподаватель

Анализ тональности текста - класс методов контент-анализа в компьютерной лингвистике, предназначенный для автоматизированного выявления в текстах эмоционально окрашенной лексики и эмоциональной оценки авторов (мнений) по отношению к объектам, речь о которых идёт в тексте.

Основной целью анализа тональности является нахождение мнений в тексте и выявление их свойств.

В современных системах автоматического определения эмоциональной оценки текста чаще всего используется одномерное эмотивное пространство: позитив или негатив (хорошо или плохо). Однако известны успешные случаи использования и многомерных пространств.

Типы классификаций:

1. Классификация по бинарной шкале. В этом случае для определения полярности документа используется два класса оценок: позитивная или негативная.

2. Классификация по многополосной шкале. За основу берется шкала, например от 1 до 5.

Существует ряд тезаурусов, специально размеченных с учётом эмоциональной составляющей. Такие словари, описанные далее, необходимы компьютерным программам при анализе тональности текста.

WordNet-Affect — это семантический тезаурус, в котором понятия, связанные с эмоциями, («эмоциональные концепты», англ. «affective concepts») представлены с помощью слов, обладающих эмоциональной составляющей («эмоциональные слова», англ. «affective words»). WordNet-Affect состоит из такого подмножества синсетов WordNet, где каждый синсет, соответствующий «эмоциональному концепту», может быть представлен с помощью «эмоциональных слов». Примером для разработки WordNet-Affect

послужило многоязычное расширение WordNet, названное WordNet Domain. В расширении WordNet Domain каждому синсету приписано не менее одной пометы предметной области (англ. «domain label»), например: спорт, политика, медицина. Всего в иерархически организованную структуру было включено около двухсот предметных помет.

SentiWordNet — это лексический семантический тезаурус, первая версия которого была разработана в 2006 году. На данный момент самой последней версией SentiWordNet является SentiWordNet 3.0, использование которой дает более чем 20 % прирост точности по сравнению с первой версией.

Методы классификации тональности:

- 1) Методы, основанные на правилах и словарях
- 2) Машинное обучение с учителем
- 3) Машинное обучение без учителя
- 4) Метод, основанный на теоретико-графовых моделях.

При использовании метода, основанного на теоретико-графовых моделях, анализ тональности разбивается на несколько этапов:

- построение графа на основе исследуемого текста,
- ранжирование его вершин,
- классификация найденных слов,
- вычисление результата.

**СЕКЦИЯ
«ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»**

ТАРГЕТИРОВАННЫЕ СОЦИАЛЬНЫЕ СЕТИ

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

Аль-Хелали Д. Т.

Данилова Г.В. – м.т.н. ассистент

В настоящее время развитие всех мировых социальных сетей можно разделить на три периода:

1. пионеры, социальные сети 90-х годов с простейшей функциональностью;
2. создание сетей (с 2000 года) с широкой функциональностью для возможности базового взаимодействия;
3. создание сетей, направленных на решение конкретных задач: поиск сотрудников для работодателя (бизнес-сети), игровые развлечения (игровые сети), и т.д.

Вслед за сетями общего типа, стали развиваться таргетированные (тематические) сети, которые использовали ту же функциональность, но в более ограниченной, конкретной нише. Сейчас этот процесс перешел в более активную стадию. Можно уже находить и регистрироваться в социальных сетях для туристов, спортсменов, меломанов, книголюбов, фотографов и т.д. Это нормальная тенденция, которая была вызвана насыщением рынка общими сетями.

Социальная сеть — социальная структура, состоящая из группы узлов, которыми являются социальные объекты (люди или организации), и связей между ними.

Термин «Социальная сеть» был введен задолго до появления Интернета и современных интернет-сетей, ещё в 1954 году социологом из «Манчестерской школы» Джеймсом Барнсом. Современное понятие в простом виде означает некий круг знакомых человека, где есть сам человек – центр социальной сети, его знакомые – ветки этой социальной сети и отношения между этими людьми – связи. Если рассматривать социальную сеть более глубоко, можно обнаружить, что связи делятся по типам: односторонние и двусторонние; сети друзей, коллег, одноклассников, однокурсников и т.д.

Во второй половине 20 века социальные сети стали активно развиваться как научная концепция, сначала они стали популярны на западе, чуть позже пришли и к нам. Затем это обычное профессиональное понятие социологов превратилось в концепцию, являющуюся одной из центральных в концепции веб 2.0, которую как понятие впервые ввел Тим О'Рейли 30 сентября 2005 года в своей статье «Tim O'Reilly – What Is Web 2.0» [1].

Первыми компьютерными социальными сетями стали группы людей, использовавшие для создания и поддержания социальных связей средства компьютерного общения, которыми стала электронная почта. Случилось это 2 октября 1971 года – день первого сообщения, отправленного на удаленный компьютер, а первыми пользователями социальной сети стали военные в сети ARPANet. Это был первый шаг к созданию Интернета и современных социальных интернет-сетей.

Следующим шагом стало изобретение IRC (англ. Internet Relay Chat – ретранслируемый интернет-чат) – сервисной системы для общения в режиме реального времени. IRC была создана в 1988 году финским студентом Ярмо Ойкариненом. Это были уже более «продвинутые» социальные сети, однако ещё далекие от современных.

Из изобретения компьютеров, электронной почты, IRC и многих других 7 августа 1991 года вытекло изобретение Интернета. Именно в этот день британский ученый Тим Бернерс-Ли впервые опубликовал первые интернет-странички и сделал тем самым следующий шаг к современным социальным сетям.

И вот в 1995 году появилась первая, приближенная к современным, социальная сеть Classmates.com, которую создал Рэнди Конрад, владелец компании Classmates Online Inc. Этот сайт помогал зарегистрированным посетителям находить и поддерживать отношения с друзьями, одноклассниками, однокурсниками и другими знакомыми людьми. Сейчас в этой сети зарегистрировано более 40 миллионов людей, преимущественно из США и Канады.

Концепция Classmates оказалась успешной и с далёкого 2005 года она развивается и уже не только в пределах этой сети появились такие мировые гиганты, как MySpace, FaceBook, Bebo и LinkedIn или гиганты рунета - Одноклассники.ру, ВКонтакте, МойМир и МойКруг [2].

Предметом данного проекта является социальная сеть для студентов. Социальная сеть, где студенты могут общаться друг с другом и с преподавательским составом, собираться в группы, классы, общины. Иметь возможность отключать все уведомления, чтобы максимально сконцентрироваться на учебе. Также, социальная сеть предоставляет все необходимые инструменты для обучения, которые раньше студенты получали из нескольких источников: OneDrive, GoogleCalendar, MSOfficeWord и др. Все пользователи будут зарегистрированы при поступлении в университет, т.е. будут иметь свои реальные, а не вымышленные имена. Главная цель – сделать процесс обучения более связанным и непрерывным.

Список использованных источников:

1. Далворт Майк. Социальные сети. Руководство по эксплуатации. – М.: Энергия, 2010. – с.15-120.
2. Тапскотт Д., Уильямс Э. Д. Викиномика. Как массовое сотрудничество изменяет все. – М.: Изд-во иностр. лит., 2009. – С. 363-384.

СПОСОБЫ ОЦЕНКИ УРОВНЯ БЕЗОПАСНОСТИ СООРУЖЕНИЙ

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

Астапович Е.И.
Парамонов А. И. – кандидат техн. наук, доцент кафедры ПОИТ

В работе кратко рассмотрены текущие подходы к уровню оценки безопасности сооружений в российских и белорусских законах и стандартах, их критика, опубликованная в научных журналах, и предложены пути устранения обозначенных недостатков.

Оценка уровня безопасности является неотъемлемым этапом при проектировании строительных сооружений. Строительное сооружение – это единичный результат строительной деятельности, предназначенный для осуществления определенных потребительских функций. В контексте данной статьи будем рассматривать строительные сооружения, подразумевающие временное или постоянное пребывание людей, такие как: жилые здания, офисные здания, станции метрополитена, площадки для организации уличных фестивалей и др.

Безопасностью сооружения можно назвать его способность выполнять свои функции без угрозы жизни и/или здоровью находящихся в нем людей. Несмотря на такую жесткую формулировку, полностью исключить наличие угрозы невозможно. В связи с этим большинство требований к безопасности сооружений разработаны с использованием стохастических подходов.

Хотя единичная угроза жизни и/или здоровью может возникнуть и в процессе обычной эксплуатации сооружения, большинство угроз возникает при образовании и развитии чрезвычайной ситуации внутри сооружения. Чрезвычайная ситуация – обстановка, сложившаяся на определенной территории в результате промышленной аварии, иной опасной ситуации техногенного характера, катастрофы, опасного природного явления, стихийного или иного бедствия, которые повлекли или могут повлечь за собой человеческие жертвы, причинение вреда здоровью людей или окружающей среде, значительный материальный ущерб и нарушение условий жизнедеятельности людей.

Наиболее распространенным способом обеспечения безопасности людей, в условиях развития чрезвычайной ситуации, является эвакуация. Эвакуация – процесс движения людей из помещения, здания, сооружения по эвакуационным путям с целью предотвращения возможного воздействия на них опасных факторов чрезвычайной ситуации.

Уровнем безопасности сооружения будем называть некоторую интегральную характеристику сооружения, определяющую безопасность его использования.

Традиционно область обеспечения безопасности сооружений регулировалась государством. Отличительной особенностью государственного регулирования безопасности сооружений является то, что в конечном итоге законодательные документы представляют собой набор конкретных требований. С точки зрения государственного регулирования можно выделить всего два уровня безопасности сооружений: «безопасный» и «не безопасный». При этом «безопасным» считается сооружение, которое удовлетворяет всем законодательным требованиям, а «не безопасным» считается сооружение, не удовлетворяющее требованиям. Хотя такой подход оправдан при государственном регулировании, где целью является вынести однозначное решение о безопасности либо небезопасности здания, при проектировании сооружения может быть полезно иметь более сравнимую оценку уровня безопасности сооружения, такую оценку, которая позволяла бы сравнивать два различных сооружения (или две версии одного и того же сооружения) по уровню безопасности.

Для достижения такой цели воспользуемся детальными требованиями государственных стандартов, в частности, расчетной моделью обеспечения пожарной безопасности в ГОСТ 12.1.004-91. Данный ГОСТ требует, что «требуемый уровень обеспечения пожарной безопасности людей с помощью систем пожарной безопасности должен быть не менее 0,999999 предотвращения воздействия опасных факторов в год в расчете на каждого человека, а допустимый уровень пожарной опасности для людей должен быть не более 10^{-6} воздействия опасных факторов пожара, превышающих предельно допустимые значения, в год в расчете на каждого человека». Уровень обеспечения пожарной безопасности определяется в соответствии с расчетной моделью как:

(1)

где Q_p – вероятность пожара в здании в год;
 $P_э$ – вероятность эвакуации людей;
 $P_{пз}$ – вероятность эффективной работы технических решений противопожарной защиты.

Вероятность пожара Q_p чаще всего вычисляется статистическими методами для зданий похожей конструкции и/или назначения.

Вероятность эвакуации ($P_э$) вычисляется по формуле:

(2)

где $P_{эп}$ – вероятность эвакуации по эвакуационным путям;
 $P_{дв}$ – вероятность эвакуации по наружным эвакуационным лестницам, переходам в смежные секции зданий.

Вероятность эвакуации по эвакуационным путям принимается равной 0,999 при расчетном времени эвакуации меньше чем необходимое время, 0 при расчетном времени эвакуации значительно больше чем

необходимое время, и вычисляется по обратно пропорциональному закону между этими двумя точками. Вероятность эвакуации по наружным эвакуационным лестницам, переходам в смежные секции зданий ($P_{дв}$) принимается в пределах 0,01-0,05. Расчетное время эвакуации вычисляется через определение линейных размеров путей эвакуации и нормирования пешеходных потоков через эти пути.

Как можно заметить, предложенный в стандарте способ определения уровня безопасности сооружения представляет собой расчетную стохастическую модель. Основой данной модели является расчет времени эвакуации, которое однозначно преобразуется в вероятность эвакуации, и нормируется путем умножения на вероятность необходимости эвакуации.

Критика данной расчетной модели неоднократно высказывалась в различных научных статьях:

- Статистический метод определения вероятности пожара в сооружении (Q_p) не подходит для уникальных объектов. При этом удельный вес данной вероятности в итоговой P_v достаточно высок, что позволяет выполнять требования стандарта, не обеспечивая достаточно надежной эвакуации и/или достаточно надежных технических средств противодействия чрезвычайной ситуации.
- Скорость движения людей при эвакуации определяется как функция плотности потока – не учитываются неоднородности скорости движения внутри потока, что значительно влияет на итоговое время эвакуации.
- Определение плотности потока как функции линейных размеров путей эвакуации не учитывает рельеф и препятствия на путях.
- Не учитывается место возникновения пожара.
- Не учитывается приоритетность эвакуации помещений в зависимости от динамики развития чрезвычайной ситуации.

Для устранения недостатка в способе расчета вероятности чрезвычайной ситуации в сооружении Q_p , представим уровень безопасности сооружения как характеристику сооружения в случае, если чрезвычайная ситуация уже наступила, т.е. примем Q_p равным единице. Что касается оставшихся недостатков, устранение их в расчетной модели стандарта является крайне сложной задачей. Для полноценной оценки степени их влияния на конечный результат, а также для выявления способов, которыми они оказывают это влияние, необходимо либо иметь хорошо структурированные статистические данные, либо проводить дорогостоящие эксперименты в реальном мире. Оба варианта не являются практичным решением проблемы.

К счастью, существует еще одна альтернатива – имитационное моделирование. Более того, замена расчетной модели эвакуации на имитационную в целом улучшит качество модели по отношению к не типовым объектам. При замене расчетной модели на имитационную логично будет определять вероятность эвакуации напрямую статистическими методами: за вероятность эвакуации можно принять среднее значение отношения успешно эвакуированных людей к общему количеству людей.

Для осуществления поставленной цели необходимо разработать имитационную модель, включающую в себя два компонента: модель движения пешеходных потоков и модель развития чрезвычайной ситуации. Область имитационного моделирования пешеходных потоков достаточно развита – существуют множество моделей поведения пешеходных потоков. К сожалению, того же нельзя сказать про модели развития чрезвычайных ситуаций.

Также стоит отметить, что описанная модель оценивала уровень безопасности сооружения по отношению к пожару в качестве чрезвычайной ситуации. Для более полной оценки уровня безопасности сооружения будем вычислять произведение вероятностей обеспечения безопасности в каждой из набора чрезвычайных ситуаций.

Таким образом, предложенная оценка уровня безопасности сооружений имеет вид:

(3)

где n – количество рассматриваемых чрезвычайных ситуаций;
 $P_{эi}$ – вероятность эвакуации в случае возникновения чрезвычайной ситуации i ;
 $P_{пэi}$ – вероятность эффективной работы технических решений, предотвращающих распространение чрезвычайной ситуации i .

Список использованных источников:

1. Система нормативных документов в строительстве. Основные положения. СНиП 10-01-2003 [Электронный ресурс]. – Режим доступа: <http://снп.рф/снп/full/187>. Дата доступа: 20.03.2018.
2. О защите населения и территорий от чрезвычайных ситуаций: Закон Респ. Беларусь, 5 мая 1998 г., №141-З. // kodeksy-by.com [Электронный ресурс]. – Режим доступа: http://kodeksy-by.com/zakon_rb_o_zashite_naseleniya_i_territorij_ot_chrezvychajnyh_situatsij/1.htm. Дата доступа: 19.03.2018.
3. ГОСТ 12.1.004-91. Система стандартов безопасности труда. Пожарная безопасность. Общие требования. — Введен 1992–07–01. — М.: Издательство стандартов, 1996. — 97 с.
4. Холщевников, В.В. Проблемы оценки безопасности людей при пожаре в уникальных зданиях и сооружениях // Журнал «Пожаровзрывобезопасность» (выпуск 4, том 12). – 2003.
5. Танклевский, Л.Т. О возможности оптимизации движения эвакуирующихся из многоэтажных зданий / Л.Т. Танклевский, С.П. Юн, А.А. Таранцев. // Журнал «Пожаровзрывобезопасность» (выпуск 1, том 14). – 2005.
6. Сметанкина, Г.И., Романченко С.А. Актуальные проблемы обеспечения пожарной безопасности общественных зданий // Современные технологии обеспечения гражданской обороны и ликвидации последствий чрезвычайных ситуаций: материалы всероссийской научно-практической конференции курсантов, слушателей, студентов и молодых ученых. – Воронеж: ФГБОУ ВО Воронежский институт ГПС МЧС России, 2016, том 1.

ОПТИМИЗАЦИЯ ПОИСКА ПУТЕЙ ТРАНСПОРТНЫХ СРЕДСТВ В ПЕРЕВОЗКАХ

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

Борисов Д.В.

Лукьянец В.Г. – к.т.н., доцент

В современном мире продолжает расти количество людей, а также увеличивается сложность транспортной инфраструктуры. Это приводит к увеличению внимания к оптимизации транспортного обслуживания перевозок. Одной из не теряющих актуальность задач, является задача поиска оптимального пути через заданные города.

Алгоритмы определения пути через множество точек делятся на точные и приближенные. Точные алгоритмы позволяют определить оптимальный путь, но за это приходится расплачиваться большим временем поиска. Зачастую для больших графов время ожидания становится настолько большим, что используют приближенные алгоритмы. Они позволяют гораздо быстрее найти необходимый путь, немного жертвуя точностью.

Рис. 1 – Города и дороги между ними образуют граф

Важной чертой транспортных перевозок является частичная или полная повторяемость часто используемых маршрутов. В таком случае можно использовать приближенные алгоритмы поиска для определения наиболее выгодной последовательности узлов графа, при этом пути между каждой парой точек уточняются на основе данных из базы. Информация о путях между точками сохраняется для каждой отсутствующей в базе пары и может использоваться повторно для поиска маршрута для другого транспорта. Поиск маршрута между парами точек осуществляется с помощью предпочитаемого алгоритма, решающего задачу о кратчайшем пути в графе. Такой подход позволяет использовать результаты предыдущих вычислений путей, тем самым экономя время и вычислительные ресурсы.

К основным преимуществам такого подхода можно отнести:

- большая точность;
- возможность комбинировать пути между точками рассчитанные точными и приближенными алгоритмами;
- возможность для увеличения точности отложено пересчитывать отдельные участки пути;
- высокая эффективность для больших графов при небольшом разнообразии возможных точек маршрута;

Основной недостаток такого подхода — большие вычислительные сложности для новых маршрутов. Если маршруты постоянно меняются и не имеют ничего общего с маршрутами, которые использовал транспорт до этого, то сохраненные пути попросту не используются. Также такой подход не будет эффективен для малых графов, на которых неплохо себя показывают точные алгоритмы.

Таким образом, в задачах оптимизации транспортного обслуживания перевозок предлагаемый алгоритм позволяет повысить эффективность поиска оптимального маршрута.

Список использованных источников:

1. Свами М., Тхуласираман К. Графы, сети и алгоритмы М.: Мир, 1984. -455 с.
2. Нечепуренко М.И., Попков В.К. Алгоритмы и программы решения задач на графах и сетях, Новосибирск: Наука. Сиб. Отделение, 1990. – 515 с.
3. Филлипс Д., Гарсиа-Диас А. Методы анализа сетей, Пер с английского – М.:Мир, 1984. – 496 с.

РАСПОЗНАВАНИЕ ЭМОЦИЙ ПО ГОЛОСУ НА ОСНОВЕ МАШИННОГО ОБУЧЕНИЯ

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

Брановицкий А.А.

Серебряная Л.В. – к.т.н., доцент

Эмоции и речь тесно взаимосвязаны и играют огромную роль в общении. В связи с этим, автоматическая и объективная диагностика эмоционального состояния человека по его речи представляет большой практический интерес. Возможность распознавания эмоций в речи важна как для исследования самой речи и эмоций, так и для улучшения качества обслуживания клиентов, например, в колл-центрах. Также идентификация эмоционального состояния является востребована в телекоммуникационной сфере, в индустрии развлечений, обучении, медицине и других сферах.

Существуют различные методы для решения задачи распознавания эмоций по голосу, на основе которых строятся системы идентификации эмоционального состояния человека. Одним из самых простых методов решения задачи является вычисление математического ожидания для частей кадра сигнала с последующим применением алгоритма K-means для классификации. Недостатком данного метода является невысокая точность распознавания эмоций. Следовательно, важной задачей является выбор алгоритма для вычисления значимых характеристик кадра сигнала и алгоритма для классификации, так как именно эти алгоритмы играют главную роль в достижении необходимой точности распознавания эмоций по голосу.

Работа типовой системы распознавания эмоций по речи может быть описана следующей последовательностью шагов:

1. Получение на вход аналогового звукового сигнала;
2. Выполнение АЦП (аналогово-цифровое преобразование) входного сигнала для получения его спектральных составляющих с помощью алгоритма ДПФ (дискретное преобразование Фурье). При необходимости повышения быстродействия системы АЦП осуществляется с помощью алгоритма БПФ (быстрое преобразование Фурье);
3. Сглаживание спектра сигнала. Может быть реализовано различными оконными фильтрами. Часто используемым фильтром является оконная функция Хэмминга.
4. Разбиение спектра сигнала на кадры;
5. Вычисление значимых характеристик и признаков кадров;
6. Распознавание эмоций на основе значимых характеристик кадра.

Выбор алгоритма MFCC для вычисления значимых характеристик кадра сигнала и нейронной сети, обученной методом обратного распространения ошибки для классификации, решают проблему достижения высокой точности распознавания эмоций по голосу.

MFCC (Мел-частотные кепстральные коэффициенты) – это представление энергии спектра сигнала. Преимущества их использования заключаются в следующем: используется спектр сигнала (то есть разложение по базису ортогональных [ко]синусоидальных функций), что позволяет учитывать волновую “природу” сигнала при дальнейшем анализе; спектр проецируется на специальную mel-шкалу, позволяя выделить наиболее значимые для восприятия человеком частоты; количество вычисляемых коэффициентов может быть ограничено любым значением, что позволяет “сжать” кадр.

Mel – это “психофизическая единица высоты звука”, основанная на субъективном восприятии среднестатистическими людьми. Зависит в первую очередь от частоты звука (а также от громкости и тембра). Другими словами – это величина, показывающая, насколько звук определённой частоты “значим” для человека. Mel на основе частоты сигнала рассчитывается по следующей формуле:

Мел-частотные кепстральные коэффициенты кадра далее поступают на вход нейронной, обученной методом обратного распространения ошибки.

Алгоритм обратного распространения ошибки – один из методов обучения многослойных нейронных сетей прямого распространения. Обучение алгоритмом обратного распространения ошибки предполагает два прохода по всем слоям сети: прямого и обратного. При прямом проходе входной вектор подается на входной слой нейронной сети, после чего распространяется по сети от слоя к слою. В результате генерируется набор выходных сигналов, который и является фактической реакцией сети на данный входной образ. Во время прямого прохода все синоптические веса сети фиксируются. Во время обратного прохода все синоптические веса настраиваются в соответствии с правилом коррекции ошибок, а именно: фактический выход сети вычитается из желаемого, в результате чего формируется сигнал ошибки. Этот сигнал, впоследствии, распространяется по сети в направлении, обратном направлению синоптических связей.

Алгоритм MFCC учитывает волновую природу звука и психофизическое восприятие звука человеком, устойчив к изменению тембра голоса, громкости и скорости произношения, что вместе с высокой точностью классификации нейронной сети, обученной методом обратного распространения ошибки, обеспечивают распознавание эмоций по голосу с точностью порядка 88-95%.

Список использованных источников:

1. Оппенгейм, А.В. Цифровая обработка сигналов / А.В. Оппенгейм, Р. Шафер. – М., Техносфера, 2012. – 1048 с.
2. Хайкин, С. Нейронные сети: полный курс / С. Хайкин. – 2-е издание. – М., Вильямс, 2016. – 1104 с.

СРАВНИТЕЛЬНАЯ ХАРАКТЕРИСТИКА КАРТОГРАФИЧЕСКИХ СЕРВИСОВ

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

Бувевич Д.И.

Таборовец В.В. - к.т.н., доцент

В настоящее время большую популярность приобрели так называемые «картографические сервисы». В основе таких сервисов лежит географическая информационная система (ГИС) – система сбора, хранения, обработки и визуализации географических данных. Картографические сервисы, взятые для анализа: Google Maps, Яндекс.Карты, OpenStreetMap.

В данной статье приводится сравнительная характеристика сервисов с точки зрения возможности их использования при разработке приложения, каким-либо образом использующего картографические данные. В приведённой характеристике сервисы рассматриваются исключительно с точки зрения разработчика, а не пользователя. Так, например, OpenStreetMap имеет возможность моментально вносить изменения в карты, а Яндекс.Карты позволяют показывать расположение общественного транспорта в режиме реального времени. Эти и подобные им функции безусловно важны для пользователя, однако, в общем случае не влияют на выбор того или иного сервиса разработчиком приложения.

В таблице 1 приведена сравнительная характеристика сервисов. Данные сервисы были выбраны исходя из их популярности, доступности и возможностей (в частности в странах СНГ). Так, например, Google Maps наиболее популярны по всему миру, однако Яндекс.Карты имеют преимущества при использовании в странах СНГ. OpenStreetMap выделяется тем, что это некоммерческий проект, то есть его использование абсолютно бесплатно.

Таблица 1

Критерий	Google Maps	Яндекс.Карты	OpenStreetMap
Покрытие, детализация	Лучшее покрытие мира, детализация в некоторых случаях хуже, чем у конкурентов	Лучшее покрытие стран СНГ	Детализация и покрытие на хорошем уровне.
Цена	До 25 000 запросов – бесплатно. Свыше 25 000 – 0.5 доллара за 1000 запросов.	До 25 000 запросов – бесплатно. Свыше 25 000 – в зависимости от тарифа. Однако даже самый выгодный тариф дороже, чем у Google Maps при тех же нагрузках.	Бесплатно
Документированность	Хорошая документация на официальном сайте, в том числе на русском языке. Множество различных видеоматериалов/курсов на английском.	Хорошая документация на официальном сайте, в том числе на русском языке.	Документация на английском языке в виде wiki-страниц. Уступает в удобочитаемости и наглядности конкурентам.
Возможность использования без программирования	Есть	Есть	Нет
Построение маршрутов	Есть (до 2 500 запросов в день бесплатно)	Есть (только платно)	Есть
Информация о текущей дорожной ситуации	Есть	Есть	Только в некоторых странах
Возможность локализации	Есть (52 языка[3])	Есть только для 4 языков[2]: русский, английский, украинский и турецкий.	Есть (около 50 языков[1])

В таблице выше были указаны пункты, в которых наблюдаются различия между сервисами. Поэтому следует отметить, что в данную таблицу не вошли следующие возможности, которыми обладают каждый из рассмотренных сервисов: Геокодер (возможность преобразования координат в адреса и наоборот), возможность настройки внешнего вида карты, Static API (возможность получения картинки с фрагментом карты, сгенерированной на стороне сервера), панорамы улиц, API для использования в мобильных приложениях, поиск по организациям.

В результате проведённого анализа можно заключить, что при выборе картографического сервиса, нужно в первую очередь опираться на требования разрабатываемого приложения. OpenStreetMap подойдёт

для случаев, когда бюджет весьма ограничен. Однако стоит отметить, что из-за более низкого качества документации, чем у соперников, разработка такого приложения может занять больше времени. Яндекс.Карты традиционно считается лучшим для использования в странах СНГ, но следует отметить, что в последнее время Google стремится восполнить этот пробел. В целом Google Maps смотрятся наиболее привлекательно с точки зрения разработчика благодаря хорошей документации, приемлемым ценам и широкому спектру возможностей.

Список использованных источников:

1. Официальная документация OpenStreetMap API [Электронный ресурс]. Режим доступа: <https://wiki.openstreetmap.org/wiki/API>
2. Официальная документация Yandex.Maps API [Электронный ресурс]. Режим доступа: <https://tech.yandex.ru/maps/mapsapi/>
3. Официальная документация Google Maps API [Электронный ресурс]. Режим доступа: <https://developers.google.com/maps/>

ДИАГНОСТИКА НЕВРОЛОГИЧЕСКИХ ЗАБОЛЕВАНИЙ НА ОСНОВЕ ОБРАЗЦОВ ПОЧЕРКА

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

Верховцов П.А.

Хмелева А.В. – канд. техн. наук, доцент

Неврологические заболевания, такие как болезнь Паркинсона, синдром дефицита внимания и гиперактивности, могут быть диагностированы на начальных стадиях на основе анализа параметров почерка либо динамики изменения параметром. Данный подход позволяет удешевить и ускорить первичную диагностику неврологических заболеваний.

При анализе почерка пациентов, страдающих неврологическими заболеваниями, было выяснено, что у 9 - 75% появляются в различной степени микрография и нечёткость контуров символов, вызванная тремором [1]. Образец почерка пациента, страдающего неврологическим заболеванием представлена на рисунке 1. Микрография — это приобретенное расстройство, характеризующееся аномально малым, стесненным почерком или прогрессирующим уменьшением почерка [2].

Рис. 1 – Почерк пациента с болезнью Паркинсона

Оба критерия хорошо поддаются автоматическому выделению и оценке. Так, для определения текущего состояния и динамики развития микрографии, достаточно несколько образцов почерка на листах одного или схожего размера, например, А4, однако, наличие линейной сетки на листе позволит получить более точные данные. Для анализа второго фактора в качестве меры может быть принято отклонение элемента рукописного символа от его скелета, либо кривизна самого скелета, если для его выделения использовалась метрика среднего отклонения от границы изображения. Сумма отклонений, деленная на общее число выделенных элементов, принимается за текущий показатель. Образец почерка после скелетизации представлен на рисунке 2.

Существуют так же и другие параметры, опущенные в данной статье в виду большей сложности расчёта и автоматизации, а также меньшей корреляцией с неврологическими заболеваниями, например, сила нажатия и количество ошибок.

Индивидуальные особенности почерка отдельного человека и погрешности при оцифровке, если сбор образца производился не в цифровом виде, приводят к существенному разбросу параметров от человека к человеку, что делает невозможным выявление небольших отклонений без анализа динамики.

Описанный подход является эффективным, однако, с его помощью может выявлять неврологические заболевания на ранних стадиях только при анализе динамики изменения параметров, что требует накопления информации об отдельном пациенте. Внедрение данного подхода в процесс диагностики неврологических заболеваний может повысить эффективность более дорогостоящих и долгих исследований и анализов.

Рис. 2 – Образец почерка после скелетизации

Список использованных источников:

1. Ziliotto A, Cersosimo MG, Micheli FE. Handwriting Rehabilitation in Parkinson Disease: A Pilot Study / Micheli FE Ziliotto A, Cersosimo MG // Ann Rehabil Med. — 2012. — no. 39. — 586 P.
2. Lerner, A.J. A Dictionary of Neurological Signs / A.J. Lerner // Springer. — 2010. — no. 10. — 221 P.

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ОЦЕНКИ НАВЫКОВ ПОЛЬЗОВАТЕЛЯ

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

Видничук В. Н

Нестеренков С.Н. – к.т.н.

Эффективность работы компании зависит от количества и квалификации персонала, работающего в ней. В большинстве случаев, в условиях быстрого роста рынка труда, существуют проблемы при подборе персонала:

- огромное количество вакансий на работу и лиц, желающих трудоустроиться;
- подбора команды персонала для выполнения работы;
- быстрота подбора;
- оценка качества работы подобранного персонала.

Отсюда вытекает проблема подбора персонала традиционными способами.

Для решения данных проблем предлагается рассмотреть математическую модель подбора персонала.

Центральным понятием в данной научной работе является понятие «Подбор персонала».

Подбор персонала - это процесс изучения профессиональных качеств и квалификации рабочего кадра, с целью установления его пригодности для выполнения работы на определенном рабочем месте и выбора из совокупности претендентов наиболее подходящего, с учетом соответствия его квалификации, специальности, личных качеств и способностей.

Для оценки навыков пользователя системы используются экспертные оценки и различные коэффициенты. Данная оценка выставляется экспертом или системой за определённый навык тестируемого. Каждому навыку выставляется своя оценка и, далее, она картелируется, путём использования различных коэффициентов. Параметр достоверности экспертной оценки, который зависит от стажа работы эксперта, от пользователей, впоследствии принявших или не принявших тестируемого на работу.

Данный коэффициент рассчитывается по следующей формуле:

Где – параметр характеризующий достоверности оценки эксперта, – параметр становления оценки от времени, – время стажа эксперта в данной области, – параметр, зависящий от количества тестируемых, принятых на работу, согласно оценке данного эксперта.

высчитывается по следующей формуле:

Где – количество всех оцененных пользователей, - количество принятых на работу в связи с этой оценкой.

Оценка за навык выставляется следующим образом:

Где – оценка за навык пользователя, – экспертная оценка навыка пользователя, – параметр достоверности оценки навыка, - количество оценок по данному навыку у пользователя, – параметр учёта данной оценки, зависящий от давности её выставления.

Зная оценки пользователя по определённым навыкам, можно сформировать его оценку квалификации в определённых областях. Для их определения можно воспользоваться следующей формулой:

Для подбора квалифицированных сотрудников, значение данной формулы у тестируемого должно стремиться к максимальному значению.

Для подбора персонала с максимальным навыком можно воспользоваться следующим выражением:

В которой параметр средней квалификации персонала должен стремиться к максимальному значению, заработная плата не должна превышать инвестиции, а время выполнения проекта не должно превышать ожидаемое.

Список использованных источников:

1. Нестеренков, С.Н. Метод определения персональных весовых коэффициентов преподавателей при распределении их нагрузки / С.Н. Нестеренков // Вести Института современных знаний. - 2015. - N 1. - С. 74-80.

МЕТОДЫ И АЛГОРИТМЫ ВЕДЕНИЯ ЭЛЕКТРОННОГО

ДОКУМЕНТООБОРОТА

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

Горбачев М.С.

Глухова Л.А. – к.т.н., доцент

Любая современная организация на каждом этапе своей деятельности вынуждена сталкиваться с ведением огромного числа документов: акты, протоколы, приказы, квитанции и т.д. Если организация использует бумажный документооборот, то он рискует столкнуться с затянутыми процессами по передаче документов, поиском документа в архивах, случайными потерями крайне важных документов. С целью устранения недостатков бумажного документооборота используются системы электронного документооборота (СЭД). Каждая СЭД может отличаться своим набором алгоритмов по работе с документами, а также, их обороту в организации. В докладе выполнен сравнительный анализ существующих алгоритмов и методов, используемых в СЭД.

Независимо от сложности и важности, любой документ проходит стадии создания, согласования, утверждения и подписи, после чего его распространяют и заносят в архив на хранение. Если документ является крайне сложным и большим, то работой по его созданию могут заниматься сразу несколько сотрудников одновременно. Также, документ может проходить различные стадии бизнес-процесса, проходя вперед по ним или возвращаясь назад, для доработки.

С целью уменьшения недостатков бумажного документооборота, в том числе затрат на принадлежности, используемых для структурирования документов, временных затрат сотрудников, которые могли бы пойти на увеличение прибыли компании, затрат на зарплаты курьеров и бумагу, используются СЭД.

СЭД – автоматизированная многопользовательская система, сопровождающая процесс управления работой иерархической организации, с целью обеспечения выполнения этой организацией своих функций. При этом предполагается, что процесс управления опирается на человеко-читаемые документы, содержащие инструкции для сотрудников организации, необходимые к исполнению [1].

Сейчас практически все организации, как малого, среднего, так и крупного бизнеса, и государственные структуры, используют различные системы для автоматизации, контроля и управления оборотом документов, что приводит к повышению производительности труда сотрудников и уменьшениям издержек организации. На территории СНГ лидерами среди СЭД являются системы Дело, DocsVision, 1С, Directum [2]. Данные СЭД в своих подходах реализуют либо устоявшиеся подходы бумажного документооборота, но адаптированного под электронные технологии, либо используют новые подходы в решении задачи управления документами.

Достоинства вышеприведенных СЭД, обеспеченные используемыми алгоритмами:

- высокая скорость передачи документов;
- контроль исполнения документов. СЭД позволяет производить отслеживание сроков исполнения документа, а также, на каком этапе выполнения (бизнес-процесса) он находится;
- использование ролей для распределения прав доступа к документам. Данное решение ограничивает доступ сотрудникам к документам, которые имеют более другие права доступа. Таким образом, сотрудники не смогут получить доступ к документам, на которые у них нет прав, что в свою очередь повышает конфиденциальность данных. Кроме того, это ускоряет и упрощает поиск документов, сужая возможный круг документов только до доступных работникам;
- хранение документов в собственной базе данных. Сочетая такой подход с распределенными базами данных (которые хранят одну и ту же информацию, из-за чего вывод из строя одной машины не повлечет долговременный простой всей системы) СЭД сильно повышают сохранность документов;
- быстрый поиск, как часть алгоритма работы с документами, так как это одна из наиболее часто используемых операций, при работе с документами вообще [3];

Недостатки, обусловленные использованием тех или иных алгоритмов вышеприведенных СЭД:

- сложная настройка систем. Большинство СЭД имеют сложную систему настроек, наладить которую могут только фирмы, специализирующиеся на оказании услуг по развертке подобных систем. Это приводит к дополнительным затратам на внедрение системы и невозможности или дороговизне (за счет содержания штата работников) поддержки использования системы;
- сложное использование систем. Из-за слабой гибкости в настройках, данные решения крайне сложны для ведения простых бизнес-процессов (ведение учета для малого бизнеса);
- низкая скорость работы самих систем (плохая оптимизация алгоритмов). Многие из приведенных вышесистем имеют большое количество проблем, связанных с медленной работой, тогда как пользователь привык к тому, что компьютерные программы должны отличаться высокой скоростью ответа и обработки.

В докладе рассматриваются принципы организации документооборота в вышеприведенных системах и их сравнение, с помощью которого можно выявить сильные и слабые стороны используемых в реальных продуктах алгоритмов.

Список использованных источников:

1. Documentautomation [Электронный ресурс]. – 2017. – Режим доступа :https://en.wikipedia.org/wiki/Document_automation
2. Мировой рынок систем электронного документооборота [Электронный ресурс]. – 2017. – Режим доступа: <http://citforum.ru/consulting/docflow/market/article1.8.200222.html>
3. Система электронного документооборота[Электронный ресурс]. – 2017. – Режим доступа: <http://www.escom-bpm.com/services/51.html>

АНАЛИЗ АРХИТЕКТУРЫ ДЕЦЕНТРАЛИЗОВАННОЙ СИСТЕМЫ БЕЗ ПОСРЕДНИКОВ

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

Горбачевский Н.А.

Серебряная Л.В. – к.т.н., доцент

Для того чтобы быть востребованными на рынке программного обеспечения, современные системы должны соответствовать определенным требованиям, таким как надежность, высокая доступность, отказоустойчивость, целостность, безопасность, быстрое действие, а также многим другими. В связи с этим большое внимание должно уделяться этапу проектирования архитектуры системы. Поскольку только тщательно проработанная и хорошо продуманная архитектура может позволить приложению соответствовать всему набору вышеперечисленных требований.

Данная работа является результатом сравнительного анализа архитектур наиболее успешных представителей децентрализованных систем без посредников на основе цепочки блоков транзакций. Анализ базируется на результатах регулярно проводимых тестов производительности, изучении документации и исходного кода приложений, а также опыте их практического использования [1]. Задачей данной работы является поиск как удачных, так и неудачных решений с целью определения принципов, которые обязательно должны быть учтены при проектировании архитектуры подобной системы. В результате данного анализа был сформирован следующий набор архитектурных подходов к созданию программной системы:

1. Принцип модульности. Система должна быть разделена на компоненты, в соответствии с выполняемыми задачами. Модули должны загружаться лишь по требованию приложения и быть полностью заменяемыми и опциональными, если это возможно. Механизмы взаимодействия между модулями также должны быть заменяемыми. Например, на первых этапах развития приложения модули могут взаимодействовать посредством прямых системных вызовов, в то время как в будущем они могут быть разделены и взаимодействовать между собой по сети [2].
2. Горизонтальное масштабирование. Даже эффективное использование всех ресурсов одного вычислительного узла не всегда позволяет обеспечить необходимые показатели быстродействия и выдержать необходимую нагрузку. В связи с этим все узлы системы должны быть гомогенными с целью обеспечения горизонтального масштабирования. Изменение количества узлов должно линейно изменять количество обрабатываемых запросов и объем хранимых данных, но не влиять на работоспособность всей системы. Использование данного подхода в сочетании с принципом модульности позволяет независимо масштабировать отдельные компоненты, а не все приложение целиком.
3. Разбиение всего набора данных на диапазоны. Для обеспечения отказоустойчивости, копия данных должна храниться на нескольких вычислительных узлах [3]. В случае выхода из строя некоторых из них, остальные узлы смогут продолжать обработку запросов для того же набора данных. В существующих системах каждый узел может обработать любой пользовательский запрос, так как он хранит полную копию всех данных. Это означает, что изменения, произведенные на одном узле, должны быть синхронно отражены на всех остальных узлах системы, иначе целостность данных будет нарушена, и последующий пользовательский запрос может не прочитать ранее записанные данные. Фактически, это означает, что обработка всех запросов должна выполняться строго последовательно, что влечёт за собой отсутствие параллелизма на уровне всей системы. Лучшим же решением является разделение всех данных на некоторое количество диапазонов и равномерное их распределение между всеми вычислительными узлами. В таком случае каждый узел будет хранить свою активную часть данных и заданное число копий данных других узлов. Однако обрабатывать пользовательские запросы он будет лишь для своего активного диапазона данных и асинхронно отображать их на других узлах, хранящих копию данного диапазона. В случае выхода какого-то узла из строя, другой узел, на котором присутствует копия активных данных первого узла, продолжит обработку пользовательских запросов уже для двух диапазонов. Это позволит сохранить целостность данных, а также обеспечить высокий уровень параллелизма на уровне всей системы.

В настоящее время еще не существует децентрализованной системы без посредников, которая бы использовала все вышеуказанные подходы. Следовательно, создание новой системы, соответствующей всем заданным принципам, является актуальной и востребованной задачей.

Список использованных источников:

1. <https://lists.hyperledger.org> [Электронный ресурс]. – Режим доступа: <https://lists.hyperledger.org/pipermail/hyperledger-fabric/2017-November/002041.html>. – Дата доступа: 19.03.2018

2. <https://martinfowler.com> [Электронный ресурс]. - Режим доступа: <https://martinfowler.com/articles/microservices.html>. - Дата доступа 19.03.2018

3. Richard L. Shuey. The Architecture of Distributed Computer Systems. / Richard L. Shuey, Ophir Frieder, David L. Spooner. - Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1997

НЕОДНОРОДНАЯ РАСПРЕДЕЛЕННАЯ ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА

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

Грачев Я.Ю.

Петровский Н.А. – к.т.н., доцент

Современные технологии предъявляют возрастающие требования к вычислительным мощностям компьютерной техники. Большинство решений предполагает использование технологий CUDA на GPU, сопроцессоров Intel Xeon Phi или решений на FPGA, но все больше внимания уделяется возможностям использования процессоров архитектуры ARM. Выпускается огромное количество мобильных устройств с этими процессорами, что делает их весьма доступными для построения распределенных вычислительных систем.

Классические вычислительные системы строятся с использованием одинаковых компонент, которые имеют специальные интерфейсы сверхскоростного обмена данными между вычислительными узлами, а также, общую память. Рассматриваемая система строится путем объединения мобильных ARM устройств через TCP/IP и центральный узел – диспетчер. В виду этого, такими свойствами, как общая память и наличие сверхскоростных интерфейсов обмена данными, такая система не обладает. При этом, каждый узел такой системы является неоднородным, имея разную вычислительную производительность на ядро, разное количество ядер, размер и скорость работы с памятью и, как следствие всего – разную вероятность выполнения вычислительной задачи за некоторый порог времени.

Главным и неоспоримым преимуществом неоднородной распределенной вычислительной системы является большое количество вычислительных узлов.

Основными недостатками такой системы являются:

- низкая скорость обмена данными;
- ограниченное количество памяти;
- ограниченность вычислительного ресурса;
- неоднородная производительность;
- вероятностное выполнение.

Эти недостатки нивелируются при решении определенного класса задач, которые требуют малое количество данных, необходимых для начала вычислений, а также, которые имеют достаточно долгое время вычисления относительно времени, затрачиваемого на передачу и распараллеливание таких вычислений. К таким задачам можно отнести вычислительные методы и криптографию.

Большинство криптографических методов требуют достаточно большое количество данных необходимых для вычислений, а также, требуют последовательного выполнения из-за зависимости текущего результата от предыдущего. Проблема невозможности распараллеливания решается при использовании специальных параллельных криптографических функций, например, семейства хэш-функции Skein [1], но это не решает проблему количества данных и времени их передачи, делая подобные вычисления неэффективными и бессмысленными. Однако, при обратной криптографии – взломе или подборе хэш-функций, необходимость передачи больших объемов данных отсутствует. Это позволяет распределять задачу на множество вычислительных узлов, аналогично тому, как работают вычислительные системы криптовалюют.

При решении задач с численными методами, каждому вычислительному узлу системы достаточно вычислять некоторую часть общей задачи, возвращая результат [2]. Но здесь, также, дополнительно, требуется учитывать, что время, которое потребуется узлу на решение его части общей задачи, должно быть больше времени, которое будет затрачено на передачу всех данных и работу с ними.

Таким образом, неоднородные распределенные вычислительные системы с вероятностным выполнением, на основе устройств, с процессорами ARM, позволяют эффективно решать определенный класс задач, а вместе с ростом производительности и уменьшением техпроцесса, в перспективе, возможности будут только расширяться.

Список использованных источников:

1. Atighehchi, K.Enache, A.Muntean, T., &Risterucci, G. (2010). An Efficient Parallel Algorithm for Skein Hash Functions. IACR Cryptology ePrint Archive, 2010, 432.

2. Dimitri P. Bertsekas and John N. Tsitsiklis. 1997. Parallel and Distributed Computation: Numerical Methods. AthenaScientific.

ПРИМЕНЕНИЕ НЕЙРОННОЙ СЕТИ ХОПФИЛДА ДЛЯ РЕШЕНИЯ ЗАДАЧИ ОПТИМАЛЬНОГО РАССЕЛЕНИЯ СТУДЕНТОВ ОБЩЕЖИТИЯ

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

Гурин Д.Н.

Медведев С.А. – к.т.н., доцент

Нейронные сети получили широкое применение в самых различных областях информационных технологий. Возможность обучения нейронной сети, а также её ключевая функция распознавания образов могут быть успешно применены для решения задачи оптимального расселения студентов в общежитии. В общем случае может решаться любая задача распределения людей или объектов по некоторым помещениям / категориям на основании набора признаков каждого из объектов.

В качестве нейронной сети, которая будет использоваться для оптимального расселения студентов, взята нейронная сеть Хопфилда. Алгоритм обучение сети Хопфилда существенно отличается от таких классических алгоритмов обучения персептронов, как метод коррекции ошибки или метод обратного распространения ошибки. Отличие заключается в том, что вместо последовательного приближения к нужному состоянию с вычислением ошибок, все коэффициенты матрицы рассчитываются по одной формуле за один цикл, после чего сеть сразу готова к работе.

За основу может быть взята как дискретная, так и непрерывная модель. Дискретная модель потребует для реализации вектора большей длины, поскольку нейроны смогут включать в себя только 2 состояния $S(t) = \{-1, 1\}$. В непрерывной модели, состояний нейрона, формально, бесконечно много, а фактически, зависит от точности вычислений компьютера. Схема нейронной сети Хопфилда представлена на рисунке 1.

Рис. 1 – Схема нейронной сети Хопфилда с тремя нейронами.

На начальном этапе необходимо определить пространство признаков, которому будут все обучаемые и распознаваемые нейронной сетью объекты. В данном случае объекты, которые представляют собой информацию о студентах, будут обладать следующими признаками:

- 1) пол студента;
- 2) факультет, на котором обучается студент;
- 3) обучение в магистратуре или на первой ступени образования;
- 4) наличие / отсутствие льгот для проживания в общежитии;
- 5) наличие / отсутствие заслуг перед университетом / общежитием;
- 6) наличие / отсутствие нарушений дисциплины и порядка в университете или общежитии.

Каждый из описанных ранее признаков может быть разбит на некоторое количество более мелких признаков, например, наличие льгот можно расписать, как наличие инвалидности, наличие статуса сироты, наличие удостоверения человека, проживающего в чернобыльской зоне, наличие справки о неполной семье и т. д.

В результате, используя данные студента, можно сформировать вектор вида $x \rightarrow \{1, 1, -1, 1, -1, 1, 1, -1, \dots\}$ размерностью S . Каждый нейрон данного вектора будет описывать один из признаков объекта. Допустим, в первом элементе, обозначающим пол студента, значение 1 будет равно мужскому, -1 – женскому. Вторым элементом будет определять степень образования: значение 1 – первая степень, -1 – магистратура и так далее. После определения пространства признаков необходимо осуществить обучение нейронной сети корректными образцами. Максимальное количество образцов можно определить по формуле (1):

$$K = S^2; \quad (1)$$

где S – число элементов вектора признаков.

Обучение нейронной сети осуществляется посредством формирования матрицы весовых коэффициентов по формуле (2):

$$w_{ij} = x_i x_j; \quad (2)$$

где S – количество обучающих образцов, – транспонированный вектор-образец, – вектор-образец.

В результате будет получена матрица размерностью $S \times S$, которая будет применяться для классификации и восстановления получаемых образцов. В полученной матрице необходимо обнулить главную диагональ.

Обучив нейронную сеть Хопфилда выборкой из корректных образцов, можно выполнять классификацию данных произвольного студента. Результаты данной операции позволят нам определить, может ли студент претендовать на заселение в общежитие, и если да, то такой блок ему подходит больше всего.

Для того чтобы обработать данные студента нейронной сетью, необходимо представить их в векторном виде. Алгоритм формирования вектора признаков был представлен выше. Зададим обозначение данному вектору, например, u . Перемножив весовую матрицу на вектор u , содержащий данные произвольного студента, и подставляя полученный результат в функцию активации (в данном случае можно использовать функцию знака), получаем вектор u' . Если этот вектор не совпадает ни с одним вектором из обучающей выборки, то необходимо выполнить данную операцию с этим вектором повторно. Вектор необходимо

подвергать итерациям восстановления до тех пор, пока он не совпадёт с одним из векторов из обучающей выборки.

Список использованных источников:

1. Хайкин, С. Нейронные сети. Полный курс. Второе издание / С. Хайкин. – Нейронные сети. – 2006.- 373 с.
2. Нейронная сеть Хопфилда на пальцах [Электронный ресурс] – Режим доступа: <https://habrahabr.ru/post/301406/>.

ПРОГРАММНОЕ СРЕДСТВО “ПЛАНИРОВАНИЕ ЛИЧНЫХ ФИНАНСОВ”

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

Давидович А. С.

Шульдова С. Г. – к.т.н., доцент

В современном мире очень важно уметь рационально распределять свои денежные ресурсы, особенно в условиях их постоянной нехватки. Всегда существовала и существует проблема безграничности человеческих потребностей и ограниченности экономических ресурсов, эта проблема особенно ярко выражается в условиях существующего экономического кризиса [1]. Поэтому необходимо уметь правильно планировать личные финансовые ресурсы.

В свою очередь финансовое планирование – отдельная наука, которая в свою очередь требует соответствующего изучения. Однако в текущей реалии обучение финансовой грамотности в школах происходит достаточно редко, тем самым, впоследствии, вызывая финансовую безграмотность населения. Что приводит к таким последствиям, когда человек оказывается в долговой яме.

Финансовое планирование – это управление процессами создания, распределения, перераспределения и использования финансовых ресурсов на предприятии или для личных целей [2]. Планирование финансов формируется в соответствии с интересами соответствующего поколения, его окружающей среды. Использование финансового планирования в личных целях возможно посредством личного финансового плана.

Личный финансовый план – это индивидуально разработанный план действий по достижению желаемых финансовых целей, включая подбор подходящих кредитных, инвестиционных, страховых, пенсионных и иных финансовых продуктов.

Личный финансовый план включает составленный с точностью до года план финансовых действий, в том числе:

- План по достижению финансовых целей, включая сроки, стоимость их выполнения, а также финансовые продукты, необходимые для этого.
- План по использованию инвестиционных продуктов, соответствующих допустимому уровню риска и срокам достижения целей, размеру вложений в эти продукты и их соотношению в вашем портфеле.
- План по использованию страховых программ, включая перечень рисков, страховые суммы, срок страхования и т.д.
- План по пенсионному обеспечению, включая подбор вариантов увеличения государственной пенсии, и подбор инструментов для формирования негосударственной пенсии.
- План по использованию кредитных инструментов, включая вид кредита, срок, объем и т.д.

Личный финансовый план позволяет создать рациональную стратегию достижения финансовых целей с помощью грамотно подобранных инвестиционных, страховых, пенсионных и кредитных продуктов, исходя из соответствующего финансового состояния.

Финансовый план может составляться как для одного человека, так и для семьи из нескольких человек.

Финансовый план составляется сроком до самой дальней цели, которую человек определяет для себя.

Также рекомендуется осуществлять периодическую корректировку личного финансового плана, в зависимости от изменившихся фактов [3].

Личный финансовый план на данный момент могут составить специалисты за определенную плату. Однако, даже если клиент и готов потратить конкретную сумму, то ему необходимо идти куда-то, вследствие чего, тратить свое время.

Исходя из изученного материала, было решено создать программное средство, которое поможет физическим лицам вести личный бюджет, анализировать, производить его оценку, а также будет помогать в составлении личного финансового плана и его корректировке.

Список использованных источников:

1. Татарникова А. О. Актуальность формирования навыков финансового планирования у населения // VIII Международная научно-практическая конференция Северо-Западного института управления РАНХиГС при Президенте РФ : тез. докл. Международной научно-практической конференции, Санкт-Петербург, 20-22 апреля 2016 г. – Санкт-Петербург : 2016. – 244с.
2. Финансовое планирование и разработка бюджета предприятия – http://globalteka.ru/books/doc_details/4524-----.html
3. Что такое личный финансовый план? – <http://www.azbukafinansov.ru/articles/index.php?article=150>

ВОССТАНОВЛЕНИЕ ДАННЫХ МЕТОДОМ ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ

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

Даниленко А.А.

Серебряная Л.В. – к.т.н., доцент

Использование всех доступных данных может значительно сказаться на результате работы прогнозной модели, поэтому важно грамотно подготовить имеющиеся данные и уметь эффективно решать проблему их отсутствия. Из-за большого объема используемой информации требуется найти максимально быстрый и оптимальный способ восполнения пробелов в данных. Применение динамического программирования может значительно улучшить классические методы решения этой проблемы, а также повысить эффективность вычислений.

Во многих приложениях невозможно исключить неполные данные, так как это приведет к искаженным и смещенным результатам или вовсе не позволит построить прогнозную модель, ведь даже широко используемые нейронные сети и опорные векторы имеют проблемы с использованием неполных данных. Игнорирование отсутствующих данных может привести к критическому сокращению обучающей выборки, что также отразится на качестве конечного прогноза. Поэтому основной задачей в процессе подготовки к работе является восстановление недостающей информации. Однако конечной целью является построение модели, что ставит определенные ограничения на скорость первоначальной обработки данных.

Задачей данной работы является обоснование и применение метода динамического программирования для решения проблемы отсутствующих данных.

Эффективность применения динамического программирования для решения данной проблемы базируется на принципе оптимальности Беллмана, который утверждает, что оптимальная стратегия зависит только от текущего состояния системы и первоначальной задачи, не зависит от предыдущих результатов, и может быть представлен следующим образом:

где γ – фактор дисконтирования, а U – вспомогательная функция.

Таким образом, динамическое программирование сводится к задаче оптимизации, при этом учитывается:

где x – уже известные данные, доступные для будущей оптимизации, а U – решение, которое будет принято при оптимизации, а g – допустимые помехи.

В контексте проблемы восстановления данных этот подход позволяет подбирать оптимальное решение для каждой подзадачи, в итоге приводя к сбалансированным данным.

Метод динамического программирования для решения глобальной проблемы отсутствующих значений используется для разбиения задачи на мелкие подзадачи – отсутствие данных в конкретной записи, с которыми намного проще справиться. Результат выполнения подзадачи сохраняется, что позволяет избежать повторения одних и тех же вычислений.

В результате, обработка данных сведется к трем возможным действиям:

1. ничего не делать (возможно в том случае, когда данные полные);
2. обратиться к памяти (если похожая проблема уже решалась);
3. предсказать значение (ранее похожая проблема не решалась).

Динамическое программирование – полезный инструмент в борьбе с недостающими данными, благодаря которому, можно значительно ускорить процесс обработки данных и заполнения пробелов в них, так как при его использовании извлекается максимум выгоды из повторяющихся записей, а также в процессе вычисления исключаются не самые оптимальные варианты.

Список использованных источников:

1. F.V. Nelwamondo, Computational intelligence techniques for missing data imputation, Ph.D. Thesis, University of the Witwatersrand, Johannesburg, 2008.
2. T. Marwala, Computational Intelligence for Missing Data Imputation, Estimation, and Management: Knowledge Optimization Techniques, Information Science Reference, USA, 2009.
3. Bertsekas, D. P.: 2005, Dynamic Programming and Optimal Control, Athena Scientific, Belmont, Massachusetts.

ОСОБЕННОСТИ КОМПЬЮТЕРНОГО РАСПОЗНАВАНИЯ НОМЕРОВ ВАГОНОВ ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА

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

Одной из актуальных задач, решаемых в автоматизированных системах фото и видеофиксации процесса взвешивания вагонов железнодорожного транспорта, является распознавание их номеров. Рассмотрены проблемные вопросы обработки кадров с изображением вагонов. Разработано программное средство, реализующее задачу распознавания.

В результате работы автоматизированной системы фото и видеофиксации процесса взвешивания железнодорожных вагонов, на носитель информации записывается последовательность кадров, фиксирующих эту процедуру. Идентифицирующим атрибутом железнодорожного вагона является его номер. Поэтому актуальным является автоматическое распознавание этого номера. В процессе решения этой задачи проводится обработка фотографии вагона на платформе взвешивания. Это изображение имеет разрешение 1920x1080 и получается видеокамерой, установленной на улице возле весовой платформы. Такое разрешение снимка позволяет применить алгоритм компьютерного распознавания номера даже в том случае, когда он занимает даже незначительную часть изображения [1].

При решении этой задачи следует учитывать ряд особенностей изображения:

- номера ж/д вагонов на фотографиях, как правило, расположены под углом, что обычно обусловлено невозможностью установки уличной камеры под прямым углом, перпендикулярно железнодорожной весовой платформе;
- насыщенность бликами изображения, полученного при восходе или закате солнца;
- номера вагонов всегда расположены в разных местах фотографии (для разных систем камеры всегда установлены по-разному);
- номера вагонов нарисованы краской с помощью трафарета, поэтому на многих фотографиях нет четких границ между цифрами и, достаточно часто, имеются подтеки краски при высыхании;
- на фотографиях помимо номеров вагонов имеются другие надписи и цифры (символы);
- номера вагонов могут содержать цифры «0» и «1», наиболее сложные для распознавания;
- цифры, из которых состоит номер, расположены очень близко друг к другу, иногда их области даже пересекают границы друг друга;
- многие цифры нарисованы прерывистой линией;
- на фоне расположения номера присутствуют шумы и помехи;
- иногда изображения настолько нечеткое, что даже человеку сложно распознать номер вагона на фотографии.

В качестве примеров на рисунках 1,2 изображены варианты представления номеров для автоматического компьютерного распознавания.

Рис. 1 – Пример фотографии со «сложным» номером для распознавания

Рис. 2 – Пример фотографии с «простым» номером для распознавания

Для решения задачи распознавания номера вагона разработано программное средство (ПС), использующее библиотеки Tesseract, Tessnet2, Asprise [2,3], применяемые для распознавания букв и цифр. Сначала, пользователь сам выделяет на обрабатываемом снимке область расположения номера вагона, которая затем, используется в качестве исходных данных для работы ПС по распознаванию. ПС использует двухэтапный статистический анализ. Для повышения точности распознавания ПС настроено и сконфигурировано только на распознавание цифр. Также, принято допущение, что номер вагона всегда имеет длину, равную 8 символам.

Эксперименты показали, что программное средство гарантирует хорошую достоверность (более 95%) распознавания на «простых» номерах ж/д вагонов и очень недостаточную (около 60%) на «сложных» номерах.

Примеры работы программного средства показаны на рисунке 3.

Рис. 3 – Примеры работы программного средства распознавания

В результате выполнения экспериментального исследования на 200 примерах, библиотеки Tesseract, Tessnet2, Asprise показали распознавания номера железнодорожного вагона около 60%.

Список использованных источников:

1. Деменковец Д.В. Система автоматической фото, видеофиксации и записи процесса взвешивания железнодорожных вагонов // Информационные технологии и системы 2017 (ИТС 2017): материалы международной научной конференции, БГУИР, Минск, Беларусь, 25 октября 2017 г. / редкол.: Л. Ю. Шилин [и др.]. — Минск: БГУИР, 2017. —305 с.

2. Википедия Свободная энциклопедия - Tesseract [Электронный ресурс] / – Режим доступа:<https://ru.wikipedia.org/wiki/Tesseract> – Дата доступа: 13.03.2018.

3. Набор API для .NET чтобы осуществлять оптическое распознавание символов и меток [Электронный ресурс] / – Режим доступа: <https://ru.products.aspose.com/ocr/net> – Дата доступа: 13.03.2018.

ЗАЩИТЫ ЛИЧНЫХ СООБЩЕНИЙ ДЛЯ СОЦИАЛЬНЫХ СЕТЕЙ

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

Демидович А.В.

Прохорчик Р.В. – ст. преподаватель каф. ПОИТ, м.т.н.

В последние годы социальные сети всё активнее используются не только для личного общения, но и для решения деловых задач. К примеру, специалисты по подбору персонала нередко используют такие ресурсы для поиска кандидатов на соответствующие позиции. Любая социальная сеть предполагает предоставление ей некоторых сведений о зарегистрированном пользователе. Но большинство пользователей не ограничивают доступ к своим данным, что говорит о том, что они предоставляют доступ к большому объему сведений, чего они изначально, возможно, и не хотели.

Учитывая то, как устроены социальные сети, параметры конфиденциальности ваших друзей непосредственно влияют и на вашу конфиденциальность. Если эти параметры недостаточно строгие, ваши записи сможет просматривать множество посторонних людей.

Однако, даже если установлены строгие параметры, опубликованная вами информация может распространяться по сети. В некоторых социальных сетях вашим контактам разрешено копировать и повторно публиковать ваши исходные записи. Посторонние люди также смогут видеть ваши частные записи, если ваш друг пометит их тегом. Стоит помнить, что вся информация остается в сети, поэтому следует публиковать только такие сообщения и фотографии, которые не могут скомпрометировать вас перед окружающими. В итоге вы не можете быть уверены, что предоставляете доступ к своим записям только соответствующему кругу лиц.

В 2016 году Facebook ввел шифрование данных и сообщений в Messenger. По умолчанию она выключена и активировать ее придется в каждом отдельном чате. При этом групповые беседы зашифровать не получится. Чтобы включить шифрование надо выбрать соответствующую иконку собеседника и выбрать пункт «Секретная переписка». Правда использование секретной переписки накладывает определенные ограничения. Например, вести ее получится только с одного устройства, а у вашего собеседника должна быть установлена последняя версия Messenger. Шифрование сообщений в Facebook Messenger должно включаться вручную для каждой переписки, оно не включено по умолчанию, как в WhatsApp, Signal или Telegram. Такой компромисс может быть продиктован желанием избежать проблем с силовыми органами.

Степень доверия к защите, предоставляемой пользователю со стороны социальных сетей, далеко не всегда может быть оправдана, когда речь о личных данных, или, тем более, строго конфиденциальных, связанных с деловыми задачами.

Цель разрабатываемого мною программного модуля направлена именно на обеспечения вспомогательного слоя защиты. Архитектура приложения представлена на рисунке 1. Это будет расширение для браузера и основывается на DOM. И как результат – будет иметь низкие накладные расходы и может найти практическое использование сегодня.

Рис. 1 – Архитектура приложения

Особенность моего приложения — в исключительном удобстве. Пользователи, которые его установили и отправляют/получают сообщения, видят в браузере сразу расшифрованный вариант, то есть нормальный текст. Шифрование и расшифровка осуществляются автоматически с помощью криптобиблиотеки в момент отправки/получения личного сообщения. При этом, по открытым каналам связи передаётся только зашифрованная информация, и для постороннего человека опубликованное сообщение будет нечитаемым.

Список использованных источников:

1. GOEL, V., AND WYATT, E. Facebook privacy change is subject of f.t.c. inquiry. <http://nyti.ms/19WVMV8>.
2. KURT OPSAHL. Facebook's eroding privacy policy: A timeline. <http://goo.gl/BkRknm>.

ОБУЧАЮЩИЙ МОДУЛЬ 'ШАХМАТЫ'

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

Дубовская Е.В.

Данилова Г.В. – м.т.н., ассистент

С 2016 года в Беларуси под патронажем Министерства спорта и туризма, Национального олимпийского комитета и государственно-общественного объединения «Президентский спортивный клуб» реализуется пилотный проект «Шахматы – школе». 300 учреждений общего среднего образования Беларуси оснащены шахматным инвентарем для проведения факультативных занятий «Шахматный всеобуч». Однако, остро стоит вопрос методического обеспечения учебного процесса. Внедрение данной компьютерной программы поможет учителям начальных классов в успешной реализации проекта.

Шахматы – настольная логическая игра со специальными фигурами на 64-клеточной доске для двух соперников, которая, считается, существует более полутора тысяч лет.

История шахматных машин старше, чем история компьютеров. Идея создать машину, играющую в шахматы, датируется ещё восемнадцатым веком. Около 1769 года появился шахматный автомат «Механический турок». Он был предназначен для развлечения королевы Марии-Терезии. Машина действительно неплохо играла – внутри неё находился сильный шахматист, который и делал ходы. Первой же машиной, которая достигла уровня шахматного мастера, была Belle (англ.), законченная в 1983 году Джо Кондоном и Кеном Томпсоном. Belle был первым компьютером, спроектированным только для игры в шахматы. Его официальный рейтинг Эло был 2250, таким образом, это была самая сильная шахматная машина своего времени. В феврале 1996 года Гарри Каспаров победил шахматный суперкомпьютер Deep Blue со счетом 4-2. Этот матч, выдающийся тем, что первую партию выиграл *Deep Blue*, автоматически став первым компьютером, победившим чемпиона мира по шахматам в турнирных условиях. *Deep Blue* вычислял 50 миллиардов позиций каждые три минуты, в то время как Каспаров -10позиций за это же время. В *Deep Blue* было 200 процессоров. С тех пор шахматные энтузиасты и компьютерные инженеры создали много шахматных машин и компьютерных программ.

Множество очень интересных и красивых задач на шахматной доске возникает при решении двух следующих комбинаторных проблем:

1. Какое максимальное число одноименных фигур (ферзей, ладей, слонов, коней, королей) можно расставить на шахматной доске так, чтобы никакие две из них не угрожали друг другу;
2. Какое минимальное число одноименных фигур (ферзей, ладей, слонов, коней, королей) можно

расставить на шахматной доске так, чтобы они держали под обстрелом все свободные поля доски. Не трудно заметить, что здесь имеется явная аналогия с рядом важных задач из теории графов. Данные задачи предлагаются для решения на шахматной доске ученикам 1-2 года обучения. Хотя за решение задачи о 1000 ферзях была назначена награда в 1 млн.\$ (на 3 сентября 2017г).

Идея программы заключается в комбинировании двух типов задач: стандартные шахматные головоломки (ответы на которые известны большинству шахматистов) и задач для обучения правилам хода фигур (доберитесь до зайца, не наступив на елки).

Рис. 1 – Игровое поле

Задача пользователя: расставить как можно больше шахматных фигур (ферзей, королей, ладей, слонов), перемещающихся по определенным правилам, на доске так, чтобы они не били друг друга. При этом на поле еще есть сгенерированные случайным образом елки. В программе реализовано 3 уровня сложности:

1 уровень (Новичок). При постановке фигур, в качестве подсказки помечаются еще и те поля, которые она пробивает, при выполнении заданий с ошибками появляется красный индикатор.

2 уровень (Ученик). При постановке фигур, только при выполнении заданий с ошибками появляется красный индикатор.

3 уровень (Студент) При постановке фигур подсказок нет.

Программа имеет прикладное значение. Её внедрение поможет педагогам и родителям воспитывать креативную, талантливую, мыслящую молодежь, способную просчитывать на несколько ходов вперед.

ОСОБЕННОСТИ ИНТЕГРАЦИИ СИСТЕМЫ ДОКУМЕНТООБОРОТА EDI С MICROSOFT DYNAMICSAX

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

Ермаченок В.А.

Таборовец В.В. – к.т.н., доцент

Взаимодействие с бизнес-партнерами, для любого предприятия, связано с необходимостью подготовки, отправки, получения, обработки и хранения большого числа документов. Управление бумажным документооборотом приводит к возникновению как временных, так и денежных затрат. Кроме того, используемый ручной ввод данных часто сопровождается возникновением ошибок. Современные информационные технологии позволяют перевести весь документооборот компании в электронный вид, используя технологию EDI. Внедрение электронного обмена информацией - это важный шаг к повышению качества взаимного сотрудничества, упрощению и увеличению скорости работы с партнерами, возможность избежать рутинных операций оформления и отправки бумажной документации.

Под аббревиатурой EDI понимают Electronic Data Interchange или Электронный Обмен Данными. Проще говоря, EDI – это отправка и получение информации с использованием компьютерных технологий. Благодаря тому, что применение технологий EDI в бизнесе удобно и практично, этот стандарт стал широко использоваться в различных отраслях экономики и социального обслуживания. Любые стандартные деловые документы, которыми, к примеру, одна FMCG компания обменивается с другой (такие как: заказ на поставку, счёт-фактура, план отгрузок, запрос о наличии товара) могут быть переданы при помощи EDI, если обе стороны провели необходимую для этого подготовку.

Стандарт EDI разработан в Американском национальном институте стандартов (ANSI). Наряду с EDI существуют и другие стандарты для электронного обмена данными. Например, EDIFACT широко используется в Европе и в автомобильной промышленности. HIPPA (закон об учете и безопасности медицинского страхования) разработан специально для соответствия деятельности учреждений здравоохранения законодательству.

Преимущества EDI:

- a. Сокращение расходов и экономия времени. Благодаря исключению ненужного этапа оформления бумажных документов, информационный поток становится более эффективным. Скажем, компьютер продавца, подключенный к системе EDI, посылает уведомление и электронные счета, исключая необходимость документального оформления счетов-фактур.
- b. Усовершенствование коммерции B2B. Система EDI позволяет быстро реагировать на деловые запросы и передавать документы с контрольным следом для гарантии точности и согласованности. Благодаря этому улучшаются взаимоотношения между деловыми партнерами. В большинстве случаев деловые партнеры совместно решают вопросы организации EDI и его многочисленных применений. В итоге, улучшается обмен общей информацией и сотрудничество между деловыми партнерами.
- c. Точность и целостность данных. Благодаря исключению лишних средств ввода данных, повышается точность их обработки, а это, в свою очередь, способствует сохранению целостности и повышению надежности бизнес-процесса. Получение более точных и полных деловых операций посредством EDI позволяет улучшить обработку информации в соответствующих приложениях. В

частности, благодаря получению заказа на приобретение посредством EDI неизменно повышается точность работы приложения для ввода заказов на стороне продавца.

Несмотря на ряд преимуществ, EDI присущи следующие недостатки:

EDI еще не получил признания в качестве идеального средства обмена информацией или ведения коммерции. Из миллионов коммерческих предприятий в США лишь менее 200000 приняли EDI;

EDI обходится дорого, поскольку внедрение и техническая поддержка этой технологии требует значительных капиталовложений. EDI носит характер двухточечной связи. Каждый устанавливаемый контакт требует специального оборудования и программного обеспечения;

Для достижения максимальной эффективности EDI требуется организация дорогих сетей с дополнительными услугами (VAN). Такие капиталовложения могут себе позволить только крупные деловые партнеры, связанные с массовым производством;

EDI представляет собой сложную для внедрения, освоения и эксплуатации систему;

Цель работы – разработка программного средства, обеспечивающего интеграцию между ERP-системой и программным средством обработки EDI-сообщений. Разработка данного программного средства предоставляет возможность обработки EDI-сообщений для автоматизированного обмена электронными сообщениями в стандартизированных форматах между бизнес-партнерами.

Список использованных источников:

1. Norbert Reekers & Steve Smithson, The Impact of Electronic Data Interchange on Interorganizational Relationships: Integrating Theoretical Perspectives.

ОБНАРУЖЕНИЕ АНОМАЛИЙ В ТЕЛЕМЕТРИЧЕСКОМ СИГНАЛЕ МАЛОГО КОСМИЧЕСКОГО АППАРАТА НА ОСНОВЕ ЭВРИСТИЧЕСКИХ АЛГОРИТМОВ ВОЗМОЖНОСТНОЙ КЛАСТЕРИЗАЦИИ

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

Ерома А.П.

Вятчин Д. А. – канд. филос. наук, доцент

Одной из важных задач обработки телеметрической информации малого космического аппарата (МКА) является оперативная оценка состояния МКА в процессе его эксплуатации, так как своевременное обнаружение аномалий в телеметрическом сигнале может предотвратить выход из строя бортовой аппаратуры МКА. В виду того, что телеметрическая информация МКА является разнородной по своей природе, а в процессе измерений могут возникнуть внешние факторы, влияющие на её точность и достоверность, перспективным является направление обработки телеметрической информации с помощью методов машинного обучения.

Целью данной работы является обнаружение аномалий в телеметрическом сигнале системы электроснабжения малого космического аппарата. На рисунке 1 представлен пример телеметрического сигнала напряжения батареи.

С целью построения образа сигнала было предложено перейти от временного представления сигнала к частотному с помощью преобразования Фурье [1].

В качестве алгоритма кластеризации был использован D-AFC-TC [3] алгоритм, использующий max-min транзитивное замыкание нечеткой толерантности, определяющее геометрическую структуру исследуемого множества. Некоторые особенности применения методов нечеткой и/или возможностной кластеризации к обработке телеметрической информации была показана в [2]. В [4] был предложен метод оценки границ множества вероятного числа нечетких кластеров в искомой кластерной структуре, а в [5] метод оценки минимального числа объектов в нечетких кластерах.

Рис. 1 – Телеметрический сигнал напряжения батареи

Рис. 2– Амплитудно-частотный спектр сигнала

Сегментация сигнала производилась методом скользящего окна. Таким образом была сформирована цепочка обработки сигнала (рисунок 3), в результате выполнения которой, искомая кластерная структура содержит разбиение сегментов сигнала на сегменты содержащие и не содержащие аномалии.

Рис. 3 – Цепочка обработки сигнала

При этом, в нормальных условиях работы оборудования все образы сигнала будут попадать в одну группу кластеров, однако, в случае появления периодических или единичных отклонений в сигнале,

соответствующие им образы попадут в другие кластеры. Появление различных типов отклонений в сигнале будет приводить к появлению новых кластеров.

В порядке вычислительного эксперимента был обработан телеметрический сигнал напряжения батареи МКА.

Таким образом, был предложен метод обнаружения аномалий в телеметрическом сигнале. Результаты обработки могут использоваться как самостоятельно, для индикации неисправностей бортовой аппаратуры МКА, так и для обучения системы нечеткого вывода, механизм формирования которой описан в [3].

Список использованных источников:

1. Сергиенко А.Б. Цифровая обработка сигналов: Учебник для вузов / Сергиенко А.Б. // СПб: Питер. – 2002 г., 606 с.
2. Ерома, А. П. Обработка телеметрических сигналов на основе эвристических алгоритмов возможностной кластеризации : дисс. ... магистра технических наук : 1-40 80 05 / А. П. Ерома ; науч. рук. Д. А. Вятчинин. - Мн.: БГУИР, 2015. - 68 с..
3. Viattchenin, D. A. Heuristic Approach to Possibilistic Clustering: Algorithms and Applications. / Viattchenin, D. A. // Springer, Berlin, Heidelberg. – 2013, 226 p.
4. Viattchenin, D. A. Estimation of bounds of the set of potential number of fuzzy clusters in a sought clustering structure / D. A. Viattchenin, A. Yaroma, A. Damaratski // Communications on Applied Electronics (2017), 6(2):1-10.
5. Viattchenin, D. A. A Method for Estimating the Least Number of Objects in Fuzzy Clusters / D. A. Viattchenin, A. Yaroma // International Journal of Electronics and Telecommunications. – 2017. – Vol 63, no. 4. – P. 341-346.

АНАЛИЗ СУЩЕСТВУЮЩИХ МОДЕЛЕЙ И МЕТОДОВ РАСПОЗНАВАНИЯ ЛИЦ

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

Журавский Р.М.

Серебряная Л.В. – к.т.н., доцент

Задача распознавания лиц имеет множество приложений в таких областях, как биометрия, организация видеоконференций, системы машинного зрения в робототехнике, интеллектуальные системы безопасности и контроля доступа и т. п. Основной трудностью данной задачи является зависимость качества результата распознавания человека по изображению лица от ракурса, положения и условий освещенности. Рассмотрим и проанализируем современные методы распознавания лиц.

Несмотря на большое разнообразие представленных методов, можно выделить общую структуру процесса распознавания лиц:

Рис. 1 - Общий процесс обработки изображения лица при распознавании

На первом этапе производится детектирование и локализация лица на изображении. На этапе распознавания производится выравнивание изображения лица (геометрическое и яркостное), вычисление признаков и непосредственно распознавание – сравнение вычисленных признаков с заложенными в базу данных эталонами. Основным отличием всех представленных алгоритмов будет вычисление признаков и сравнение их совокупностей между собой.

Существующие методы распознавания лиц:

- Метод гибкого сравнения на графах (Elastic graph matching);
- Нейронные сети;
- Скрытые Марковские модели (СММ, НММ);
- Метод главных компонент или principal component analysis (PCA).

1) Метод гибкого сравнения на графах (Elastic graph matching)[1]:

Преимущество - до 95-97% эффективность распознавания, даже при наличии различных эмоциональных выражений и изменении ракурса лица до 15 градусов.

Недостатки:

- Высокая вычислительная сложность процедуры распознавания;
- Низкая технологичность при запоминании новых эталонов;
- Линейная зависимость времени работы от размера базы данных лиц.

2) Нейронные сети:

Основное преимущество - сверточная нейронная сеть, которая является логическим развитием идей таких архитектур нейронной сети как когнитрона и неокогнитрона позволяет получить 96% точность распознавания[2]. Успех обусловлен возможностью учета двумерной топологии изображения.

Недостатки:

Добавление нового эталонного лица в базу данных требует полного переобучения сети на всем имеющемся наборе, а это достаточно длительная процедура;

Проблемы математического характера, связанные с обучением: попадание в локальный оптимум, выбор оптимального шага оптимизации, переобучение и т. д.

3) Скрытые Марковские модели (СММ, НММ):

Преимущество - СММ используют статистические свойства сигналов и учитывают непосредственно их пространственные характеристики. Элементами модели являются: множество скрытых состояний, множество наблюдаемых состояний, матрица переходных вероятностей, начальная вероятность состояний. Каждому соответствует своя Марковская модель. При распознавании объекта проверяются сгенерированные для заданной базы объектов Марковские модели и ищется максимальная из наблюдаемых вероятностей того, что последовательность наблюдений для данного объекта сгенерирована соответствующей моделью [3].

Недостатки:

- Необходимо подбирать параметры модели для каждой базы данных [4];
- СММ не обладает различающей способностью, то есть алгоритм обучения, только максимизирует отклик каждого изображения на свою модель, но не минимизирует отклик на другие модели [4];
- Не применяются в коммерческих целях.

4) Метод главных компонент или principal component analysis (PCA) [5]:

Основное преимущество - значительное уменьшение размерности пространства признаков таким образом, чтобы оно как можно лучше описывало «типичные» образы, принадлежащие множеству лиц. Полученный один раз на обучающей выборке изображений лиц набор собственных векторов используется для кодирования всех остальных изображений лиц, которые представляются взвешенной комбинацией этих собственных векторов. Используя ограниченное количество собственных векторов, можно получить сжатую аппроксимацию входному изображению лица, которую затем можно хранить в базе данных в виде вектора коэффициентов, служащего одновременно ключом поиска в базе данных лиц[6].

Основной недостаток - является очень чувствительным к входным данным. Изменение освещенности, угол поворота головы и эмоциональное выражение лица сильно меняют результат распознавания. Это обуславливается тем, что задачей алгоритма является наилучшая аппроксимация входных данных, а не их распределение по классам.

Для большинства современных систем автоматического распознавания лиц основной задачей является задача сравнения данного изображения лица с набором изображений лиц из базы данных. Характеристики систем автоматического распознавания лиц в этом случае, оцениваются путем определения вероятностей ошибочного отказа в распознавании (для изображения лица, присутствующего в базе, принимается решение как о неопознанном лице) и ошибочного распознавания.

Выбор метода идентификации лиц для создания практических систем автоматического распознавания лиц, должен основываться на ограничениях конкретной системы, и определяется непосредственно на этапе проектирования.

Учитывая вышеизложенное, для повышения качества результата, перспективным может являться создание гибридных методов, использующих преимущества и нивелирующих недостатки рассмотренных выше различных частных подходов.

Список использованных источников:

1. L.Wiskott, J.-M. Fellous, N. Krüger, and C. von der Malsburg, "Face Recognition by Elastic Bunch Graph Matching," Technical Report IR-INI 96-08, Institut für Neuroinformatik, Ruhr-Universität Bochum, D-44780 Bochum, Germany, 1996.
2. Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, Lior Wolf, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification" 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 2014.
3. Гультяева, Т.А. Скрытые марковские модели с одномерной топологией в задаче распознавания лиц / Т.А. Гультяева, А.А. Попов; НГТУ. — 2006.
4. Samaria F. S., Face Recognition Using Hidden Markov Models// PhD Thesis, University of Cambridge, 1994.
5. Шерстобитов Александр Иванович, Федосов Валентин Петрович, Приходченко Владислав Александрович, Тимофеев Митрий Витальевич Распознавание лиц на групповых фотографиях с использованием алгоритмов сегментации // Известия ЮФУ. Техническиенауки. 2013.
6. P.N. Belhumeur ; J.P. Hespanha ; D.J. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection", IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997.

АНАЛИЗ МОДИФИКАЦИЙ ГЕНЕТИЧЕСКОГО АЛГОРИТМА

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

Игнатъев Н.С.

Ярмолик В.Н. – д.т.н., профессор

Генетические алгоритмы (ГА) - изначально возникли в результате наблюдений и попыток копирования процессов, происходящих в природе, в частности естественного отбора живых организмов. С точки зрения программных и информационных технологий, ГА используются как стратегия решения задач оптимизации, основанная на имитировании биологической эволюции. Программы, построенные на базе таких алгоритмов, дают довольно хорошие результаты и используются в различных сферах науки и инженерии.

Задача классической модели генетического алгоритма формализуется таким образом, чтобы её решение могло быть закодировано в виде генотипа (вектора генов фиксированной длины). Некоторым образом создаётся множество генотипов начальной популяции, которые будут оценены с использованием фитнес функции, в результате чего с каждым генотипом ассоциируется определённое значение приспособленности, которое определяет насколько хорошо фенотип, им описываемый, решает поставленную задачу. С помощью пропорционального отбора формируется промежуточный массив, из которого случайным образом выбираются два родителя. Далее производится одноточечный кроссинговер, и созданные два потомка мутируют (одноточечная мутация) с заданной вероятностью. Мутировавшие потомки занимают места своих родителей. Процесс продолжается до тех пор, пока не будет достигнут критерий окончания алгоритма.

В модификации СНС для кроссинговера, выбирается случайная пара родителей, но не допускается, чтобы между ними было малое хеммингово расстояние. При скрещивании производится однородный кроссинговер: к потомку переходит по половине битов каждого родителя. Для нового поколения выбираются N лучших различных особей среди родителей и детей. При этом дублирование генотипов не допускается. В модели СНС размер популяции относительно мал — около 50 особей. Это оправдывает использование однородного кроссинговера и позволяет алгоритму сойтись к решению. СНС противопоставляет агрессивный отбор агрессивному кроссинговеру, однако, малый размер популяции быстро приводит ее к состоянию, когда создаются только более или менее одинаковые генотипы. В таком случае СНС применяет cataclysmic mutation: все генотипы, кроме самого приспособленного, подвергаются сильной мутации (изменяется около трети битов). Так алгоритм перезапускается и далее продолжает работу, применяя только кроссинговер. Таким образом, алгоритм СНС довольно быстро сходится из-за того, что в нем нет мутаций, следующих за оператором кроссинговера, используются популяции небольшого размера, и отбор особей в следующее поколение ведется и между родительскими особями, и между их потомками.

Идея гибридных алгоритмов заключается в сочетании генетического алгоритма с некоторым другим

классическим методом поиска, подходящим в данной задаче. В каждом поколении все сгенерированные потомки оптимизируются выбранным методом и затем заносятся в новую популяцию. Тем самым получается, что каждая особь в популяции достигает локального оптимума, вблизи которого она находится. Далее, производятся обычные для ГА действия: отбор родительских пар, кроссинговер и мутации. На практике гибридные алгоритмы оказываются очень удачными. Это связано с тем, что вероятность попадания одной из особей в область глобального максимума обычно велика. После оптимизации такая особь будет являться решением задачи. Известно, что генетический алгоритм способен быстро найти во всей области поиска хорошие решения, но он может испытывать трудности в получении из них наилучших. Обычный оптимизационный метод может быстро достичь локального максимума, но не может найти глобальный. Сочетание двух алгоритмов позволяет использовать преимущества обоих.

В модели генитор используется специфичный способ отбора. Вначале, как и полагается, популяция инициализируется, и ее особи оцениваются. Затем выбираются случайным образом две особи, скрещиваются, причем, получается только один потомок, который оценивается и занимает место менее приспособленной особи в популяции, а не одного из родителей. После этого снова случайным образом выбираются две особи, и их потомок занимает место родительской особи с самой низкой приспособленностью. Таким образом, на каждом шаге в популяции обновляется лишь одна особь. Процесс продолжается до тех пор, пока пригодности хромосом не станут одинаковыми. В данный алгоритм можно добавить мутацию потомка после его создания. Критерий окончания процесса, как и вид кроссинговера и мутации, можно выбирать разными способами.

Для повышения эффективности работы ГА создано множество модификаций. Они связаны с применением иных методов селекции, с модификацией генетических операторов, с преобразованием функции приспособленности, а также, с различными способами кодирования параметров задачи в форме хромосом. Описанные выше примеры не исчерпывают многообразие областей и задач применения ГА. Эта область сейчас является перспективной и требует ещё долгого изучения.

Список использованных источников:

1. Эволюционные вычисления [Электронный ресурс]. – Режим доступа к источнику: <https://www.intuit.ru/studies/courses/14227/1284/info>
2. Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы // Под ред. В.М. Курейчика. — 2-е изд., испр. и доп. — Москва: ФИЗМАТЛИТ, 2006. — 320 с.

РЕАЛИЗАЦИЯ ГОЛОВОЛОМКИ «КУБИК РУБИКА»

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

Камеко В.В.

Данилова Г. В. – м. т. н., ассистент

В настоящий момент, в связи со спросом на головоломки различных видов, возникла потребность в их симуляторах, позволяющих собирать головоломки на компьютерах. В качестве головоломки для реализации была выбрана самая известная из них – «Кубик Рубика». В программе будут реализованы все функции данной головоломки на языке программирования Delphi.

Известный факт, что головоломки положительно влияют на развитие пространственного и логического мышления, а также, являются отличным антистрессом. Одна из таких головоломок, «Кубик Рубика», не только стала самой знаменитой головоломкой и собрала целое движение «спидкуберов», но и поставила довольно интересную задачу математикам и программистам всего мира. В настоящее время существует достаточно мало приложений эмулирующих эту замечательную игрушку, как и не существует единого формата хранения состояния «Кубика».

«Кубик Рубика» – механическая головоломка, изобретённая в 1974, представляющая из себя куб $3 \times 3 \times 3$ с окрашенными в разные цвета гранями. Задача игрока – собрать «кубик» в первоначальное состояние из перемешанного.

Рис. 1 – Различные вариации головоломки Кубика Рубика.

Существуют также различные вариации данной головоломки, например: головоломки с произвольными размерами ($N \times M \times K$), кубики других многогранников (додекаэдр, тетраэдр, усеченные пирамиды, призмы, октаэдр, ромбододекаэдр, кубооктаэдр), а также многомерные аналоги.

В программе будут реализованы следующие функции:

- Возможность сохранения и загрузки состояний кубика.
- Поддержка нескольких сохранений.
- Возможность изменения состояния кубика, путем вращения граней.
- Поворот самого кубика для лучшего представления о его положении в пространстве.

Для реализации сохранений будет спроектирован специальный формат файла, поддерживающий хранение состояния кубика, но не всю историю его изменения, для экономии размера файла и упрощения взаимодействия пользователя с программой.

Для полного погружения в игру планируется добавить очки и время, потраченное на сборку кубика.

Планируется добавить фоновую музыку и главное меню, где пользователь сможет выбирать способы взаимодействия с программой. В планируемые функции также входит добавление возможности выбора размера кубика и цвета его граней.

Управление будет реализовано через кнопки интерфейса и движений мышью для поворота кубика или его граней. Интерфейс и другие изображения по возможности будут нарисованы.

Программу планируется создавать на игровом движке с открытым исходным кодом «Castle Engine», для облегчения работы с созданием интерфейса пользователя и 3D-объектами. В программе будут использованы кнопки для взаимодействия с пользователем, различного вида картинки, а также 3D-модель кубика. Исходный код, по возможности, будет разбит на модули со схожей функциональностью.

Список использованных источников:

1. https://ru.wikipedia.org/wiki/Кубик_Рубика – Кубик Рубика. Материал из Википедии
2. https://ru.wikipedia.org/wiki/Пирамидка_Мефферта – Пирамидка Мефферта. Материал из Википедии
3. <https://castle-engine.io> – Бесплатный open-source 3D и 2D игровой движок

АЛГОРИТМЫ И СИСТЕМЫ РАСЧЕТА БУКМЕКЕРСКИХ КОЭФФИЦИЕНТОВ НА СПОРТИВНЫЕ СОБЫТИЯ

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

Канунников И.С.

Парамонов А.И. – канд. техн. наук, доцент

Для эффективной работы букмекерских контор существует такая структура в их организации, как аналитический отдел, главная задача которого – рассчитать и грамотно расставить базовые коэффициенты на события. Аналитик, применяя математический аппарат теории вероятности, статистического анализа и численных методов, выставляет коэффициент возможности того или иного события. Безошибочно работающий аналитический отдел – неременный залог прибыльности букмекерской конторы.

На начальном этапе деятельность аналитического отдела связана со сбором информации, такой как статистические показатели игроков, команд и прочее. Затем выполняется её обработка и формирование спортивной линии. Обычно, за несколько дней до события, формируются начальные коэффициенты, которые затем корректируются таким образом, чтобы ставки были разделены в пропорции 50 на 50 (для упрощенного варианта события с двумя возможными исходами – например, победа или поражение команды). Предположим, требуется вычислить первичный коэффициент на домашнюю победу футбольного клуба А над клубом Б. Формирование коэффициентов упрощенно проходит по следующему алгоритму: пусть Клуб А выиграл 40% домашних встреч. Если перевести это в коэффициенты, то коэффициент получим $x_1 = (1/0,4) = 2,5$. Пусть Клуб Б проиграл в выездных встречах 20% игр. Получается, что на проигрыш коэффициент должен составить $x_2 = (1/0,2) = 5$. Усредняем два полученных значения и получаем коэффициент победы Клуба А над Клубом Б $x_3 = (x_1 + x_2)/2 = (2,5 + 5)/2 = 3,75$.

С целью автоматизировать процесс формирования спортивной линии и получить доступ к большим объемам статистической информации, разрабатывается специализированное программное обеспечение. На сегодняшний день доступен широкий спектр программ для расчета спортивных ставок. Стоимость данного вида программного обеспечения варьируется от бесплатных до тысяч долларов. Рассмотрим ряд наиболее известных и популярных из них.

Калькулятор Келли – программа, в основе которой используется математический аппарат критерия Келли [1]. Калькулятор для расчета прогноза на матч рассчитывает возможность выигрыша по коэффициенту, выставленному букмекером и вероятностью исхода, определенной самим игроком.

Footbet – приложение для расчета вероятности исхода футбольного матча на основе статистических данных прошлых встреч команды [2]. Для каждой команды вычисляется определенный коэффициент,

который зависит от класса команды, места проведения встречи, результатов последних игр, текущего места команды в турнирной таблице, количества отсутствующих игроков основного состава и т.д., причем, все факторы различаются между собой по приоритетности, следовательно, входят в суммарный коэффициент с определенными «весами». Затем вычисляется суммарный коэффициент двух команд, и по схеме Байеса определяется долевое участие каждой команды в суммарном коэффициенте. Полученные значения переводятся в процентные соотношения и предлагаются в качестве вероятности победы той или иной команды. Причем учтено, что команда с наилучшими возможными показателями выигрывает у команды с наихудшими возможными показателями в 96% случаях, что подтверждается статистическими данными.

Keys to the Match – система от IBM для прогнозирования результатов игр в теннис [3]. На рынке с 2013 года как предиктивный аналитический инструмент для четырех турниров Большого шлема: Уимблдона и Открытых чемпионатов Австралии, Франции и США. Для каждого игрока система анализирует его уникальный стиль игры и дает рекомендации по трем ключевым показателям (keys), которых должен достигнуть теннисист, чтобы увеличить свои шансы на выигрыш в текущем матче. IBM не раскрывает, какие именно прогностические алгоритмы используются в системе.

Cortana Intelligence Suite – система искусственного интеллекта от Microsoft [4]. Систему активно тестируют на прогнозировании футбольных матчей. Случаев прогнозирования тенниса не известно. Для прогнозирования результатов система использует комплексный анализ двух факторов: статистические данные (соотношение побед и поражений команды, количество забитых голов, место проведения матча, погодные условия) и данные веб-поиска и социальных сетей (так называемый «коллективный разум»). Шанс выигрыша каждой команды определяется в процентном соотношении, рассчитывается вероятность ничьей в матчах для каждой игры отборочного тура.

OhMyBet! – сервис с российскими корнями [5]. Авторы – выпускники Физтеха и ВМиК МГУ. По опубликованным данным сервис позволил выделить ряд признаков игры в теннис на основе собранных данных о 825 000 сыгранных матчей в турнирах с 2000 по 2014 года. Были обработаны и зафиксированы такие признаки как: результат матча, покрытие, турнир, количество эйсов, количество двойных ошибок, процент выигрыша на первой и второй подаче, средняя скорость подачи каждого игрока, возраст игроков, оценочная мотивация выигрыша, предыдущие встречи игроков, травмы, время отдыха между матчами и т. д. К признакам игр применили алгоритмы машинного обучения. Алгоритм, используемый в системе, создатели не раскрывают, однако из анализа опубликованных работ и наблюдением за поведением самой системы можно предположить, что это нейронная сеть.

Исследовательский интерес к прогностическим моделям для спортивных игр привел к появлению сервисов, предлагающих пользователям результаты такого прогнозирования. На основе выполненного обзора можно сделать предположение, что основными математическими методами прогнозирования результатов выступают иерархические марковские модели и алгоритмы машинного обучения. Развитие подобных алгоритмов и систем анализа спортивных событий находит свое применение не только в играх на ставках, в букмекерских конторах, но и при подготовке спортсменов к соревнованиям, выявлению их тренда физической формы и результатов. Именно поэтому ведущие фирмы ИТ сектора активно выступают на рынке программного обеспечения по аналитике спортивных событий.

Список использованных источников:

1. Kelly, J.L. Jr. A New Interpretation of Information Rate // Bell System Technical Journal. – 1956. – №35. – С. 917-926.
2. Kahneman, D. Judgment under Uncertainty: Heuristics and Biases // Cambridge University Press. — 2005. — С. 21
3. Belson, K. As Tennis Stats Proliferate, Software Tries to Make Sense of It All // New York Times [Электронный ресурс]. – Режим доступа: <https://www.nytimes.com/2013/08/26/sports/tennis/as-tennis-stats-proliferate-software-tries-to-make-sense-of-it-all.html> - 30.03.2018
4. Microsoft. Introduction to Cortana Intelligence Suite [Электронный ресурс]. – Режим доступа: <https://social.technet.microsoft.com/wiki/contents/articles/36688.introduction-to-cortana-intelligence-suite.aspx> - 30.03.2018
5. Машинное обучение для прогнозирования тенниса. Часть 2 [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/307422/#43> - 30.03.2018

ГОЛОСОВОЕ УПРАВЛЕНИЕ ПЕРСОНАЛЬНЫМ КОМПЬЮТЕРОМ

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

Кива В.С.

Лукьянец В.Г. – к.т.н., доцент

Статья посвящена характеристике сути голосового управления и возможности применения этой технологии на персональном компьютере. Рассмотрены основные реализованные алгоритмы распознавания речи.

В настоящее время проблема разработки и создания эффективных систем распознавания речи и голосового управления персональным компьютером имеет большую актуальность. Совместное использование таких систем является основой полнофункционального голосового интерфейса, объем применения которого, очень широк на практике. Исследования в области интерфейса речи ведутся многими учеными, а разработка осуществляется крупнейшими компьютерными организациями, включая Microsoft, Intel и IBM. Использование распознавания команд, посредством голосового управления на персональном компьютере, может иметь широкий диапазон и может быть реализовано во многих областях общества и промышленности.

На современном этапе развития технологий трудно представить полный переход к голосовому управлению, что станет возможно только тогда, когда персональный компьютер научится полностью понимать человеческую речь, а не просто ограниченный набор команд. Но наличие режимов голосового управления в различных приложениях и операционных системах действительно актуально, потому что не все люди имеют возможность свободно использовать свои руки. В принципе, управление голосом с помощью компьютера упрощает работу с ним, потому что команды передаются быстро и без усилий через микрофон.

Ведущие компании по разработке программного обеспечения, разработке операционных систем и внедрению новых высокотехнологичных устройств понимают все преимущества включения голосового управления в список возможностей своих продуктов, поэтому в этом направлении ведется интенсивная работа. Среди операционных систем, созданных Microsoft, функция голосового управления доступна только в версии Windows 8. Те же функции поддерживаются операционными системами, созданными Apple, особенно благодаря применению Siri [1].

Чтобы настроить голосовые команды в операционной системе, на персональном компьютере должна быть установлена соответствующая программа. Взаимодействие с компьютером может быть создано не только с помощью клавиатуры и мыши. Теперь стали доступны и голосовые команды. Существует специальный инструментарий, который позволяет вам это делать. Его функция включает в себя не только написание текста для диктовки или декодирование аудиозаписей. Через голосовые команды можно запускать приложения и использовать их [2].

Основой для создания систем голосового управления являются алгоритмы и методы распознавания речи.

Наиболее распространенными и эффективными являются следующие методы:

1. MFCC (Mel-Frequency Cepstral Coefficients) – это метод кепстральных коэффициентов, который заключается в вычислении коэффициентов спектра Фурье, наложения на полученный спектр набора фильтров шкалы мел, выполнения логарифмирования измененного спектра и реализации дискретного косинусного преобразования.
2. LPCC (Linear Predictive Cepstral Coefficients) – это метод кепстральных коэффициентов линейного предсказания, который основывается на вычислении коэффициентов авторегрессионной модели для каждого фрейма аудио сигнала.
3. PLP (Perceptual Linear Predictive) – это метод коэффициентов перцептивного линейного предсказания, который отличается от метода LPCC тем, что учитываются особенности восприятия различных частот человеком - перед вычислением параметров авторегрессионной модели сигнал проходит определенную предобработку. Вычисленный мгновенный спектр Фурье преобразуется в спектр на шкале барков, после чего выполняется операция свертки маскирующих кривых критических полос с полученным спектром для получения эффекта маскировки частоты. Далее производится аппроксимация кривой громкости и кепстральная обработка [3].

Признанные методы анализа приведенных методов показали, что большинство усилий разработчиков сосредоточено на частотных характеристиках выделения человеческого вокального канала и, следовательно, на качественных методах, основанных на обработке кепстрального сигнала.

Настройка системы распознавания речи подразумевает способность управлять шумодавлением, определять время задержки и результаты распознавания, а также, процент слияния нескольких результатов распознавания. Итоговый результат возникает в окне и позволяет редактировать речевую базу без специального знания системы. Настройки распознавания голоса на персональном компьютере позволяют настроить следующие параметры: автозапуск с системой Windows; ввод журнала событий и ошибок и сохранение его в файле; определить, как сохранить журнал ошибок и событий и управлять виджетами [4].

Программное обеспечение и виджеты для голосового управления реализуются в Microsoft Visual Studio с поддержкой интегрированной среды разработки Windows Forms и WPF с использованием Visual C #, являются реализацией языка C# и предназначены для демонстрации основных функций речевого модуля Microsoft SpeechAPI.

Анализ рынка программ показывает, что лидером среди программ, которые позволяют управлять компьютером на русском языке, по качеству и точности распознавания является программа Tuple. Есть, также такие программы как: RealSpeaker, Web Speech, Горыныч, и т.д., но все они менее точны или у них не полная интеграция с ОС.

Таким образом, в результате исследования были рассмотрены и проанализированы наиболее успешные методы распознавания речевых команд и имеющееся эффективное программное обеспечение для управления голосом на персональном компьютере.

Список использованных источников:

1. Самые совершенные программы управления голосом [Электронный ресурс]. – Режим доступа: https://www.gkh11.ru/news/samye_overshennye_programmy_upravlenija_golosom/2016-01-28-1770. – Дата доступа: 24.03.2018.
2. Управление голосом и жестами на компьютере [Электронный ресурс]. – Режим доступа: <https://bursin.ru/upravlenie-golosom-i-zhestami-na-kompyutere/>. – Дата доступа: 22.03.2018.
3. Кравченко, К.В. Автоматизированная система голосового русскоязычного управления операционной системой Windows / К.В. Кравченко, Р.А. Дьяченко // Современные проблемы науки и образования. – 2014. – № 3.
4. Как управлять компьютером при помощи голосовых команд? [Электронный ресурс]. – Режим доступа: <http://nastroyse.ru/programs/review/upravlenie-kompyuterom-s-pomoshhyu-golosa.html>. – Дата доступа: 23.03.2018.

ПРОГНОЗИРОВАНИЕ КЛИМАТИЧЕСКИХ УСЛОВИЙ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИЙ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ

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

Клещиков А.С.

Медведев С.А. – к.т.н., доцент

Влияние прогноза климатических условий для общества, бизнеса, сельского хозяйства заставляет уделять большое внимание данному вопросу. Последние несколько лет показывают потрясающие достижения в области прогнозирования климата. Все системы и методы, разработанные до сих пор, используют температуру поверхности моря в качестве основного фактора, использующегося для прогноза. Затем уже используются статистические и математические модели для дальнейшего прогноза климата. Однако есть и другой подход к прогнозированию климатических условий, который основывается на исторических метеорологических показателях региона, таких как осадки, скорость ветра, точка росы, температура и т.д.

Основная цель данной статьи - как использовать технологии интеллектуального анализа данных (в частности, алгоритма К-ближайших соседей), как разработать систему, которая использует исторические данные для прогнозирования климатических условий заданного региона, города или страны на месяцы вперед.

Интеллектуальный анализ данных - это недавняя разработка в области очень больших баз данных и хранилищ данных. Он используется для обнаружения скрытых полезных паттернов в огромных базах данных. Интеллектуальный анализ данных может быть классифицирован по его методам на три основных типа: поиск ассоциативных правил, кластерный анализ и классификация/предсказывание.

Алгоритм К-ближайших соседей – это алгоритм классификации, который основывается на формуле дистанции в евклидовом пространстве. Данная формула используется для выяснения близости между неизвестными образцами с известными классами. Неизвестный образец присваивается тому классу, который является наиболее распространенным среди К соседей данного образца, классы которых уже известны.

Функция вычисления дистанции в евклидовом пространстве выглядит следующим образом:

(1)

Шаги алгоритма К-ближайших соседей выглядят следующим образом:

вычисление расстояния между новым образцом и известными образцами по формуле (1);

сортировка полученных расстояний так, чтобы и выборка К образцов с наименьшими значениями расстояний;

новому образцу присваивается класс с наибольшим количеством элементов в выборке.

Для применения алгоритма К-ближайших соседей для прогнозирования климатических условий необходимо указать четыре параметра: текущая дата (дата, от которой будет отсчитываться прогноз) (CD), количество дней для прогноза (N), значение К алгоритма и атрибуты для прогноза.

Затем, для вычисления расстояния в евклидовом пространстве, исходный массив данных разбивается на последовательности. Каждая последовательность имеет размер S, который рассчитывается как произведение количества дней прогноза N на количество атрибутов прогноза. Базовой последовательностью являются записи за предыдущие N дней для выбранных атрибутов. Общее количество последовательностей (Т) вычисляется как общее количество записей делить на количество записей в базовой последовательности. Дальнейшее вычисление расстояний можно выразить с помощью псевдокода:

```
While i < Общее_Количество_Записей  
For j = 0 to S
```

```

Sum = Sum + Sqr (Базовая_Последовательность[j] - Все_Записи[i])
i++
End For
Distance = Sqrt (Sum)
End While

```

Затем полученные расстояния сортируются по возрастанию и выбирается K наименьших расстояний. Итоговый прогноз вычисляется как среднее значение K образцов для выбранного атрибута и дня.

Данный метод показывает очень точные результаты, которые могут использоваться в реальной жизни. Так, для дискретных значений таких показателей как туман, град, снег, гололедица, гроза и т.д. на небольших наборах данных точность составила свыше 90%. На больших массивах данных точность повышается до 95%.

Недостатком данного метода является тот факт, что он не учитывает глобальные изменения климата (ENSOevents). Однако, данный метод прекрасно работает для областей, которые не подвержены таким изменениям.

Список использованных источников:

1. Data Mining for Climate Change and Impacts – A.R.Ganguly, K.Steinhaeuser - 2008 IEEE International Conference on Data Mining Workshops.

2. Метод ближайших соседей [Электронный ресурс]. – 2018. – Режим доступа: https://www.ibm.com/support/knowledgecenter/ru/SSLVMB_24.0.0/spss/base/idh_idd_knn_variables.html Дата доступа: 21.03.2018.

МЕТОДЫ ВОЗВЕДЕНИЯ ЧИСЛА В СТЕПЕНЬ ПО МОДУЛЮ

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

Короткевич А.В.

Ярмолик В.Н. – д.т.н., профессор

Одной из наиболее важных операций в асимметричной криптографии является возведение числа в степень. Т.к. операция выполняется в конечном поле, то фактически задача сводится к нахождению . Очевидно, что простейшим способом решения является выполнение умножения, однако, такой способ неприемлем, когда речь идет о числах большой разрядности. Эффективные методы выполнения возведения числа в степень по модулю будут рассмотрены в данной работе.

Существует два способа уменьшения времени выполнения операции возведения в степень в конечном поле. Во-первых, это уменьшение времени выполнения умножения двух элементов группы. Вторым способом является уменьшение требуемого числа операций умножения. В идеале оба подхода должны быть использованы одновременно [1].

В качестве решения, использующего уменьшение числа операций умножения при возведении в степень, можно привести следующий алгоритм:

Алгоритм 1 –Бинарное возведение в степень справа налево

INPUT: Целые числа a и b .

OUTPUT: a^b .

1) $a^0 = 1$.

2) While do:

2.1) If нечетное then $a^b = a \cdot a^{b-1}$.

2.2) $b = b / 2$.

2.3) If then $a^b = (a^2)^{b/2}$.

3) Return a^b .

Приведенный алгоритм использует последовательные операции умножения и возведения в квадрат, что фактически тоже является умножением. Пусть n – длина в битах бинарного представления числа a , а b – число единиц в данном представлении. Тогда, согласно алгоритму, потребуется возведений в степень и операция умножения. Если a является случайным числом, то число возведений в степень будет приблизительно $\log_2 b$, а умножений – b . Таким образом, алгоритм позволяет сократить число операций с b для самого простого решения до приблизительно $\log_2 b$, что является хорошим показателем.

Алгоритм 1 вычисляет a^b всякий раз, когда b является нечетным. Для некоторых значений a выражение может быть вычислено более эффективно, чем для случайного a . Алгоритм 2 выполняет бинарное возведение в степень слева направо, которое заменяет операцию $a^b = a \cdot a^{b-1}$ (для случайного a) на $a^b = (a^2)^{b/2}$ (для фиксированного a).

Алгоритм 2 –Бинарное возведение в степень слева направо

INPUT: Целое a и положительное целое b .

OUTPUT: a^b .

1) $a^0 = 1$.

2) For from down to do:

2.1) $a^b = a^b \cdot a^2$.

2.2) If b is odd, then $a^b = a^b \cdot a$.

3) Return .

Пусть n – длина в битах бинарного представления числа a , m – число единиц в данном представлении. Тогда, согласно алгоритму, потребуется n возведений в степень и операция умножения на a . Число возведений и умножений остается таким же, как и в алгоритме 1, но в алгоритме 2 умножение всегда производится на фиксированное значение a . Если a имеет особую структуру, то это умножение может быть гораздо более легким, чем умножение двух случайных значений.

Оптимизацией приведенных алгоритмов являются оконные методы возведения в степень, которые на одной итерации обрабатывают более одного бита значения степени одновременно [2]. Разновидность такого алгоритма может иметь следующий вид:

Алгоритм 3 –Возведение в степень слева направо с помощью окна ширины

INPUT: Целые a и n , где n для некоторого k .

OUTPUT: a^n .

1) Предварительные вычисления.

1.1) $a \leftarrow a \bmod m$.

1.2) For from n to 1 do: $a \leftarrow a^2 \bmod m$. (Таким образом, $a \leftarrow a^{2^k} \bmod m$)

2) $a \leftarrow a^k \bmod m$.

3) For from n down to 1 do:

3.1) $a \leftarrow a^2 \bmod m$.

3.2) $a \leftarrow a^2 \bmod m$.

4) Return a .

В алгоритме 4, по сравнению с алгоритмом 3, уменьшается количество предварительных вычислений, а также сокращается среднее число используемых операций умножения (исключая возведение в квадрат). называется шириной окна.

Алгоритм 4 –Возведение в степень с использованием скользящего окна

INPUT: Целые a , n , k , m .

OUTPUT: a^n .

1) Предварительные вычисления.

1.1) $a \leftarrow a \bmod m$.

1.2) For from n to 1 do: $a \leftarrow a^2 \bmod m$.

2) $a \leftarrow a^k \bmod m$.

3) While do:

3.1) If then do: $a \leftarrow a^2 \bmod m$.

3.2) Else найти наибольший набор бит такой, что $a \leftarrow a^{2^k} \bmod m$ и do: $a \leftarrow a^2 \bmod m$.

4) Return a .

Отметим, что метод скользящего окна для данного возведения в степень требует трех операций умножения, соответствующих $a \leftarrow a^2 \bmod m$ и $a \leftarrow a^k \bmod m$. А алгоритму 2 понадобится 4 операции умножения для тех же самых a и n .

Рис. 1 – Производительность алгоритмов возведения числа в степень по модулю

Два алгоритма из рассмотренных являются оконными, потому для них необходимо определить оптимальную ширину окна. Все исследования проводились для чисел различных размерностей, которые ограничиваются размерами эллиптических групп, рекомендованных NIST кривых P-192, P-224, P-256, P-384, P-521 (число обозначает размер поля в битах). Из полученных результатов можно сделать вывод о том, что оптимальной шириной окна для алгоритма 3 является значение 5. Оптимальное значение ширины скользящего окна (алгоритм 4) приблизительно равняется 4.

После нахождения оптимальных значений для оконных алгоритмов перейдем к сравнительному анализу производительности возведения в степень по модулю всеми рассмотренными алгоритмами для чисел различной величины. График такой зависимости можно найти на рисунке 1.

Помимо описанных алгоритмов на график также включено возведение в степень по модулю стандартным методом платформы .NET. Полученные результаты позволяют сделать вывод о том, что бинарные методы обладают худшей производительностью, а оконные – лучшей. Самым лучшим методом возведения числа в степень по модулю является алгоритм с использованием скользящего окна ширины 4. Именно он и будет использован в разрабатываемой криптосистеме. Стоит заметить, что данный алгоритм на больших эллиптических группах превосходит по производительности соответствующий метод из платформы .NET, что является хорошим результатом.

Список использованных источников:

1. Menezes, A. Handbook of applied cryptography / A. Menezes, P. Van Oorschot, S. Vanstone – CRC Press, 1997.
2. St Denis, T. BigNum math: implementing cryptographic multiple precision arithmetic / T. St Denis, G. Rose, S. Vanstone – Syngress Publishing, Inc, 2006.

ВНЕДРЕНИЕ МОБИЛЬНЫХ ТЕХНОЛОГИЙ В ПРОЦЕСС ОБУЧЕНИЯ

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

Кротов Е.М.

Данилова Г.В. – м.т.н., ассистент

Отсутствие какой-либо практики использования мобильных телефонов в процессе обучения является упущением. Один мобильный гаджет может с легкостью стать заменой сразу нескольким нужным предметам. Акцентирование внимания на внедрение мобильных технологий для оптимизации ведения учебы может повлиять в лучшую сторону на успеваемость учащихся.

В настоящее время абсолютное большинство людей пользуется мобильными телефонами. Гаджет стал неотъемлемой частью нашей жизни и используется ежедневно. Этот простой, функциональный и эффективный атрибут позволяет человеку оставаться на связи, где бы он ни был, не ограничиваясь лишь сотовой связью. Различные мессенджеры, социальные сети стали тем самым фактором, без которого сегодня трудно представить общение молодежи.

Рис. 1 – Статистика использования различных гаджетов в интернете

В то же время, преподаватели до сих пор вынуждены использовать бумагу для проведения тестов и контрольных работ. В случае, если у преподавателя много студентов, он вынужден таскать кучу макулатуры, что не только повышает риск утери чьих-то работ, но и нагружает излишним физическим трудом. Осуществлять проверку больших объемов тестов достаточно сложно и долго, при этом преподаватель сильно устает и может сам допустить ошибку, что в последствии опечалит студента и вызовет споры.

Беря во внимание данные факторы, встаёт вопрос об интеграции учебного процесса с мобильными гаджетами. Вместо огромных кип листочков с проверочными тестами, студенты могут без особых усилий выполнить контрольную прямо у себя в телефоне. В свою очередь, приложение позволит автоматизировать проверку тестовых элементов контрольной. Преподаватель же избавляется от собираний листочков, получает в своём личном кабинете приложения уже проверенные тестовые задания и просматривает удобную статистику ответов для анализа труднодоступных вопросов.

Анализируя предметную область автоматизации тестирования средствами мобильных систем, была поставлена задача спроектировать и реализовать мобильное приложение для упрощения процесса тестирования и анкетирования среди студентов ВУЗов.

Система должна реализовывать следующие компоненты:

- Создание тестовых и контрольных работ преподавателем по заданной теме.
- Выполнение тестовых заданий студентами, предварительно зарегистрированными в системе.
- Просмотр итоговых результатов со статистикой для анализа пробелов в материале и знаниях студентов преподавателем.
- Возможность скачивания материалов для самостоятельного изучения или повторения студентами.
- Возможность просмотра оценок в текущем семестре за практические и лабораторные работы.

Разработанное приложение даст возможность оптимизировать процесс обучения, связанный с

практическими работами и контролем знаний, позволит студентам избавиться от необходимости приносить ноутбуки вместе с книгами и конспектами. Вся необходимая информация и средства будут всегда вместе с ними. Кроме того, для преподавателей снижается нагрузка, которая в дальнейшем может быть распределена на актуализирование и структуризацию преподаваемого материала.

Список использованных источников:

1. Данилова, Г. В. Программное средство управления формированием IT-компетенций / Г. В. Данилова // Дистанционное обучение – образовательная среда XXI века : материалы IX международной научно-методической конференции (Минск, 3-4 декабря 2015 года). – Минск : БГУИР, 2015. – С. 300 - 301.

СИСТЕМА КОНТРОЛЯ И ОЦЕНКИ СОСТОЯНИЯ ПРОЦЕССА РАЗРАБОТКИ КОМПЛЕКСНЫХ ПРОГРАММНЫХ СРЕДСТВ ДЛЯ ТЕЛЕКОММУНИКАЦИОННОЙ ОТРАСЛИ

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

Кулаковский С.А.

Бранцевич П.Ю. – к.т.н., доцент

Показателем успешности реализации проекта является достижение поставленных перед проектом целей в рамках заданных ограничений по финансовым, материальным, человеческим и временным ресурсам. Наличие четкой системы организации и контроля выполнения проекта является ключевым аспектом его успешного завершения, особенно в высокотехнологичных, инновационных и динамичных отраслях, к которым относятся телекоммуникации. Наряду с другими этапами жизненного цикла проекта, особую важность представляет организация процесса разработки программного обеспечения и контроль состояния данного процесса, позволяющего заблаговременно внести необходимые коррективы и принять своевременные управленческие решения.

Телекоммуникационная отрасль на современном этапе развития представляет собой реализацию взаимодействия целого ряда аппаратно-программных комплексов на базе широкого стека информационных технологий, включающего виртуализацию, использование облачных решений, микросервисы, организацию межконтинентальных и космических каналов связи. Кроме инфраструктурных процессов, перед поставщиками телекоммуникационных услуг стоят задачи обработки, хранения и анализа огромных объемов данных, реализации внешних и внутренних информационных ресурсов, направленных на обеспечение бесперебойного доступа миллионов пользователей к предоставляемым сервисам. Некоммерческая ассоциация TMForum (TeleManagementForum) стандартизирует отрасль посредством разработки рекомендаций, описывающих концепции построения отраслевого программного обеспечения (описание интеграционных интерфейсов, моделей данных) [2]. Единая терминология, наличие списка бизнес-процессов и референтная карта приложений с описанием основных функций, позволяют уже на ранних этапах сформировать набор высокоуровневых требований и применять каскадную модель разработки программного обеспечения для реализации проекта. Типизация имплементационных задач позволяет поставщику наработать необходимую экспертизу, продуктивизировать проектное решение и обеспечить необходимую точность экспертной оценки планируемых работ в каждом из телекоммуникационных доменов.

Система контроля и оценки состояния процесса разработки является механизмом количественной оценки, которая позволяет в любой момент времени оценить ключевые показатели процесса разработки, и на основании полученной информации принять необходимые управленческие решения. К таким решениям относятся: изменение состава команды разработки, переоценка сроков реализации проекта, оценка эффективности использования проектных ресурсов и их достаточности для успешного завершения проекта. Реализация основывается на адаптации ряда существующих подходов и методик, которая позволяет повысить точность проводимой оценки.

Для оценки трудоёмкости планируемого объема работ применяется метод WBS (WorkBreakDownStructure) – иерархическая декомпозиция всего объема работ до уровня, на котором задача может быть оценена и спланирована [1]. При проведении иерархического разбиения работ используется группировка задач по логическим доменам, где требуется привлечение разработчиков с однородным уровнем экспертизы и схожих технических навыков с учётом рекомендаций консорциума TMForum.

Формирование и учёт состава команды разработки происходит на основании ресурсного плана проекта, учитывающего не только доступность каждого ресурса (разработчика) с учётом выходных, праздничных дней, отпусков, но, также, и определённые показатели, характеризующие производительность и вовлеченность каждого разработчика:

Вовлеченность – характеристика, определяющая степень привлечения разработчика непосредственно к написанию программного кода. Разработчик может частично привлекаться на проект либо решать задачи в нескольких логических доменах, либо только частично заниматься непосредственно разработкой;

Производительность – коэффициент, определяющий способность разработчика решать задачи относительно оценки, определённой технической лидером. Определяется на основании опыта разработчика в конкретном домене и в ходе проекта корректируется на основании накопленной статистики решения

проектных задач.

Для оценки текущего состояния процесса разработки используется расширение метода освоенного объема (EVT – EarnedValueTechnique) [1]. В разрезе количества задач и их трудоёмкости с учётом реализованного объема работ определяется текущий статус процесса разработки в любой момент времени. На основании известных характеристик команды разработки и накопленных статистических данных производится расчёт планируемых дат окончания запланированных работ.

Система контроля и оценки состояния процесса разработки использует адаптацию каскадной модели разработки ПО, совмещая фазы разработки и тестирования. Привлечение команды тестирования для верификации реализованных функций на этапе разработки позволяет обнаружить большую часть дефектов на более ранних этапах. Введение количественного соотношения «входящих» задач (обнаруженные дефекты, улучшения либо необходимые дополнительные изменения) к «исходящим» задачам с учётом основного объема работ и характеристик команды разработки, позволяет на базе экстраполяции оценить ожидаемую дату завершения работ и потенциальный сдвиг сроков разработки.

Список использованных источников:

1. Руководство к своду знаний по управлению проектами (Руководство PMBOK). – М.: Олимп-Бизнес, 2018 – 590с.
2. TM Forum. [Electronic resource] - <https://www.tmforum.org/>. Dateofaccess: 21.03.2018

АЛГОРИТМ DESX КАК СРЕДСТВО ПАРАЛЛЕЛЬНОГО ПОТОКОВОГО ШИФРОВАНИЯ БОЛЬШИХ ОБЪЁМОВ ДАННЫХ

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

Куцейко Р.Ю.

Ярмолик В.Н. – д.т.н., профессор

В настоящее время существует множество отраслей деятельности человека, которые сталкиваются с необходимостью хранения и обработки огромного количества цифровой информации на различных носителях. В случае необходимости защиты такого большого объема данных с помощью криптографии, наибольшей проблемой становится время, необходимое на процесс шифрования и дешифрования. В связи с тем, что скорость процессорной обработки информации на сегодняшний день растёт медленно, для решения поставленной задачи необходимо использовать параллельные вычисления, обеспечивающие многократное ускорение обработки и алгоритмы, поддающиеся разбиению на параллельные процессы. В качестве такого алгоритма можно рассматривать алгоритм потокового шифрования DESX. Процесс его работы поддается разбиению на отдельные параллельные потоки, а относительно простая реализация позволяет производить преобразования на высокой скорости.

DESX – блочный потоковый алгоритм шифрования, разработанный на базе стандарта DES (DataEncryptionStandard). Основное отличие от других модификаций заключается в добавлении процедуры отбеливания ключа шифрования для увеличения криптографической стойкости [1]. В основе алгоритма шифрования лежит преобразование сетью Фейстеля, которое представлено на рисунке 1.

На схеме наглядно изображен один из последовательных раундов преобразования входного блока данных. Блок разбивается на старшую и младшую части, после чего они обрабатываются отдельно. Вычисляется образующая функция F и происходит суммирование по модулю два для определения частей блока, который будут участвовать в следующем раунде шифрования. В зависимости от избранной стратегии шифрования (электронная кодировочная книга, сцепление блоков шифрованного текста, обратная связь по шифрованному тексту и обратная связь по выходу) можно добиться различных результатов шифрования со своими уникальными особенностями [2].

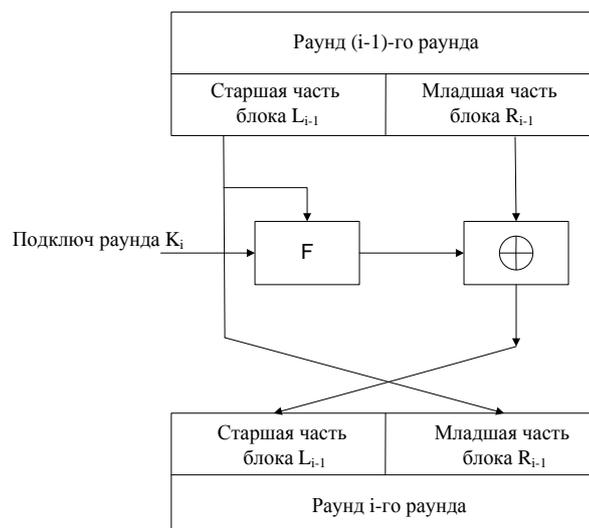


Рис. 1 –Раунд сети Фейстеля

Используя стратегию шифрования электронной кодировочной книги можно получить следующие преимущества:

- высокая скорость шифрования;
- легко поддается разбиению на параллельные процессы, что ещё больше повышает скорость шифрования [3];
- обладает приемлемыми показателями криптографической стойкости;

Основным недостатком можно считать более низкую, по сравнению с другими стратегиями шифрования, криптографическую стойкость. Три упомянутые стратегии шифрования (сцепление блоков шифрованного текста, обратная связь по шифрованному тексту и обратная связь по выходу) решают описанную проблему, однако гораздо хуже поддаются разбиению на параллельные процессы, вследствие необходимости учитывать на текущем шаге результат предыдущих вычислений. По этой причине данные подходы сильно уступают в скорости работы электронной кодировочной книге. Одной из возможностей решения описанной проблемы является разбиение данных на большие блоки, которые независимо будут шифроваться на основе предложенных стратегий. Однако, в таком случае, одинаковые блоки большого размера шифруются одинаковым образом.

Использование алгоритма шифрования DESX по стратегии электронной кодировочной книги позволяет достичь очень высокой скорости шифрования больших объемов данных и легко поддается разбиению на параллельные процессы, что делает такой подход легко масштабируемым и настраиваемым.

Список использованных источников:

1. Аграновский А. В, Хади Р. А. Практическая криптография (серия «Аспекты защиты»), М.: Солон-Пресс, 2002. 254 с.
2. Деднев, М. А. Защита информации в банковском деле и электронном бизнесе / М.А. Деднев. – М.: Кудиц-образ, 2004. – 186с.
3. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си / Брюс Шнайер. – М.Триумф, 2003. – 816 с Урядов В.Н.

АЛГОРИТМ РАБОТЫ СИСТЕМЫ УЧЕТА ЗАДОЛЖЕННОСТИ ПО БАНКОВСКИМ КРЕДИТНЫМ КАРТАМ

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

Ламчановский А.Г.

Бахтизин В.В. – к.т.н., профессор

Требования банков к функциональности систем учета клиентской задолженности по кредитным картам постоянно повышаются. Банки выводят на рынок кредитные карты с новыми условиями, увеличивается количество клиентов. Для учета и взыскания задолженности банки используют специальные информационные автоматизированные системы.

Просроченная задолженность – это непогашенная в срок задолженность по основному долгу или плановым процентам за пользование кредитом, а также иным платежам по кредитному договору. С момента возникновения просрочки по выплате по кредиту банк начисляет штрафы и пени, размер и принцип расчета которых указываются в кредитном договоре.

До сих пор не существует оптимального алгоритма решения проблемы взимания просроченной

задолженности. Существует множество нерешенных задач, связанных с управлением проблемными активами. Каждый банк самостоятельно выбирает пути работы с просроченной задолженностью в соответствии с его структурой, практикой взаимодействия подразделений банка, объемом и спецификой кредитного портфеля.

С клиентами, имеющими просроченную задолженность, работают специальные подразделения банков.

Предлагаемый алгоритм взыскания просроченной задолженности состоит из следующих шагов.

1) Отправить клиенту SMS-сообщение с информацией о задолженности по кредитной карте и просьбой оплатить. Если задолженность будет погашена в течение 3-х дней, перейти к шагу 7. Иначе перейти к шагу 2.

2) Сотрудник колл-центра банка звонит клиенту, продолжается отправка информационных SMS-сообщений. Если задолженность будет погашена в течение 60 дней от образования просроченной задолженности, перейти к шагу 7. Иначе перейти к шагу 3.

3) Сотрудник колл-центра банка связывается с доверенными лицами (как правило, родственниками) клиента. Клиенту банка отправляют SMS-сообщения и сообщения по электронной почте с просьбой погасить долг. Если задолженность будет погашена в течение 120 дней от образования просроченной задолженности, перейти к шагу 7. Иначе перейти к шагу 4.

4) Сотрудники банка выезжают по месту проживания и/или регистрации, а также работы клиента банка, и просят погасить задолженность, общаются с родственниками, коллегами и начальством клиента. Если клиент соглашается погасить просроченную задолженность, перейти к шагу 7. Иначе перейти к шагу 5.

5) Банк обращается в государственные органы с целью защиты своих законных интересов. Если клиент погашает просроченную задолженность, перейти к шагу 7. Иначе перейти к шагу 6.

6) Судебные приставы посещают клиента, и описывают его имущество в пользу банка.

7) Завершить алгоритм.

Автоматизированные информационные системы учета задолженности, прежде всего, предназначены для поддержки стадии переговоров с клиентом, так как эта стадия является самой эффективной на практике. С другой стороны, те же системы могут использоваться для учета работы с клиентами и на более поздних стадиях в целях упрощения формирования общей стратегии взыскания. Методы работы сотрудников банка, связанных с взысканием просроченной задолженности, и внешних подрядчиков разделяются в зависимости от стадии работы.

Данный алгоритм удовлетворяет требования законодательства Республики Беларусь. Его использование позволяет четко разграничить роли подразделений банка, повысить эффективность работы сотрудников, сократить время выполнения задач за счет автоматизации процессов.

Система учета задолженности по банковским картам должна взаимодействовать с другими системами банка. Для этих целей банки разрабатывают системы самостоятельно или покупают комплектные готовые решения для работы банка.

Примеры внешних систем, представленных на рынке Республики Беларусь:

- EPAM.Debt Collection – это гибко настраиваемая система, предназначенная и для розничных банков;

- Системные технологии Collection – система для предупреждения возникновения задолженности и управления процессами взыскания.

Список использованных источников:

1. Налогообложение, учет и отчетность в коммерческом банке // Методический журнал «Бухгалтерский учет просроченной задолженности физических лиц». – 2011. - №4 – С. 5-10.

2. Мирошниченко, Ю. В. Работа с просроченной задолженностью // InfoBank.by [Электронный ресурс] – 2017. - Режим доступа: <http://infobank.by/rabota-s-prosrochennoj-zadolzhennostju.html>

3. Лопатин, Ю. В. Аналитический отчет «ИССЛЕДОВАНИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ В БАНКОВСКОМ СЕКТОРЕ РЕСПУБЛИКИ БЕЛАРУСЬ» // Научно-технологическая ассоциация "Инфопарк" by [Электронный ресурс] - 2010. - Режим доступа: <http://infopark.by/sites/default/files/old/docs/7660.pdf>

ИСПОЛЬЗОВАНИЕ МНОГОКРАТНЫХ ТЕСТОВ ДЛЯ ПСЕВДОИЩЕРПЫВАЮЩЕГО ТЕСТИРОВАНИЯ ОЗУ

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

Леванцевич В.А.

Ярмолик В.Н. – д.т.н., профессор

Исчерпывающее тестирование (Exhaustive Testing) характеризуется максимальной эффективностью обнаружения неисправностей цифровых устройств и ошибок программного обеспечения [1–2]. Однако в силу большой временной сложности подобных тестов в настоящее время используются их различные аппроксимации и в первую очередь такие, как псевдоисчерпывающие тесты (Pseudoexhaustive Tests) [3]. Характерной особенностью псевдоисчерпывающих тестов является то, что их сложность, определяемая количеством тестовых наборов, существенно меньше сложности исчерпывающих тестов.

Еще одной эффективной аппроксимацией исчерпывающего тестирования является вероятностное тестирование (Random Testing) и его многочисленные модификации [4].

Существенным недостатком управляемых вероятностных тестов является необходимость перебора потенциальных кандидатов в тестовые наборы и вычисления для них характеристик, являющихся критериями для включения либо не включения их в вероятностный тест, что увеличивает вычислительную сложность формирования подобных тестов [1]. С целью уменьшения такой сложности широко обсуждаются и используются многократные тесты для запоминающих устройств [5].

Главным достоинством многократных вероятностных тестов является использование некоторой обобщающей характеристики для теста в целом, а не для тестового набора в отдельности. Это позволяет значительно уменьшить вычислительную сложность построения подобных тестов.

Для повышения эффективности повторного применения маршевых тестов радикальным подходом может быть как изменение последовательности адресов, применяемых в каждом последующем маршевом тесте, так и изменение начального состояния памяти [4].

В качестве анализируемого ОЗУ будем рассматривать запоминающее устройство, состоящее из N однокбитных запоминающих ячеек. Для тестирования подобных устройств, как правило, используются только маршевые тесты как единственно возможное решение для тестирования современных ОЗУ большой емкости [6]. Общая структура маршевых тестов состоит в последовательном обращении ко всем запоминающим ячейкам ОЗУ путем генерирования их адресов.

В качестве адресных последовательностей будем рассматривать только случайные адресные последовательности, каждая из которых формирует одну из неповторяющихся последовательностей адресов ОЗУ. В качестве маршевого теста используем простейший тест $\{\hat{1}(w0); \hat{1}(r0, w1)\}$, выполняющий инвертирование нулевого содержимого ОЗУ. Процедура тестирования ОЗУ будет состоять в многократном применении приведенного теста.

Для примера рассмотрим множество для тестирования k из N ячеек памяти, приведенное на рис. 1, состоящее из наборов двоичных векторов, называемых орбитами.

Рис. 1. Множество орбит для $k=2$

Видно, что для $k=2$ существуют только две орбиты. Исчерпывающий тест, т. е. генерирование всевозможных комбинаций (00, 01, 10, 11) из двух бит, обеспечивается двумя орбитами (1–2) в любой их последовательности. С увеличением значения k до трех получим шесть различных орбит (рис. 2).

Рис. 2. Множество орбит для $k=3$

Исчерпывающий тест для $k=3$ обеспечивается орбитами 1–4–5 и 2–3–6. Следует отметить, что число орбит три является минимальным для обеспечения исчерпывающего теста, состоящего из восьми двоичных комбинаций (000, 001, 010, ..., 111).

Общее количество орбит для произвольного значения k определяется как $Q_{\text{tot}} = k!$ [5].

Максимальная оценка Q_{max} количества орбит, необходимых для формирования исчерпывающего теста для произвольного значения k , содержит 2^k тестовых наборов. Применение одной орбиты независимо от ее вида обеспечивает $k+1$ двоичных векторов, каждый из которых состоит из k бит. В худшем случае каждая последующая орбита будет обеспечивать как минимум один новый двоичный вектор по отношению к множеству векторов, сгенерированных ранее использованными орбитами.

Тогда максимальная оценка Q_{max} будет вычисляться как

$$Q_{\text{max}} = 1 + (2^k - (k + 1)) = 2^k - k.$$

Минимальное количество орбит Q_{min} для формирования исчерпывающего теста определяется соотношением

$$Q_{\text{min}} = \binom{k}{\lceil k/2 \rceil}.$$

Фактически, чтобы обеспечить исчерпывающий тест для k ячеек ОЗУ, необходимо генерирование всех двоичных векторов из $\lceil k/2 \rceil$ единиц и $\lfloor k/2 \rfloor$ нулей, количество которых максимально по сравнению с другими комбинациями нулей и единиц. Это следует из свойства числа сочетаний из k по r для $k > 1$, $0 < r < k$:

$$\binom{k}{\lceil k/2 \rceil} > \binom{k}{r}.$$

Таким образом, Q_{min} будет равняться количеству двоичных векторов из $\lceil k/2 \rceil$ единиц и $\lfloor k/2 \rfloor$ нулей (1).

Для оценки среднего числа Q_{ave} орбит, которое необходимо для формирования исчерпывающего теста на произвольных k ячейках ОЗУ сформулируем необходимое условие генерирования исчерпывающего теста на произвольных k ячейках ОЗУ в виде следующего утверждения: необходимым условием для формирования исчерпывающего теста в произвольных k из N ячеек ОЗУ является генерирование для этих ячеек множества орбит, в которых все двоичные коды, содержащие $\lfloor k/2 \rfloor$ единиц и $k - \lfloor k/2 \rfloor$ нулей, будут сгенерированы хотя

бы по одному разу.

Тогда задача определения среднего числа количества орбит Q_{ave} , которые необходимы для формирования исчерпывающего теста на произвольных k ячейках ОЗУ, сводится к классической задаче собирателя купонов (CouponCollector'sProblem) и среднее количество Q_{ave} случайной выборки купонов (орбит) для получения всех купонов как минимум по одному разу согласно соотношению [7]

$$Q_{ave} = 1 + \frac{Q_{min}}{Q_{min} - 1} + \frac{Q_{min}}{Q_{min} - 2} + \dots + \frac{Q_{min}}{2} + Q_{min} = Q_{min} \sum_{n=1}^{Q_{min}} \frac{1}{n}.$$

Численные значения Q_{ave} для различных k приведены на рисунке 3.

Рис.3. Численные значения Q_{ave}

Выводы. Приведенные численные характеристики минимальной, максимальной и средней кратности многократного теста, позволяют сделать вывод о реальности применения многократных маршевых тестов с изменяемыми адресными последовательностями. для псевдоисчерпывающего тестирования современных ОЗУ.

Список использованных источников:

1. An Orchestrated Survey on Automated Software Test Case Generation / S. Anand [etal.] // Journal of Systems and Software. – 2014. – Vol. C-39, № 4. – P. 582–586.
2. Ярмолик, С.В. Управляемые вероятностные тесты / С.В. Ярмолик, В.Н. Ярмолик // Автоматика и телемеханика. – 2012. – № 10. – С. 142–155.
3. Mrozek, I. Iterative Antirandom Testing / I. Mrozek, V. Yarmolik // Journal of Electronic Testing: Theory and Applications (JETTA). – 2012. – Vol. 9, no. 3. – P. 251–266.
4. Ярмолик, С.В. Управляемое случайное тестирование / С.В. Ярмолик, В.Н. Ярмолик // Информатика. – 2011. – № 1(29). – С. 79–88.
5. Ярмолик, С.В. Многократные неразрушающие маршевые тесты с изменяемыми адресными последовательностями / С.В. Ярмолик, В.Н. Ярмолик // Автоматика и телемеханика. – 2007. – № 4. – С. 126–137.
6. Goor, A.J. Testing Semiconductor Memories, Theory and Practice / A.J. Goor. – Chichester, UK: John Wiley & Sons, 1991. – 536 p.
7. Flajolet, P. Birthday paradox, coupon collectors, caching algorithms and self-organizing search / P. Flajolet, D. Gardy, L. Thimonier // Discrete Appl. Math. – 1992. – No. 39. – P. 207–229.

ПРОГРАММНО-АЛГОРИТМИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ОБРАБОТКИ ИНФОРМАЦИИ В ТОРГОВОЙ ДЕЯТЕЛЬНОСТИ

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

Левашенко П.С.

Лукьянец В.Г. – к.т.н., доцент

В данном докладе проанализированы способы и средства автоматизации обработки информации в торговой деятельности. Выявлены и проанализированы основные проблемы обработки информации в торговой деятельности. В результате анализа моделей, методов и средств, и выявления недостатков было принято решение разработать программное обеспечение, обеспечивающее обработку информации на предприятии.

Актуальность данной проблемы заключается в том, что стремительное развитие товарных и финансовых рынков явилось мощным толчком к интенсивному нарастанию процессов информатизации во всех сферах жизни общества. Изменились подходы к оценке роли информации и информационному обслуживанию производственно-хозяйственной, управленческой деятельности и различных категорий пользователей.

Эффективное управление торговым предприятием в современных условиях невозможно без использования компьютерных технологий. Белорусские предприятия в области применения этих технологий значительно уступают зарубежным конкурентам.

Иностранные предприятия, как правило, имеют опыт модернизации и внедрения не одного поколения ИС. В развитых западных странах происходит смена уже четвертого поколения ИС. На белорусских предприятиях зачастую используют системы первого или второго поколения.

Руководители многих белорусских предприятий имеют слабое представление о современных компьютерных интегрированных системах и предпочитают содержать большой штат собственных программистов, которые разрабатывают индивидуальные программы для решения стандартных управленческих задач.

Внедрение системы автоматизации на предприятиях торговли, как и любое серьезное преобразование на предприятии, является сложным и зачастую болезненным процессом. Тем не менее, некоторые проблемы,

возникающие при внедрении системы, достаточно хорошо изучены, формализованы и имеют эффективные методологии решения. Заблаговременное изучение этих проблем и подготовка к ним значительно облегчают процесс внедрения и повышают эффективность дальнейшего использования системы. Целью автоматизации торговли является повышение эффективности управления предприятием, и как следствие увеличение объема продаж и рост прибыльности.

Всем известно, что торговля, в отличие от производства, не создает материальные ценности, а выступает в качестве посредника между производителем и потребителем, доставляя товары покупателям и возвращая производственному предприятию деньги за товар. Являясь посредником, оно пропускает через себя финансовые и товарные потоки, оставляя себе процент за услуги.

Рис. 1 – Торговое предприятие как посредник

Business Engine функционирует относительно подразделений предприятия по схеме клиент — операция — товар. Клиенты, взаимодействуя между собой, инициируют начало операций, которые приводят к перемещению товаров и денег или только денег. Каждая операция состоит из нескольких подопераций (транзакций), которые имеют свой набор и последовательность (например, получил товар, перечислил деньги, отправил товар и т. д.). В свою очередь каждая транзакция инициирует перемещение денег по счетам и товаров по складам. В экономических отношениях каждый клиент имеет одну или несколько ролей, которые добавляют ему возможности осуществления различного рода операций.

Рис. 2 –Business Engine

Вся мощь Business Engine заключается в том, что он представляет собой программно-реализованную экономическую модель с необходимыми для всех торговых предприятий операциями. А вот на базе какого типа торговли, товара или структуры подразделений предприятия реализовать это программное решение — решает само предприятие торговли.

Прежде чем приступать к внедрению системы автоматизации на предприятии, необходимо ответить на два вопроса: действительно ли предприятию нужна автоматизация и если все же нужна, то насколько обширной она должна быть. Рассмотрим предприятие ОАО «Белбакалея», которое занимается мелкооптовой и розничной торговлей

При выборе схемы автоматизации в данной работе одну из главных ролей играет актуальные потребности предприятия. Потребности предприятия очевидны.

Во-первых, потребность в наличии полной и достоверной информации о товаре, в обеспечении достаточной скорости и надлежащего качества выполнения складских операций, в организации системы контроля персонала и т.п.

Вторая потребность в существенном ускорении оборота склада, многоуровневой аналитике, автоматизации нетиповых складских операций.

Положительные стороны автоматизации – это то, что она позволит решить все вышеперечисленные проблемы и позволит:

- Увеличить пропускную способность складского помещения, а также позволит в реальном времени отслеживать остатки, положение товаров на складе, повысить эффективность управления;
- Автоматизация поможет проводить небольшие маркетинговые исследования, изучать рынки сбыта и пробовать различные политики ценообразования.

С помощью автоматизации будет проводиться детальный учет номенклатуры в разрезе качества, партий, сроков годности, серийных номеров.

Предметной областью решаемой задачи является работа товароведов, а именно автоматизация учета заявок, формирование отчетности и контроль за наличием товаров на складе. Товаровед занимается персональным учетом товара на складе, его своевременным заказом. Через него проходят потоки информации. Это входящая, а затем уже переработанная исходящая информация. В настоящее время товароведы вручную строят отчет по продажам товара, подсчитывают проданный товар и товар на складе, затрачивая на это много времени. Но с помощью разрабатываемого программного средства товароведы смогут сэкономить много времени.

Предполагается, что если осуществить разработку программного обеспечения, обеспечивающего обработку информации на предприятии ОАО «Белбакалея» с дальнейшим её внедрением в деятельность предприятия, то возрастет эффективность работ, а именно автоматизация учета заявок, формирование отчетности, контроль за наличием товаров на складе, упрощение работы, связанной с поиском, редактированием и добавлением информации.

В настоящее время на предприятии ОАО «Белбакалея» рабочее место товароведов частично автоматизировано. Т.е. товаровед пользуется некоторыми программными продуктами, которые не автоматизируют его труд полностью. Нет единого интегрированного решения, которое бы увеличило эффективность работы. Под интегрированным решением понимается система обработки информации. Эта система необходима для полной автоматизации работы складского учета, чтобы у товароведов освободилось больше времени для маркетинговых исследований рынка, а грамотно проведенные маркетинговые исследования увеличат продажи, и, следовательно, прибыль магазина. Сейчас существует огромный ряд

программных решений, которые автоматизируют складской учет. Предприятию в первую очередь требуется относительно не дорогой программный продукт.

В результате анализа моделей, методов и средств, и выявления недостатков было принято решение разработать программное обеспечение, обеспечивающее обработку информации на предприятии ОАО «Белбакалея» с учетом вышеуказанной информации. Разработанное программное обеспечение разработано для товароведа, который является работником ОАО «Белбакалея». Данный программный продукт может быть использован в других организациях торговой сферы.

Список использованных источников:

1. В.В. Брага, Автоматизированные информационные технологии в экономике: учеб. для вузов / Н.Г. Бубнова, Л.А. Вдовенко, В.И. Гусев; под ред. Г.А. Титоренко. – М.: ЮНИТИ, 2003. – 399 с.
2. Гради, Буч. Объектно-ориентированный анализ и проектирование. – М.: Бином, 1998.
3. Петров, В.Н. Информационные системы: пособие для вузов / В.Н. Петров. – СПб. и др.: Питер, 2003. – 687 с.

ИДЕНТИФИКАЦИЯ РУКОПИСНЫХ ПОДПИСЕЙ С ИСПОЛЬЗОВАНИЕМ НЕЙРОСЕТЕЙ

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

Липский В.Г.

Данилова Г.В. – м.т.н., ассистент

Системы верификации подписи нацелены на идентификацию личности человека, распознавая его рукописную подпись. Несмотря на многие достижения в данной области за последние несколько десятилетий, создание систем, которые могут различать подлинные подписи и фальшивки, по-прежнему является трудной задачей, о чем свидетельствуют высокие коэффициенты ошибок, полученные при тестировании данных систем на больших наборах данных. Для того, чтобы успешно решить данную проблему, можно использовать искусственные нейросети.

Искусственная нейронная сеть (ИНС) — математическая модель, а также её программное или аппаратное воплощение, построенная по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма [1]. Это понятие возникло при изучении процессов, протекающих в мозге, и при попытке смоделировать эти процессы. Первой такой попыткой были нейронные сети У. Маккалока и У. Питтса. После разработки алгоритмов обучения, получаемые модели стали использовать в практических целях: в задачах прогнозирования, для распознавания образов, в задачах управления и др.

При распознавании образов в качестве самих образов могут выступать различные по своей природе объекты: символы текста, изображения, образцы звуков и т.д. При обучении сети предлагаются различные образцы образов с указанием того, к какому классу они относятся. Образец, как правило, представляется как вектор значений признаков. При этом совокупность всех признаков должна однозначно определять класс, к которому относится образец. В случае если признаков недостаточно, сеть может соотнести один и тот же образец с несколькими классами, что неверно. По окончании обучения сети ей можно предъявлять неизвестные ранее образы и получать ответ о принадлежности к определённому классу.

ИНС представляет собой систему соединённых и взаимодействующих между собой простых процессоров (искусственных нейронов). Такие процессоры обычно довольно просты (особенно в сравнении с процессорами, используемыми в персональных компьютерах). Каждый процессор подобной сети имеет дело только с сигналами, которые он периодически получает, и сигналами, которые он периодически посылает другим процессорам. И, тем не менее, будучи соединёнными в достаточно большую сеть с управляемым взаимодействием, такие по отдельности простые процессоры вместе способны выполнять довольно сложные задачи.

Нейронные сети не программируются в привычном смысле этого слова, они обучаются. Возможность обучения — одно из главных преимуществ нейронных сетей перед традиционными алгоритмами. Технически обучение заключается в нахождении коэффициентов связей между нейронами. В процессе обучения нейронная сеть способна выявлять сложные зависимости между входными данными и выходными, а также выполнять обобщение. Это значит, что в случае успешного обучения сеть сможет вернуть верный результат на основании данных, которые отсутствовали в обучающей выборке, а также неполных и/или «зашумленных», частично искажённых данных.

Системы верификации подписей работают с изображениями данных подписей и это является задачей распознавания образов. С такими задачами хорошо справляются свёрточные нейронные сети [2]. 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. Таким образом, используя свёрточные нейронные сети, можно добиться верной идентификации подписи в 80%-90% случаях. Однако главным минусом данного подхода является то, что для обучения нейросети необходим достаточно большой набор изображений с подписью человека, чья личность будет идентифицироваться.

Данные системы могут быть успешно применены в различных сферах. Например, в университете для

проверки того, что преподаватель действительно сам поставил свою подпись в зачетке или ведомости, а не студент подделал его подпись или для проверки того, что студент сам писал определенную проверочную работу. Для этого необходимо собрать достаточно большое количество отсканированных подписей преподавателей и студентов, потом с помощью них обучить нейросеть и сохранить обученную модель. В дальнейшем при проверке подлинности подписи нужно будет только получить ее изображение и с помощью обученной модели нейросети идентифицировать ее.

Список использованных источников:

1. Мак-Каллок У. С., Питтс В. Логическое исчисление идей, относящихся к нервной активности // Автоматы / Под ред. К. Э. Шеннона и Дж. Маккарти. — М.: Изд-во иностр. лит., 1956. — С. 363—384.
2. Галушкин А. И. Синтез многослойных систем распознавания образов. — М.: Энергия, 1974. — 440 с.

МОДЕЛИ И РЕАЛИЗАЦИЯ ДИСТАНЦИОННОГО УПРАВЛЕНИЯ СКЛАДСКОЙ ДЕЯТЕЛЬНОСТЬЮ

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

Лукашевич К.В.

Лукьянец В.Г.. – к.т.н., доцент

Профессиональное управление складом и его логистикой неизбежно требует глубокого анализа, позволяющего наиболее точно оценить неопределенность ситуации с помощью современных методов исследования. В связи с этим существенно возрастает приоритет и роль анализа, основным содержанием которого является комплексное системное изучение состояния предприятия.

Система управления складом (сокр. WMS от англ. Warehouse Management System — система управления складом) — система управления, обеспечивающая автоматизацию и оптимизацию всех процессов складской работы профильного предприятия.

Существуют следующие, основные, модели управления складской деятельностью:

- 1) получение/прием товара – прием, проверка соответствия поставки сопроводительным документам и целостности товара;
- 2) хранение товара – определение мест хранения поступающих на склад товарно-материальных ценностей (ТМЦ) для товара, сортировка, построение оптимальных маршрутов, размещение грузов в зоне хранения;
- 3) комплектация товара – отбор товара из зоны хранения, комплектация и упаковка;
- 4) отгрузка товара – формирование партий отгрузки и отгрузки товара клиентам.

Вне зависимости от реализации, будет ли это десктопное приложение или веб, фактически WMS системы могут существовать только когда создана инфраструктура. А это, в первую очередь, мощный сервер, RAID-массив. Однако, используя технологию Blockchain, можно сократить все эти издержки, абсолютно не потеряв в производительности.

Blockchain – это способ хранения данных или цифровой реестр транзакций, сделок, контрактов. Всего, что нуждается в отдельной независимой записи и, при необходимости, в проверке. В блокчейне можно хранить данные о выданных кредитах, правах на собственность, нарушении правил дорожного движения, товарах. То есть практически обо всем. Главным его отличием и неоспоримым преимуществом является то, что этот реестр не хранится в каком-то одном месте. Он распределён среди нескольких сотен и даже тысяч компьютеров во всем мире. Любой Пользователь этой сети может иметь свободный доступ к актуальной версии реестра, что делает его прозрачным абсолютно для всех участников.

Существуют следующие типы блокчейн:

– Открытый блокчейн (англ. public blockchain) — блокчейн, в котором не существует ограничений на чтение данных блоков (при этом данные могут быть зашифрованы) и ограничений на отсылку транзакций для включения в блокчейн.

– Закрытый блокчейн (англ. private blockchain) — блокчейн, в котором прямой доступ к данным и к отправке транзакций ограничен определенным узким кругом организаций.

– Общедоступный (инклюзивный) блокчейн (англ. permissionless blockchain) — блокчейн, в котором не существует ограничений на личность обработчиков транзакций (т. е., пользователей, которые могут создавать блоки транзакций).

– Эксклюзивный блокчейн (англ. permissioned blockchain) — блокчейн, в котором обработка транзакций осуществляется определенным списком субъектов с установленными личностями.

Естественно, для бизнеса необходимо использовать только закрытый или эксклюзивный тип. Блокчейн в управлении складской деятельностью поможет решать следующие задачи:

- Распределенность, безопасность, скорость обработки транзакций
- Борьба с мошенничеством и ошибками доставки
- Прозрачность цепи поставок

Разрабатываемое программное обеспечение с использованием технологии блокчейн позволит решить

все задачи, поставленные перед разработчиком программного обеспечения для реализации дистанционного управления складской деятельностью.

Список использованных источников:

1. «Логистика». / Под. Ред Б.А. Аникина М., 2014 г.2. «Логистика». А.Д. Чудаев М., 2011г.
2. «Основы логистики» /Под. Ред Л.Б. Миротина и В.И Сергеева. М.: ИНФРА –М, 2013г.
3. Дыбская В. В. Логистика для практиков. «Эффективные решения в складировании и грузопереработке». – М.: ВИНТИ РАН, 2006. 264с., гл.6,7

УНИФИЦИРОВАННЫЙ ИНТЕРФЕЙС ДЛЯ СОЗДАНИЯ КРОССПЛАТФОРМЕННЫХ ВЕБ-ОРИЕНТИРОВАННЫХ SMARTTV ПРИЛОЖЕНИЙ

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

Максимченко А.В.

Таборовец В.В. – к.т.н., доцент

В современном мире при появлении SmartTV телевизоров вопрос о создании для них веб-приложений становится все более и более актуальным. За последнее время SmartTV платформ появилось большое количество, и все они в большинстве случаев по-разному работают. Это приводит к решению следующей задачи при написании приложений: способность приложения работать на всех платформах и операционных системах SmartTV телевизоров одинаковым образом. Одним из вариантов решений этой проблемы является унифицирование интерфейсов различных SmartTV платформ и операционных систем для платформо-независимого создания веб-ориентированных SmartTV приложений.

SmartTV или ConnectedTV - 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 SmartTV было выявлено, что каждая система предоставляет свой набор характерных настроек для взаимодействия с ней. Это можно увидеть наглядно при реализации воспроизведения видео контента, управления приложением посредством мыши и клавиатуры, взаимодействие приложения с Интернет.

На глубокое изучение и понимание вариантов реализации этих функций в приложении требуется большое количество времени. Это увеличивает процесс разработки и вероятность допущения ошибки в написании приложения под различные операционные системы и платформы в целом.

Каждая операционная система SmartTV, такая, к примеру, как WebOS, Orsey, Tizen, Netcast, требует детального разбора и понимания своих характерных методов для создания видео-плеера и работы с видео-материалом, такой как воспроизведение, пауза, режим прототки в прямом и обратным направлении, на определенную позицию, регулирование громкости, подсчет времени длительности просматриваемого видео-материала и текущего времени, задание качества видео потока и многое другое.

Схожая проблема возникает и при реализации взаимодействия SmartTV телевизора и разрабатываемого приложения с использованием клавиш на пульте телевизора. Каждая операционная система SmartTV предоставляет свой собственный набор кодов клавиш, которые приложение должно уметь правильно обрабатывать: определять, к какой операционной системе они относятся и реагировать должным образом. Это, как правило, создает массу неудобств, поскольку большинство кодов клавиш разные и это ведет к путаницы в реализации различных интерфейсов. Примеры разнообразия кодов клавиш можно наглядно увидеть на ресурсах [2] и [3].

Часто при реализации веб-приложений для SmartTV 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 теле-платформы для успешной реализации этих функций.

Решения этих многих проблем получилось достичь путем реализации унифицированного интерфейса, который инкапсулирует логику взаимодействия приложения с телевизором, распознаёт текущую платформу SmartTV, и предоставляет разработчикам открытый интерфейс, позволяющий реализовать функции для работы со SmartTV телевизорами без углубления в работу самого интерфейса.

Унифицированный интерфейс реализует сервис, который позволяет создать видеоплеер, управлять приложением при помощи контроллеров мыши и клавиатуры, осуществлять различные проверки телевизора на работоспособность, доступ к сети и многое другое для каждой платформы SmartTV. Затем 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 SmartTV интерфейс позволяет абстрагироваться от знаний о конкретных интерфейсах различных платформ и операционных систем телевизоров с технологией SmartTV 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.
2. HandlingControlKeyEvents | SamsungDevelopers [Электронный ресурс]. - Электронные данные. - Режим доступа: <http://developer.samsung.com/tv/develop/legacy-platform-library/art00046/index>.
3. LG | webOSTVDeveloper | RemoteControl [Электронный ресурс]. - Электронные данные. - Режим доступа: <http://webostv.developer.lge.com/design/webos-tv-system-ui/remote-control>.

БАЛАНСИРОВАНИЕ НАГРУЗКИ В WEB ПРИЛОЖЕНИЯХ

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

Малич К.В.

Куликов С. С. – к.т.н., доцент

В настоящее время современные приложения работают с постоянно растущими объемами входящих данных. Для обеспечения поддержки нормального функционирования приложения существуют различные методы. К таким методам можно отнести балансирование нагрузки.

Балансирование нагрузки представляет собой распределение входящих запросов между несколькими сетевыми устройствами. Это позволяет поддерживать отказоустойчивость, предоставляет возможности для горизонтального масштабирования.

В Web приложениях в качестве балансировщиков нагрузки используют несколько Web серверов, все используемые сервера называются фермой серверов. Пример топологии с несколькими серверами представлен на рисунке 1:

Рис. 1 – использование балансировщика нагрузки.

При такой топологии вводится понятие Прокси, который позволяет распределять нагрузку между серверами. Прокси – сервис, который принимает все входящие запросы и распределяет их между серверами в ферме в соответствии с их загруженностью. Как правило, прокси использует очереди для хранения входящих запросов.

Также использование балансировщиков нагрузки позволяет справиться с проблемой отказа одного из серверов. В таком случае вся нагрузка ложится на остальные сервера, но приложение продолжает свою работу, когда как в случае с единственным сервером, приложение бы прекратило свое корректное функционирование.

Также к положительным сторонам использования балансировщиков нагрузки можно отнести простоту горизонтального масштабирования, это значит, что для сокращения нагрузки на существующие сервера достаточно добавить новый сервер, который также будет обрабатывать входящие запросы.

Данный подход является актуальным в наши дни, и большинство крупных систем используют балансирование нагрузки для поддержания работоспособности.

ЭМПИРИЧЕСКАЯ МОДОВАЯ ДЕКОМПОЗИЦИЯ С ИСПОЛЬЗОВАНИЕМ ГРАФИЧЕСКИХ ПРОЦЕССОРОВ NVIDIA, ПОДДЕРЖИВАЮЩИХ ПРОГРАММНО-АППАРАТНУЮ АРХИТЕКТУРУ CUDA

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

Мелещня Д.В.

Бранцевич П.Ю. – к.т.н., доцент

Эмпирическая модовая декомпозиция является ключевой частью преобразования Гильберта-Хуанга. В силу того, что этапы преобразования включают построение огибающих с помощью интерполяционных полиномов, вся процедура представляет достаточно сложную вычислительную задачу. Для сокращения времени декомпозиции предлагается использовать графические процессоры, реализующих CUDA-совместимую архитектуру, а также, предлагаются подходы по работе с памятью, методы глобальной синхронизации и полной редукции на графическом процессоре, позволяющие

ускорить декомпозицию сигнала на модовые функции.

Эмпирическая модовая декомпозиция (EMD – Empirical Mode Decomposition) – итеративная процедура, ставящая в соответствие исходному сигналу набор эмпирических мод (IMF – Intrinsic Mode Functions). Модовая декомпозиция, или просеивание, сводится к последовательности следующих этапов. По локальным экстремумам строятся верхняя и нижняя огибающие. После этого, вычисляется разность между средним значением огибающих и исходным сигналом. Далее, если остаток удовлетворяет критерию остановки, он считается очередной модовой функцией, в противном случае разность принимают за исходный сигнал и алгоритм повторяется. После нахождения IMF, ее вычитают из исходного сигнала и, если разность не является монотонной функцией либо меньше некоторого порогового значения, то алгоритм просеивания повторяется.

Ключевой этап декомпозиции – построение огибающих на основе локальных экстремумов. Для интерполяции на участках между экстремумами используются кубические сплайны. Для того чтобы найти одну модовую функцию, в среднем, необходимо 6-8 операций просеивания, или 12-16 построений огибающих. Таким образом, процесс просеивания сам по себе может выполняться достаточно долго. При этом, в зависимости от исходного сигнала, для нахождения всех внутренних модовых функций необходимо выполнить несколько итераций просеивания. Экспериментальные исследования показали, что в случае разложения реальных сигналов количество получаемых модовых функций растет с увеличением длины исходного сигнала. Подытоживая данные факты можно сделать вывод, что в случае длинных сигнальных реализация, эмпирическая модовая декомпозиция требует много времени и вычислительных ресурсов, количество в лучшем случае зависит линейно от количества точек исходного сигнала.

Для того чтобы сократить время преобразования сигнала предлагается использовать графический процессор (GPU – Graphics Processing Unit). OpenCL и CUDA являются наиболее популярными фреймворками для вычислений общего назначения на GPU (GPGPU – General-Purpose Computing For GPU), поскольку предоставляют возможность писать программы для графического процессора основываясь на стандарте языка Си и поддерживаются крупными производителями оборудования. В общем оба фреймворка реализуют схожую модель памяти и вычислений, однако в силу того, что адаптеры NVIDIA для GPGPU доступны как облачные сервисы, использование CUDA является более удобным с точки зрения разработки и дальнейшего использования.

В реализации преобразования Гильберта-Хуанга первый этап – нахождение локальных экстремумов – выполняется на центральном процессоре (CPU – CentralProcessingUnit). Далее полученный массив точек копируется в глобальную память GPU. При этом выделяется только один участок памяти для хранения минимумов и максимумов, а также сервисных структур данных, идентифицирующих принадлежность точек исходной кривой тому или иному интервалу интерполяции. В дальнейшем, для доступа к массивам используется указатели – каждый с соответствующим смещением. При частом выделении памяти, которое характерно для ННТ с большим количеством итераций просеивания, такой подход позволяет сократить суммарное время расчета на 20 - 25%, поскольку каждая операция выделения памяти ведет к дополнительным накладным расходам.

Первым этапом после того, как в память графического устройства скопированы все данные, является экстраполяция на интервалах между экстремумами. Для этого используется сплайны Катмула-Рома. Суть метода заключается в том, что для каждого интервала исходной функции находится свой полином, описывающий кривую на данном участке. При этом значение в некоторой точке произвольного интервала определяется по следующей формуле:

где

– базисные функции Эрмита.

За первый проход на основе локальных экстремумов вычисляются коэффициенты интерполирующих полиномов. Полученные коэффициенты сохраняются в глобальную память устройства. Вместо того, чтобы хранить коэффициенты в виде массива структур, для коэффициентов при одинаковой степени неизвестного всех интервалов выделяется свой массив. Таким образом, обеспечивается выравнивание данных в памяти. А поскольку особенностью CUDA-совместимой архитектуры является объединение доступа к локальной памяти, это позволяет сократить задержки при чтении и наиболее эффективно использовать кэш первого уровня L1.

За второй проход интерполяции на основе полученных на предыдущем этапе коэффициентов вычисляется значения функции в точках между интервалами. При нахождении значения функции в заданной точке используется схема Горнера. В соответствии с этим правилом многочлен n-й степени:

представляется в виде:

Это позволяет сократить количество операций умножения с 7 до 3, а с учетом того, что необходимо строить верхнюю и нижнюю огибающие – с 14 до 6. После этого находится среднее значение огибающих, которое вычитается из исходного сигнала.

Последним этапом в итерации просеивания является определение, найдена ли очередная модовая функция. В качестве критерия останова используется среднеквадратичная разность между исходным сигналом и возможной модой. Если она не превышает заданного значения, очередная внутренняя модовая функция считается найденной. Для нахождения среднеквадратичной разности используется достаточно часто применяемый в параллельной обработке данных подход – редукция. Однако в классическом варианте заключительный этап производится на центральном процессоре, что может быть не совсем эффективным в случае с длинными сигнальными реализациями. Поэтому в предложенном алгоритме модовой декомпозиции все итерации редукции производятся на графическом адаптере, а для синхронизации всех потоков используется перезапуск ядра, так как CUDA поддерживает только синхронизацию на уровне группы потоков, но не содержит примитивов синхронизации всех потоков. Еще одним архитектурным решением, позволяющим сократить время вычисления, является использование разделяемой памяти внутри блоков. Доступ к разделяемой памяти осуществляется гораздо быстрее чем к глобальной, поэтому каждый блок работает с копией данных в разделяемой памяти.

На рисунке 1 представлено сравнение трех реализаций преобразования Гильберта-Хуанга: для CPU, GPU и вариант для GPU с оптимизацией чтения из глобальной памяти, редукции и использованием разделяемой памяти.

Рис. 1 – Сравнение зависимости времени декомпозиции сигнала от его длины при запуске на графическом и центральном процессорах (по оси ординат время в миллисекундах, по оси абсцисс количество точек)

Экспериментальные исследования проводились с использованием процессора IntelXeonE5450 и графического адаптера NvidiaTeslaK80.

Список использованных источников:

1. Роджерс, Д. Математические основы машинной графики / Д. Роджерс, Дж. Адамс. – М.: Мир. – 2001. – 604 с.
2. Huang, N.E. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis / N.E. Huang, Z. Shen, S.R. Long, M.C. Wu, H.H. Shih and other. – 1998. – Т.454. – с.903 – 995.
3. Huang E., An Introduction to Hilbert-Huang Transform: A Plea for Adaptive Data Analysis. Research Center for Adaptive Data Analysis.
4. Sanders, J. CUDA by Example: An Introduction to General-Purpose GPU Programming / J. Sanders, E. Kandrot. – Paperback, – 2010 – 279 p.
5. Документация CUDA Toolkit // NVIDIA [Электронный ресурс]. – 2018. – Режим доступа: <http://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html> Дата доступа: 18.01.05

МОДЕЛЬ ПОСТРОЕНИЯ ОБЩИХ ТРЕБОВАНИЙ ПО ОБРАБОТКЕ БАНКОВ ДАННЫХ В ЗАДАЧАХ СОПРОВОЖДЕНИЯ СИСТЕМ

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

Моженкова Е.В.

Парамонов А.И. – к.т.н., доцент

Изменения программного обеспечения необходимы для адаптации к повышенным функциональным требованиям и различным системным конфигурациям, вызванным этими изменениями. Корпоративная информационная система (КИС) становятся все более сложной по мере роста и развития, поэтому поддержание такой системы является основной задачей для отрасли. Одной из наиболее актуальных проблем в компьютерной индустрии является необходимость поддерживать и улучшать программный продукт с более высокими темпами и с меньшими затратами.

На основе результатов, которые получены в исследованиях [1-2], можно сформулировать одну из главных целей обработки банков данных в задачах сопровождения КИС – определение области расширения автоматизации в задачах сбора и анализа данных клиента по выявленной проблеме с целью дальнейшей локализации бизнес-процесса системы на стороне разработчика. Эта проблема особенно актуальна в свете разнородности структур баз данных КИС. На рисунке 1 представлена контекстная диаграмма модели сбора общих требований к программному средству обработки банков данных в нотации IDEF0.

Модель включает внешние и внутренние факторы, влияющие на требования к программному средству обработки банков данных. Контекстная диаграмма состоит из четырех процессов:

– «Определение логики построения структуры данных» – информационная структура предприятия определяет эксплуатационные характеристики КИС и уровень безопасности обрабатываемых в системе данных;

– «Составление структуры БД КИС» – бизнес-процессы производственного процесса влияют на логику построения структуры данных;

– «Проектирование модели обработки банков данных» – сведения о метаданных БД являются источником набора сущностей и связей между ними, которые должна обработать модель для построения алгоритма сбора необходимых данных;

– «Формирование общих требований к разработке ПС» – выходной информацией контекстной диаграммы является техническое задание на разработку ПС обработки банков данных.

Каждый процесс отвечает за определенную область функционирования модели. Управляющими стрелками диаграммы являются: устав предприятия, модель угроз безопасности КИС, а также стандарты управления данными – ГОСТ 7.70-2003, ГОСТ 34.321-96 [3, 4].

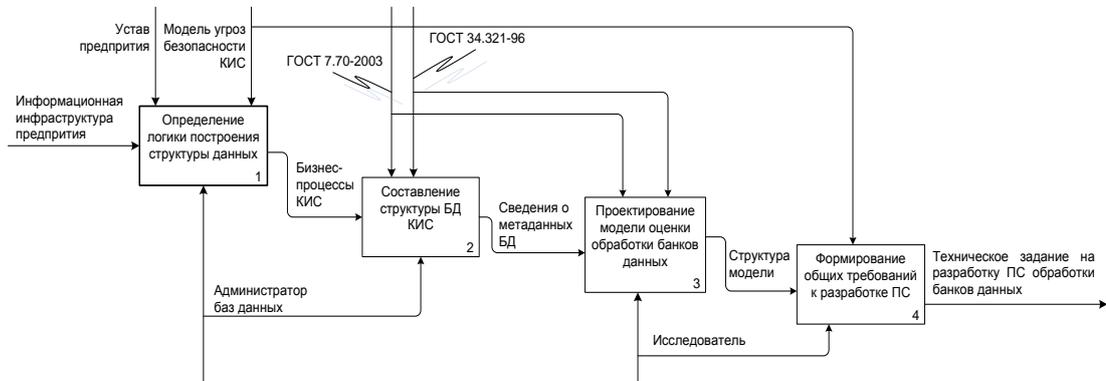


Рис. 1 – Функциональная модель сбора общих требований обработки банков данных

Вышеобозначенные процессы влияют на модель оценки обработки банков данных, и, следовательно, влияют на процессы: оценка структуры КИС, оценка структуры БД, определение набора обрабатываемых таблиц, определение набора данных для обезличивания.

Построение функциональной модели позволило формализовать общие требования к разработке программного средства обработки банков данных. Одной из актуальных и востребованных задач становится построение математической модели оценки обработки банков данных различных архитектур КИС, которая в дальнейшем будет внедрена в логику работы программного средства.

Список использованных источников:

1. Моженкова, Е. В. Проблемы и задачи при работе с корпоративными вычислительными сетями / Е. В. Моженкова, А.И. Парамонов // Компьютерные системы и сети: материалы 53-й научной конференции аспирантов, магистрантов и студентов (Минск, 2 – 6 мая 2017 г.). – Минск: БГУИР, 2017. – С. 101 – 103.
2. Моженкова, Е. В. Актуальность проблем и задач сопровождения корпоративных информационных систем / Е. В. Моженкова, А. И. Парамонов // Информационные технологии и системы 2017 (ИТС 2017) = Information Technologies and Systems 2017 (ITS 2017): материалы междунар. науч. конф. (Республика Беларусь, Минск, 25 октября 2017 года) / редкол.: Л. Ю. Шилин [и др.]. – Минск: БГУИР, 2017. – С. 270 - 271.
3. ГОСТ 7.70-2003 Описание баз данных и машиночитаемых информационных массивов.
4. ГОСТ 34.321-96 Информационные технологии. Система стандартов по базам данных. Эталонная модель управления данными.

ПОСТРОЕНИЕ ИСКУССТВЕННОЙ НЕЙРОННОЙ СЕТИ ДЛЯ РАСПОЗНАВАНИЯ ОБРАЗОВ

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

Моисеенко И.В.

Серебряная Л.В. – к.т.н., доцент

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 временных рядов, цель которого - прогнозирование их дальнейшего развития.

Успешно зарекомендовавшим себя инструментом распознавания является искусственная нейронная сеть (ИНС). ИНС возникли на основе знаний о функционировании нервной системы живых существ. Они представляют собой попытку использования процессов, происходящих в нервных системах, для выработки новых технологических решений [1].

Существует много видов искусственных нейронных сетей с разными архитектурами (рис. 1), подходящих для распознавания образов, однако алгоритм их построения будет схож.

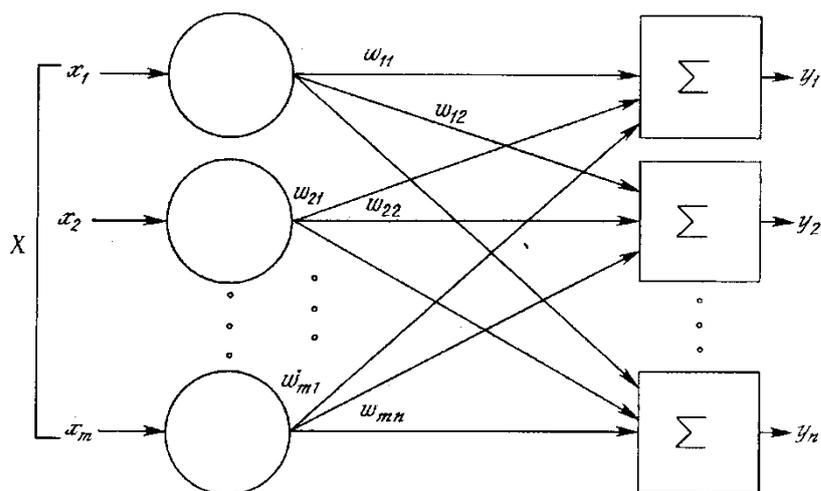


Рис. 1 - Модель искусственной нейронной сети

Построение ИНС подразделяется на следующие этапы:

1) Сбор данных для обучения:

Набор данных для обучения должен быть репрезентативным (данные должны иллюстрировать истинное положение вещей в предметной области) и непротиворечивым (противоречивые данные в обучающей выборке приведут к плохому качеству обучения сети). При необходимости применяются методы искусственного расширения обучающей выборки [3].

2) Выбор топологии сети:

Постановка задачи и имеющиеся данные для обучения определяют тип сети. Для обучения с учителем необходимо наличие для каждого элемента выборки "экспертной оценки". Если такой оценки нет (или получить ее невозможно), выбором будет ИНС, обучающаяся без учителя.

3) Подбор параметров обучения и характеристик сети:

На данном этапе происходит подбор количества слоев ИНС, кол-во нейронов в каждом слое. Чем больше суммарное число связей между нейронами, тем выше способности ИНС к обобщению. Количество нейронов в выходном слое, как правило, равно количеству определяемых классов [4]. При этом устанавливается соответствие между выходом нейронной сети и классом, который он представляет. Когда сети предъявляется некий образ, на одном из её выходов должен появиться признак того, что образ принадлежит этому классу. В то же время на других выходах должен быть признак того, что образ данному классу не принадлежит.

4) Обучение ИНС:

В процессе обучения сеть просматривает обучающую выборку. При обучении с учителем выборку делят на следующие части: обучающая (60%), валидационная (20%) и тестовая (20%). Обучающая выборка подается сети для обучения, а валидационная используется для расчета ошибки сети. При обучении сети предлагаются различные образцы образов с указанием того, к какому классу они относятся. Образец, как правило, представляется как вектор значений признаков. При этом совокупность всех признаков должна однозначно определять класс, к которому относится образец. В случае, если признаков недостаточно, сеть может соотнести один и тот же образец с несколькими классами, что неверно.

5) Тестирование качества обучения:

Тестирование качества обучения нейросети необходимо проводить на примерах, которые не участвовали в её обучении.

Наиболее подходящие существующие архитектуры ИНС для решения задачи распознавания поведенческих шаблонов на графиках временных рядов [5]:

1. Сеть Хопфилда:

Структура сети Хопфилда представляет собой однослойную рекуррентную сеть, замкнутую обратными связями [6]. Внутренняя организация сети Хопфилда, обрабатывая зашумленный и искаженный образ, подстраивает собственные параметры до тех пор, пока не приходит к некоторому стационарному состоянию (одному из эталонов) и, следовательно, позволяет решать задачи классификации, фильтрации и восстановления образов.

Преимущество - для обучения сети Хопфилда нужны только эталонные значения, на основе которых однократно подбирается матрица весовых коэффициентов.

Недостатки:

- относительно небольшой объем памяти;
- достижение устойчивого состояния не гарантирует правильный ответ сети.

2. Сеть Кохонена:

Сеть Кохонена состоит из двух слоев, причем работа сети организована таким образом, что для каждого входного образа только один из нейронов выходного слоя примет единичное значение [6]. Алгоритм обучения сети основывается на вычислении разностей между вектором синаптических связей и входным вектором и на пересчете весовых коэффициентов для нейрона, соответствующего наименьшей из

вычисленных разностей. Из этого следует, что сеть Кохонена позволяет разделить множество входных образов на несколько групп в соответствии с характерными признаками.

Преимущество - позволяет разделить множество входных образов на кластеры, обладающие некоторыми общими свойствами.

3. Многослойный персептрон:

Многослойный персептрон является одной из базовых и самых распространенных моделей нейронных сетей, позволяющих решать широкий спектр задач, одна из которых - задача распознавания образов. Многослойный персептрон относится к сетям, обучающимся на основе алгоритма обратного распространения ошибки, весовые коэффициенты подстраиваются итерационно.

Преимущество - может формировать на выходе произвольную многомерную функцию при соответствующем выборе количества слоев, диапазона изменения сигналов и параметров нейронов [1].

Недостатки:

- требуется обучающая выборка значительного объема, содержащая множество образцов каждого типа;

- для каждого из обучающих примеров в процессе обучения многократно происходит пересчет значений всех нейронов сети и связей между ними.

Выбор конкретной архитектуры ИНС для распознавания временных рядов должен быть основан на имеющемся наборе обучающих данных, требованиях к времени обучения, а также ограничений объема памяти необходимого для работы и обучения ИНС. Исходя из вышеизложенного, целесообразной является разработка такой архитектуры ИНС, которая удовлетворит требованиям задачи распознавания и использует преимущества существующих ИНС.

Список использованных источников:

1. Серебряная, Л. В. Методы и алгоритмы принятия решений: учебно-методическое пособие / Л. В. Серебряная, Ф. И. Третьяков. – Минск: БГУИР, 2016. – 64 с.
2. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume II-3/W3, 2013 CMRT13 - City Models, Roads and Traffic 2013, 12 – 13 November 2013, Antalya, Turkey, 13 - 16 с.
3. Dictionary of Philosophy of Mind [Электронный ресурс] / McCulloch, Warren Sturgis - May 2004. - Режим доступа: <https://sites.google.com/site/minddict/mcculloch-warren-sturgis>. - Дата доступа: 17.03.2018
4. Artificial Neural Network [Электронный ресурс] / Wikipedia. – Режим доступа: https://en.wikipedia.org/wiki/Artificial_neural_network. – Дата доступа: 18.03.2018.
5. Найман, Э. Малая энциклопедия трейдера / Э. Найман. – М.: Альпина Паблишер, 2003. – 378 с.
6. Хайкин, С. Нейронные сети: полный курс / С. Хайкин. – М.: Вильямс, 2006. – 104 с.

АНАЛИЗ ПОБОЧНЫХ ДАННЫХ СОСТОЯНИЯ СИСТЕМЫ И ДАННЫХ РЕЗУЛЬТИРУЮЩИХ ТОЧЕК СИСТЕМЫ

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

Мялик И.С.

Куликов С.С. – к.т.н., доцент

Уже сейчас большинство программных системы обладают теми или иными возможностями мониторинга собственной работоспособности. Одним из видов такого мониторинга является сохранение данных состояния системы или данных результата выполнения каких-либо ключевых функций, т.е. протоколирование определенных, чаще всего, скрытых от пользователя, сведений о системе. Как правило, протокол работы сохраняется в файлы на сервере или в базы данных с определенной моделью для будущей обработки. В данный момент не до конца решенным остается вопрос обработки большого объема протоколов работы системы в реальном времени.

Допустим, перед нами стоит задача обработки большого объема протоколов работы системы размером в несколько ТБ (например, системы показа рекламы).

Одна из возможных стратегий может быть описана следующим образом:

1. Простейшей эвристикой определяются «нужные пользователи», например, по числу заходов на сайт за неделю.
2. Теперь для каждого сайта считается его посещаемость «нужными пользователями» и всеми пользователями, определяется доля «правильных пользователей» от числа всех посетителей сайта.
3. Далее сортируются сайты по убыванию рассчитанного показателя, выбираются, например, 300 сайтов, начиная с начала, на которых находится целевая аудитория.

Вероятнее всего результатом решения этой задачи будут веб-сайты содержащие один и тот же тип информации. Можно предположить, что результатом будут веб-сайты по аренде и продаже жилья. Таким образом, определяется новый этап задачи, а именно: каким образом разместить рекламные баннеры. На каких сайтах следует разместить аренду домов в Минске, а на каких продажу квартир в Ингушетии. Для выполнения этой задачи, необходимо работать с инструментами Big Data и иметь доступ к данным для их анализа.

Инструмент Big Data

1. Необходимо выбрать место для хранения протоколов работы системы, им может быть файловая система или база данных.
2. Далее следует эти данные обрабатывать, сортировать по разным параметрам, находить веб-сайты в верхних 100\300\1000 строчках рейтинга, определять долю целевых пользователей в общем трафике. Далее будет использована парадигма MapReduce и искусственные алгоритмы вида:
 - фильтрации;
 - оптимизации: соединения «map-join» с использованием распределенного кэша;
 - оптимизации: применение «combiner», как одно из требований реализации «reduce»;
 - соединения «reduce-join», к примеру, для поиска рейтинговых сайтов;
3. Хотя данное заключение верно не только для Big Data, но также и для большого объема данных становится крайне критично, для этого необходимо построить некий процесс предобработки данных, с помощью машинного обучения.

Машинное обучение

Первый этап работы – информационная очистка и предобработка данных:

- работа с аномальными значениями;
- удалить без информативные данные, такие как: технические посещения и т.д.;
- нормализовать URL-адреса;
- работа с пропущенными, при трассировке, значениями.

Второй этап – превращение данных, в понятные для модели векторы.

На этом этапе существует множество подходов, но нет однозначной методики их применения. Было выделено несколько общепринятых подходов для решения этой задачи:

- проанализировать домен и сгруппировать веб-сайты по нему;
- скачать описательную часть страницы и проанализировать ее;
- скачать весь код страницы и постараться выявить тематику страницы;
- связать посещения пользователем сайтов в цепочку и определить логику в последовательности переходов.

Последний этап – применение машинного обучения.

Для применения машинного обучения нету определенного подхода, это наиболее творческий вариант, требующий применения здравого смысла и умения итерационно улучшать выбранное решение.

Список использованных источников:

Real-WorldMachineLearning. Henrik Brink, Joseph W. Richards.

КОНТРОЛЬ КАЧЕСТВА РАЗРАБОТКИ ПРОГРАММНЫХ СРЕДСТВ

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

Новицкий А.А.

Бахтизин В.В. – профессор каф. ПОИТ, к.т.н., доцент

Работы по контролю качества разработки востребованы в течение всего жизненного цикла программных средств (ПС). Выбор оптимальных для проекта работ по контролю качества (КК) отвечает целям повышения надежности итогового продукта, уменьшения издержек, повышения рентабельности процесса разработки и является актуальной темой. Повсеместное внедрение программных продуктов и рост сложности используемых технологий и ПС становится причиной роста цены ошибки в продукте. Поэтому не теряет актуальности вопрос дальнейшей оптимизации процессов контроля качества (КК) разработки программных средств.

В данном докладе рассматривается классификация тестирования в зависимости от необходимости выполнения программного кода. Это позволяет четко выделить как минимум одну пригодную для анализа группу работ. По данному критерию методы тестирования делятся на статические, не требующие запуска программного кода, и динамические, непосредственно взаимодействующие с функционирующим приложением или его моделью.

Статические виды тестирования включают работы по верификации требований и анализу программного кода, вследствие чего могут обеспечивать наиболее раннее выявление различных проблем, потому обязательны к включению в план разработки [1].

Динамические методы тестирования в основном обеспечивают обнаружение дефектов функциональности в ходе проверки тестовых сценариев или же отвечают задачам проверки устойчивости системы к превышению пределов нормального функционирования, закрытости от влияния извне. Подходы к динамическому тестированию наиболее разнообразны, что создает дополнительные трудности в выборе работ для полного избыточного покрытия функций разрабатываемого ПС.

Независимо от наличия автоматизации процесса динамического тестирования общим вариантом критерия оценки покрытия является покрытие, основанное на спецификации или требованиях. Главное требование состоит в покрытии некоторого утвержденного минимума – набора требований. Разница в механизме доступа обуславливает специфику проведения тестирования, предпочтительного для каждого из методов тестирования. Таким образом, рассматривать следует критерии эффективности ручного и

автоматизированного тестирования.

Механизм планирования и оценки покрытия разрабатываемого ПС тестированием может быть выбран в соответствии с потребностями системы и имеющейся документации. С учетом этого выбор применяемых методов и техник тестирования строго не регламентируется. План, методы и техники тестирования могут определяться командой на этапе планирования, исходя из субъективных мнений, предпочтений, доступных ресурсов [3]. Таким образом, задача формирования полного неизбыточного плана работ по КК разработки, пригодного для внедрения в проекты, процессы которых выстроены с использованием популярных моделей и методологий разработки на базе некоторых систем планирования процессов, может быть рассмотрена с аналитической точки зрения.

Отдельно при разработке в каждый момент времени стоят следующие задачи:

- оценка количества ошибок, оставшихся в проекте, и их критичность;
- оценка количества ошибок, оставшихся в выбранном компоненте, и их критичность;
- оценка времени, за которое текущая версия станет стабильной, т.е. количество ошибок и их критичность в данной версии будут меньше заданного порога.

Для решения данных задач можно использовать два подхода – изучение исходного кода программного продукта и применение моделей оценки надежности (МОН) ПС. Использование МОН позволяет посредством построения вероятностной модели случайных процессов и использования различных статистических методов получить оценки различных метрик (ожидаемое число ошибок, вероятность ошибки).

В докладе рассматривается возможность использования техник вероятностного (статистического) тестирования с целью создания репрезентативного сценария тестирования. В качестве некоторого критерия оценки эффективности использования данных техник тестирования можно принять достижение надежностью некоторого приемлемого для конкретного проекта значения [4].

На основе полученного сценария и располагая некоторыми данными для анализа с места планируемого внедрения плана работ, предлагается более предметно рассчитывать рентабельность автоматизации работ в процессе тестирования с учётом планируемой к использованию модели/методологии разработки с точки зрения затрат финансов и времени [2].

Список использованных источников:

1. Royce, W. W. Managing the Development of Large Software Systems / Winston W. Royce // Article / Proceedings of IEEE WESCON 26 : Article / The Institute of Electrical and Electronics Engineers, Inc. – 1970.
2. Extreme Programming Explained: Embrace Change, 2nd Edition: Book / Addison-Wesley. – NY, 2004.
3. Липаев, В. В. Человеческие факторы в программной инженерии: рекомендации и требования к профессиональной квалификации специалистов : Учебник / В. В. Липаев. – М.: Синтег, 2009. - 328 с.
4. Sayre, K. Improved Techniques for Software Testing Based on Markov Chain Usage Models: Dissertation / K. Sayre. – Knoxville: The University of Tennessee, 1999. –128с.

ОПТИМИЗАЦИЯ АВТОМАТИЗИРОВАННОГО ОБНАРУЖЕНИЯ УЯЗВИМОСТЕЙ К SQL-ИНЪЕКЦИЯМ В WEB-ПРИЛОЖЕНИЯХ

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

Оношко Д.Е.

Бахтизин В.В. – к.т.н., доцент

Широкое распространение web-приложений как способа реализации программных средств, в сочетании с их доступностью неограниченному кругу пользователей предъявляет повышенные требования к качеству web-приложений. По данным OWASP [1], по состоянию на 2017 г. среди уязвимостей web-приложений наиболее распространёнными остаются SQL-инъекции. Обнаружение таких уязвимостей вручную является трудоёмким процессом, что создаёт необходимость в разработке методов их автоматизированного обнаружения.

Предложенная в [2] модель обнаружения уязвимостей предполагает выделение в web-приложении процедур, параметры каждой из которых в дальнейшем подвергаются оценке. Однако, несмотря на относительную простоту модели, ввиду значительного объёма исходных кодов, присущего современным web-приложениям, при реализации метода обнаружения уязвимостей на основе этой модели возникает необходимость оптимизации процесса назначения оценок.

Очевидным способом оптимизации является исключение из рассмотрения тех частей web-приложения, которые не оказывают влияния на результаты обнаружения уязвимостей. В основу такой оптимизации предлагается положить разделение модели на 2 уровня: внутриволеводный и межволеводный.

Межволеводный уровень модели предлагается представить в виде графа зависимостей, отражающего характер взаимосвязей между отдельными процедурами web-приложения. Упрощённый пример такого графа представлен на рисунке 1.

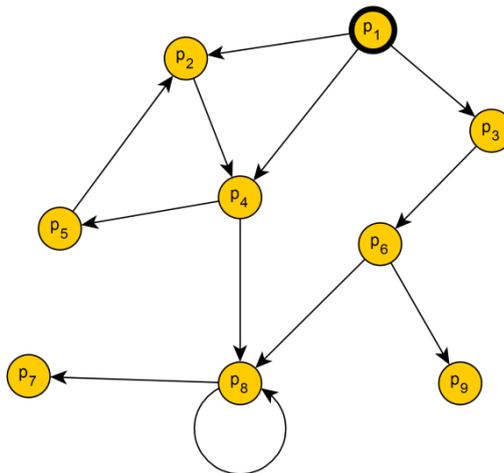


Рис. 1 – Упрощённый пример графа зависимостей

Такое представление структуры web-приложения позволяет очевидным образом:

- определять порядок анализа отдельных процедур web-приложения;
- выявлять взаимную рекурсию с целью исключить заикливание программного средства, реализующего метод обнаружения уязвимостей, — анализатора;
- исключать из рассмотрения недостижимый код.

Начинать анализ следует с процедур, не имеющих исходящих дуг в графе зависимостей. В примере это процедуры p_7 и p_9 . Точка входа web-приложения p_1 (на рисунке выделена) подвергается анализу в последнюю очередь, а полученные её параметрами оценки позволяют дать ответ на вопрос о фактическом наличии или отсутствии в web-приложении уязвимостей к SQL-инъекциям.

Выделение межпроцедурного уровня позволяет оптимизировать процесс обнаружения уязвимостей за счёт отказа от включения во внутривпроцедурный уровень модели web-приложения информации о ряде процедур, а также выбора оптимального порядка накопления такой информации. При наличии соответствующей поддержки со стороны анализатора также представляется возможным распараллеливание анализа процедур, не имеющих взаимных зависимостей.

Список использованных источников:

1. OWASPTop 10 2017. The Ten Most Critical Web Application Security Risks. [Электронный ресурс.] — Режим доступа: https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf. — Дата доступа: 27.11.2017.
2. Бахтизин В. В. Модель обнаружения уязвимостей в web-приложениях / В. В. Бахтизин, Д. Е. Оношко // Докл. БГУИР. — 2016. — №1 (95). — С. 5–11.

ВИЗУАЛИЗАЦИЯ БОЛЬШИХ ДАННЫХ (BIGDATA)

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

Панкратьев А.С., Пилинко Н.А., Голубев К.А.

Лапицкая Н.В. – к.т.н, доцент

В настоящее время объем необходимых для обработки данных стремительно растут, однако общепотребительные методы их представления и обработки требуют много ресурсов и времени, а также недостаточно просты для понимания. Одним из решений этой проблемы может стать представление данных в графическом виде на пространственно-временной поверхности.

Цель: представить массивные структуры данных на пространственно-временной поверхности.

Рассмотрим использование данного метода на примере представлении комплексной статистики по пожарам на карте.

Входными данными будем считать:

- База данных МЧС РБ по пожарам за последние 11 лет;
- GET-параметры с выбранными пользователем фильтрами;

В пространственной плоскости S данные представляются согласно административно-территориальному делению.

Разобьем плоскость на m частей, где S_i - представление на карте частотной и количественной информации за период.

Каждый S_i является функцией. $S_i = f(x, y, z, c)$, где x, y, z - географические координаты, c - полученный

цвет, является функцией $C(Si1, Si2, \dots, Sin)$

В свою очередь Si -ую часть можно разбить на N частей, где N - количество ситуаций.

Каждый Sij -ый элемент хранит количество повторений j -ой ситуации в i -ом регионе. Найдем максимальное число повторений каждой ситуации

$$Mj = \text{Max}(Mij);$$

Каждой ситуации сопоставим насыщенный цвет λ_j - данный цвет соответствует 100%-ому объему j -ой ситуации.

Для каждой j -ой ситуации i -ого региона найдем его цвет χ_{ij} путем осветления базового цвета λ_j на $\eta\%$, где

$$\eta = 100 - (Sij / Mj) * 100$$

После чего найдем цвет каждого i -ого региона путем смешивания χ_{ij} -ых цветов методом суммы их RGB-представления

$$R_i = \frac{1}{N} \sum_{j=1}^N r_{ij}$$

$$G_i = \frac{1}{N} \sum_{j=1}^N g_{ij}$$

$$B_i = \frac{1}{N} \sum_{j=1}^N b_{ij}$$

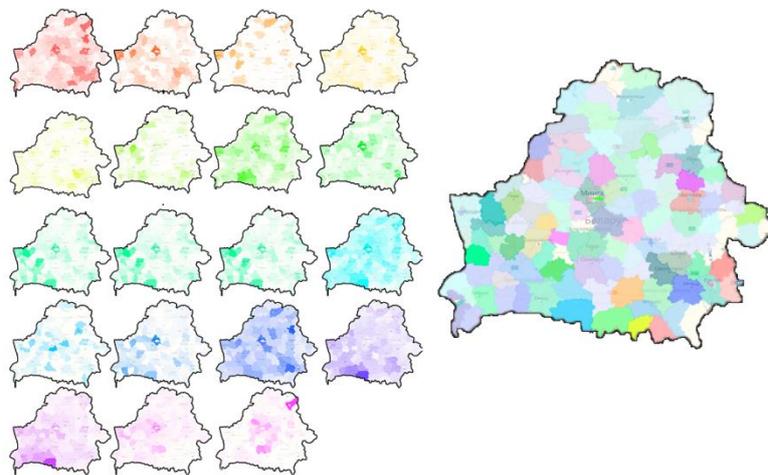


Рис 1. Пример смешивания цветов

После получения всех входных параметров функции Si , она наносится на пространственно-временную поверхность.

Разработанное решение позволяет эффективно визуализировать многомерную информацию, обладает интуитивно понятным интерфейсом. Разработанный код легко модифицируется для использования в самых разных сферах. Применение технологии смешивания цвета усиливает восприятие и анализ информации

Список использованных источников:

1. Noab Iliinsky Designing Data Visualizations 20 с.
2. Chun-houh Chen, Wolfgang Härdle, Antony Unwin "Handbook of Data Visualization" 10-25 с
3. Geoff McGhee Tooling Up for Digital Humanities Seminar May 6, 2011

РАСПОЗНАВАНИЕ ВИДОВ ДЕЯТЕЛЬНОСТИ ЧЕЛОВЕКА С ПОМОЩЬЮ СМАРТФОНА

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

Парамонова А.К.

Серебряная Л.В. – к.т.н., доцент

Когда мы смотрим на человека, мы сразу понимаем, чем он занят в данный момент. Даже если мы видим человека впервые, наш мозг мгновенно сравнивает наблюдаемые действия с тысячами увиденных ранее и выдает нужное

совпадение. Подобные процессы, но в меньшем масштабе, могут происходить и в компьютере. Используя алгоритмы машинного обучения, можно научить смартфон распознавать специфическую деятельность человека и совершенствовать это умение по мере поступления новых данных.

Проблема распознавания видов деятельности человека привлекает научное компьютерное сообщество с 1980-х годов благодаря возможности обеспечения персонализированной поддержки приложений и её связи со многими областями, такими как медицина, взаимодействие с человеком и компьютером и социология. Также задача охватывает такие проблемы, как распознавание намерений, оценка поведения, оценка местоположения и предоставление услуг на основе местоположения. В данной работе будет рассмотрен процесс подготовки набора данных к анализу, обучения нейронной сети и тестирования качества распознавания видов деятельности человека.

В качестве объекта исследования был выбран набор данных, собранный исследователями в Фордхемском университете в 2012 году с помощью различных смартфонов на базе Android. Данные были собраны у 36 добровольцев в возрасте от 19 до 48 лет, которые выполняли 6 основных видов деятельности: 3 неподвижных вида (стоять, сидеть, лежать) и 3 подвижных (идти, подниматься по лестнице, спускаться по лестнице). Набор данных содержит более миллиона записей, каждая из которых состоит из 6 атрибутов: идентификатор волонтера, вид деятельности, момент времени записи измерений и показания акселерометра для трех осей координат: X, Y, Z. Показания фиксировались каждые 50 миллисекунд, таким образом каждую секунду создавалось 20 записей.

Срез собранных данных для волонтера №12, который выполнял пробежку:

12, Jogging, 17593991982000, 0.69, -7.31, -1.18;

12, Jogging, 17594031625000, -9.43, 3.95, 10.99;

12, Jogging, 17594111459000, 5.43, 5.43, -0.15;

12, Jogging, 17594151589000, -6.21, 14.33, -5.63;

12, Jogging, 17594195626000, -3.21, 16.04, -6.51;

Для решения поставленной задачи в качестве объекта обучения была выбрана сверточная нейронная сеть. Выбор обоснован тем, что классической задачей для такого вида нейронных сетей является распознавание образов на изображениях, которая является близкой к выбранной задаче распознавания видов деятельности человека по виду входных и выходных параметров. Входными параметрами являются многомерные структуры данных. Нейронная сеть производит свертку данных и анализирует их как единое целое. Выходным параметром является N-пространственный вектор, где N – количество классов, которые распознаёт нейронная сеть.

В задаче распознавания видов деятельности человека входные параметры представляют собой структуру данных, содержащую 3 канала для 3 осей координат. Каждый канал является вектором, который содержит 90 элементов, что эквивалентно 4,5 секундам наблюдения. Таким образом будет производиться анализ данных не в конкретный момент фиксирования показаний акселерометра, а в динамике за выбранный промежуток. Выходным параметром является вектор с 6 элементами для 6 видов деятельности, в котором содержатся данные о том, с какой вероятностью человек занимался каждым видом деятельности.

Для представления данных в удобном для обучения нейронной сети виде данные были разбиты на срезы по 90 записей с шагом в 10 и отсортированы по волонтерам и видам деятельности. Полученный набор данных разбит в отношении 70/30. 70% набора – обучающая выборка, 30% – выборка для тестирования.

Сверточные нейронные сети состоят из различных слоёв: сверточные, подвыборки и слоя обычной нейронной сети – полносвязного персептрона. Ниже приведено краткое описание каждого слоя.

Слой свёртки – основной блок сверточной нейронной сети. Представляет собой матрицу коэффициентов для каждого входного канала. Разбивает входной канал на фрагменты и подает на выход результаты матричного произведения для каждого фрагмента.

Слой подвыборки призван уменьшить объем обрабатываемых данных. Он разбивает входные матрицы на фрагменты и преобразует каждый фрагмент с помощью нелинейного преобразования. Например, с помощью функции максимума слой выбирает из каждого фрагмента максимальное значение коэффициентов и отправляет на выход.

Полносвязный персептрон представляет собой классическую нейронную сеть, которая также может содержать несколько слоёв. Сеть выполняет функцию классификации и определяет вероятность того, что исследуемый объект принадлежит к какому-либо классу.

Обычно сеть состоит из большого количества слоёв. Первым слоем всегда является сверточный слой, затем несколько чередующихся между собой слоёв подвыборки и свертки, в конце находится полносвязный персептрон. Также на архитектуру оказывают влияние такие требования, как скорость работы и вероятность правильной классификации объекта. Для данной задачи установлены следующие требования: выдача результата в течение 1 секунды с вероятностью правильной классификации 70%. Для решения задачи распознавания видов деятельности была спроектирована сверточная нейронная сеть, содержащая 2 сверточных слоя и 1 слой подвыборки. Данной решение позволяет получить приемлемое соотношение скорости работы и качества распознавания. На рисунке 1 представлена архитектура сверточной нейронной сети для задачи распознавания видов деятельности человека с учетом указанных требований.

Обучение нейронной сети произведено с помощью метода обратного распространения ошибки, который является классическим алгоритмом обучения нейронных сетей. Для этого на вход нейронной сети подавались данные из обучающей выборки, полученный результат сравнивался с ожидаемым результатом, затем веса элементов нейронной сети подвергались корректировке. Процесс обучения продолжался до тех пор, пока не была использована вся обучающая выборка.

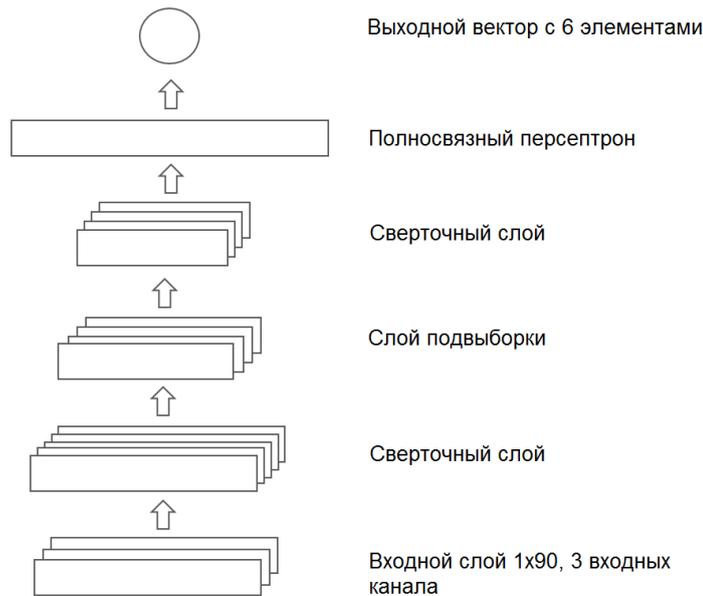


Рис. 1 – Архитектура спроектированной нейронной сети

Процесс тестирования качества нейронной сети заключался в получении результатов классификации для данных, которые не участвовали в процессе обучения. Был произведен подсчет количества случаев, когда результат оказывался правильным. Для этого была использована тестовая выборка. Наилучший результат, который был получен в результате тестирования, - более 80% случаев правильного определения вида деятельности человека.

Таким образом, была решена задача распознавания видов деятельности человека с помощью смартфона с использованием сверточной нейронной сети.

Список использованных источников:

1. Гудфеллоу, Я. Глубокое обучение (серия адаптивных вычислений и машинного обучения) – TheMITPress, 2016 – 534-540 с.
2. Рашка, С. Машинное обучение на языке Python – PacktPublishing, 2015 – 354 с.
3. Баят, А. Исследование по распознаванию человеческой деятельности с использованием данных акселерометра от смартфонов // 11-я Международная конференция по мобильным системам и повсеместным вычислениям– Онтарио, Канада, 2014. – 2-8 с.
4. Лаборатория WISDM. [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://www.cis.fordham.edu/wisdm/dataset.php> Дата доступа: 20.03.2018.

ПРОГРАММНОЕ СРЕДСТВО ДЛЯ ОРГАНИЗАЦИИ ЗАЩИЩЕННОГО ХРАНЕНИЯ ДАННЫХ НА ВНЕШНИХ НОСИТЕЛЯХ

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

Пашковский С.М.

Прохорчик Р.В. – ст. преподаватель каф. ПОИТ, м.т.н.

Сегодня стоит острая проблема хранения данных в электронном виде. Основные тому причины человеческий фактор и массовое подключение электронных устройств к глобальной сети Интернет. Под человеческим фактором понимается: беспечность людей, незнание и/или не соблюдение элементарных правил безопасности, возможность обмана человека. Данные факторы сложно решить какими-либо техническими средствами. Проблемы, связанные с массовым подключением к Интернету, можно решить при помощи технических и организационных средств. Одним из таких средств является перенос данных на внешнее хранилище данных.

Перенос информации с компьютера на внешний, съёмный носитель данных является неплохим методом повышения защищенности данных. Обычно человек пользуется секретными файлами (с документами, с личной информацией, с научными разработками и т.д.) редко, поэтому их можно перенести на съёмный носитель и подключать его только в случаях необходимости. Это существенно сокращает время, в течении которого злоумышленник может украсть информацию.

Однако, существуют потенциальные уязвимости, связанные с хранением данных на съёмных носителях:

- Хищение или утеря носителя;
- Попытка чтения данных с носителя вредоносной программой в то время, когда носитель подключено

к компьютеру.

Для предотвращения описанных уязвимостей предлагаются следующие методы, повышающие сохранность данных.

Во-первых, шифрование данных на этом носителе. Это позволит предотвратить утечку информации вследствие хищения или утери носителя данных.

Во-вторых, контроль программ (процессов), которые имеют доступ на чтение, запись информации на защищенный носитель. Т.к. на пользовательском компьютере могут присутствовать вредоносные программы, которые будут пытаться считать данные с подключаемых устройств, необходимо требовать от пользователя подтверждение, о том, что данная программа может работать с носителем.

Для реализации данной концепции защиты данных необходимо разработать соответствующее программное средство. Оно будет состоять из двух отдельных компонентов, способных взаимодействовать между собой: драйвера операционной системы и приложения для ввода пользовательских настроек.

Основным компонентом системы является драйвер. Он осуществляет всю необходимую работу по шифрованию, дешифрованию информации, контролю доступа программ к носителю. Выбор пал на драйвер, а не на простое пользовательское приложение, т.к. получение доступа к драйверу и его модификация сложнее для вредоносных программ [1]. Наличие драйвера позволяет любой программе записывать и читать данные из любой программы, что очень удобно для пользователя. При создании простого приложения пользователь мог бы работать с носителем только из него. Однако, доступность к носителю из любой (потенциально вредоносной) программы является угрозой безопасности. Поэтому необходимо создать механизм контроля доступа программ к носителю.

Контроль доступности носителя будет осуществляться путём хранения разрешенных приложений. При попытке программы (процесса) прочесть или записать данные на защищенный носитель система будет запрашивать разрешение пользователя на эти действия. Драйвер будет хранить разрешенные программы только в течение его работы (следовательно, только в оперативной памяти). После перезагрузки операционной системы эта информация будет стираться, и пользователь снова должен будет дать доступ необходимым программам. Разрешённые программы не будут сохраняться на диск, т.к. у злоумышленника появится возможность изменить эти настройки на диске.

Шифрование данных будет производиться на основе стандарта СТБ 34.101.31-2011 [2]. Выбран симметричный блочный алгоритм шифрования, т.к. он обеспечивает хорошую скорость, а в данном случае скорость имеет большое значение, т.к. может производиться чтение/запись больших объемов информации.

Ключ для каждого устройства будет свой. Он будет храниться в файле в хешированном виде. В качестве алгоритма хеширования должен использоваться один из надёжных на данный момент алгоритмов (например, SHA-256).

Список используемых источников:

1. Руссинович М., Соломон Д. *Внутреннее устройство Microsoft Windows. 6-е изд.* – СПб.: Питер, 2013 – 800с.
2. СТБ 34.101.31-2011 "Информационные технологии. Защита информации. Криптографические алгоритмы шифрования и контроля целостности".
3. *WindowsDriverKitDocumentation [Электронный ресурс]. – Режим доступа: <https://docs.microsoft.com/windows-hardware/drivers/index> - Дата доступа: 22.03.2018.*

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ИЗУЧЕНИЯ ИНОСТРАННЫХ СЛОВ НА ОСНОВЕ ИНТЕРВАЛЬНОГО МЕТОДА

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

Подлужный П.Н.

Хмелева А.В. – канд. техн. наук, доцент

В наше время, знать иностранный язык жизненно необходимо. Для некоторых – это возможность расширения кругозора, а для других – необходимое условие для работы. Но изучение языка требует достаточно много времени и усилий, которые можно сэкономить, если использовать методы и техники, которые эффективнее всего используют эти ресурсы. Возможно, не самым лучшим, но одним из лучших методов для изучения новой информации является интервальный метод.

Интервальные повторения — техника удержания в памяти, заключающаяся в повторении запомненного учебного материала по определённым, постоянно возрастающим интервалам [1]. Идея, что интервальные повторения можно использовать для улучшения процесса обучения, впервые была предложена в книге «Психология обучения», написанной профессором Алеком Мейсом в 1932 году. В 1939-м Spitzer протестировал эффект методики на студентах в Айове [2]. Spitzer исследовал метод на более чем 3600 студентах и доказал его эффективность. Это достигается за счёт того, что повторение слова происходит в тот момент, когда оно вот-вот будет забыто.

Наиболее известным и распространённым методом, реализующим метод интервального повторения для запоминания различной информации, являются карточки. Т.е. на одной стороне находится вопрос (в случае с изучением слов — слово на иностранном языке), а на другой стороне — ответ на него (в случае с

изучением слов — перевод слова). Так можно изучать и запоминать не только иностранные слова, но всё, что вам необходимо. Когда вы начинаете забывать слово (имеются различные исследования, которые устанавливают периоды времени, через которое это происходит) — программа предоставляет вам карточку с ним, вы отвечаете и, в случае правильного ответа карточка с данным словом будет предложена вам через больший, по сравнению с предыдущим, период времени. Так как в нашей жизни присутствуют всяческие стрессы и другие помехи работы нашего мозга, то некоторые карточки будут забываться до того, как программа предложит вам их повторить. В этом случае, время следующего повторения будет меньше — предыдущий период.

Существуют хорошие программы, которые используют данный метод. Например, “Anki” и “Memrise”. Но имеется один минус в их работе. Если вы пропустили несколько дней и у вас скопилось много карточек, то вам, возможно, не захочется их изучать, и вы просто закончите использовать данную программу и на этом ваше изучение остановится.

Для его преодоления, в разрабатываемом приложении, будет введён рейтинг слова, который будет использовать интервальный метод. Рейтинг будет напрямую связан с периодами интервального метода. Пользователь может регулярно повторять слова и тогда, рейтинг будет возрастать. При своевременном повторении слово будет оставаться в памяти, а его рейтинг возрастать, увеличивая периоды повторения. Если пользователь не повторил карточку, то её рейтинг будет уменьшен и с ним, уменьшится период, через который пользователю будет необходимо его повторить.

Пользователю не будет знать, нужно ли именно сейчас повторить данное слово. Он будет пользоваться программой тогда, когда ему это удобно. Это снижает эффективность интервального метода, но при этом, мотивация пользователя не будет снижена. При повторении слова, ему лишь будет показан рейтинг. Если же пользователь будет повторять слово раньше, чем наступит период следующего повторения, то рейтинг будет изменяться на меньшее значение. При слишком частом повторении слова, ему будет выведено сообщение о том, что слово усвоено им достаточно на данном этапе, и он может пока не концентрироваться на нём.

Для поддержания мотивации в приложение необходимо ввести разнообразные варианты повторения слова:

- Тесты;
- Рукописный ввод слова;
- Экзамен, главной особенностью которого будет ограниченное время;

Для английского языка будет присутствовать режим изучения неправильных глаголов.

Все эти варианты будут так же изменять рейтинг слов, но по своим правилам.

Данное приложение, основываясь на интервальном методе, предоставляет пользователю возможность нерегулярного изучения слов, что позволяет не терять мотивации и продолжать изучать слова в удобном ему режиме.

Список используемых источников:

1. «Human Memory: Theory and Practice», Alan D. Baddeley, 1997
2. Spitzer, H. F. (1939). Studies in retention. Journal of Educational Psychology, 30, 641—657

АЛГОРИТМ ПРИМЕНЕНИЯ СЕМАНТИЧЕСКОЙ СЕТИ ДЛЯ ПОИСКА ОТВЕТА НА ВОПРОС

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

Потараев В.В.

Серебряная Л.В. – к.т.н., доцент

Семантический анализ является довольно эффективным методом обработки информации. Семантическая сеть позволяет осуществлять различные способы обработки данных. Рассмотрим представление текстовых данных в виде семантической сети и алгоритм получения ответа на вопрос, основанный на использовании данной сети.

В современных автоматизированных системах хранятся и обрабатываются значительные объёмы информации. Поэтому актуальным является повышение эффективности автоматизированной обработки данных, как по скорости, так и по точности обработки. Одним из инструментов улучшения эффективности обработки данных является учёт их смысловой структуры, то есть семантический анализ.

Многие информационные системы предназначены для поиска ответа на запрос [1]. Рассмотрим задачу ответа на вопрос, сформулированный на естественном языке. В русском языке выделяют пять основных видов вопросов: закрытые, открытые, риторические, переломные, вопросы для обдумывания [2].

Целью данной работы является разработка алгоритма поиска ответа на открытый вопрос.

Открытый вопрос – это вопрос, требующий разъяснения [2], например: «что?», «кто?», «где?», «как?», «сколько?», «почему?». Алгоритм, позволяющий отвечать на различные типы вопросов, может быть использован для упрощения работы с информационной системой. Если система способна найти ответ на различные типы вопросов, то у пользователя есть больше возможностей сформулировать вопрос в удобной для него форме.

Одним из инструментов для осуществления семантического анализа данных являются базы знаний.

База знаний – это компонент экспертной системы, предназначенный для хранения долгосрочных данных, описывающих определенную предметную область, и правил, описывающих целесообразные преобразования данных этой области [3].

Модель базы знаний, основанная на семантической сети, наиболее соответствует современным представлениям об организации долговременной памяти человека [4]. Если учесть, что человеческая память представляет информацию довольно эффективно, то и её искусственная реализация в виде семантической сети также, вероятно, будет эффективной.

Количество типов отношений в семантической сети определяется её создателем, исходя из конкретных целей. В реальном мире их число стремится к бесконечности. Каждое отношение является, по сути, предикатом (утверждением), простым или составным [5].

В качестве узлов семантической сети могут выступать понятия, которые являются подлежащими и дополнениями в предложениях текста, а в качестве узлов – сказуемые. Построенная таким образом семантическая сеть может быть использована при разделении текстов на категории (классы) [6].

Рассмотрим модель, имеющую большее количество типов отношений. Предположим, необходимо найти ответ на вопрос определения места «где?». Используемая при этом семантическая сеть может содержать следующие типы отношений:

- 1) Слово – форма слова.
- 2) Подлежащее - сказуемое.
- 3) Синоним.
- 4) Место.

Этот список типов отношений можно расширить, но для решения поставленной задачи он достаточен в данном виде.

На основе перечисленных типов отношений можно создать семантическую сеть. Процесс её создания можно автоматизировать, используя словари (синонимов, существительных и т.п.) и учитывая начальную форму слова [7].

Для предложения «На улице идёт сильный дождь» получится сеть, представленная на рисунке 1.

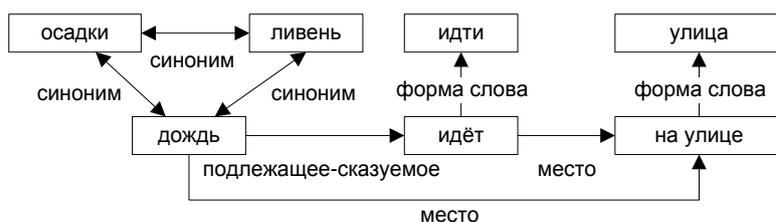


Рис. 1 – Пример семантической сети

При построении сети соответствующие предлоги добавляются в узел, означающий форму существительного. Это позволит строить на основе данной сети предложения, содержащие слова в корректной форме.

В данной сети нет типа отношения, в котором бы участвовало слово «сильный». Поэтому это слово не добавлено в сеть. Для простоты определения начальной формы слов в её качестве можно использовать «обрезанную» форму, полученную с использованием алгоритма стемминга. При определении каждого из типов связей могут быть использованы различные алгоритмы.

Итак, рассмотрим процесс нахождения ответа на вопрос с использованием данной сети. Поставленный вопрос - «Где идут осадки?».

«Где» - вопрос определения места. Покажем, что для построения предложения-ответа на вопрос данного типа достаточно найти в сети фрагмент, связывающий слова вопроса и место, связанное со словами найденного фрагмента.

Фрагмент сети, связывающий подлежащее вопроса «осадки» и сказуемое вопроса «идут» (точнее, его начальную форму «идти»), – «осадки», «дождь», «идёт», «идти». Здесь повторяются различные формы одного слова, а также синонимы. При построении ответа необходимо отфильтровать подобные повторения. Так, слово фрагмента, связанное с другими словами фрагмента лишь связью «синоним», лучше пропустить. Среди форм одного слова лучше выбрать форму, отличную от начальной. Данные правила можно менять в зависимости от поставленной задачи.

В итоге в ответ будут добавлены слова, согласованные между собой – «дождь идёт». Место, связанное с наибольшим числом слов найденного фрагмента – «на улице». Таким образом, в ответ на вопрос «где идут осадки» можно сформировать предложение: «дождь идёт на улице».

Для ответа на вопрос необходимо заранее построить семантическую сеть с нужными типами связей. Обобщив рассмотренный способ получения ответа на открытый вопрос, получим следующий алгоритм:

- 1) Определить тип вопроса.
- 2) Найти фрагмент сети, связывающий подлежащее и сказуемое вопроса. Для этого может быть использован алгоритм поиска в ширину. Если в вопросе нет подлежащего (либо сказуемого), то в качестве фрагмента сети можно взять некоторую окрестность сказуемого (либо подлежащего).

3) Убрать в найденном фрагменте сети синонимы и повторы разных форм одного слова.

4) В зависимости от типа вопроса, выбрать дополнительные слова, связанные со словами найденного фрагмента типом связи, соответствующим типу вопроса.

Ситуация, когда поиск фрагмента сети, связывающего подлежащее и сказуемое, занимает слишком много времени, может быть обработана различным образом. Например, работа алгоритма может быть прекращена без возврата ответа, либо при построении ответа может быть использована некоторая совокупность слов, связанных в сети со словами запроса. Аналогично может быть обработан и случай, когда фрагмент сети не найден.

Итак, получение ответа на вопрос при помощи семантической сети является вполне автоматизируемой задачей. Результат во многом зависит от способа построения сети и точности выделения смысловых связей между словами. Хранение различных форм слова в семантической сети позволяет строить корректные предложения на естественном языке. Предложенный алгоритм может быть использован в информационных системах для получения ответа на разные типы вопросов, представленные в разной форме. Для получения корректного ответа в большем числе ситуаций данный алгоритм может быть модифицирован.

Список использованных источников:

1. Информационные системы. [Электронный ресурс]. – Электронные данные. – Режим доступа: http://www.umpro.ru/index.php?page_id=17&art_id_1=261&group_id_4=72. Дата доступа: 19.03.2018.

2. Сурин, Н. Риторика: Учебное пособие / Н. Сурин – М.: МГИУ, 2007. – 246 с.

3. Базы знаний экспертных систем. [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://daxnow.narod.ru/index/0-18>. Дата доступа: 19.03.2018.

4. 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/record/01007574162>.

5. 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.

6. Потараев, В.В. Метод классификации текстовой информации на основе семантической сети / В.В. Потараев // Апробация. – 2016. – №1. – С. 56-58.

7. Потараев, В.В. Алгоритм построения семантической сети и её применение / В.В. Потараев // Информационные технологии и системы 2017 (ИТС 2017): материалы международной научной конференции. – Минск: БГУИР, 2017. – с. 144-145.

ОЦЕНКА ЭФФЕКТИВНОСТИ ВНЕДРЕНИЯ АВТОМАТИЗАЦИИ ТЕСТИРОВАНИЯ

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

Радюш Н.Г.

Серебряная Л.В. – к.т.н., доцент

Тестирование программного обеспечения является одним из основных методов управления качеством. Наиболее распространённым является ручное тестирование, однако в последнее время всё большую популярность набирает автоматизированное тестирование. Оно обладает следующими преимуществами: позволяет уменьшить количество человеческих ресурсов занятых на проекте, а также сводит к минимуму негативное влияние человеческого фактора на качество программного продукта. Тем не менее автоматизация тестирования не всегда может быть экономически эффективной, т.к. использование этого подхода без должной оценки и расчёта эффективности ведет к тому, что автоматизация становится нерентабельной и нецелесообразной.

Целью данной работы является исследование способов оценки экономической эффективности внедрения автоматизированного тестирования.

Для того, чтобы оценить экономическую эффективность внедрения автоматизации тестирования предлагается использовать метод расчёта периода возврата инвестиций (ROI), который является показателем рентабельности вложений и вычисляется как отношение дохода к затратам [1]. Инвестиции в разработку решения по автоматизации начинают себя окупать, как только это значение в процентном соотношении превосходит 100%.

При оценке эффективности автоматизации под экономической выгодой чаще всего понимается снижение трудозатрат функциональных тестировщиков, путем сокращения количества человек, занятых на проекте. Другими факторами, влияющими на эффективность автоматизации, являются:

- анализ и поддержка автоматизированных скриптов;
- стоимость подготовки тестовых наборов [2].

Для получения данных, которые необходимы для расчёта эффективности применения автоматизации, выбирается интервал проведения тестирования, который чаще всего составляет месяц, реже – неделя или итерация в жизненном цикле проекта. Далее для выбранного интервала рассчитываются затраты на тестирование вручную и тестирование с помощью автоматизированных тестов. Данный расчёт производится в человеко-часах.

При расчете стоимости тестирования вручную для каждого интервала учитывается, что данные затраты со временем уменьшаются, из-за замещения их автоматизированным тестированием.

При расчете затрат на разработку автоматизированных тестов учитывается объём тест кейсов,

подлежащих автоматизации, и количество инженеров по автоматизации тестирования, занятых на проекте, – это позволяет рассчитать время, необходимое на реализацию тестов, полностью замещающих ручное тестирование [2].

После этого рассчитываются суммарные затраты, т.е. сумма затрат, полученных в результате первых двух вычислений. Затем для сравнения добавляется стоимость тестирования без использования автоматизированных тестов.

Эффективностью использования автоматизации, или экономией трудозатрат, является разница между накопленными затратами на тестирование вручную и тестированием с помощью скриптов на данном интервале.

На рисунке 1 представлена разница между накопленными затратами с и без автоматизации на одном из проектов:

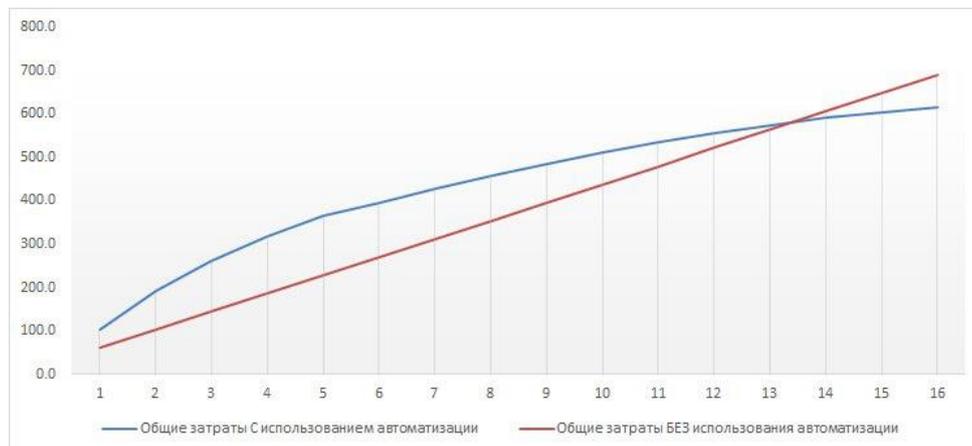


Рис. 1 - Разница между накопленными затратами с и без автоматизации

По данным рисунка 1, становится понятно, что в данном случае автоматизация имеет экономическое обоснование. На первоначальном этапе затраты на написание автотестов значительно выше, чем на ручное тестирование. Однако, со временем, после активной стадии разработки скриптов, издержки на их поддержку становятся незначительными, в то время как рост затрат на тестирование вручную остается постоянным. В конечном итоге, экономический эффект от внедрения автоматизации наступает достаточно быстро.

В рамках выполнения данной работы была проведена оценка эффективности внедрения автоматизированного тестирования на одном из проектов компании, предоставляющей услуги по автоматизации тестирования, методом расчёта периода возврата инвестиций (ROI). В результате проведенных расчётов было доказано, что данный метод может являться одним из способов оценки экономической эффективности и целесообразности внедрения автоматизации.

Также необходимо выделить следующее направление дальнейших исследований - метод расчёта рентабельности инвестиций с точки зрения эффективности использования ресурсов, как способ оценки экономической эффективности внедрения автоматизации тестирования, т.к. только детальный анализ может помочь сделать осознанный выбор, а также выявить плюсы и минусы выбранного подхода.

Список использованных источников:

1. Steven M. Bragg, Business Ratios and Formulas
2. Evaluating the Effectiveness of Test Automation [Электронный ресурс] / Test Automation – Электрон. дан. – Режим доступа: <https://www.a1qa.com/blog/evaluating-the-effectiveness-of-test-automation/> – Загл. С экрана – Яз. англ.

AWS СЕРВИСЫ ПРИ РАЗРАБОТКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА

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

Рылеев Е.К.

Таборовец В. В. – к. т. н., доцент

В докладе будут рассмотрены основные сервисы AWS, которые используются при разработке программного обеспечения основанного на облачных сервисах Amazon и предназначенные для поддержки электронного документооборота в информационных сетях.

Сервисы Amazon являются наиболее распространенными и стабильными при разработке программного обеспечения на языках, не создаваемых компанией Microsoft на текущий момент, т.к. практически все они поддерживают репликацию и автомасштабирование. Также AWS имеет довольно

демократичные цены, чем обуславливается частое размещение pet-проектов на данных сервисах.

Управление AWS осуществляется как с помощью веб интерфейса (AWS console), так и с помощью Command Line Tools. В консоли собраны все сервисы AWS, но функциональность настройки несколько обрезана. В командной строке же можно более гибко настроить тот или иной сервис, так же доступны закрытые в консоли функции.

Amazon Elastic Compute Cloud (EC2)

EC2 — это облачный сервис, предоставляющий виртуальные сервера (AmazonEC2 Instance), 2 вида хранилищ данных, а так же балансировщик нагрузки (LoadBalancer).

EC2 — это не что иное, как сервис, предоставляющий VPS в настоящем облаке, где сервер может легко мигрировать между нодами, а хранилище легко может быть расширено до почти безразмерного. EC2 позволяет запускать уже заранее сконфигурированные серверы с предустановленными ОС: AmazonLinux, RedHatEL, SuseES, Windows, OracleEL.

Так же возможно создавать свои образы (AMI — AmazonMachineImage) и использовать любой Linux. Есть возможность настроить защиту доступа к серверам. EC2 инстансы объединяются в группы безопасности (SecurityGroups) с возможностью ограничения доступа по портам с IP или подсетей [1].

Балансировка нагрузки и автомасштабирование являются очень важными функциями EC2. Вы можете создать правила при которых станет возможно автоматически увеличить количество серверов, например, если один или несколько серверов не справляются с нагрузкой. Контроль за здоровьем серверов ведёт ещё один сервис AWS — AmazonCloudWatch. С помощью этого сервиса можно создавать разного рода проверки — checks — с помощью которых контролируются важнейшие показатели работы ОС.

AmazonSimpleStorage (S3)

AmazonS3 это сервис для хранения данных в файлах. Указано, что предоставляется безразмерное пространство для хранения файлов размером от 1 байта до 5 Терабайт. Предоставляется 99.999999999% гарантия целостности и 99.99% гарантия доступности файлов в год.

Файлы хранятся в отдельных бакетах (bucket), в которых можно создавать директории и поддиректории. Бакеты хранятся в разных регионах (Region). Доступны следующие регионы: US Standard, US West (Oregon), US West (Northern California), EU (Ireland), Asia Pacific (Singapore), Asia Pacific (Tokyo), South America (Sao Paulo), и GovCloud (US) [2].

К бакетам можно применять разного рода политики безопасности: делать их приватными, публичными, а также разделять права между пользователями. Например, можно открыть вебсайт: bucketname.s3-website-us-east-1.amazonaws.com и хранить там статический контент.

S3 может логировать запросы и складывать отчёты в отдельный бакет. Это удобно при расследовании, когда много пользователей/приложений имеют доступ к сервису. Загрузка, удаление и другие операции доступны по REST или SOAP, так же возможно шифрование канала передачи данных с S3.

Интересной деталью является то, что можно встроить BitTorrent протокол заменой http, как основного протокола скачивания файлов.

S3 так же предполагает версиюность файлов. Всегда можно восстановить файл предыдущей версии, т.е. откатиться до нужного состояния.

Amazon Relational Database Service (RDS)

RDS — это сервис баз данных, который выносится на отдельную машину. Проще говоря, это отдельные VPS серверы, оптимизированные для работы с базами данных.

В Amazon RDS доступны следующие Системы Управления Базами Данных: MySQL community edition, Oracle Database Standard Edition One, Oracle Database Standard Edition, Oracle Database Enterprise Edition. Дисковое пространство RDS инстанса так же заказывается клиентом. Минимальный размер стораджа — 5 Гб. Все RDS инстансы работают на 64 битной платформе [3].

Существует возможность гибко настроить доступ к серверу БД с помощью групп безопасности. Доступ возможно дать отдельным адресам/подсетям или же группам безопасности EC2 и всем серверам, которые в неё входят. Это полезно, например при автомасштабировании, когда все экземпляры приложения поднимаются в одной группе и имеют доступ к серверу БД.

Так же можно настроить репликацию между серверами баз данных через консоль или утилиты командной строки. RDS поддерживает мгновенные слепки (Snapshot) и автобекап, давая возможность быстро и качественно восстановить данные. Если же случаются проблемы с аппаратным обеспечением, RDS автоматически перенесёт ваш хост на здоровую ноду.

При выходе обновлений, СУБД могут быть автоматически пропатчены и перезагружены. Клиенты уведомляются заблаговременно. Стоит заметить, что root доступа к СУБД нет. Возможности хранения встроенных процедур и тонкие настройки осуществляются через API и утилиты командной строки.

SimpleQueueService (SQS)

SQS — сервис для построения очередей событий. Требуется такая очередь, например, когда разделены приложения создания имейла и его посылка. Тогда создаётся элемент очереди с телом письма, хедерами и т.п, а приложение, отправляющее почту, считывает элементы из очереди и рассылает их. Лимитов по количеству очередей и по количеству элементов в очередях Amazon не предоставляет [4].

SQS может использоваться вместе с сервисом SNS, который является своего рода маршрутизатором сообщений: можно настроить SNS таким образом, чтобы он принимал все сообщения идущие от всех

микросервисов и автоматически посылал их в нужные SQS, подписанные на сообщения определенного типа.

Применение при разработке программного обеспечения электронного документооборота

Применение AWS-сервисов при разработке программного обеспечения электронного документооборота с микросервисной архитектурой позволяют значительно упростить разработку сообщения между микросервисами, процесс развертывания. Позволяют легко настроить посылку каких-либо уведомлений её пользователям путем внедрения использования SNS-SQS связки. Использование EC2-инстансов типа t2 позволяет значительно уменьшить стоимость работы приложения. Также использование AWS-сервисов упрощает задачу балансировки нагрузки на приложения, так как они имеют встроенные балансировщики и у разработчиков остается единственная задача – выбор типа балансировщика в зависимости от типа сервиса. Использование RDS позволяет легко создавать реплики работающей базы данных, что также значительно снижает нагрузку и увеличивает скорость работы приложения. Использование S3 позволяет хранить неограниченное количество документов и обеспечит их доступность в любое время.

Список использованных источников:

1. Документация AWSEC2 [Электронный ресурс]. – Режим доступа: <https://aws.amazon.com/documentation/ec2/>. Дата доступа: 04.04.2018.
2. Документация AWSS3 [Электронный ресурс]. – Режим доступа: <https://aws.amazon.com/documentation/s3/>. Дата доступа: 04.04.2018.
3. Документация AWSRDS [Электронный ресурс]. – Режим доступа: <https://aws.amazon.com/documentation/rds/>. Дата доступа: 04.04.2018.
4. Документация AWSSQS [Электронный ресурс]. – Режим доступа: <https://aws.amazon.com/documentation/sqs/>. Дата доступа: 04.04.2018.

МИКРОПРОЦЕССОРНАЯ СИСТЕМА ДЛЯ ФОРМИРОВАНИЯ ИСПЫТАТЕЛЬНОГО МАГНИТНОГО ПОЛЯ ПРОМЫШЛЕННОЙ ЧАСТОТЫ

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

Селедец В.Н.

Леванцевич В.А. – м.т.н., старший преподаватель

Надежность и достоверность работы радиоэлектронной аппаратуры зависит от её устойчивости к воздействию различного рода помех. Среди этих помех можно выделить группу помех, обусловленных магнитными полями промышленной частоты. Их воздействие может привести к нарушению нормального функционирования или выходу из строя радиоэлектронной аппаратуры.

Силовой характеристикой магнитного поля является его напряженность H , которая измеряется в амперах на метр.

Для оценки влияния магнитных полей промышленной частоты на работоспособность аппаратуры разработан ряд стандартов, определяющих требования к оборудованию для проведения испытаний на устойчивость к воздействию магнитных полей, значений напряженностей, а также методику проведения испытаний. Согласно ГОСТ Р 50648-94 [1] для создания испытательного магнитного поля используется индукционная катушка Гельмгольца. Для получения магнитного поля требуемой напряженности величина регулируемого переменного тока, протекающего через катушку, может достигать 30 ампер.

Использование автотрансформаторов [1] для задания такой величины тока не позволяет автоматизировать процесс проведения испытаний, а также не обеспечивает требуемую точность установки тока из-за ступенчатой регулировки тока. Применение аналоговых усилителей также ограничено из-за высокой рассеиваемой мощности на транзисторах выходного каскада.

Представляют интерес усилители, работающие в ключевом режиме [2]. В таких усилителях транзистор всё время или разомкнут (выключен), или замкнут (включен), поэтому рассеиваемая транзистором мощность минимальна.

Если на вход такого усилителя подать импульсный сигнал постоянной частоты и переменной скважности, то меняя длительность импульсов можно менять среднее напряжение на выходе. Такой принцип формирования сигнала называется широтно-импульсной модуляцией (ШИМ). Для формирования напряжения промышленной частоты на вход усилителя надо подать ШИМ-сигнал показанный на рисунке 1.

Для реализации поставленной задачи предлагается использовать микропроцессорную систему на базе микроконтроллера Atmega32 фирмы Atmel.



Рис. 1. ШИМ-сигнал

ШИМ-сигнал для получения синусоиды формируется с помощью аппаратного ШИМ реализуемого на двух таймерах-счетчиках микроконтроллера. Для этого в память микроконтроллера записывается 256 значений таблицы периода синуса частоты 50Гц, рассчитанной по формуле (1):

$$y(i) = \left(1 + \sin \left(\frac{2\pi \times i}{N} \right) \right), \quad (1)$$

где N - количество точек на период формирования синуса;
 i - i -я точка периода.

Timer1 микроконтроллера определяет частоту генерации синуса. Он осуществляет счет от 0 и до числа, записанного в регистр порога сравнения счетчика. После этого возникает прерывание, и счет опять начинается с 0. При обработке прерывания выбирается следующее число из таблицы синуса. Это число умножается на коэффициент K , который задает амплитуду синуса. Полученный результат записывается в регистр ШИМ второго счетчика Timer2, который формирует ШИМ-сигнал.

Timer2 работает в режиме быстрой ШИМ [3]. Этот режим предназначен для генерации ШИМ-импульсов повышенной частоты. В этом режиме используется однонаправленная работа счетчика. Временная диаграмма режима быстрой ШИМ приведена на рисунке 2.

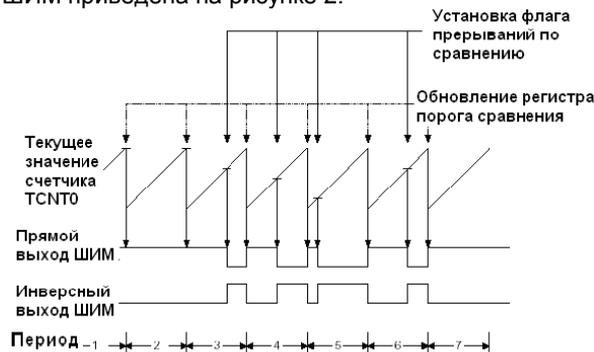


Рис. 2. Временная диаграмма быстрой ШИМ

Обновление регистра порога сравнения происходит всякий раз при переполнении счетчика. Разрядность ШИМ равна 8, при этом частота ШИМ составляет около 15 КГц. Выбор такой частоты ШИМ обусловлен тем, что на частотах ниже 4 КГц, слышны акустические шумы, вызванные механическими колебаниями индукционных катушек испытательной камеры.

Для стабилизации амплитуды синусоидального сигнала, генерируемого ШИМ, в ходе работы устройства, используется ПИД-регулятор [4].

Данный тип регуляторов характеризуются высокой скоростью и точностью работы, а также способностью устранять статическую ошибку регулирования. *Статическая ошибка* - это разность величин регулируемого параметра в исходном и конечном (после окончания регулирования) состояниях равновесия системы.

ПИД-регулятор формирует управляющий сигнал, являющийся суммой трёх слагаемых, первое из которых пропорционально разности входного сигнала и сигнала обратной связи (сигнал рассогласования), второе — интеграл сигнала рассогласования, третье — производная сигнала рассогласования. Рассчитать значение управляющего сигнала можно по формуле (2):

$$u(t) = P + I + D = K_p e(t) + K_i \int_0^t e(r) dr + K_d \frac{de}{dt} \quad (2)$$

где K_p , K_i , K_d — коэффициенты усиления пропорциональной, интегрирующей и дифференцирующей составляющих регулятора соответственно.

Для функционирования ПИД-регулятора необходимо наличие значения измеряемой величины и значения установки. Значение установки задаётся пользователем, оно же является заданным значением амплитуды синусоидального сигнала. Значением измеряемой величины является текущее значение

амплитуды синусоидального сигнала, которое измеряется и преобразуется аналого-цифровым преобразователем микроконтроллера. Структурная схема функционирования ПИД-регулятора в системе приведена на рисунке 3.



Рис. 3. Структурная схема функционирования ПИД-регулятора в системе

Выводы. Была разработана микропроцессорная система, позволяющая автоматизировать процесс проведения испытаний и повысить точность установки параметров испытательного магнитного поля. Программное средство для системы реализовано на языке C в среде AtmelStudio.

Список использованных источников:

1. Устойчивость к магнитному полю промышленной частоты.: ГОСТ Р 50648-94. Введен 3.03.94 Постановлением Госстандарта России.
2. П. Шкритек. Справочное руководство по звуковой схемотехнике. М: «Мир», 1991, с.226-234.
3. А.В. Естифеев, Микроконтроллеры AVR семейств Tiny и Mega фирмы «ATMEL»– М.: Издательский дом «Додэка – XXI», 2004, с. 86-9.7
4. ПИД-регулятор. Общие принципы [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://www.owen.ru/41523665>.

СРЕДСТВО КОММУНИКАЦИИ ПРЕПОДАВАТЕЛЯ И СТУДЕНТА

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

Скрипко О.А.

Данилова Г.В. – м.т.н., ассистент

В настоящее время растёт тенденция использования различных информационных систем и сервисов в сфере обучения. Нередко имплементация той или иной системы ведёт к возрастанию эффективности обучения молодёжи и к снижению затрат на человеко-часы преподавателей. Внедрение средства коммуникации преподавателя и студента сможет не только повысить эффективность общения, но и наладить диалог между ними.

Человечество постоянно эволюционирует. Появляется всё больше технологий и средств информатизации, которые, в конечном счёте, становятся неотъемлемой частью жизни людей.

В след за возрастанием необходимых знаний, учебная программа в ВУЗах меняется. Сегодня чаще всего можно наблюдать классическую методику преподавания, которая характеризуется разбиением учебного процесса на три сообщающиеся ветки: лекционная читка, практическое закрепление и самостоятельное обучение.

Однако, в настоящее время использование классической методики не удовлетворяет как студентов, так и преподавателей. Студенты с трудом воспринимают монотонные лекции, практические работы опережают теоретический материал, а информацию в интернете не всегда получается систематизировать и актуализировать.

В результате, на первых порах держаться лишь отличники, закалённые схожей программой в школах. Остальным приходится прикладывать титанические усилия при изучении.

При исследовании информационных технологий в учебном процессе была найдена брешь в коммуникации между студентом и преподавателем. Во время лекции поступает много сложного материала, который всегда порождает вопросы. В таком случае студенту необходимо попросить помощи у преподавателя за разъяснением непонятого места в материале.

Однако поток студентов на ПОИТ насчитывает порой более 200 человек. В том случае, если каждый начнёт обращаться к преподавателю, у последнего голова пойдёт кругом. Потому преподаватели просят студентов задавать вопросы прямо в аудитории и посреди учебного процесса, что на деле оказывается неработающей методикой. Преподавателя перебивают, теряется мысль и ход изложения материала, а студенты тихо обсуждают умственные способности вопрошающего.

В другом случае, когда преподаватель после каждого труднопонятого места в материале спрашивает поток о непонятных местах, стоит тишина. Студенты или боятся спрашивать, поскольку внутри себя считают вопрос глупым, либо сами не понимают, что они не понимают. Последнее – наиболее чаще встречающийся случай. Пока студент не поймёт, зачем ему рассказали про интегралы и не применит их на практике, он не узнает пробелов в своих знаниях о них.

Целью данного дипломного проекта является веб-приложение по обеспечению коммуникации между

студентом и преподавателем. Преподаватель сможет выкладывать свои материалы для ознакомления студентами. В свою очередь, студенты смогут задавать вопросы по материалу, просто выделив непонятный участок мышкой. При этом выделенный участок помечается определённым цветом и открывается ветка обсуждения вопроса, в которой могут участвовать как студенты, так и преподаватели.

На основе этой системы будут построены и другие элементы системы:

- Преподаватель сможет оценивать хорошие вопросы для дальнейшего поощрения студента во время экзамена или зачёта, при чём преподаватель сможет либо отображать свою оценку студенту, либо скрыть её для всех, кроме себя.

- Преподаватель практических и лабораторных работ будет иметь возможность ведения ведомости оценок, которые играют роль при итоговых экзаменах. Такую ведомость смогут просматривать как студенты, так и преподаватель лекционного материала.

- Система фильтрации повторяющихся вопросов. Если при создании вопроса студентом такой же вопрос уже имеется в базе, то вопрос создан не будет, а студент перенаправляется на страницу обсуждения данного вопроса.

- Система разбиения лекций по датам. На основе уже имеющегося расписания занятий преподаватель сможет определять конкретную тему лекции к конкретному дню в расписании. При чём при выборе определённой даты будет открыт материал, который был прикреплен преподавателем.

С помощью вышеописанного веб-приложения планируется повысить эффективность подачи материала, наладить диалог между преподавателем и студентом. В конце концов, нужные вопросы всегда всплывают уже после лекции. Никогда не поздно задать их, сидя дома, перед этим правильно их сформулировать. В свою очередь, преподаватель имеет возможность анализировать непонятные моменты в своём материале и подстраивать его для более лёгкого усвоения.

Список использованных источников:

1. Данилова, Г. В. Программное средство управления формированием IT-компетенций / Г. В. Данилова // Дистанционное обучение – образовательная среда XXI века: материалы IX международной научно-методической конференции (Минск, 3-4 декабря 2015 года). – Минск: БГУИР, 2015. – С. 300 - 301.

2. Живицкая, Е. Н. Информационная система подготовки IT-специалистов / Е. Н. Живицкая, Г. В. Данилова // Информатизация образования. – 2017. – № 1 (79). – С. 54–72.

МЕТОДЫ МАШИННОГО ОБУЧЕНИЯ В РЕКОМЕНДАТЕЛЬНЫХ СИСТЕМАХ БАНКОВСКОЙ СФЕРЫ И ОБЛАСТИ БИЗНЕС-АНАЛИЗА

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

Стасюк А.В.

Медведев С.А. – к.т.н., доцент

Информация всегда играла чрезвычайно важную роль в жизни человека.

Общеизвестно высказывание о том, что тот, кто владеет информацией, тот владеет и миром. Иное сообщение стоит дороже жизни. По преданию, 13 сентября 490 года до н.э. греческий воин-гонец, прибежавший из Марафона в Афины, не останавливаясь в пути, упал замертво, но донес весть о победе над персами.

С давних времен сбор и систематизация сведений об окружающем мире помогали человеку выживать в нелегких условиях – из поколения в поколение передавался опыт и навыки изготовления орудий охоты и труда, создания одежды и лекарств. Информация постоянно обновлялась и дополнялась – каждое изученное явление позволяло перейти к чему-то новому, более сложному.

С течением времени роль информации в жизни человека становилась все существеннее. Нужно было изучать и понимать уже не только законы природы, но и понятия и ценности человеческого общества – литературу, искусство, архитектуру и т.д. Сейчас, в первой половине 21-ого века роль информации в жизни человека является определяющей – чем больше навыков и знаний он имеет, тем выше ценится как специалист и сотрудник, тем больше имеет уважения в обществе.

Важность принятия правильного решения финансовым аналитиком или бизнесменом, готовящимся вложить крупную часть своего капитала в дело, сложно переоценить. Иногда на кон ставятся экономики не только предприятий и холдингов, а стран и регионов.

Анализ текста.

Анализ текста, определение многих его параметров – семантический анализ, извлечение сущностей, анализ тональности текста и так далее – есть одна из основных задач данного приложения. Определяющий функционал.

Получение этих параметров приведет к возможности частичного ранжирования соответствующим интересам клиента.

Семантический анализ - этап в последовательности действий алгоритма автоматического понимания текстов, заключающийся в выделении семантических отношений, формировании семантического представления текстов. Один из возможных вариантов представления семантического представления -

структура, состоящая из "текстовых фактов" [1]. Семантический анализ в рамках одного предложения называется локальным семантическим анализом. В общем случае семантическое представление является графом, семантической сетью, отражающим бинарные отношения между двумя узлами - смысловыми единицами текста. Глубина семантического анализа может быть разной, а в реальных системах чаще всего строится только лишь синтаксико-семантическое представление текста или отдельных предложений.

Тональность — это эмоциональное отношение автора высказывания к некоторому объекту (объекту реального мира, событию, процессу или их свойствам/атрибутам), выраженное в тексте. Эмоциональная составляющая, выраженная на уровне лексемы или коммуникативного фрагмента, называется лексической тональностью (или лексическим сентиментом). Тональность всего текста в целом можно определить как функцию (в простейшем случае сумму) лексических тональностей составляющих его единиц (предложений) и правил их сочетания[2].

Извлечение сущностей из текста (Entity Extraction).

Типичная задача извлечения информации: просканировать набор документов, написанных на естественном языке, и наполнить базу данных выделенной полезной информацией.

С данной задачей на данный момент доволно неплохо справляются коммерческое решение Rosette и целая система библиотек для NLP – Spacy. Правда оба решения имеют неплохой процент качества работы именно с текстами на английском языке, хотя и поддерживают другие языки. Еще одна проблема – большой объем ресурсов, необходимый для работы Spacy. Один «поднятый» instance Spacy потребляет 2 Гб оперативной памяти. Так как в рамках приложения решается задача возможности заведения для каждого клиента своего словаря – появляется проблема с количеством доступной оперативной памяти.

Обучение классификатора. SVM.

В общем, задача машинного обучения сводится к получению набора выборки данных и, в последствии, к попыткам предсказать свойства неизвестных данных. Если каждый набор данных — это не одиночное число, а например, многомерная сущность (multi-dimensional entry или multivariate data), то он должен иметь несколько признаков.

Машинное обучение представляет собой обучение выделению некоторых свойств выборки данных и применение их к новым данным. Вот почему общепринятая практика оценки алгоритма в Машинном обучении — это разбиение данных вручную на два набора данных. Первый из них — это обучающая выборка, на ней изучаются свойства данных. Второй — контрольная выборка, на ней тестируются эти свойства.

SVM. Основная идея метода — перевод исходных векторов в пространство более высокой размерности и поиск разделяющей гиперплоскости с максимальным зазором в этом пространстве. Две параллельных гиперплоскости строятся по обеим сторонам гиперплоскости, разделяющей классы. Разделяющей гиперплоскостью будет гиперплоскость, максимизирующая расстояние до двух параллельных гиперплоскостей. Алгоритм работает в предположении, что чем больше разница или расстояние между этими параллельными гиперплоскостями, тем меньше будет средняя ошибка классификатора.

Итогом проделанной работы стала рекомендательная система, ранжирующая новости в соответствии с требованиями пользователя-клиента данного сервиса. Разработаны бизнес-правила, разработана и реализована «гибкая» архитектура приложения, позволяющая без исправления кода настраиваться под требования нового клиента.

Список использованных источников:

- 1) Building Machine Learning Systems with Python. Luis Pedro Coelho, Willi Richert. Построение систем машинного обучения на языке Python. Луис Педро Коэльо, Вилли Ричарт.
- 2) Machine Learning with Spark: Create scalable machine learning applications to power a modern data-driven business using Spark. NickPentreath.

ЗАЩИТА ДАННЫХ В ОБЛАЧНОМ ХРАНИЛИЩЕ

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

Сыч Ю.В.

Прохорчик Р.В. – ассистент каф. ПОИТ, м.т.н.

Облачные хранилища данных – сервисы, предоставляющие возможность хранить свои файлы на удаленных серверах, а также получать к ним доступ из любой точки мира, где есть доступ в Интернет. В условиях роста необходимости хранения информации, данные сервисы стали популярны среди пользователей. В настоящее время множество компаний также хранит большую массу информации, необходимая для принятия решений. Она накапливается в различных источниках и хранилищах, превращаясь в опыт предприятия. Однако, отправляя свои файлы в облачное хранилище, пользователи не задумываются о защите информации в облаках от доступа третьих лиц. Стоит ли всецело доверять держателям облачных сервисов, полагаясь на предпринимаемые ими меры защиты информации – личное дело каждого.

Опасность хранения файлов в облачном хранилище заключается в том, что пользователь передает файлы на хранение третьему лицу с неизвестными, по отношению к пользователю, намерениями. А опасность именно файлов, как объекта информации, в том и заключается, что с него можно сделать копию и

пользователь об этом факте никак не узнает. Данные компаний о заказчиках, счётах, денежных оборотах, хранящиеся в облаке, могут быть подвержены утечке, вследствие чего компании могут оказаться в крайне плохом положении.

В качестве защиты данных в облачном хранилище поставлена задача создать расширение, которое способно шифровать данные при их загрузке в сервис, а также обратная операция – расшифровка данных при скачивании из сервиса. Пользователь задаёт ключ и алгоритм для шифрования. Для создания расширения выбран браузер Chrome, а в качестве сервиса облачного хранилища используется Dropbox. Основным алгоритмом шифрования выбран аес (симметричный алгоритм блочного шифрования), обеспечивающий высокую скорость шифрования данных. В данный момент реализовано:

- Шифрование данных при загрузке на сервис перетягиванием (draganddrop) файлов в облако;
- Расшифровка данных при скачивании с главной страницы сайта;
- Выбор алгоритма шифрования у пользователя;
- Сохранение ключа для шифрования пользователя.

Принцип работы расширения для защиты облака:

- При попытке загрузки в сервис (скачивании из сервиса) данных, запрос на загрузку будет отменён;
- Полученные данные шифруются выбранным алгоритмом шифрования ключом пользователя;
- Формируется запрос отправки/скачивания зашифрованных данных;
- Отправка запроса (загрузка/скачивание данных).

Основные преимущества расширения для защиты облака:

- работа на всех популярных операционных системах, таких как: Windows, Mac, Linux;
- поддержка русского языка – управлять расширением просто, а все функции понятны;
- ключ и алгоритм для шифрования имеется только на стороне клиента, таким образом, третьи лица

практически теряют возможность получить доступ к данным пользователя.

Основные недостатки расширения:

- данные шифруются непосредственно перед отправкой или загрузкой, а не на лету. Таким образом необходимо ожидать шифрования(расшифровки) данных, а затем отправку или скачивание, вследствие чего теряется скорость работы с облачным хранилищем;

- программа может работать только в браузере Chrome версии 4.0 и выше.

В дальнейшем необходимо реализовать шифрование данных при загрузке на сервис через форму облачного хранилища, расшифровку данных при скачивании из формы открытого файла, создание вспомогательного окна процесса шифрования.

В целом при совершенствовании расширения необходимо сформировать адаптивность для различных браузеров (IE, Firefox, Opera) и возможность шифрования для других сервисов облачных хранилищ, таких как Google Диск, OneDrive, Яндекс.Диск.

Список использованных источников:

1. Статья Способ удобного шифрования данных в облаке (собственными средствами) // Habrahabr [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/241720/>
2. Статья Шифрование данных в облаке, 2016: Исследование Gemalto и Ponemon [Электронный ресурс]. – Режим доступа: http://www.tadviser.ru/index.php/Статья:Шифрование_данных_в_облаке

СТАТИСТИЧЕСКИЙ ПОДХОД К УПРАВЛЕНИЮ РЕСУРСАМИ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ

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

Толкачёв А.В.

Куликов С.С. – к.т.н., доцент

В настоящее время облачная модель поставки и функционирования сетевых приложений и сервисов является доминирующей на рынке и продолжает демонстрировать стабильный рост. Использование методов статистического моделирования для управления ресурсами облачного вычислительного кластера может значительно повысить эффективность функционирования системы и минимизировать накладные расходы на балансировку нагрузки.

Облачные вычисления – подход, при котором ресурсы поставляются в виде услуги, могут быть сданы в аренду и предоставлены пользователям через сеть по запросу. Для максимально эффективной утилизации вычислительных ресурсов активно используются технологии виртуализации. Общая схема облачного вычислительного кластера приведена на рисунке 1.

В рамках одного физического сервера (хоста) при помощи специального программно-аппаратного комплекса (гипервизора) функционирует множество виртуальных машин, выступающих в качестве среды функционирования ПО. Гипервизоры связаны с системой мониторинга, ответственной за динамическое перераспределение нагрузки между физическими хостами. Трансфер виртуальной машины между хостами является достаточно ресурсоёмкой операцией, так как требует дополнительные сетевые ресурсы и вычислительные ресурсы для синхронизации [1]. Управление ресурсами, базирующееся на основе модели статистических предсказаний, позволяет минимизировать накладные расходы на миграцию виртуальных

машин между физическими хостами, избегая преждевременного перераспределения вычислительных мощностей, необходимость в которых обусловлена текущими показаниями мониторинга. Данный подход позволяет давать более объективную оценку текущей нагрузки на вычислительный кластер и предоставлять альтернативные схемы управления ресурсами с учётом наиболее вероятного изменения потребления ресурсов с течением времени [2, 3].

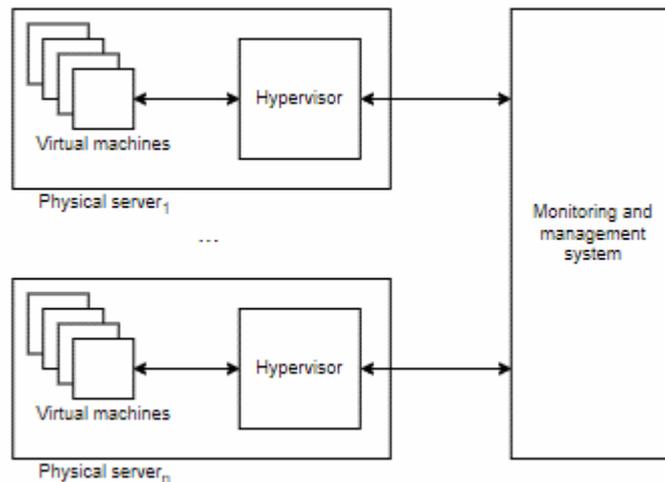


Рис. 1 – Общая схема облачного вычислительного кластера

Использование статистических предсказаний в процессе управления облачными ресурсами так же упрощает соблюдение соглашения об уровне предоставления услуг (англ. ServiceLevelAgreement, SLA). Статистическое планирование использования ресурсов позволяет предварительно резервировать дополнительные вычислительные мощности для поддержания метрик доступности и функционирования сервиса (время доступа, общая доступность и т.п.) на заявленном уровне [2].

Список использованных источников:

1. Ворожцов А.С., Тутова Н.В., Тутов А.В. Динамическое распределение вычислительных ресурсов центров обработки данных // Т-COMM: Телекоммуникации и транспорт. – 2016. – Том 10. - №7. – С.47-51
2. Resource Central: Understanding and Predicting Workloads for Improved Resource Management in Large Cloud Platforms. [Электронный ресурс] – Режим доступа: <https://www.microsoft.com/en-us/research/publication/resource-central-understanding-predicting-workloads-improved-resource-management-large-cloud-platforms/>
3. Michael Borkowski, Stefan Schulte, Christoph Hochreiner Efficient Resource Management Technique for Performance Improvement in Cloud Computing // IEEE/ACM 9th International Conference on Utility and Cloud Computing. – 2016.

ДЕНОРМАЛИЗАЦИЯ КАК СРЕДСТВО УЛУЧШЕНИЯ ПРОИЗВОДИТЕЛЬНОСТИ БАЗ ДАННЫХ

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

Хадарович П.В.

Медведев С.А. – к.т.н., доцент

В настоящее время количество высоконагруженных приложений неизменно увеличивается. Чтобы справиться с возрастающей нагрузкой, производится либо улучшение аппаратной части машины, обрабатывающей запросы пользователей, либо добавление нескольких новых машин для обработки запросов. Но применение некоторых методов оптимизации производительности баз данных, позволит улучшать аппаратную часть или внедрять новые машины значительно реже. Денормализация является одним из таких методов. При ее грамотном применении, можно существенно увеличить скорость обработки данных, что в свою очередь, уменьшит время обработки запросов.

Денормализация – намеренное преобразование структуры базы данных в состояние, которое не соответствует критериям нормализации баз данных. Архитектура базы данных, соответствующая критериям нормализации, способствует ее улучшению пониманию и дальнейшему сопровождению. Но при больших объемах данных, данный подход может существенно снизить производительность обработки данных. В таких случаях, денормализация способна уменьшить время обработки данных.

Применение денормализации имеет смысл тогда, когда происходит потеря производительности по следующим причинам:

- в запросе присутствует слишком много операций объединения таблиц. В таких случаях, производится

объединение нескольких таблиц в одну, что позволяет сократить количество операций объединения [1];

- в запросе происходят сложные вычисления, возникающие, например, при использовании групповых и агрегатных функций. Чтобы избежать выполнения таких вычислений каждый раз при извлечении данных, применяется предварительное вычисление необходимых значений во время операции вставки новой записи в таблицу с последующим сохранением полученного значения в отдельной колонке, принадлежащей этой же таблице. Такой подход демонстрирует высокую производительность для данных, которые необходимо часто извлекать, но которые редко изменяются [2].

Основными недостатками денормализации является появление избыточной информации. Также ее применение негативно сказывается для данных, которые часто обновляются. Кроме того, нельзя недооценивать тот факт, что денормализация ухудшает понимание модели данных и ее дальнейшее сопровождение.

Денормализация является очень действенным способом для улучшения производительности обработки данных. Она часто находит свое практическое применение в различных моделях данных. Но у нее есть свои недостатки. Прежде чем применять денормализацию, для начала нужно попробовать другие способы оптимизации производительности. И в том случае, когда они не приносят должного результата, стоит обратиться к денормализации.

Список использованных источников:

1. Grant Fritchey, *Sql Server 2012 Query Performance Tuning* – 470с.
2. <https://habrahabr.ru/post/64524/> - электронный ресурс

АВТОМАТИЗАЦИЯ ТЕСТИРОВАНИЯ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

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

Хильчук А.С.

Куликов С.С. – к.т.н., доцент

На данный момент во многих сферах общества используется множество программных продуктов. Также наблюдается тенденция к использованию мобильных приложений и отход от веб-вариантов приложений. Поэтому требования к стабильности и удобству использования этих приложений становятся выше.

Тестирование ПО – это процесс исследования, испытания программного продукта, по результатам которого можно выявить ситуации, в которых поведение программы является неправильным, нежелательным или не соответствующим спецификации. [1]

Для снижения затрат на проведение ручного тестирования внедряется её автоматизация. Автоматизированное тестирование ПО – это процесс верификации программного обеспечения, при котором основные функции и шаги теста, такие как запуск, инициализация, выполнение, анализ и выдача результата, выполняются автоматически, что помогает сократить время тестирования и упростить его процесс. [2]

Одной из главных задач при внедрении автоматизации тестирования на проекте по разработке мобильных приложений является выбор фреймворка для разработки тестов. Самым популярным из них – Appium [3], который является платформой для кроссплатформенной автоматизации тестирования с открытым исходным кодом. С помощью данного программного решения можно выполнять тестирование нативных, гибридных и мобильных веб-приложений на платформе iOS и Android. Управление элементами приложения и мобильным устройством осуществляется с использованием WebDriverJSONWireProtocol. Данный фреймворк является инструментом для тестирования по методу чёрного ящика, следовательно, нет необходимости в рекомпиляции приложения, а также в доступе к исходному коду. Данный фреймворк доступен для языков программирования Java, .Net (C#), Python, Ruby, JavaScript и PHP. К минусам можно отнести невозможность параллельного запуска тестов для платформы iOS, ограниченную поддержку жестов, сложность в настройке, поддержку версии Android только 4.1 и выше и достаточно низкую скорость работы.

К популярным фреймворкам по разработке тестов для мобильных приложений можно отнести Robotium [4]. Данная платформа с открытым исходным кодом позволяет тестировать нативные и гибридные Android-приложения по методу чёрного ящика, а значит для запуска теста, как и с Appium, от разработчика потребуется только исполняемый арк-файл. Но стоит учесть, что данная платформа подойдёт только для приложений, где переход к следующей итерации разработки не влечёт за собой сильного изменения пользовательского интерфейса, в противном случае разработчику тестов будет необходимо переписать вплоть до 90% кода тестов. Так же к минусам можно отнести поддержку только языка Java в качестве языка программирования тестов, а также поддержку только Android-приложений.

К группе часто используемых платформ для разработки автоматизированных тестов под мобильные приложения также относится Espresso от компании Google [5]. Основной API данного решения невелик, но так как это проект с открытым исходным кодом, код может быть расширен разработчиком тестов. Отличительной особенностью данной платформы является синхронизация с главным UI-потокот Android-приложения. Большинство фреймворков запускают тесты в отдельном потоке, вследствие чего возникают проблемы со скоростью и могут возникнуть критические ошибки в самих тестах вследствие обновления пользовательского интерфейса приложения. Большинство разработчиков игнорируют этот факт и ставят задержки в потоке

(sleep), что является достаточно грубым нарушением принципов построения логики работы тестов. С Espresso разработчику тестов не придётся осуществлять методы синхронизации, так как фреймворк осуществляет её сам посредством интеграции с `MainUIThread`.

Для написания автоматизированных тестов для iOS-приложений также используют платформу XCUITest [6], которая разрабатывается компанией Apple и поставляется вместе со средой разработки XCode. Также её можно использовать для проведения юнит-тестов и тестов производительности iOS-приложений. Как и Espresso, данная платформа потребует доступ к исходному коду тестируемого приложения. По этой причине, разработчик тестов имеет возможность написания тестов только на языках программирования Swift или Objective-C и поэтому больше используется разработчиками приложения, нежели сотрудниками отдела тестирования. Так как это не кроссплатформенное решение, скорость выполнения тестов в значительной степени выше по сравнению с Appium. К дополнительным плюсам можно отнести низкую степень появления ложного результата теста. Недостатком является сложность связки с системами непрерывной интеграции для автоматического запуска.

Таким образом, для мобильных приложений, будь это приложение интернет-банкинга, интернет-магазина или новостного портала, очень важна правильность работы, устойчивость к некорректным действиям пользователя и другим негативным факторам. Автоматизация тестирования в значительной степени ускоряет процесс проведения тестов, которые позволяют удостовериться в том, что приложение соответствует установленному техническому заданию, за достаточно короткий промежуток времени, выполняя проверки на различных мобильных окружениях и их конфигурациях.

Список использованных источников:

1. Гленфорд Майерс, Том Баджетт, Кори Сандлер. Искусство тестирования программ, 3-е издание (The Art of Software Testing, 3rd Edition.) — М.: «Диалектика», 2012. — 272 с.
2. Про Тестинг [Электронный ресурс] // protesting.ru: Тестирование Программного Обеспечения. URL: <http://www.protesting.ru/automation> (дата обращения: 12.03.2018).
3. Appium [Электронныйресурс] // appium.io: Appium – AutomationforApps. URL: <http://www.appium.io> (дата обращения: 12.03.2018).
4. Robotium [Электронныйресурс] // robotium.com: UserscenariotestingforAndroid. URL: <https://github.com/RobotiumTech/robotium> (дата обращения: 13.03.2018).
5. Espresso [Электронный ресурс] // developer.android.com: Espresso. URL: <https://developer.android.com/training/testing/espresso/index.html> (дата обращения: 13.03.2018).
6. XCUITest [Электронный ресурс] // developer.apple.com: Testing with Xcode. URL: https://developer.apple.com/library/content/documentation/DeveloperTools/Conceptual/testing_with_xcode/chapters/09-ui_testing.html (дата обращения: 13.03.2018).

ВЕБ-ПРИЛОЖЕНИЕ ОКАЗАНИЯ УСЛУГ СТРАХОВОЙ КОМПАНИИ

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

Чернышев Н.С.

Фадеева Е.Е. – ассистент

В последнее время веб-технологии активно развиваются и набирают популярность. На сегодняшний день сложно представить нашу жизнь без компьютера, телефона или сети Интернет.

Веб-приложение — клиент-серверное приложение, в котором клиентом выступает браузер, а сервером — веб-сервер. Логика веб-приложения распределена между сервером и клиентом, хранение данных осуществляется, преимущественно, на сервере, а обмен информацией происходит по сети.

Одним из преимуществ подхода с использованием веб-приложения является тот факт, что клиенты не зависят от конкретной операционной системы, поэтому веб-приложения являются кроссплатформенными средствами. Вместо того, чтобы писать различные версии веб-приложения для Microsoft Windows, MacOS, Unix и других систем, приложение создается один раз и может быть развернуто на любой системе. В связи с этим, отсутствует необходимость установки специализированных программных средств и разработки множества вариантов одного приложения для каждой платформы на мобильные устройства и персональные компьютеры, так как любой современный браузер может быть использован для работы с веб-приложением, что уменьшает затраты времени и средств на разработку этого приложения.

Также веб-приложение может выступать в качестве клиента других служб, например, базы данных или другого веб-приложения, расположенного на другом сервере.

Страхование — одна из важнейших категорий общественных отношений. В широком смысле оно включает различные виды страховой деятельности, которые в комплексе обеспечивают страховую защиту.

С древних времён объективная потребность в страховании обуславливается тем, что убытки подчас возникают вследствие разрушительных факторов, не подконтрольных человеку, таких как стихийные бедствия или различного рода чрезвычайные ситуации. В подобной ситуации невозможно взыскивать убытки с кого-либо и заранее созданный страховой фонд может быть источником возмещения полученного ущерба.

В общем и целом, страхование — система страховой защиты от возможного наступления различного рода рисков. Оно представляет собой способ возмещения страховщиком убытков пострадавшим путем их

распределения между всеми страхователями. Страховщик формирует общий страховой фонд за счет страховых взносов всей совокупности страхователей, средства которого при наступлении страховых событий направляет на страховые выплаты. Размер страхового покрытия определяется по соглашению сторон. На стоимость влияет и количество оказываемых дополнительных услуг.

В условиях современного общества страхование превратилось во всеобщее универсальное средство страховой защиты всех форм собственности, доходов и других интересов предприятий, организаций, юридических и физических лиц. Страхование играет все большую роль в обеспечении благополучия каждой семьи и каждого человека. С его помощью человек создает финансовые гарантии и обеспечивает средствами себя и членов своей семьи в случае наступления возможных, но незапланированных жизненных ситуаций. Наверное, нет такого человека, который не пользовался бы услугами страхования или не прибегал однажды к его помощи.

Страховые компании предлагают широкий спектр услуг по страхованию различных рисков, страхованию транспортных средств, страхованию имущества, страхованию гражданской ответственности, страхованию жизни и здоровья и других видов.

Развитие страхования идет вместе с техническим прогрессом и подкрепляется различными законами и нормативными актами, так как затрагивает практически все сферы жизни.

Для того, чтобы сделать процесс страхования более удобным и легким, предпринята попытка автоматизации данного процесса. Таким образом, веб-приложение позволит облегчить оказание услуг страховой компанией, а также взаимодействие между клиентами и агентами страховой компании.

Благодаря современным технологиям, существует достаточное количество различных веб-приложений и ресурсов в сети Интернет, которые предоставляют информацию и услуги по страхованию.

Данная тема была выбрана как следствие тенденции активного развития веб-программирования и автоматизации процессов страхования. Актуальность выбранной темы обусловлена бурным развитием страхового рынка, новых форм и видов страхования в условиях современных экономических отношений.

Список использованных источников:

1. HibernateORMDocumentation [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://docs.jboss.org/hibernate/orm>
2. SpringFrameworkDocumentation [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://spring.io>

ОБЛАСТИ ПРИМЕНЕНИЯ АЛГОРИТМОВ РАСПОЗНАВАНИЯ ОБЪЕКТОВ

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

Чечётко А.Л.

Пармонов А. И. – канд. техн. наук, доцент

В настоящее время для решения многих практических задач используются системы компьютерного зрения. Среди них системы видеонаблюдения, системы помощи водителю и другие. В разных системах могут применяться различные алгоритмы и методы идентификации и распознавания объектов. Это связано с различной спецификой наблюдаемых объектов, а именно освещённостью сцены, относительности размеров объектов на сцене, частичным или полным перекрытием объектов и другими. В данном докладе рассмотрены области применения основных известных алгоритмов компьютерного зрения.

Гистограмма направленных градиентов (Histogram of Oriented Gradients, HOG) – дескрипторы особых точек, которые используются в компьютерном зрении и обработке изображений с целью распознавания объектов. Данная техника основана на подсчете количества направлений градиента в локальных областях изображения. Основной идеей алгоритма является допущение, что внешний вид и форма объекта на участке изображения могут быть описаны распределением градиентов интенсивности или направлением краев. Реализация этих дескрипторов может быть произведена путём разделения изображения на маленькие связанные области, именуемые ячейками, и расчетом для каждой ячейки гистограммы направлений градиентов или направлений краев для пикселей, находящихся внутри ячейки. Комбинация этих гистограмм и является дескриптором. Для увеличения точности локальные гистограммы подвергаются нормализации по контрасту. Дескриптор HOG имеет несколько преимуществ над другими дескрипторами. Как обнаружили Далал и Триггс, грубое разбиение пространства, точное вычисление направлений и сильная локальная фотометрическая нормализация позволяют игнорировать движения пешеходов, если они поддерживают вертикальное положение тела. Дескриптор HOG является хорошим средством нахождения людей на изображениях [1].

Метод Виолы-Джонса (Viola-Jones object detection) – алгоритм, позволяющий обнаруживать объекты на изображениях в реальном времени. Его предложили Паул Виола и Майкл Джонс в 2001 году. Хотя алгоритм может распознавать различные классы изображений, основной задачей при его создании было обнаружение лиц. У метода имеется множество реализаций, в том числе в составе библиотеки компьютерного зрения OpenCV – функция `cvHaarDetectObjects()` [2]. Алгоритм находит лица с высокой точностью и низким количеством ложных срабатываний.

Алгоритм сегментации MeanShift группирует объекты с близкими признаками. Пиксели со схожими

признаками объединяются в один сегмент, на выходе получаем изображение с однородными областями [3].

Фильтр Калмана – эффективный рекурсивный фильтр, оценивающий вектор состояния динамической системы, используя ряд неполных и зашумленных измерений [4]. Назван в честь Рудольфа Калмана. Фильтр широко используется в инженерных и эконометрических приложениях: от радаров и систем технического зрения до оценок параметров макроэкономических моделей. Калмановская фильтрация является важной частью теории управления, играет большую роль в создании систем управления. Совместно с линейно-квадратичным регулятором фильтр Калмана позволяет решить задачу линейно-квадратичного гауссовского управления. Фильтр Калмана и линейно-квадратичный регулятор – возможное решение большинства фундаментальных задач в теории управления. Во многих приложениях компьютерного зрения размерность вектора состояния объекта превосходит размерность вектора данных наблюдения. И при этом фильтр Калмана позволяет оценивать полное внутреннее состояние объекта. Благодаря пошаговой природе алгоритма, он может в реальном времени отслеживать состояние объекта (без прогнозирования, используя только текущие замеры и информацию о предыдущем состоянии и его неопределенности).

Фильтр частиц является одним из самых популярных методов оптимальной фильтрации. В сравнении с часто применяемыми для задач компьютерного зрения расширенными фильтрами Калмана (ЕКФ) фильтры частиц не зависят от методов линеаризации или аппроксимации. Обычный ЕКФ плохо справляется с существенно нелинейными моделями, а также в случае шумов системы и измерений, сильно отличающихся от гауссовых, поэтому были разработаны различные модификации, такие как UKF (unscented KF), QKF (Quadrature KF) и т. п. Следует отметить, что в свою очередь фильтры частиц более требовательны к вычислительным ресурсам. Со времени первого описания алгоритма (Н. Гордоном, Д. Салмондом и А. Смитом) он нашел применение в различных областях — навигации, робототехнике, компьютерном зрении [5].

Список использованных источников:

1. Кулинкович, В.А. Применение методики гистограмм направленных градиентов для классификации дактилоскопических изображений / Минск: журнал Белорусского государственного университета. Математика. Информатика. – 2016. – С. 53-59.
2. Буй, Тхи Тху Чанг Распознавание лиц на основе применения метода Виолы–Джонса // Буй Тхи Тху Чанг, Фан Нгок Хоанг, В.Г. Спицын / Томск: журнал «Известия Томского политехнического университета». – № 5. – 2012. – С. 54-59.
3. Comaniciu D. Mean shift: A robust approach towards feature space analysis // Comaniciu D., Meer P. / IEEE Trans. Pattern Analysis and Machine Intelligence, 24. – 2002. – p. 603–619.
4. ТГПУИм. Л.Н. Толстого. Фильтр Калмана [Электронный ресурс] – Режим доступа: <http://poivs.tsput.ru/Math/ProbabilityAndStatistics/ProbabilityTheory/RandomProcesses/KalmanFilter>
5. Волков, В.А. Численное решение задач нелинейной фильтрации на основе алгоритмов фильтра частиц // В.А. Волков, И.А. Кудрявцева / М.: журнал московского авиационного института «Труды МАИ». Выпуск № 89 – 2013. – С. 18.

СИСТЕМА ПРЕДОСТАВЛЕНИЯ ТУРИСТИЧЕСКИХ УСЛУГ С ИСПОЛЬЗОВАНИЕМ НЕЙРОСЕТЕВЫХ ТЕХНОЛОГИЙ

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

Чигир В.С.

Хмелева А.В. – к.т.н., доцент

В докладе рассмотрены задачи, возникающие перед турагентствами при предоставлении туристических услуг, проанализированы существующие варианты их решения. По результатам рассмотрения их недостатков предложено создание системы предоставления туристических услуг с использованием нейросетевых технологий.

Эффективное использование информации, поступающей из различных источников в сфере туризма, является важной и непростой задачей. Сотрудники турагентств вынуждены предлагать продукт клиентам и принимать решения по добавлению туров, основываясь на частичной, неполной и неточной информации. Необходим эффективный механизм управления знаниями в быстро меняющейся среде для задачи продвижения туристского продукта. Современные туристские проекты достаточно высокого уровня сложности связаны с самыми разнообразными секторами экономики, могут включать все виды транспорта, проживания, организации питания, развлечений, продажи дополнительных товаров и услуг.

При принятии решения по предоставлению услуг, когда клиент просит подобрать ему турпродукт, команда, как правило, основывает свой выбор на интуиции, учитывая состояние рынка туризма. При решении сложных вопросов вряд ли можно полагаться и на интуицию одного человека или даже целой команды. Кроме того, представления, достаточно адекватные на данный момент, могут слишком быстро или просто незаметно устареть, а в результате привычное предложение, как считалось еще недавно, наилучших услуг уже будет противоречить запросам нового клиента. К тому же, привлекательность турпродукта является во многом субъективным обстоятельством, которое зависит от оценки туристом качества предлагаемых услуг[1]. Следует обратить внимание на такие индивидуальные различия туристов, как семейное положение, подходящие погодные условия, требовательность к уровню комфорта и качеству услуг, активность на отдыхе, уровень информированности и мобильности, желание получить максимум впечатлений от общения или стремление уединиться от окружающего мира.

Таким образом, в сфере туризма складывается новая ситуация, которую достаточно трудно даже

описать в деталях, а тем более проанализировать, не прибегая к современным информационным технологиям. Приводимые выше проблемы могут быть решены путем использования искусственных нейронных сетей, так как необходимо учитывать разнообразные не унифицированные факторы и совершать принятие решения на основе значений данных факторов [2]. В целях повышения доступности системы она проектируется в виде веб-сервиса.

Целью предлагаемой системы является создание веб-приложения для анализа разнообразных данных в сфере туризма и управления решениями предоставления туристических услуг на основе этого анализа. Основными пользователями системы являются:

- турагентства, сотрудники которых следят за анализом данных и результатами при выборе турпродуктов для потребителей;
- клиенты, которые хотят получить туристический продукт.

Предполагается, что турагентства будут использовать систему для полного процесса предоставления туристических услуг, включающий в себя анализ данных и принятие решений. Исходя из этого, первое, что нужно иметь в виду – это удобство пользования для сотрудников турагентств. Для обеспечения данной характеристики предусматриваются следующие возможности:

- управление турами, новостями;
- агрегирование данных о пользователях: предпочтения в отдыхе, личные данные, предыдущий выбор турпродуктов;

- создание обучающих данных для искусственной нейронной сети и ее обучение;

- определение наиболее подходящих турпродуктов для пользователя путем работы нейронной сети.

Несмотря на то, что первыми пользователями системы предполагаются турагентства, так же не менее важно предусмотреть удобство для клиентов; для этого проектируются следующие функции:

- поиск туров по следующим критериям: даты начала и окончания отдыха, страны, города, цены;

- личный кабинет пользователя;

- возможность купить или забронировать туры, билеты;

- отображение списка стран по регионам;

- информация о стране: описание, климат, температурный режим, советы для путешественников;

- просмотр карты мира;

- возможность оставлять отзывы на туры.

Таким образом, проектируемое программное средство предлагает решение и автоматизацию основных задач, возникающих при принятии решений о выборе турпродукта для клиента при предоставлении туристических услуг.

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. – 2006.

РАСПОЗНАВАНИЕ ДОРОЖНЫХ ЗНАКОВ С ПРЕДВАРИТЕЛЬНОЙ ЛОКАЛИЗАЦИЕЙ НА ИЗОБРАЖЕНИИ

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

Шадраков Р.Н.

Серебряная Л.В. – к.т.н., доцент

В настоящее время в различных сферах жизни идет автоматизация множества процессов с целью уменьшения необходимого для их выполнения времени, рабочей силы и количества совершаемых ошибок. Многие компании разрабатывают системы с использованием искусственного интеллекта для решения широкого класса задач, например, системы автоматического пилотирования, системы безопасности и другие системы, связанные с необходимостью принятия решений. Человек может упустить какую-либо деталь или потерять концентрацию в нужный момент, что может привести к нежелательным последствиям. Программное средство способное в режиме реального времени распознавать дорожные знаки и оповещать о них водителя, может увеличить безопасность участников дорожного движения и снизить количество нарушений ПДД.

При разработке программного средства для распознавания дорожных знаков в режиме реального времени, необходимо обеспечить высокую скорость обработки кадра, например, для достижения скорости обработки 20 кадров в секунду, нужно выполнять процесс распознавание в пределах 50 миллисекунд. Дорожные знаки делятся на несколько категорий, в зависимости от чего имеют разное цветовое оформление и форму. Большинство знаков имеют форму эллипса, прямоугольника и треугольника. Также встречаются знаки особой формы, например, знаки, предупреждающие о переезде через железную дорогу. Основными цветами являются белый, красный, синий, желтый и оранжевый. Так как знаки различных категорий имеют множество отличий, процесс распознавания сразу на целом изображении занимает слишком много времени.

Процесс распознавания дорожного знака на изображении можно разделить на два основных этапа: обнаружение знака и его классификация. С учетом современных вычислительных мощностей, процесс классификации дорожного знака выполняется сравнительно быстро, а основной вычислительной сложностью

обладает его обнаружение на изображении. Среди методов обнаружения объекта на изображении можно выделить:

- использование цветowych фильтров;
- выделение и анализ контуров;
- сопоставление с шаблоном;
- поиск ключевых точек;
- использование методов машинного обучения.

В связи с тем, что заранее неизвестно какой из множества знаков находится на изображении, использование метода сопоставления с шаблоном или поиска ключевых точек нерационально. Большинство алгоритмов обнаружения дорожных знаков используют преобразования Хафа, которое позволяет эффективно находить описанные параметризованными кривыми фигуры на изображении. Однако, данный метод очень чувствителен к качеству изображения и наличию шумов на нем, что приводит к увеличению времени выполнения процесса обнаружения. Следовательно, необходимо предварительно обработать исходное изображение, путем сужения области поиска и удаления посторонних шумов.

Цветовое оформление знака зависит от его категории, а это означает, что нельзя ограничиться простым использованием цветowych фильтров для поиска областей изображения, где может находиться дорожный знак. Однако, большинство знаков имеет выраженный контур, следовательно, можно использовать метод выделения границ, путем поиска мест резкого изменения градиента яркости, после чего проверять найденные линии-границы на соответствие необходимым геометрическим фигурам. В случае, если фигура является допустимой, то можно начать процесс распознавание области изображения, находящейся внутри найденной границы.

Из-за разнообразия признаков у дорожных знаков в разных категориях стоит использовать нейронную сеть, однако для ее обучения понадобится объемная обучающая выборка. Необходимо рассматривать случаи, когда дорожный знак частично закрыт, что приводит к искажению его формы, а также учитывать уровень освещения на изображении в зависимости от времени суток.

Выделение областей с потенциальным дорожным знаком, позволяет сократить время, необходимое на распознавание дорожных знаков на изображении. Для улучшения процесса распознавания знаков при использовании в режиме реального времени стоит добавить механизм отслеживания положения ранее обнаруженного знака на соседних кадрах, чтобы избежать его повторного распознавания.

Список использованных источников:

1. Попов Е.Ю., Крыжановский Д.И. Алгоритм распознавания дорожных знаков ограничения скорости // Современные научные исследования и инновации. 2012. № 6 [Электронный ресурс]. – Режим доступа: <http://web.snauka.ru/issues/2012/06/14717>. – Дата доступа: 11.01.2018.
2. Якимов П.Ю. Предварительная обработка цифровых изображений в системах локализации и распознавания дорожных знаков // Компьютерная оптика. 2013;37(3). – 401-405 с.
3. Якимов П.Ю. Отслеживание дорожных знаков в видеопоследовательности с использованием скорости автомобиля // Компьютерная оптика. 2015;39(5). – 755-800 с.

ГЕОИНФОРМАЦИОННАЯ СИСТЕМА ДЛЯ ОЦЕНКИ УРОВНЯ РАДИОАКТИВНОГО ЗАГРЯЗНЕНИЯ

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

Шамына А.Ю., Ардяко А.Д.

Лапицкая Н.В. – к.т.н., доцент

Сегодня проблема загрязнения окружающей среды радионуклидами стоит достаточно остро. Радиоактивное загрязнение в результате аварии на ЧАЭС является наиболее тяжелой экологической проблемой Беларуси. При этом отсутствуют современные автоматизированные средства информирования населения о радиоактивном загрязнении территории. Также отсутствует средство, позволяющее установить связь между плотностью радиоактивного загрязнения и видами деятельности, безопасных при текущей плотности радиоактивного загрязнения.

Основной целью данной научной работы является создание геоинформационной системы для оценки уровня радиоактивного загрязнения территории РБ, которая может эксплуатироваться на мобильных устройствах. При этом пользователь системы не обязан знать тонкости радиозоологии, изучать тематические карты, анализировать информацию.

В программном средстве предусмотрена реализация целого ряда функций. Одной из ключевых является функция определения динамики загрязнения территории различными радионуклидами по годам. Реализована возможность ввода координат точки различными способами для анализа радиационной обстановки. Исходя из плотности загрязнения и его вида, пользователю предоставляются рекомендации о видах деятельности, которые не рекомендуется производить на данной территории. Также доступен краткий справочник основных терминов и понятий радиационной безопасности.

В дополнение, для большей наглядности, реализовано графическое отображение данных о плотности в виде графиков и на картографической подложке.

Также присутствует функция, позволяющая определять нахождение пользователя на территории, на которой установлен контрольно-пропускной режим. Посещение данных территорий без специального разрешения запрещено и влечет за собой административную ответственность.

Одной из ключевых проблем данной работы является проблема соотношения данных о радиоактивном загрязнении территории с фактическими географическими координатами.

Большинство карт о радиоактивном загрязнении, которые представлены в свободном доступе, являются некачественными отсканированными изображениями с низким разрешением. В процессе поиска подходящего источника внимание было обращено на «Атлас современных и прогнозных аспектов последствий аварии на Чернобыльской АЭС на пострадавших территориях России и Беларуси (АСПА Россия–Беларусь)» [1]. В нем представлены карты фактического, ретроспективного и перспективного радиоактивного загрязнения территорий, пострадавших от аварии на ЧАЭС. Карты в нем сегментированы по областям и изотопам, которые составляют радиоактивное загрязнение. Подобное представление географической информации является неприемлемым для программной работы, поэтому была произведена работа по оцифровке и векторизации этих источников информации о радиоактивном загрязнении.

В связи с тем, что данные, которые определяют радиоактивное загрязнение территории, представлены в прямоугольной системе координат картографической проекции Гаусса-Крюгера, которая основана на системе координат 1942 года, возникла необходимость разработки алгоритма перехода для указанных систем географических координат. При загрузке данных о местоположении с GPS-приемника передача данных о местоположении в программу осуществляется в системе координат WGS-84.

Преобразование географических систем координат WGS84 в СК-42 осуществляется при помощи 7-параметрического преобразования Гельмерта [2].

Программное средство было внедрено в работу ГУ «Республиканский центр по гидрометеорологии, контролю радиоактивного загрязнения и мониторингу окружающей среды», что подтверждается справкой о внедрении. Данную систему может использовать не только население для получения справочной информации, но и специалисты при проведении выездных работ на загрязненной радионуклидами территории.

Следует отметить, что область применения данного программного средства не ограничивается только территорией РБ. Она может распространяться на любые страны и ограничивается лишь информационным наполнением.

Список использованных источников:

1. Атлас современных и прогнозных аспектов последствий аварии на Чернобыльской АЭС на пострадавших территориях России и Беларуси (АСПА Россия–Беларусь) / Под ред. Ю.А. Израэля и И.М. Богдевича. – Москва–Минск: Фонд «Инфосфера»–НИА-Природа, 2009. – 140 с.
2. ГОСТ Р 51794-2001. Аппаратура радионавигационная глобальной навигационной спутниковой системы и глобальной системы позиционирования. Системы координат. Методы преобразований координат определяемых точек [Текст]. – М.: Госстандарт России, 2002. – 11 с.
3. Шамына, А. Ю. Разработка геоинформационной системы для оценки уровня радиоактивного загрязнения территории/ А. Д. Ардяко, А. Ю. Шамына // Управление защитой от чрезвычайных ситуаций: безопасность человека и общества: совершенствование системы реагирования и управления защитой от чрезвычайных ситуаций: сб. материалов I Международной очной научно-практической конференции. - Минск: УГЗ, 2017. - 150 - 151 с.

СХЕМА ВЗАИМОДЕЙСТВИЯ КЛИЕНТА И WEB-СЕРВЕРА

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

Шарков Д.С.

Хмельёва А.В. – к.т.н., доцент

«Клиент-серверная» архитектура является основополагающим принципом работы в сети Интернет. Принцип построения данной архитектуры основан на разделении на два процесса, которые взаимодействуют друг с другом. Как только вы подключаетесь к сети Интернет со своего компьютера или мобильного, или любого другого устройства, имеющего возможность работать с Web-пространством, вы становитесь «клиентом». Предположим, вы зашли на какой-то сайт и хотите скачать с него документ. Нажимая по ссылке с нужным документом, формируется и отправляется запрос на Web-сервер с определенными данными. Web-сервер обрабатывает данный запрос и отправляет ответ на «клиент». Данную схему можно изобразить графически следующим образом:

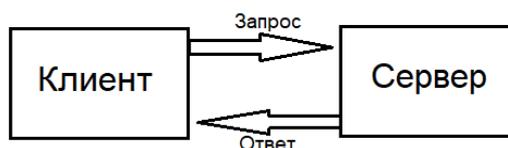


Рис. 1 - Взаимодействие Web-сервера и Web-клиента

Базовой технологией WWW является протокол передачи данных под названием HyperText Transfer Protocol (протокол передачи гипертекста), сокращенно HTTP. HTTP протокол является одной из основополагающих Web. Термин Web был определен Тимом Бернерс-Ли [1] и состоял из трех основных составляющих:

- URI (UniformResourceIdentifier) – путь к уникальному ресурсу в сети.
- HTML (HyperTextMarkupLanguage) – стандартизированный язык разметки.
- HTTP (HyperTextTransferProtocol) – протокол передачи данных в сети интернет.

Работа с документами Web-сервера осуществляется при помощи браузеров, таких как Firefox, Mozilla, IE и другие. При этом браузер становится клиентом по отношению к серверу, формирует запросы в соответствии с правилами и отправляет их серверу. Инициатором запроса всегда является клиент. Сервер может обращаться к другим серверам в процессе обработки запроса. Сервер может отослать свою или чужую страницу HTML, адрес которой запросил клиент.

Стандартизированным языком, при помощи которого происходит общение между клиентом и сервером, является XML (eXtensible Markup Language) – расширяемый язык разметки. Для коммуникации посредством протокола HTTP используется ряд методов [2]. Наиболее используемые из них: GET, POST, PUT, DELETE. Метод GET используется для получения информации или какого-либо ресурса. Взаимодействие при помощи метода POST аналогично использованию метода GET с тем отличием, что при помощи метода POST можно передавать какие-то данные на сервер, т.е. добавляется еще один параметр: «Тело» запроса с данными, передаваемыми на сервер. Метод PUT аналогичен методу POST. При этом метод PUT используется для добавления новых ресурсов на сервер, а метод POST используется для обновления уже существующих данных на сервере. Метод DELETE используется для удаления каких-либо данных с сервера. При использовании любого из методов используется URL (единый указатель ресурса).

При отправке данных на сервер любым методом передаются не только сами данные, введенные пользователем, но и ряд переменных, называемых переменными окружения, характеризующими клиента, историю его работы, пути к файлам и т. п.: REMOTE_ADDR (IP-адрес компьютера, отправляющего запрос), REMOTE_HOST (имя компьютера, с которого отправлен запрос), HTTP_REFERER (адрес страницы, ссылающейся на текущий скрипт), REQUEST_METHOD (метод, который был использован при отправке запроса), QUERY_STRING (информация, находящаяся в URL после знака вопроса), SCRIPT_NAME (виртуальный путь к программе, которая должна выполняться), HTTP_USER_AGENT (информация о браузере, который использует клиент), CONTENT_TYPE (при использовании метода передачи POST в этой переменной хранится строка с названием типа данных, присланных браузером).

Список использованных источников:

1. Википедия [Электронный ресурс]. – Электронные данные. – Режим доступа: https://ru.wikipedia.org/wiki/Всемирная_паутина.
2. Общий принцип работы Интернет приложений [Электронный ресурс]. – Электронные данные. – Режим доступа: http://eos.ibi.spb.ru/umk/5_15/5/5_R1_T1.html.

ВЕБ-ПРИЛОЖЕНИЕ ПО СОЗДАНИЮ И УПРАВЛЕНИЮ ПЛАНОМ ХИМИЧЕСКИХ ЭКСПЕРИМЕНТОВ ДЛЯ РОБОТИЗИРОВАННЫХ СТАНЦИЙ

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

Шевцов Д.О., Шевцова Т.А.

Куликов С.С. – к.т.н., доцент

Разработано веб-ориентированное приложение, написанное на языке Java с использованием Java-фреймворка GoogleWebToolkit. Для реализации пользовательского интерфейса использовали библиотеку Sencha. В качестве локальной базы использовали MongoDB, а централизованной – OracleDB.

Актуальной задачей современной аналитической химии является решение проблем автоматизации методов химического анализа. Возрастающая потребность в полной или, по крайней мере, частичной механизации и автоматизации аналитического процесса напрямую связана с постоянно растущей необходимостью выполнения огромного числа рутинных анализов в самых разных сферах деятельности человека – медицине, фармацевтике, службах контроля окружающей среды, криминалистике и др. [1].

Благодаря непрерывному развитию техники появляются все новые специализированные устройства, способные заменить человека на всех стадиях химического анализа. Важным при автоматизации аналитического процесса и последующем его управлении является установление связи между всеми стадиями анализа, а также между химиком и техническим устройством, которое осуществляется посредством компьютерных программ.

Разработанное приложение призвано решить проблему интеграции современной автоматизированной техники в химические лаборатории и ориентировано на связь между химиком и робототехникой. Подобные

приложения призваны стать удобной платформой, позволяющей легко конфигурировать и проводить химические эксперименты.

Разработанное приложение реализовано при помощи языка Java с использованием Java-фреймворка GoogleWebToolkit. Преимущество веб приложения – независимость от клиентской архитектуры: для работы с приложением достаточно иметь устройство с браузером и доступ в сеть. Сконфигурированные эксперименты, содержащие в себе информацию о разработке экспериментальных методик, подборе условий анализа, а также результаты эксперимента, хранятся централизованно, способствуя возможности интеграции различных лабораторий в единый исследовательский центр.

При создании приложения решался ряд задач, а именно: реализация веб-интерфейса для создания плана химического эксперимента; возможность сохранения и загрузки созданного пользователем химического эксперимента; возможность экспорта в настольное приложение для последующего проведения эксперимента автоматическими роботизированными станциями; возможность импорта эксперимента из настольного приложения с возможностью представления результатов в удобном виде.

В качестве фреймворка использовали GoogleWebToolkit (GWT), который позволяет создавать Ajax-приложения. Основными преимуществами GWT для разработки данного приложения являются встроенный компилятор из Java в JavaScript, простой механизм удалённого вызова процедур (RPC), динамические и многоразовые компоненты пользовательского интерфейса (виджеты), заранее разработанные классы (drag'n'drop и др.), ряд доступных от третьих лиц библиотек (от Google и др.).

В качестве одной из таких библиотек-расширений использовали библиотеку Sencha – это набор стандартных виджетов, панелей и др. инструментов для представления пользовательского интерфейса.

В качестве локального хранилища промежуточных версий экспериментов использована нереляционная база данных MongoDB. На финальной стадии планирования эксперимента (непосредственного перед его выполнением) данные экспортируются в настольное приложение, где и происходит его выполнение. Для хранения финальных версий экспериментов использовали базу данных Oracle DB.

Работа с приложением начинается с авторизации, после её успешного прохождения предоставляется выбор между созданием, загрузкой или копированием существующего эксперимента. При создании нового эксперимента необходимо выбрать рабочее пространство – микропланшет с определённым количеством лунок и объёмом в каждой из них.

Создание плана проведения химического эксперимента происходит в виртуальных микропланшетах. Вкладка «DefineRecipes» содержит выбранные виртуальные микропланшеты с лунками, в которые можно добавлять различные растворы и вещества. Анализ и операции могут быть применены либо ко всему виртуальному микропланшету, например, перемешивание или нагревание, либо к отдельным лункам, например, флуоресцентный анализ. Рисунок 1 демонстрирует добавление терефталевой кислоты к диапазону лунок A1-H1, в качестве способа заполнения выбран градиентный с шагом 10 мкл.

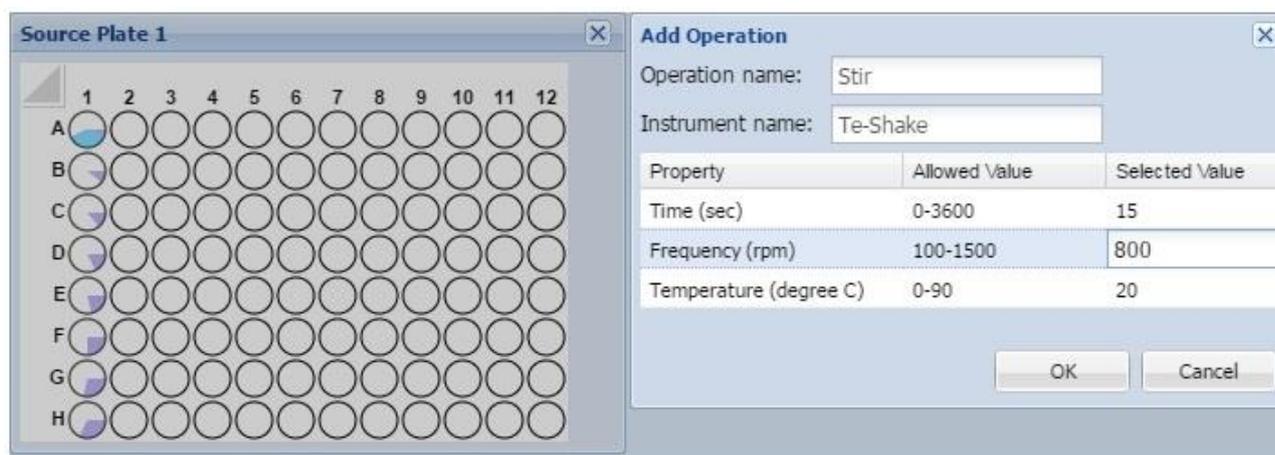


Рисунок 1. Микропланшет и добавление к нему операции «перемешивание» с выбором устройства и параметров проведения.

Вкладка «ExperimentOverview» предоставляет возможность провести обзор эксперимента (рисунок 2).

Experiment Data		Select Materials	Define Mixtures	Select Operations	Define Recipe	Define Analysis	Experiment Overview	Instruction List
Plate Name	Row	Column	Action	Class	Action Content	Amount	Units	
Plate 1	A	1	Add	Reagent	C8H6O4 (Terephthalic acid)	20	µl	
Plate 1	A	1	Add		FeSO4 0.05M	100	µl	
Plate 1	A	1	Operation		Stir			
Plate 1	B	1	Add	Reagent	C8H6O4 (Terephthalic acid)	30	µl	
Plate 1	C	1	Add	Reagent	C8H6O4 (Terephthalic acid)	40	µl	
Plate 1	D	1	Add	Reagent	C8H6O4 (Terephthalic acid)	50	µl	
Plate 1	E	1	Add	Reagent	C8H6O4 (Terephthalic acid)	60	µl	
Plate 1	F	1	Add	Reagent	C8H6O4 (Terephthalic acid)	70	µl	
Plate 1	G	1	Add	Reagent	C8H6O4 (Terephthalic acid)	80	µl	
Plate 1	H	1	Add	Reagent	C8H6O4 (Terephthalic acid)	90	µl	

Рисунок 2. Вкладка «ExperimentOverview»

Последняя вкладка в приложении – «InstructionList» – содержит список последовательных инструкций, которые будут использованы роботами для получения результатов эксперимента. Она предназначена для заключительной проверки правильности созданного экспериментального плана.

После окончательной проверки полное описание и последовательность проведения операций эксперимента сохраняется и экспортируется в лабораторию, в которой он проводится в автоматическом режиме.

Список использованных источников:

- 1.Отто, М. Автоматизация анализа и производственный анализ / М. Отто // Современные методы аналитической химии. – М., 2008. – Гл. 7. – С. 436-478.
- 2.Yuan, L. Automation in new frontiers of bioanalysis: a key for quality and efficiency / Y. Long, J. Qin C // J. Bioanalysis. – 2012. – Vol. 4, № 23. – P. 2759-2762.
- 3.Dunkerley, S. A general robot control system for automatic chemical analysis / S. Dunkerley, M. J. Adams // Laboratory Automation and Information Management – 1997. – Vol. 33, № 2. – P. 93-105.
- 4.North, N. Robotics and Laboratory Automation in Pharmaceuticals Analysis / N. North // Encyclopedia of Analytical Chemistry. – 2006.

ИСПОЛЬЗОВАНИЕ ШТРИХ-КОДОВ ДЛЯ АВТОМАТИЗАЦИИ РАБОТЫ С ТЕСТОВЫМИ ЗАДАНИЯМИ

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

Шендерович В.А.

Деменковец Д.В. – ассистент кафедры ПОИТ

Наряду с компьютерным тестированием, в учебном процессе сохраняет актуальность такая форма контроля знаний, при которой тестовые задания выдаются студентам в печатном виде. Разработанный программный продукт “Генератор тестов” предназначен для автоматизации подготовки тестовых заданий и обработки полученных ответов с применением штрих-кодов. Данное программное средство позволит снизить трудоемкость и повысить качество тестирования, а также систематизировать тестовые задания в базе данных вопросов и ответов.

Программное средство “Генератор тестов” реализует основные функции, необходимые для формирования базы данных вопросов и ответов по учебному предмету с разбивкой по темам и подготовки на этой основе тестовых заданий. Перед формированием тестов пользователь программы осуществляет ввод в базу данных вопросов и ответов. После заполнения базы появляется возможность подготовки, редактирования, сохранения и распечатки тестовых заданий, создаваемых путем случайной выборки вопросов по определенной теме с заданными преподавателем параметрами. В частности, могут варьироваться: число вариантов, количество и сложность вопросов, а также повторяемость вопросов в различных вариантах.

На данном этапе разработки “Генератор тестов” обеспечивает следующие функции:

- добавление в базу новых тем, переименование и удаление существующих;
- добавление в существующие темы новых вопросов, редактирование и удаление существующих вопросов;
- добавление к существующим вопросам новых ответов, редактирование и удаление существующих ответов;
- создание и сохранение случайно сгенерированных тематических тестов с заданием количества вариантов в тесте и количества вопросов по каждому уровню сложности, а также удаление уже сохранённых тестов;

- генерация на основе теста готового для печати документа MSWord, содержащего варианты теста и ключи к ним, а также штрих-коды, содержащие идентификаторы соответствующих вариантов и теста;
 - загрузка ранее созданных тестов путем выбора их из списка, указания идентификатора теста или указания идентификатора одного из вариантов теста, в том числе в автоматическом режиме с использованием сканера штрих-кодов;
 - сохранение всех данных в базе данных, ее экспорт, импорт и полная очистка.
- Пример работы программы представлен на рисунке 1.

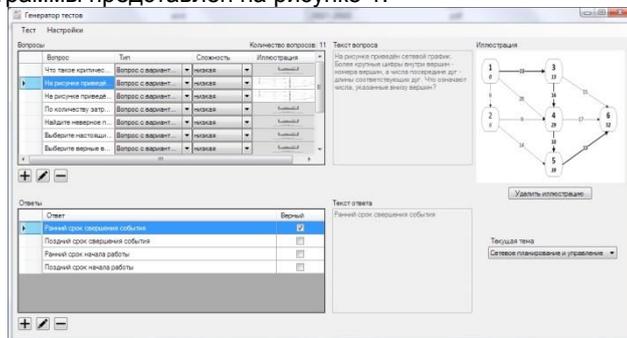


Рис. 1 – Программное средство “Генератор тестов”

В дальнейшем предполагается значительное расширение функциональных возможностей рассматриваемого программного продукта на базе его ключевой особенности - применения штрих-кодов, что позволяет автоматизировать работу с ранее напечатанными тестовыми заданиями.

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://elilb.osu.ru/bitstream/123456789/175/1/2681-2685.pdf>.

ПРОГРАММНОЕ СРЕДСТВО «РАСПИСАНИЕ ЗАНЯТИЙ БГУИР» ДЛЯ ПЛАТФОРМЫ ANDROID

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

Шишея И.Ю.

Данилова Г.В. – м.т.н., ассистент

В настоящее время информационные технологии всё сильнее интегрируются в жизнь большинства пользователей, многие вещи теперь намного удобнее делать с помощью мобильных и десктопных приложений. Появилось много программных средств под различные мобильные устройства, например, всевозможные приложения электронных расписаний. Большинство существующих приложений созданы на основе клиент-серверной архитектуры. Существует несколько приложений расписания занятий для студентов БГУИР'a. Текущие приложения лишены возможности своевременного оповещения учащихся об изменениях в расписании.

Электронное расписание имеет очень много плюсов по сравнению с обычными бумажными расписаниями:

1. Все данные, которые могут понадобиться, находятся в одном месте и легкодоступны. Пользователю не нужно тратить время на то, чтобы добраться до места, где находится расписание, всю информацию можно получить сразу, имея только доступ к компьютеру. Это играет большую роль, потому что студент, скорее всего, не захочет тратить много времени.
 2. Пользователь может в реальном времени получать актуальные данные. Вся информация является актуальной на тот момент, когда пользователь решит воспользоваться расписанием.
 3. Вся информация обновляется и добавляется централизованно. Все действия предоставляемой с информацией производятся из одного источника, это исключает все возможные неточности в предоставляемой информации. Также программное средство становится единственным источником данных.
- Существует множество таких программных средств, реализованных на различных платформах. Сейчас многие расписания доступны на специальных сайтах, например, расписание городского транспорта. Но не всегда у пользователя есть доступ к браузеру. К тому же, не всегда мобильные версии таких сайтов удобны в

использовании. Поэтому появились мобильные версии расписаний. Практически у каждого человека есть телефон на базе операционных систем Android или iOS. Мобильные приложения разрабатываются с учетом специфики конкретной мобильной платформы, поэтому удобны в использовании. Пользователи могут в пару кликов зайти в приложение и узнать всю необходимую информацию, даже если у них отсутствует подключение к сети интернет, так как зачастую используется локальное хранилище данных, что делает приложение независимым от каких-либо внешних условий.

В основе таких мобильных приложений лежит клиент-серверная архитектура. Основная идея заключается в том, что приложение формируется из двух больших частей: клиентской части (в данном случае ее роль играет само мобильное приложение) и серверной. Серверная часть отвечает за предоставление программного интерфейса к данным, которые будет использовать клиентская часть. Также она содержит базу данных, которая хранит всю требуемую информацию. Задача клиентской части состоит в том, чтобы взаимодействовать с серверной частью, получать требуемую информацию и корректно ее отображать пользователям. Также она может содержать свою собственную локальную базу данных, чтобы в случае отсутствия подключения к серверу дать пользователю возможность использовать приложение в полной мере в оффлайн режиме. Приложение также может предоставлять пользователю функции push-уведомлений для информирования пользователя о каких-либо нововведениях или событиях, которые могут возникать. Клиент-серверное взаимодействие на платформе Android осуществляется при помощи библиотеки Dagger2. Она позволяет легко реализовать REST – архитектурный стиль взаимодействия компонентов распределенного приложения в сети.

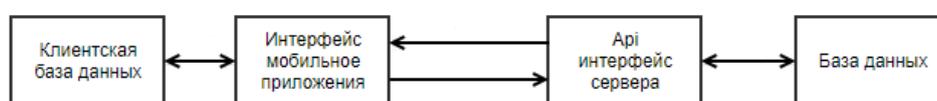


Рис. 1 – Схема взаимодействия приложения с серверной частью

Таким образом, описанное программное средство позволит облегчить взаимодействие студентов с расписанием и другой полезной информацией об университете. Большинство студентов использует телефоны на операционной системе Android, что позволит охватить большое количество пользователей.

Список использованных источников:

1. Медникс З., Дорнин Л., Мик Б., Накамура М. П78 Программирование под Android. 2-е изд. – СПб.: Питер, 2013. – 560 с.
2. П. Дейтел, Х. Дейтел, Э. Дейтел, М. Моргано Android для программистов: создаём приложения. – СПб.: Питер, 2013. – 560 с.: ил.

ПРИМЕНЕНИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ В УЧЕБНОМ ПРОЦЕССЕ ВЫСШИХ УЧЕБНЫХ ЗАВЕДЕНИЙ

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

Шульга Е.С.

Бранцевич П. Ю. – к.т.н., доцент

Рассмотрены особенности внедрения новейших информационных технологий в высшее образование, что приводит к облегчению выполнения рутинных действий участниками учебного процесса, а также дает возможность использовать прогностические и оптимизационные математические модели.

В современном мире продолжают процессы повсеместного внедрения новейшего аппаратного обеспечения и программных средств во все сферы человеческой жизнедеятельности. Университеты не могут оставаться в стороне от данных процессов. Информатизация и компьютеризация осуществляются как формальным образом: действиями управляющих структур, – так и неформальным, когда субъекты образовательного процесса начинают использовать некоторый набор аппаратно-программных средств с единственной целью: облегчить выполнение рутинных задач, переложить их осуществление на компьютеры и алгоритмы. Можно зафиксировать некоторое отставание административного аппарата от деятельности отдельных людей, но это и объяснимо: субъектам учебного процесса необходимо решать свои насущные проблемы, что они и делают наиболее быстрым и удобным образом, в то время как управленческие структуры начинают внедрение каких-либо средств только после рассмотрения возможных вариантов и долговременной апробации выбранного решения.

Результаты информатизации образования могут внедряться как в качестве сопутствующих для традиционного процесса обучения, так и в качестве новейших дидактических средств. В рамках разработки одного из вопросов дидактики: отношения «студент – учебный материал», – предлагаются новые обучающие средства, которые не требуют присутствия преподавателя. Информационно-компьютерные технологии бросают вызов традиционной педагогике. Прежде всего это обнаруживается в том, что создание программного обеспечения для учебного процесса является алгоритмизацией деятельности педагога. И

данный процесс может завершиться успешно, ведь преподаватели не являются владельцами знания, но только хранителями и ретрансляторами, а значит данные функции можно переложить на машины и алгоритмы [1].

Но данный процесс еще далек от завершения. В вузах нашей страны по-прежнему широко применяется традиционный процесс обучения, который основывается на отношениях преподавателей и студентов. Процесс образования по своей сути пока что остаётся процессом взаимодействия человека с человеком. Пока не завершена информационная революция, в текущем переходном периоде интерес представляют задачи облегчения рутинных действий, которые вынуждены выполнять участники учебного процесса.

Современное высшее образование характеризуется массовостью. Проблема управления его участниками порождает значительные информационные потоки. Сбор, обработка информации являются необходимыми составляющими работы со студентами. Каждое подразделение, каждый сотрудник вуза обрабатывает свою часть этого массива данных.

Ограничимся проблемами, с которыми сталкиваются в своей работе преподаватели. Основная часть информации, порождаемая процессом преподавания, относится к проверяющему и контролирующему блоку работы. Зачастую сбор и обработка данной информации производится в аналоговой форме, то есть без использования специальных устройств или программного обеспечения. Оцифровка же этой информации способствует повышению удобства доступа, пользования ею, ее сохранности, долговечности. Кроме того, к собранной информации могут применяться математические модели, результаты которых можно использовать для прогнозирования характеристик работы и их оптимизации. Например, в работе [2] рассматривается деятельность преподавателя с точки зрения систем массового обслуживания. Такое моделирование возможно только при наличии исходных данных, например, таких как количество студентов в группе, время приема работы у одного студента, количество проверенных работ за одну пару, статистика принятых и отклонённых работ. Результатами применения модели могут быть прогнозы о том, успеет ли проверить преподаватель работу конкретного студента, потребуются ли дополнительные занятия. Кроме этого, по каждому студенту можно осуществлять сбор информации об успешности выполнения заданий, что может быть использовано при ранжировании студентов в рейтинге, при оценивании преподавателем работы в семестре на экзамене.

Таким образом, развитие информационных технологий оказывает влияние на эволюцию высшего образования, причем как в области непосредственно образовательных средств, так и в области сопутствующих процессу обучения технологий. Современная белорусская высшая школа несколько запаздывает с их внедрением. Однако их развертывание должно значительно облегчить управление потоками данных, особенно для преподавателей. Одним из результатов их внедрения является возможность построения различных проективных, прогностических и оптимизационных моделей.

Список использованных источников:

1. Гусаковский, М. А. Университет как центр культуропорождающего образования. Изменение форм коммуникации в учебном процессе / М. А. Гусаковский, Л. А. Яценко, С. В. Костюкевич. Минск: БГУ, 2004.
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.

WEB-СЕРВИС КАК СРЕДСТВО УПРОЩЕНИЯ ОБМЕНА ИНФОРМАЦИЕЙ МЕЖДУ РАЗНЫМИ ПЛОЩАДКАМИ

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

Медведев С.А., Юревич Д.Ю.

Медведев С.А. – к.т.н., доцент

Web-сервис (служба) – программа, которая организывает взаимодействие между сайтами. Информация с одного портала передается на другой. Информация в интернете разнородна. Сайты управляются разными системами. Используются разные протоколы передачи и шифрования. Веб-сервисы упрощают обмен информацией между разными площадками.

Наглядным примером интеграции веб-сервисов служит запуск ОАО «Авиакомпания «Белавиа» приложения OnlineTimeTable. OnlineTimeTable – программной системы, основанной на технологии Web-service и предоставляющей всем заинтересованным лицам простой и эффективный способ получения оперативной информации о прилёте / вылете и состоянии рейсов, выполняемых ОАО «Авиакомпания «Белавиа». Эта информация легко может быть размещена на любых сайтах, а также интегрирована с приложениями [1].

Веб-служба - программная система, идентифицируемая строкой URI, чьи общедоступные интерфейсы определены на языке XML. Описание этой программной системы может быть найдено другими программными системами, которые могут взаимодействовать с ней согласно этому описанию посредством сообщений, основанных на XML и передаваемых с помощью интернет протоколов.

Используемые стандарты [1]:

- XML: Расширяемый язык разметки, предназначенный для хранения и передачи структурированных данных;

- SOAP: Протокол обмена сообщениями на базе XML;

- WSDL: Язык описания внешних интерфейсов веб-службы на базе XML.

Преимущества Web-сервисов [2]:

- создание необходимых условий для взаимодействия программных компонентов вне зависимости от платформы;

- Веб-сервисы основываются на открытых стандартных протоколах и за счет внедрения XML обеспечивается простота формирования и настройки веб-сервисов;

- применение HTTP гарантирует взаимодействие систем посредством межсетевого доступа.

Недостатки Web-сервисов [2]:

- невысокая производительность и большой объем трафика в сравнении с системами RMI, CORBA, DCOM за счет использования XML-сообщений в разрезе текста;

- неопределенный уровень безопасности, поскольку все современные веб-сервисы должны внедрять кодирование и требовать авторизации пользователя. Хватит ли здесь наличия HTTPS или необходимы более надежные протоколы, как XML Encryption, SAML и т.д., – решается в ходе разработки.

Веб-сервисы могут использоваться во многих сферах [3]:

1) B2B-транзакции.

Интеграция процессов идет сразу, без участия людей. Например, пополнение каталога интернет-магазина новыми товарами. Их привозят на склад, и кладовщик отмечает в базе данных приход. Автоматически информация передается в интернет-магазин. И покупатель вместо пометки “Нет на складе” на карточке товара видит его количество.

2) Интеграция сервисов предприятий.

Если в компании используются корпоративные программы, то веб-сервис поможет настроить их совместную работу.

3) Создание системы клиент-сервер.

Сервисы используются, чтобы настроить работу клиента и сервера. Это дает преимущества:

- можно продавать не само программное обеспечение, а делать платным доступ к веб-сервису;

- легче решать проблемы с использованием стороннего ПО;

- проще организовывать доступ к контенту и материалам сервера.

Успешная интеграция Web-сервисов на различных площадках позволит существенно упростить обмен информацией между ними.

Список использованных источников:

1. <https://belavia.by/webservice/>.

2. Э. Ньюкомер., Веб-сервисы. Для профессионалов. Июнь 2003 г. – 256с.

3. <https://semantica.in/blog/chto-takoe-veb-servis.html>.

СЕКЦИЯ «ВЫСШАЯ МАТЕМАТИКА»

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ ОБЪЁМНЫХ ТЕЛ И ПОВЕРХНОСТЕЙ

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

Крук М.И., Юркевич А.С., Яковлевич П.О.

Баркова Е.А. – канд. физ-мат. наук, доцент

Разбор и анализ различных методов визуализации трехмерных поверхностей средствами информационных технологий сегодня является актуальной проблемой.

На практике и в производстве используются поверхности 2-го порядка, такие как: сфера (небесные тела в астрономии), однополостный гиперболоид (используется в строительной технике), цилиндрические поверхности (трубы, тоннели, цистерны), конические поверхности (всевозможные, конические купола, воронки и т.д.). При создании, телескопов, радио и телеантенн используется параболоид вращения.

Архитекторы и дизайнеры используют поверхности второго порядка при создании своих объектов. Это и фонтаны разных форм, и дизайнерские интерьеры, и архитектурные решения. При строительстве водонапорных башен, маяков, телевизионных мачт и других сооружений находят применение прямолинейные образующие.

В медицине широко используются тела вращения. Посмотрите хотя бы на бионические протезы. И робототехника при создании очередного человекоподобного механизма не обходится без таких тел. Такое широкое применение обусловлено особыми свойствами тел вращения: тут и специфическое отражение лучей света, и обтекаемость, и гладкость поверхности, и просто эстетическое удовлетворение.

На данный момент, основным средством, позволяющим не только строить трехмерные поверхности, но и автоматизировать различные расчеты являются математические пакеты.

В своей работе мы использовали созданный ранее математический пакет, основанный на симбиозе уже существующих методов: метода сетки полигонов и метода триангуляции (рис1).

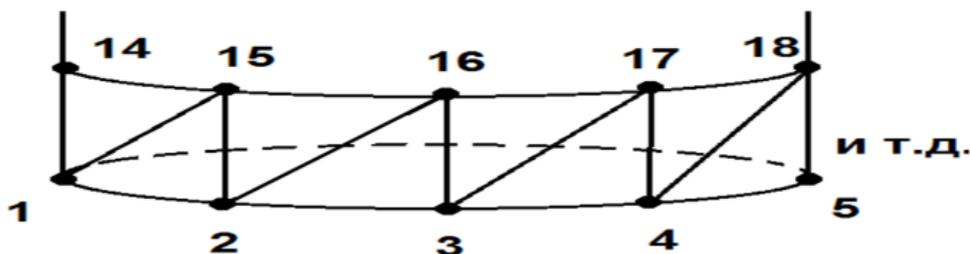


Рис 1.

С помощью полигонов отображают кривые линии, поверхности или тела. Полигоны, принадлежащие одному геометрическому объекту, для удобства объединяют в сетки. Полигональная сетка является совокупностью полигонов для отображения некоторого геометрического объекта. Она состоит из полигонов поверхностей его граней. Представление геометрических объектов полигонами является одним из наиболее рациональных методов для их отображения.

Метод триангуляции заключается в аппроксимации треугольными пластинами поверхности моделируемого объекта, отстоящими от нее на расстоянии, не превосходящем некоторой заданной величины. Все треугольные пластины должны стыковаться между собой, а их вершины лежат на поверхности. С набором треугольных пластин легче работать, чем с поверхностью общего вида. Таким образом триангуляция тела представляет собой совокупность треугольников, которые получены путем триангуляции поверхностей его граней.

По такому плану моделируются и тела вращения, описанные в разработанном нами математическом пакете.

Программа способна моделировать широкий спектр фигур, среди которых не только шар, конус, эллипсоид, однополостный гиперболоид, эллиптический цилиндр, и эллиптический параболоид, но также и гиперболоический параболоид, лента Мебиуса, тор, геликоид, бутылка Клейна, в том числе в форме «фигуры 8» (рис 2.).

Также программа способна моделировать пересечение поверхностей (или тел) с плоскостью. Пользовательский режим предоставляет возможность моделирования тел с исходным уравнением, настройку режима вращения тел: автоматическое или ручное. Помимо этого, предусмотрены статические положения тела (проекция). Возможен выбор цвета фона и поверхности тела, а также изменение скоростивращения.

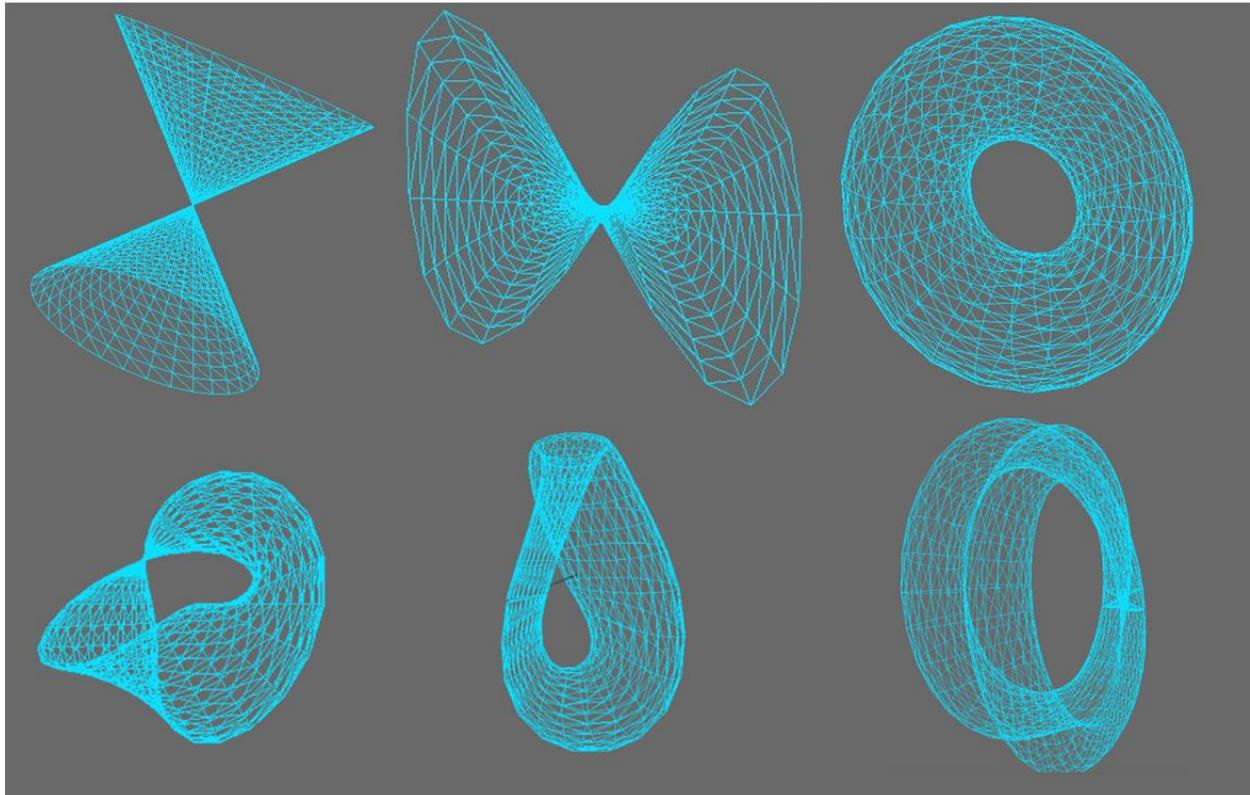


Рис. 2.

Автоматизированное построение различных тел и поверхностей позволяет нам рассмотреть с разных сторон тот или иной 3D объект, расширить знания, изучить более точно его свойства. Наша программа будет особенно полезна студентам и преподавателям, для визуализации трёхмерных объектов в курсе аналитической геометрии и различных приложений теории интегрирования в курсе математического анализа. В дальнейшем мы планируем расширять возможности нашей разработки, как с математической, так и с пользовательской стороны.

Список использованных источников:

1. DirectX 9 уроки программирования на C++, Горнаков С. Г.;
2. Конспект лекций по высшей математике: Письменный Д.;
3. Технология моделирования тел вращения в пространстве / В. В. Фесько, А. С. Орлова // Современные проблемы математики и вычислительной техники : сборник материалов X Республиканской научной конференции молодых ученых и студентов, 24-24 ноября 2017 года. – Брест: БГТУ, 2017. - С. 62-65.

О ЗАДАЧЕ ИОСИФА ФЛАВИЯ

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

Карамач Н.А., Кот З.К.

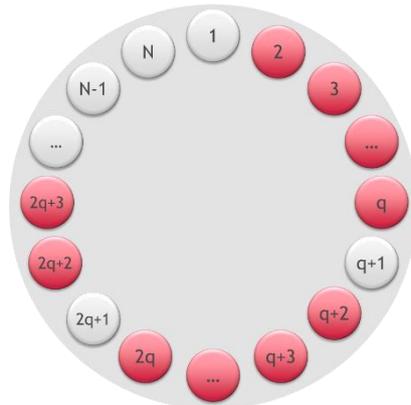
Баркова Е. А. – кандидат физ.мат.наук, доцент

Давным-давно, примерно в 1 век н.э., во времена Иудейской войны один ничем не примечательный отряд евреев был загнан римлянами в пещеру с узким входом. Пленникам было предложено сдаться, но вместо этого они решили умереть, не сдавшись врагу: встав в круг, воины договорились, что каждые два воина будут убивать третьего, пока не погибнут все. В составе такого отряда оказался Иосиф Флавий – хитрый историк и военачальник. Он быстро высчитал необходимые для себя и своего товарища места для того, чтобы выжить и сдаться в плен. Так и возникла знаменитая задача Иосифа Флавия, смысл которой заключается в зачеркивании чисел, расставленных по кругу.

Заинтересовавшись задачей Иосифа Флавия, мы решили изменить постановку задачи, решить её и на основании этого получить общий метод решения подобных задач.

Условие новой задачи:

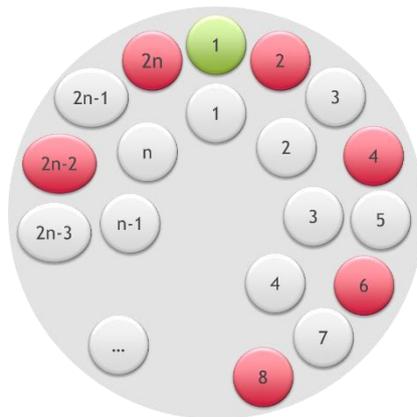
В круге расставлены N чисел. Проходя по кругу, 1-е число пропускается, вычеркиваются числа $2, 4, 6, \dots$. Число 1 остается нетронутым. Далее исключаются числа $3, 5, 7, \dots$ и так далее.



Для решения задача разбивается на несколько этапов:
 Анализ частных случаев с четным и нечетным количеством чисел, расставленных по кругу;
 Вывод рекуррентных соотношений;
 Обобщение полученных соотношений и вывод частной формулы.

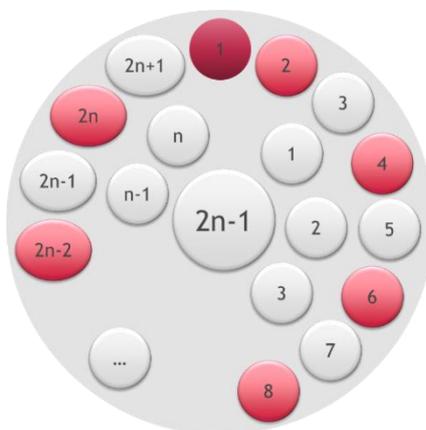
Четный случай:

Допустим, что изначально в круге имеется $2n$ чисел. После первого прохода по кругу остаются числа с номерами $1, 3, 5, \dots, 2n-3, 2n-1$. Следующий проход по кругу будет начинаться с номера 1 . Видно, что в круге осталось n чисел, каждое i -е из которых равно $2i-1$. На основе этого выводится рекуррентное соотношение для четного случая:



Нечетный случай

Теперь в круге расставлено $2n+1$ чисел. Поначалу, все происходит по подобному сценарию. Однако, в этом случае число 1 также вычеркивается для того, чтобы получить ситуацию с n числами. На этот раз рекуррентное соотношение имеет вид



Рекуррентные соотношения

Объединяя получившиеся соотношения с уравнением, получаем следующую систему:

Частная формула

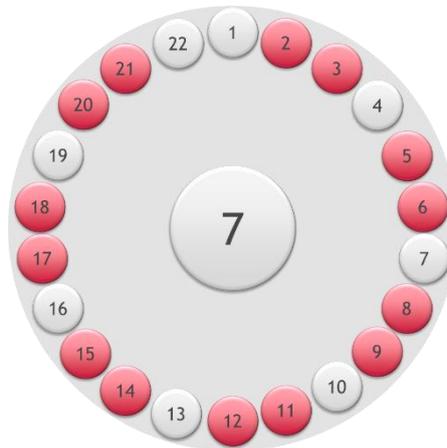
Рекуррентное соотношение, полученное ранее, позволяет составить таблицу первых значений (Рисунок – 4). Несложно заметить, что если сгруппировать значения по степеням 2-ки (выделено цветом), то в каждой группе всегда будет начинаться с 1, а затем увеличиваться на 2.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	1	1	3	1	3	5	7	1	3	5	7	9	11	13	15	1

Итак, запишем в виде, где – наибольшая степень 2-ки, непревосходящая, а – то, что остается. Тогда решением полученной системы рекуррентных соотношений будет следующим:

Условие второй задачи

Рассмотрим ситуацию, когда в круге стоят натуральные числа от 1 до n , вычеркнем числа 2,3, пропустим число 4 и вычеркнем следующие два числа и т.д. Будем так вычеркивать до тех пор, пока не останется одно число. Это число и будет решением задачи (Рисунок - 4).



Частный случай

Расставим в круге соответственно 6, 7, 8 чисел. Чтобы число n осталось после первого круга, оно должно иметь вид (т.е. при делении на три давать остаток 1). Мы делим на тройки чисел, два из которых вычеркиваем, значит после удаления остается число, т.к. удаляется чисел.

Чтобы число n осталось невычеркнутым и после второго круга, нужно чтобы делилось на 3. Предположим, что после второго круга остается чисел (т.к. вычеркиваются чисел). Значит тогда должно делиться на 3 и так далее, пока после очередного круга не останется меньше 3 чисел и тогда наверняка число n будет вычеркнуто последним. Значит, общая формула для n чисел будет выглядеть следующим образом:

Итак, чтобы число n было вычеркнуто последним, оно должно быть представлено в виде двух слагаемых, одно из которых степень тройки, а другое некоторое четное число.

Частное решение

Представим в виде. Множитель g нам необходим, чтобы представить некоторые числа. Например, если мы возьмем число 40, выделим максимальную степень тройки - 27, получим что, откуда четное число. Также, можно заметить, что – это количество чисел, оставшихся после зачеркивания полного количества кругов (Можно использовать формулу: Итак, если представлено в виде, то решением всегда будет являться число 1. Это и будет основа нашего рекуррентного соотношения.

Решение задачи сведением к случаю, рассмотренному ранее

Рассмотрим пример, когда в круге стоят 17 чисел (Рисунок – 5). Проследим цепочку вычеркивания чисел. После того, как мы вычеркнем первую пару чисел в круге, останется 15 чисел. Следующий шаг—

вычеркиваем еще 2 числа. В круге остается 13 чисел, вычеркиваем следующую пару и т.д., пока в круге не останется 9 чисел. В нашем случае, как только в круге осталось 9 чисел, начертим второй круг и пронумеруем числа снова от 1 до 9. А результат мы знаем заранее: он равен 1. Так как 1 стоит напротив числа 13, то оно и является решением. Такое решение основано на понятии о рекуррентном соотношении, когда последующий член выражается через предыдущий. В нашем случае решение более сложного выражается через решение более легкого. Это помогает быстро найти решение для любого количества чисел.

Итак, решением задачи является число 13. С одной стороны, где $l(4)$ — количество пар чисел, которые были вычеркнуты до тех пор, пока в круге не осталось 9 чисел. С другой стороны, .



Запишем формулу в общем виде:

Закключение

Итак, мы получили формулу, по которой определяется место числа, вычёркиваемого последним в случае, когда вычеркивается каждое второе число - , а также формулу для случая, когда подряд вычеркивается два числа - . Способ, который мы применили для решения данной задачи, позволяет вывести общую формулу для случая, когда подряд вычеркивается любое количество чисел - q:

ГИПОТЕЗА РИМАНА

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

Паньков В.А., Чадович В.А.

Борисенко О.Ф. – к.ф.м.н., доцент

В 2000-м году Математический институт им. Клея обозначил список из семи математических задач, решение которых не было найдено на протяжении многих лет. Единственной решенной из этого списка задачей остается гипотеза Пуанкаре, доказанная еще в 2003г. Мы бы хотели рассмотреть другую задачу - гипотезу Римана. Гипотеза была сформулирована в 1859 году немецким математиком Бернхардом Риманом и имеет богатую историю, однако так и не была доказана или опровергнута. Более того, до сих пор нет ни намёка на то, чтобы установить её истинность или ложность.

История этой гипотезы берёт свои корни из так называемой Базельской задачи, сформулированной задолго до Римана. Вопрос ставился следующим образом: «Выразить в замкнутом виде бесконечный ряд обратных квадратов». На тот момент уже было известно, что гармонический ряд расходится, а в случае ряда обратных квадратов каждый новый член ряда значительно меньше предыдущего, поэтому «базельский ряд» заинтересовал математиков. Задача была решена в 1735 году Леонардом Эйлером. Потрясающий ответ имел вид $\pi^2/6$. Эйлерово решение дало не только замкнутое выражение для ряда из обратных квадратов, но в качестве побочного продукта еще и замкнутые выражения для многих рядов с чётными степенями знаменателя. Однако для нечётных степеней результат до сих пор неизвестен. Таким образом, мы подошли к ряду:

$$1 + \frac{1}{2^N} + \frac{1}{3^N} + \frac{1}{4^N} + \frac{1}{5^N} + \dots$$

До этого момента N рассматривали только в качестве натурального числа, однако Риман задался вопросом, что если N – комплексное число. Таким образом, Риман сформулировал свою дзета-функцию:

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}$$

Гипотеза Римана гласит: все нетривиальные нули дзета-функции имеют вещественную часть, равную одной второй.

Чтобы разобраться в формулировке, стоит уточнить, что тривиальные нули дзета-функции – это все отрицательные четные целые значения аргумента (т.е. -2, -4, -6 и т.д.). Изначально данная функция рассматривалась на промежутке, где реальная часть больше единицы, а ряд, представленный функцией, сходится. Однако функция допускает аналитическое продолжение на другую половину комплексной плоскости без единицы (т.к. когда реальная часть равна единице, ряд - гармонический). Именно благодаря аналитическому продолжению функции возможно рассматривать её поведение при подаче отрицательных аргументов. Так были найдены тривиальные нули, а те нули, которые не удалось найти этим способом, стали называть нетривиальными.

Вопрос распределения нулей дзета-функции напрямую связан с темой распределения простых чисел. Еще в первой половине XVIII века Л. Эйлер установил связь между простыми числами и вышеупомянутым рядом, являющимся частным случаем дзета-функции реальной частью больше единицы и мнимой равной нулю.

$$\zeta(s) = \prod_p (1 - p^{-s})^{-1}$$

Это является одним из основных свойств дзета-функции, связывающей её с простыми числами. Стоит отметить, что Риман установил, что данное свойство работает для всех s , у которых $\text{Re}(s) > 1$.

Есть множество оснований считать, что распределение нетривиальных нулей дзета-функции напрямую связано с распределением простых чисел на некотором отрезке числовой прямой. Ярким примером служит теорема о распределении простых чисел. Одна из ее наиболее важных формулировок имеет вид:

$$J(x) = \text{Li}(x) - \sum_{\rho} \text{Li}(x^{\rho}) - \ln 2 + \int_x^{\infty} \frac{dt}{t(t^2 - 1) \ln t}$$

И она верна лишь в том случае, если верна гипотеза Римана.

Достоверно известно только одно: все нетривиальные нули расположены в полосе, где реальная часть принимает значения от нуля до единицы. Гипотеза Римана лишь уточняет это свойство. В свое время Риман активно искал нетривиальные нули на т.н. критической полосе, где реальная часть равно одной второй, и успешно нашел некоторые из них. До сих пор создаются алгоритмы, призванные просеивать миллионы возможных вариантов аргументов с этой полосы. Однако математического доказательства до сих пор нет.

Математическое сообщество разбилось на два лагеря: кто-то активно ищет доказательство достоверности Гипотезы, а кто-то – способ её опровергнуть. Несмотря на это множество работ и математических теорий основываются на том, что Гипотеза верна. Даже не зная, верна гипотеза или нет, математики все равно продолжают писать статьи, основывающиеся на её истинности. Вот почему гипотеза Римана является одной из самых важных проблем математики – проблемой тысячелетия.

Список использованных источников:

1. Джон Дербшир. Простая одержимость. Бернхард Риман и величайшая нерешенная проблема математики, 2003.
2. Простое число. Математическая энциклопедия (в 5 томах). — М.: Советская Энциклопедия, 1977. — Т. 4.

МАТЕМАТИКО-СИНТАКСИЧЕСКАЯ ТЕОРИЯ КАК СРЕДСТВО ОПТИМИЗАЦИИ АЛГОРИТМОВ РАБОТЫ С ЕСТЕСТВЕННЫМИ ЯЗЫКАМИ

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

Щуцкий Н.А.

Борисенко О.Ф. - кандидат физ.-матем.наук, доцент

В данной научной работе были исследованы процессы и результаты, относящиеся к генерации текста посредством цепей Маркова первого порядка и рекуррентных нейронных сетей. Марковские цепи позволяют установить статистическую закономерность связи тех или иных слов, в результате имея алгоритм, способный к использованию только уже бывших в статистике паттернов. В то же время нейронная сеть на основе статистики позволяет установить закономерности построения предложений, то есть частично восстановить синтаксические правила языка, что позволяет ей работать значительно эффективнее на новых данных. На основе данного исследования было решено заняться модификацией данного процесса, что в последствии привело к появлению математико-синтаксической теории.

Основные постулаты, справедливые в контексте данной теории таковы:

1. Слово - некоторым образом закодированная информация, представимая в виде набора чисел.
2. Лексика - группа лексических аргументов(слов), для которой справедливы аксиомы теории групп.
3. Синтаксис языка – множество разрешённых в данном языке операций склеивания.
4. При склеивании нескольких лексических аргументов происходит объединение в один лексический аргумент, не представимый словом, при это порядок аргумента повышается на единицу.
5. Порядок аргумента – число операций склеивания, уже произведённых над данным аргументом.

Для простоты применения данной теории введём абстрактный синтаксис, включающий 4 части речи:

1. Объекты – лексические аргументы, являющиеся некоторыми сущностями или телами.
2. Действие – лексические аргументы, являющиеся взаимодействиями объектов.
3. Свойства – лексические аргументы, описывающие параметры объектов или действий.
4. Конструкции – лексические аргументы, связывающие между собой все остальные части речи.

Множество допустимых операций склеивания может быть определено рядом правил:

1. Аргументы 0-ого порядка одной части речи склеиваются в аргумент той же части речи.
2. Аргументы 1-ого порядка склеиваются в объект или действие 2-ого порядка.
3. Аргументы 2-ого порядка склеиваются в аргумент 3-его порядка – простое предложение.
4. Аргументы 3-его порядка склеиваются в аргумент 4 порядка – сложное предложение.
5. Аргументы 3-его и 4-ого порядков склеиваются в аргумент 5 порядка – текст.

Оптимизация формирования синтаксиса на нейронных сетях происходит в 3 этапа обучения:

1. Заменяем лексические аргументы обучающей выборки на 4 части речи абстрактного языка.
2. На основе данной выборки обучаем нейронную сеть, дополняя стартовый абстрактный синтаксис.
3. Заменяем абстрактные части речи на естественные, проводим частичное переобучение.
4. Проводим обучение, заменяя естественные части речи на слова естественного языка.

В итоге на основе модели синтаксиса конкретного языка могут быть построены синтаксически корректные и связанные тексты, однако их смысловая наполненности всё ещё будет целиком зависеть от некоторой обучающей выборки из конкретных слов, на основе которой будут определены возможные связи. Таким образом задача оптимизации сводится к формированию полного набора операций синтаксиса языка за наименьшее время, что происходит в результате действий, обратных аппроксимации, по отношению к результатам каждой из предыдущих этапов обучения, начиная с теоретически воссозданного абстрактного стартового синтаксиса, что в последствии уменьшает число требуемых вычислений до регулируемого уровня.

Список использованной литературы:

1. Deep learning / Ian Goodfellow, Yoshua Bengio, Aaron Courville – Cambridge, MA: MIT press [2017] – 373p.
2. Mathematical linguistics / Andras Kornai – Cambridge, MA: Springer [2008] – 201p.
3. Contemporary abstract algebra / Joseph A. Gallian – Boston, MA; Brooks Cole [2016] – 49p.

ФРАКТАЛЫ В ПРИРОДЕ

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

Смирнов А.Д., Даукшис В.В.

Новик Ю.Ф. – магистр физико-математических наук

Человек с самого начала своего существования наблюдал за уникальными природными явлениями, такими как молния, облака, горы, снежинки, морозные узоры на окне. У него возникали вопросы, как это устроено, ведь все это выходит за пределы евклидовой геометрии. Невозможно описать камень или облако с помощью треугольников, кружков и

прямых. В таких случаях нам приходят на помощь фракталы.

Говоря простым языком, Фрактал – это сложная геометрическая фигура, обладающая свойством самоподобия.

Фракталы встречаются всюду: начиная от бактерий и заканчивая Вселенной.

В компьютерной графике фракталы применяются для построения изображений природных объектов. Они создаются исходя из математических расчетов, но базовым элементом является математическая формула.

В физике при моделировании нелинейных процессов, а это такие, как турбулентное течение жидкости, облака, пламя и сложные процессы диффузии-адсорбции, также используются фракталы.

В медицине фракталы используются для описания систем внутренних органов. Таким образом, исследования, связанные с фракталами, меняют многое из привычных представлений о нашем мире.

Нами была создана программа, которая строит различные виды фракталов (рис.1а). Данная программа была создана на языке Delphi. Для построения фракталов мы рассматривали алгоритмы рекуррентных формул. Рисование на форме осуществлялось свойством Canvas. Программа способна построить кривую Коха, треугольник Серпинского, множество Мандельброта, драконову ломаную и фрактальное дерево (рис. 1б).

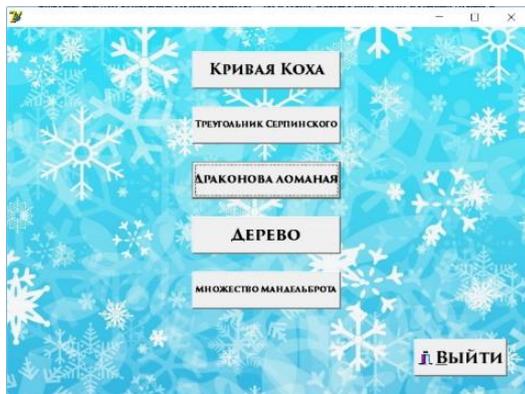


Рис. 1.а – Главное меню

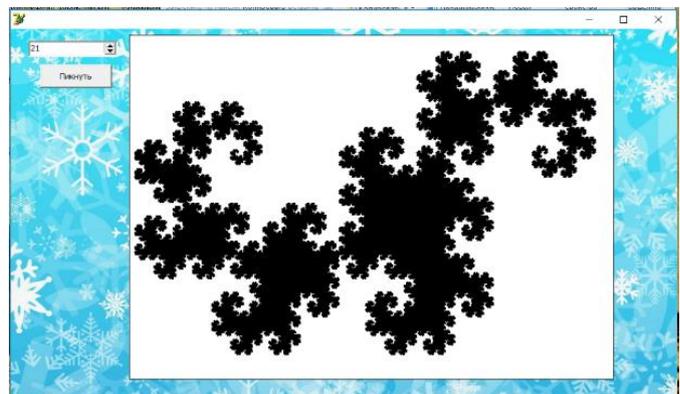


Рис. 1.б – Построение драконовой ломаной

Также в программе пользователь имеет возможность выбрать порядок (возраст) фрактала, чтобы четче понять, как строятся различные виды фракталов. Доступна функция приближения.

В дальнейшем мы планируем расширять возможности программы. Будут добавлены построения новых фрактальных фигур, анимационное построение, а также функции пользовательского построения своего уникального фрактала.

Программа представляет собой учебное пособие для начинающих программистов, гейм-дизайнеров, а также для углубленного изучения в области математики и физики.

Список использованных источников:

1. Фрактальная геометрия природы: Б. Мандельброт.

СЕКЦИЯ «ФИЗИКА»

АНАЛИЗ МЕТОДОМ СКАНИРУЮЩЕЙ ЗОНДОВОЙ МИКРОСКОПИИ МИКРОРЕЛЬЕФА ПОКРЫТИЙ НА СТЕКЛЕ, НАНЕСЕННЫХ ИОННО-АССИСТИРОВАННЫМ ОСАЖДЕНИЕМ

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

Богдевич П.С., Мойсейчик Е.С., Пшеничный Д.С., Холупко И.С.

Ташлыкова-Бушкевич И.И. – к.ф.-м.н., доцент

В настоящей работе были проведены исследования поверхностей тонких пленок сплава Al-1.0 ат. % Cr, полученных методом ионно-ассистированного осаждения. В процессе работы был разработан метод исследования данных образцов инструментами АСМ и алгоритм анализа конечных данных. Разработанный алгоритм был успешно применен для анализа изменения топографии поверхности образцов в зависимости от условий их получения.

Изучение структуры поверхности играет важную роль в понимании механизмов формирования свойств металлов и их сплавов. Так как поверхностный слой металлов имеет особое физико-химическое состояние, то его свойства (макро- и микроскопические) могут иметь значительные отличия в сравнении со свойствами его объемного образца. В связи с этим, всё больший интерес проявляется к исследованиям поверхностных свойств материалов, практическая значимость которых заключается в дальнейшей возможности получать материалы, свойствами поверхности которых можно управлять.

Исследование характеристик микроструктуры поверхности металла, особенно характеристик, определяемых вдоль вертикальной оси, требует соответствующих методов. Одним из перспективных вариантов является метод сканирующей зондовой микроскопии [1].

Основой метода сканирующей зондовой микроскопии является анализ топографии поверхности образца механическим зондом с размером острия порядка 10 нм. Выделяют три типа сканирующей зондовой микроскопии, имеющие свойственные им типы взаимодействия [1]:

- сканирующая туннельная микроскопия (электрическое взаимодействие);
- атомно-силовая микроскопия (механическое взаимодействие);
- магнитно-силовая микроскопия (магнитное взаимодействие).

В данной работе методом атомно-силовой микроскопии (АСМ) исследовались особенности морфологии поверхности тонких пленок сплава Al-1.0 ат. % Cr, сформированных ионно-ассистированным осаждением. Универсальность метода АСМ, современного и высокоточного метода оценки качества поверхности изделий, достигается путем возможности наблюдения микроструктуры поверхности в естественных условиях эксплуатации: на воздухе и в жидкостях, а также метод АСМ является надежным инструментом для оценки шероховатости поверхности материалов, расширяя диапазон измерений до субнанометрового масштаба [1].

Эксперименты по ионно-ассистированному осаждению металлических пленок на стекло проводились с использованием вакуумного резонансного плазменно-дугового источника ионов при отсутствии ускоряющего напряжения и при $U=3$ кВ (вакуум 10^{-2} Па) [2]. В качестве электродов в установке использовался сплав Al-1.0 ат. % Cr. Время осаждения покрытий составляло 3, 6 и 9 ч (при $U=0$) и 10 ч (при $U=3$ кВ). Расчетная скорость осаждения покрытий составляла $\sim 0,1$ нм/мин. В соответствии с методикой предложенной Р. Sigmund [3] скорость «кристаллизации» (охлаждения каскадов) составила 10^{12} - 10^{13} К/с.

Изучение топографии поверхности образцов и определение ее шероховатости было выполнено на атомно-силовом микроскопе NT-206 с использованием зондов CSC-38. Значение шероховатости поверхности (σ) определялось по усредненным данным, полученным с 7 площадок размером 20×20 мкм², выбранных произвольным образом. Для типичных 2D-АСМ изображений фольг было выполнено построение профиля поверхности вдоль 11 горизонтальных линий, расположенных на расстоянии 2 мкм друг от друга, используя программу Surface Explorer [4] и были построены 11 соответствующих гистограмм распределения элементов рельефа поверхности по высоте. Далее по результатам усреднения для каждого 2D-АСМ изображения была построена результирующая гистограмма распределения структурных элементов поверхности по высоте и определена средняя высота неровностей профиля.

В качестве параметров, характеризующих шероховатость поверхности образца, были выбраны три величины: два «высотных» – (среднеарифметическая шероховатость) и (высота неровностей профиля) – и один «шаговый» параметр (среднее значение шага между выступами) [5]. Для унифицированной оценки неоднородности твердой поверхности был рассчитан коэффициент σ .

Как видно на рисунке 1а, все полученные гистограммы распределения средних измеренных высот и впадин рельефа поверхности исследуемой тонкой пленки унимодальны – имеют ярко выраженный максимум, обладают положительной асимметрией (большая доля рельефа приходится на выступы), связанной с островковым механизмом роста покрытия [6], и по форме практически повторяют гауссово распределение.

Согласно данным АСМ, нанесение пленки Al-1.0 ат. % Cr в беспотенциальном режиме ($U=0$) в течение трех часов приводит к увеличению значения σ до 8.9 нм, что в 3.5 раза больше по сравнению с шероховатостью исходной стеклянной подложки. Увеличение времени нанесения покрытия с 3 до 9 ч приводит к росту шероховатости уже сформированной пленки до 21.6 нм при том, что на гистограммах мы наблюдаем снижение величины максимума и более равномерное распределение высот и впадин. Это объясняется тем, что на этапе поверхности пленки характеризуется наличием мелких «островков» – участков с большой высотой, но маленькой площадью, между которыми находятся участки с низкой высотой, рис.1а. К

моментам времени и островки уменьшаются за счет заполнения пленкой участков между ними с увеличением высоты последних. Анализируя данные рисунка 1а, полученные при $U=3$ кВ, на этапе установлено, что при подаче ускоряющего потенциала степень неоднородности поверхности значительно уменьшается при практически неизменном k , что хорошо согласуется с данными на рисунке 1б. Указанному режиму получения пленки соответствует минимальное значение параметра k . Обнаруженное снижение величины параметра k в зависимости от условий ОПАСИ (режима и времени осаждения покрытий) свидетельствует, что параметр k не является в достаточной степени информативным, чтобы давать количественную оценку изменения высотных характеристик микрорельефа (топографии) поверхности в продольном направлении при осаждении покрытий.

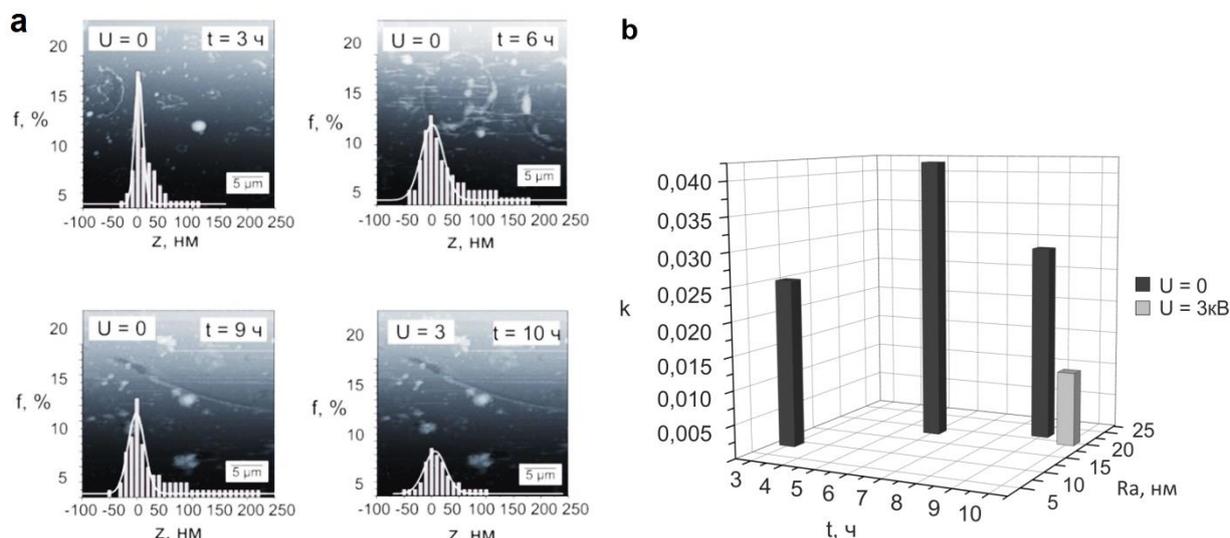


Рис. 1 – 2D - АСМ изображения топографии поверхности пленок сплава Al-1.0 ат. % Cr, нанесенных на стекло, в зависимости от условий осаждения и соответствующие гистограммы распределения высот и впадин рельефа поверхности (а); зависимость значений коэффициента k от параметра Ra и времени осаждения покрытия (б)

Рассчитанный нами параметр k как отношение $k = \frac{Ra}{t}$ позволяет сразу же дать качественную оценку эволюции морфологических неоднородностей поверхности в зависимости от условий формирования покрытия и получить данные о количественном соотношении поперечной и продольной шероховатости изучаемого микрорельефа. Последнее имеет важное практическое значение для условий эксплуатации изделия. Поскольку шероховатость во многом определяет смачивающие свойства материалов, то важно знать, какая доля жидкой фазы в случае гидрофильных образцов проникает в углубления поверхности. Таким образом, можно сделать вывод, что коэффициент k является дополнительной количественной мерой степени неоднородности поверхности и может быть использован для контроля морфологических особенностей поверхности материалов с заданными служебными характеристиками.

Авторы благодарны Яковенко Ю. С. (БГПУ) за помощь при проведении экспериментов с использованием АСМ.

Список использованных источников:

1. Миронов, В. Л. Основы сканирующей зондовой микроскопии / под ред. В. Л. Миронова. - Нижний Новгород: Институт физики микроструктур РАН, 2004. - 114 с.
2. Ташлыков, И. С., Белый, И. М. Патент РФ №2324. 1С1 ВУ, С23 С4/12. С4/18, С14/16. Офиц. бюл. гос. пат.ведом. 1, 30 (1999).
3. Sigmund, P. Energy density and time constant of heavy-ion-induced elastic-collision spikes / P. Sigmund // Appl. Phys. Lett. - 1974. - Vol. 25. - P. 169-171.
4. Официальный сайт ОДО «Микротестмашины». [Электронный ресурс]. – SurfaceExplorer. – Режим доступа: <http://microtm.com>. – Дата доступа: 20.03.2018.
5. Калинин, Б. А. Измерение топографии модифицированной поверхности материалов: Лабораторная работа / Б. А. Калинин, Н. В. Волков. - М.: МИФИ, 2008. - 32 с.
6. Фельдман, Л., Майер, Д. Основы анализа поверхности тонких пленок. – М.: Мир, 1989. – 344 с.

ВЛИЯНИЕ СВЧ-ИЗЛУЧЕНИЯ НА ЖИВЫЕ ОРГАНИЗМЫ

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

Бурштын М.В., Авельчук Ю.А.

Ташлыкова-Бушкевич И.И. – к. ф.-м. н., доцент

Данное исследование представляет результаты экспериментов, выполненных при изучении влияния СВЧ-излучения на живые организмы на примере растений. Собранное на основе диода Д604 устройство позволяет оценивать мощность СВЧ-излучения, испускаемое мобильными телефонами при исходящих/входящих звонках в зависимости от поколения беспроводной связи.

В современном мире человека окружает огромное множество различных электронных приборов, каждый из которых излучает электромагнитные волны определённой частоты. Среди данного многообразия излучений, наибольший интерес представляет СВЧ-излучение. Под сверхвысокими частотами (СВЧ) принято понимать участок электромагнитного спектра с частотами колебаний, лежащими приблизительно между 30 МГц и 3000 Гц, и с длинами волн соответственно между 10 м и 0,1 мм [1]. Несмотря на то, что его влияние на живые организмы до сих пор не изучено до конца, свойства микроволнового излучения [2] используются повсеместно: для бесконтактного нагрева тел и для радиолокации (большая интенсивность излучения), в средствах связи (малая интенсивность излучения).

Чтобы оценить, какой уровень СВЧ-излучения каждый человек в среднем получает от некоторых электронных приборов, был произведён ряд экспериментов. В первом эксперименте, выполненном нами, оценивался уровень экранизации электромагнитных волн СВЧ-печи: для этого необходимо поместить мобильный телефон внутрь микроволновой печи, закрыть печь и позвонить на телефон. Если абонент телефона, размещённого внутри микроволновой печи недоступен, это свидетельствует о том, что данная модель СВЧ-печи хорошо экранирует электромагнитные волны, излучаемые внутри неё, и является безопасной для живых организмов.

Наиболее распространённым средством связи в наше время является мобильный телефон. Для того, чтобы определить, какой уровень СВЧ-излучения получает человек при использовании телефонов, было собрано небольшое устройство, схема которого представлена на рисунке 1. Для этого устройства был использован диод Д604 - кремниевый, точечный, детекторный мощностью 1-2 Вт. В случае, если диод зафиксирует волны в диапазоне от 2,4...4 см светодиод, подключённый к нему, загорится.

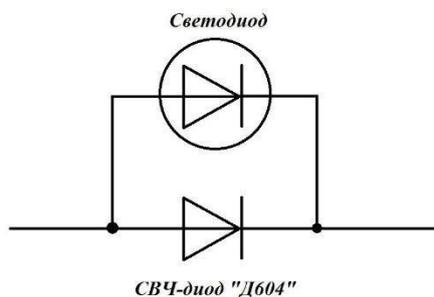


Рис. 1 – Схема прибора, реагирующего на СВЧ-излучение определённой частоты

Выполненный опыт заключался в следующем: на сотовый телефон производился звонок с другого сотового телефона, при этом диод размещали около каждого из телефонов. Опыт был произведён с 2G, 3G и 4G связями. Из полученных результатов можно сделать следующие выводы: 3G и 4G связи оказались наиболее безопасными, т.к. светодиод устройства не загорелся, а при использовании 2G связи он горел достаточно ярко. Таким образом, можно сделать вывод о том, что чем старше поколение связи, тем сильнее излучение, испускаемое телефоном.

Результаты, проведенного опыта, подтверждают то, что человек постоянно поглощает СВЧ-излучение. В данной работе рассмотрены негативные аспекты влияния СВЧ-излучения на живой организм. Из курса физики известно, что СВЧ-излучение большой интенсивности способно нагревать тела, в том числе и живые организмы [2-7]. Однако ток крови позволяет уменьшить нагрев (это относится к органам, богатым кровеносными сосудами). В свою очередь органы, не содержащие кровеносных сосудов, например, хрусталик глаза, не способны регулировать нагрев. В частности тепловой нагрев хрусталика приводит к его помутнению и в дальнейшем разрушению.

Следует учесть и влияние микроволнового излучения на пищу, приготовленную в СВЧ-печи: в результате воздействия электромагнитного излучения на вещество возможна ионизация молекул, т.е. атом может приобрести или потерять электрон, – а это меняет структуру вещества. Излучение приводит к разрушению и деформации молекул пищи. Микроволновая печь создает новые соединения, не существующие в природе, называемые радиолитическими. Радиолитические соединения создают молекулярную гниль — как прямое следствие радиации.

Также СВЧ-излучение приводит к ослаблению клеток нашего организма. В результате облучения

электромагнитными волнами клеточные мембраны ослабевают [7,8]. Следствие чего является то, что клеточные мембраны не могут предохранить клетку от проникновения вирусов, грибов и других микроорганизмов, также подавляется естественный механизм самовосстановления.

Для того, чтобы оценить влияние СВЧ-излучения, нами был поставлен следующий эксперимент: два одинаковых цветка на протяжении месяца поливали водой, разогретой в микроволновке, и обыкновенной водой. В результате наблюдения за комнатными растениями существенных различий между цветами мы не выявили. На основании чего был сделан вывод, что еда, разогретая в СВЧ-печи, не представляет опасности для живых организмов.

В настоящее время человечество неспособно полностью отказаться от использования СВЧ-волн, к тому же часть из них имеет природное происхождение. Поэтому каждому человеку необходимо самому следить за уровнем, окружающих его электромагнитных волн, уменьшая его, где это возможно (например, используя более современные модели сотовых телефонов и СВЧ-печи с высокой степенью экранизации). В настоящее время учёные всего мира ищут способ безопасного использования электромагнитных волн.

Список использованных источников:

1. Лебедев, И. В. Техника и приборы СВЧ / И. В. Лебедев – Москва: Высшая школа, 1970. – 9 с.
2. Википедия [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Микроволновое_излучение. – Дата доступа: 01.04.2018
3. Лебедев, И. В. Техника и приборы СВЧ / И. В. Лебедев – Москва: Высшая школа, 1970. – 9 с.
4. Станков, В.М. Инфаркты и страсть: смена мировоззрения / В.М. Станков – Одесса: АО Бахва, 2009. – 98 с.
5. Бударков, В.А. Краткий радиозоологический словарь / Федеральная целевая программа «Гос. поддержка интеграции высшего образования и фундамент. науки на 1997-2000 годы». – Саранск: Изд-во Мордовского ун-та, 2000. – 256 с.
6. Миллиметровые волны и фотосинтезирующие организмы / А. Х. Тамбиев [и др.]; под ред.: Ю. В. Гуляева, А. Х. Тамбиева. – М: Радиотехника, 2003. – 176 с.
7. Андреева, А.П. Влияние СВЧ-излучения малой мощности на гемоглобин / А.П. Андреева, М.Г. Дмитриева, С, А. Ильина – 1971. – 121 с. – (Серия «Электроника СВЧ»; вып. 11)
8. Харвей, А. Ф. Техника сверхвысоких частот. Т.2 / А. Ф. Харвей. М.: Советское радио, 1965. – 784 с.

АНАЛИТИЧЕСКАЯ ОЦЕНКА СТРУКТУРНЫХ ОСОБЕННОСТЕЙ ТОНКИХ МЕТАЛЛИЧЕСКИХ ПЛЕНОК, ФОРМИРУЕМЫХ МЕТОДОМ ИОННО-ПЛАЗМЕННОГО НАПЫЛЕНИЯ

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

Маргун Ю.С., Поваляева Н.И., Степанцов Е.В., Макаревич Я.С., Петрушкевич В.Р.

Ташлыкова-Бушкевич И.И. – к.ф.-м.н., доцент

Исследована микроструктура поверхности тонких металлических пленок Al и его бинарных сплавов с хромом и железом, сформированных ионно-ассистированным осаждением на стеклянных подложках, методом растровой электронной микроскопии (РЭМ). Проведен анализ полученных РЭМ изображений методом случайных секущих.

В настоящее время устойчивый научный интерес сохраняют технологии обработки поверхностей твердых тел, позволяющие решать прикладные практические задачи, связанные с улучшением физико-химических свойств поверхности имеющегося материала. Ионно-плазменные технологии обработки поверхности [1] широко используются для экономичной модификации и придания требуемых характеристик поверхностям твердых тел.

Для управления свойствами тонкопленочных металлических покрытий при их осаждении на подложку необходима диагностика поверхности системы покрытие / подложка. В данной работе обсуждаются экспериментальные результаты исследования с помощью РЭМ микроструктуры Al и его сплавов Al-Cr и Al-Fe, полученных методом ионно-ассистированного осаждения (ОПАСИ).

Эксперименты по ионно-ассистированному осаждению металлических пленок на стекло проводились с использованием вакуумного резонансного плазменно-дугового источника (вакуум 10^{-2} Па) [2] в двух режимах: при отсутствии ускоряющего потенциала и при $U=3$ кВ. В качестве электродов установки использовался чистый алюминий и сплавы Al-1.0 ат. % Cr, Al-1.5 ат. % Fe. Время осаждения покрытий составляло 3, 6, 9 ч (при $U=0$) и 10 ч (при $U=3$ кВ) при скорости осаждения покрытия ~ 0.1 нм/мин. В соответствии с методикой, предложенной Р. Sigmund [3], скорость «кристаллизации» (охлаждения каскадов) имела значение 10^{12} - 10^{13} К/с. Изучение микроструктуры поверхности образцов было выполнено с применением растрового электронного микроскопа марки LEO-1455 VP.

Данные исследования поверхности образцов методом РЭМ показали, что на поверхности тонких пленок присутствует микрокапельная фракция из наносимого покрытия. Для количественного анализа геометрии обнаруженных структурных элементов на поверхности рассматриваемых металлических пленок алюминия и его сплавов использовался один из стандартных методов стереометрической металлографии – метод случайных секущих [4].

По результатам статистической обработки РЭМ изображений были получены таблицы следующих

параметров: среднего измеренного диаметра, величины удельной поверхности и объемной доли микрокапельной фракции с относительной погрешностью 8 %. Объем выборки $x=20$. По полученным данным построены гистограммы распределения по размерным группам среднего измеренного диаметра наблюдаемых структур у наносимых металлических пленок, где по оси абсцисс отложены значения выбранного параметра каждой размерной группы, а по оси ординат – количество объектов с данным значением параметра, рисунок 1, 2. Установлено, что размер и плотность присутствующей на поверхности тонких металлических пленок микрокапельной фракции зависит от режима и времени осаждения покрытия. Как видно на рисунке 1, распределения средних измеренных диаметров микрокапельной фракции по размерным группам хорошо описываются бимодальным распределением: можно выделить два характерных максимума в каждом случае.

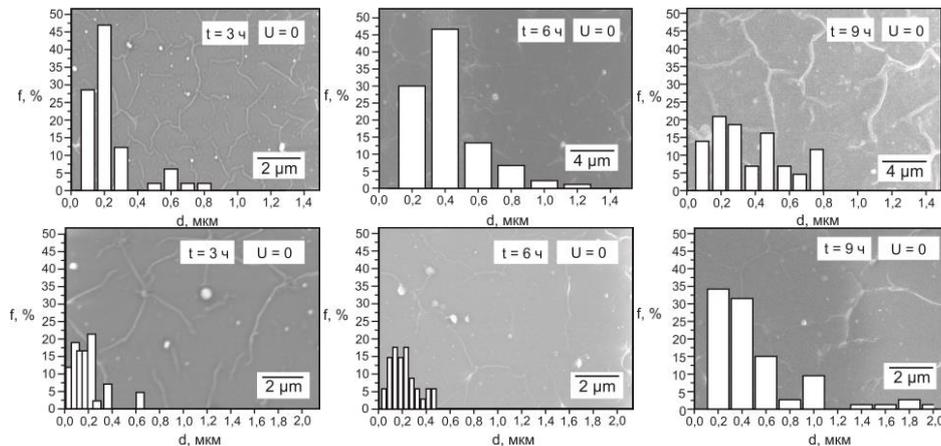


Рис. 1 - Распределение частиц микрокапельной фракции на поверхности тонких пленок сплавов Al-1.0 ат. % Cr (а-в) и Al-1.5 ат. % Fe (г-е) по размерным группам при $U=0$ в зависимости от времени осаждения покрытия

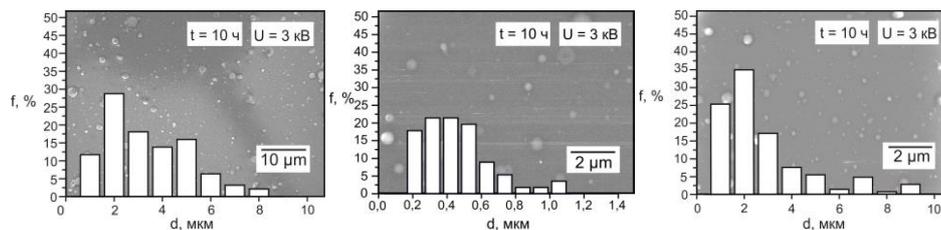


Рис. 2 - Распределение частиц микрокапельной фракции на поверхности тонких пленок Al (а) и сплавов Al-1.0 ат. % Cr (б) и Al-1.5 ат. % Fe (в) по размерным группам при $U=3$ кВ в зависимости от времени осаждения покрытия

Найдено, что средний измеренный диаметр микрокапельной фракции для пленок, полученных при $U=0$, возрастает в диапазоне от 0.2 ± 0.02 мкм до 0.4 ± 0.02 мкм для пленки Al - 1.0 ат. % Cr и в диапазоне от 0.18 ± 0.02 мкм до 0.52 ± 0.02 мкм для пленки Al - 1.5 ат. % Fe при увеличении времени осаждения покрытия с 3 до 9 часов. На поверхностях пленок, осажденных при $U=3$ кВ, формируются микрокапельные фракции значительно более крупные и частые, средний диаметр которых достигает значения 3.0 ± 0.02 мкм для системы Al/стекло при времени осаждения $t=10$ ч. Также для тонких пленок Al - 1.0 ат. % Cr и Al - 1.5 ат. % Fe наряду с увеличением d характерно возрастание объемной доли микрокапельной фракции V в зависимости от условий формирования пленок.

Шероховатость поверхности изделия является одним из ключевых параметров, определяющих условия его эксплуатации. На рисунке 3 представлены 3D гистограммы, наглядно демонстрирующие зависимость средней арифметической шероховатости (R_a) [5] металлической пленки, формируемой методом ОПАСИ, от наиболее вероятных размеров среднего диаметра микрокапельной фракции и условий формирования покрытия.

Таким образом, варьируя условия формирования покрытия методом ОПАСИ и изменяя элементный состав электродов можно получать покрытия с требуемой шероховатостью, а, значит, прогнозируемыми эксплуатационными свойствами. Результаты исследований поверхностной микроструктуры тонких металлических пленок инструментами стереометрической металлографии позволяют количественно оценить вклад физических параметров эксперимента на характер формируемого рельефа поверхности и степень ее шероховатости, а, значит, дать оценку качества формируемого металлического покрытия.

Полученные экспериментальные данные позволяют сделать вывод, что выбор параметров при формировании покрытия методом ионно-ассистированного осаждения позволяет управлять процессами структурообразования покрытий, которые в конечном итоге определяют функциональные свойства пленок.

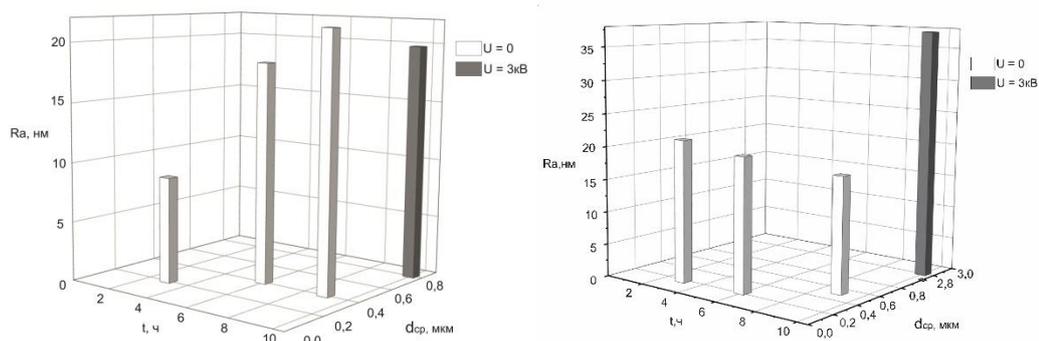


Рис. 3 – Зависимость средней арифметической шероховатости поверхности и среднего диаметра микрокапельной фракции пленок сплавов Al-1.0 ат. % Cr (а) и Al-1.5 ат. % Fe (б), формируемых на стеклянной подложке методом ОПАСИ от режима и времени осаждения покрытия

Авторы выражают признательность Яковенко Ю. С. (БГПУ) за помощь при выполнении исследования и обсуждение результатов.

Список использованных источников:

1. Пичугин В.Ф. Материаловедение поверхности и тонких пленок. Томск: Изд-во ТПУ, 2008. 173 с.
2. Ташлыков, И. С., Белый, И. М. Патент РБ №2324. 1С1 ВУ, С23 С4/12. С4/18, С14/16. Офиц. бюл. гос. пат.ведом. 1, 30 (1999).
3. Sigmund, P. Energy density and time constant of heavy-ion-induced elastic-collision spikes / P. Sigmund // Appl. Phys. Lett. - 1974. - Vol. 25. - P. 169–171.
4. Салтыков, С. А. Стереометрическая металлография / С. А. Салтыков. — М.: Металлургия, 1976. — 272 с.
5. Ташлыкова - Бушкевич, И.И. Свойства поверхности тыльных и лицевых контактов фотоэлектронных преобразователей, осаждаемых пассивно и при ионном ассистировании / И.И. Ташлыкова-Бушкевич, Ю.С. Яковенко, И.А. Бушкевич, О.Г. Бобрович, И.С. Ташлыков // Материалы и структуры современной электроники: сб. науч. тр. VII Междунар. науч. конф., посвящ. 50-летию каф. физики полупроводников и нанозлектроники, Минск, 12–13 окт. 2016 г. / редкол: В. Б. Оджаев (отв. ред.) [и др.]. — Минск: Изд. центр БГУ, 2016. — С. 123–126.

ОПТИМИЗАЦИЯ СОЛНЕЧНЫХ БАТАРЕЙ С ПОМОЩЬЮ РОБОТИЗИРОВАННОЙ СИСТЕМЫ, УЧИТЫВАЮЩЕЙ ТРАЕКТОРИЮ ДВИЖЕНИЯ СОЛНЦА

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

Петрушко Е.А.

Ташлыкова-Бушкевич И. И. - к. ф.-м. н., доцент.

Альтернативная энергетика развивается стремительными темпами. Наиболее популярными являются ветровая и солнечная энергетика. В данной статье будет рассмотрен один из способов модификации солнечных батарей.

Целью настоящего проекта является подтверждение опытным путём энергетической и экономической целесообразности использования роботизированных систем, учитывающих траекторию перемещения Солнца.

Солнечная энергия широко используется во многих областях: для естественного освещения помещений, при нагреве воды, сушке, иногда даже для приготовления пищи. Но наиболее значимое её использование – это производство электричества или тепла.

Классическая солнечная электростанция представляет собой совокупность солнечных батарей, установленных на неподвижной несущей конструкции. В течение светового дня Солнце перемещается по небесной сфере, в результате чего меняется угол падения солнечных лучей на поверхность солнечных батарей. Необходимым и обязательным условием получения максимальной мощности солнечной электростанции является угол падения солнечных лучей на поверхность солнечных батарей. В данном случае угол падения солнечных лучей должен быть максимально близок к θ . Стационарно установленные солнечные батареи не позволяют отслеживать перемещение Солнца, в результате чего, солнечная электростанция теряет часть мощности, что увеличивает сроки окупаемости и ухудшает рентабельность подобных проектов. Поэтому было принято разработать роботизированную систему, учитывающую траекторию движения Солнца.

Для исследования и сравнения в работе была проведена оценка среднесуточной мощности стационарной батареи (Табл. 1). Контрольные значения напряжения, силы тока и мощности были зарегистрированы каждый час на протяжении светового дня.

Таблица 1. Оценка среднесуточной мощности стационарной батареи

Время t, ч	Сила тока I, А	Напряжение U, В	Мощность P, Вт
	0,11	11,41	1,25
	0,15	12,56	1,88
	0,21	13,30	2,79
	0,27	13,79	3,72
	0,32	14,04	4,49
	0,28	14,01	3,92
	0,22	13,82	3,04
	0,17	12,64	2,15
	0,15	12,22	1,83
	0,11	11,97	1,32
	0,10	11,44	1,14

Теперь рассмотрим роботизированную систему, учитывающую траекторию движения Солнца. Внешний вид и схема вращения системы представлены на рисунках 1 и 2. Зеленой стрелкой указано вращение системы в горизонтальной плоскости, а бордовой в вертикальной плоскости.

Для реализации проекта были использованы следующие комплектующие: солнечные модули, шаговые двигатели, аппаратная платформа ArduinoNanoV3, кросс-плата расширения (shield) для аппаратной платформы с DuPont-коннекторами, аккумуляторы Li-ion формата 18650. Теперь следует объяснить назначение данных компонентов.

На кросс-плате были выполнены все соединения компонентов. К ней подключены Arduino (управляет вращением системы) и сервоприводы (обеспечивают данное вращение в двух плоскостях). Аккумуляторы накапливают часть энергии, которая обеспечит работу системы.

Несущая конструкция представляет собой подвижную раму. В основании лежит горизонтальная платформа, которая вращает систему в горизонтальной плоскости относительно сторон света. На данную поворотную платформу крепится вертикальная металлическая рама, которая, в свою очередь, меняет угол наклона солнечной батареи в вертикальной плоскости относительно горизонта.

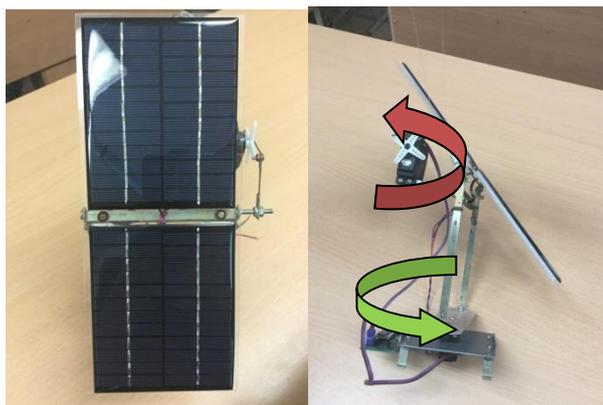


Рис. 1 - Внешний вид системы

Рис. 2- Схема вращения системы

Контрольные значения напряжения, силы тока и мощности (Табл. 2) регистрировались каждый час на протяжении светового дня.

Таблица 2. Оценка среднесуточной мощности роботизированной системы

Время t, ч	Сила тока I, А	Напряжение U, В	Мощность P, Вт
	0,26	13,11	3,41
	0,28	13,43	3,76
	0,29	13,57	3,94
	0,31	14,04	4,35
	0,33	14,10	4,65
	0,32	14,05	4,50
	0,30	14,02	4,20
	0,29	13,98	4,05
	0,28	13,72	3,84
	0,25	13,37	3,34
	0,23	13,14	3,02

Проведём сравнительную характеристику стационарной и роботизированной батарей (Табл. 3). Примем во внимание, что средняя рыночная стоимость стационарной системы составляет 8400 у.е., а роботизированной, с учетом изготовления несущей конструкции, 10000 у.е.

Таблица 3. Сравнение стационарной батареи и роботизированной системы

Тип	Стационарная	Роботизированная
Стоимость, руб.	15960	19000
Номинальная мощность	5000	5000
Среднесуточная мощность	3050	4800
Сроки окупаемости, лет	2,72	2,06

Анализируя данные, приведенные в таблице, можно сделать следующие выводы. Использование роботизированных систем, учитывающих траекторию движения Солнца, является экономически целесообразным методом оптимизации солнечных электростанций. Среднесуточная мощность роботизированной системы на 57% больше стационарной. Короткие сроки окупаемости говорят о высокой рентабельности проекта. Исходя из анализа опыта высокоразвитых стран, проект является высокорентабельным в том случае, когда сроки его окупаемости находятся в пределах 2-4 лет.

Список используемых источников:

1. Ермашкевич, В.Н. Возобновляемые источники энергии Беларуси: прогноз, механизмы реализации / В.Н. Ермашкевич, Ю.Н. Румянцева. // Учебное пособие. – Мн.: НО ООО «БИП-С», 2004. – 121 с.
2. Mukund, R. Ветровые и солнечные энергетические установки. Проектирование, анализ и эксплуатация / Mukund R. Patel // Учебное пособие. – CRC Press, 2005. – 472 с.
3. Раушенбах, Г. Справочник по проектированию солнечных батарей / Г. Раушенбах // Справочник по энергетике. - Энергоатомиздат, 1983. – 360 с.
4. Ландсберг, Г. С. Элементарный учебник физики. Том 2. Электричество и магнетизм / Г. С. Ландсберг // Учебник для студентов вузов. - Физматлит, 2001. – 480 с.
5. Официальный сайт Министерства энергетики Республики Беларусь. [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://www.minenergo.gov.by>. – Дата доступа: 16.03.2018

МОДЕЛИРОВАНИЕ ДИФРАКЦИИ ФРЕНЕЛЯ

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

Королёв Д.Ф.

Григорьев А. А. — к.ф.-м.н., доцент

Дифракция электромагнитных волн — явление, проявляющееся в отклонении от геометрической оптики при неоднородностях среды, сравнимых с длиной волны. Рассмотрим источник света S рядом с которым находится экран с круглым отверстием, центром которого является точка O. В данной работе построена модель дифракции света в точке M, лежащей на прямой, соединяющей S и O.

Метод Френеля заключается в разбиении поверхности фронта волны на зоны с разностью равной половине длины волны. Тогда колебания в точке M, возбуждаемые двумя соседними зонами, противоположны по фазе и ослабляют друг друга. Амплитуда, возбуждаемая зоной Френеля, зависит от угла площади поверхности зоны и от угла между нормалью к поверхности зоны и прямой, направленной в точку M, потому наибольшую амплитуду возбуждают первые зоны Френеля. При открытой малой части первой зоны наблюдается дифракция Фраунгофера, при открытых нескольких зонах — дифракция Френеля, при большом количестве открытых зон, дифракцией, зачастую, можно пренебречь. Для демонстрации дифракции Френеля выберем параметры системы, при которых останутся открытыми три и четыре зоны Френеля. Изобразим дифракционные картины и построим графики интенсивности по их разрезам (см. рис. 1 - 2):

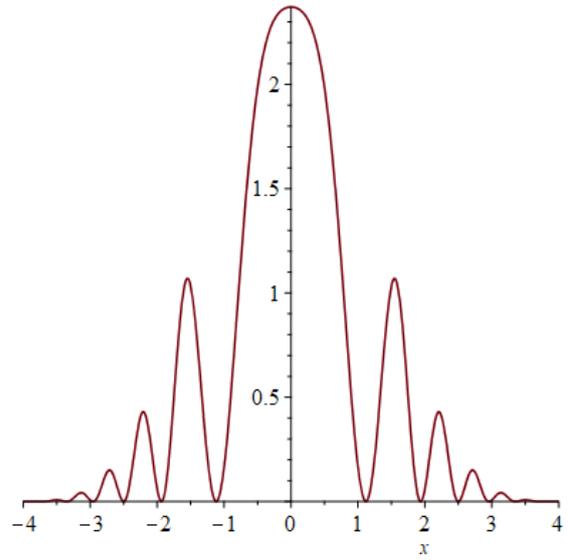
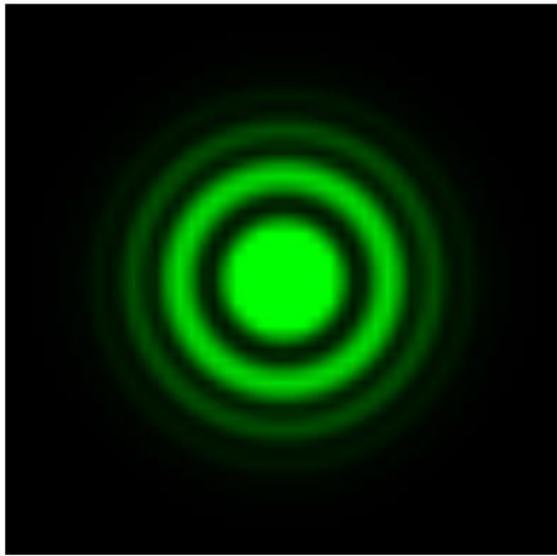


Рис. 1 - Дифракция при трёх открытых зонах Френеля

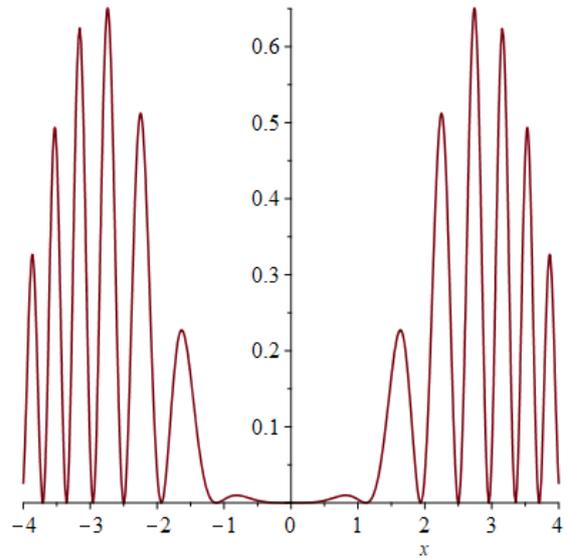
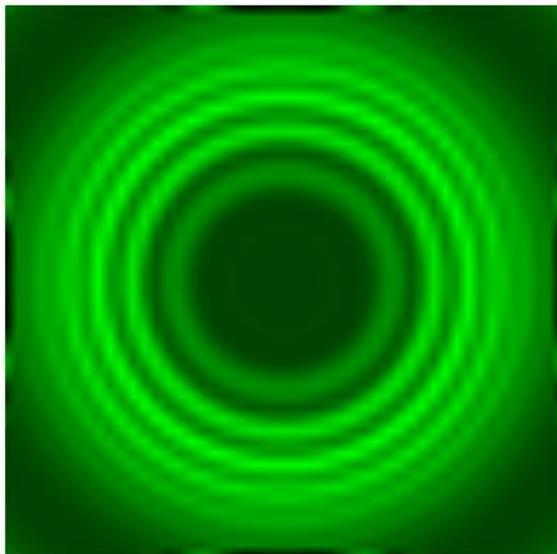


Рис. 2 - Дифракция при четырёх открытых зонах Френеля

Нечётному количеству открытых зон Френеля соответствует дифракционная картина с максимумом освещённости в центре и множеством вторичный минимумов и максимумов, чётному — картина с минимумом освещённости в центре.

Список использованных источников:

1. Савельев И.В. Курс общей физики. М.: Наука. 1970г. Т.1—281 с.
2. Ландсберг Г.С. Оптика. М.: ФИЗМАТЛИТ. 2003г. 138 с.

СЕКЦИЯ «ИНФОРМАТИКА»

МЕТОДЫ ОПТИМИЗАЦИИ ДОСТАВКИ БОЛЬШИХ ОБЪЕМОВ ТЕЛЕМЕТРИИ В УСЛОВИЯХ УЗКИХ КАНАЛОВ СВЯЗИ

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

Александров А.А., Пилецкий И.И.

Пилецкий И.И. – к.ф-м.н., доцент

В данной работе рассмотрены некоторые вопросы, связанные с передачей больших объемов данных в центр обработки данных с использованием узких каналов передачи информации, а также обзор методов оптимизации доставки телеметрической информации с транспорта.

Для успешного использования технологий BigData на транспорте необходимо решить проблему доставки больших объемов телеметрической информации в центры обработки данных. Для городского общественного транспорта, где покрытие сигналом мобильных сетей GSM/HSPA/LTE приближается к 100%, проблем с отправкой больших объемов данных не возникает (в крайнем случае данные можно накапливать на борту транспортного средства и передавать их во время нахождения транспорта в зоне уверенного приема сигналов мобильных сетей, например, когда транспортное средство находится в депо), то для транспорта, такого, как грузовые автомобили, поезда, грузовые самолёты, есть определенная проблема в доставке больших объемов данных.

Одним из методов оптимизации доставки больших объемов данных в условиях узких каналов связи является метод буферизации передаваемых данных. В местах недостаточного уровня сигнала мобильных операторов для устойчивого канала передачи данных между базовой станцией оператора сотовой связи и установленным оборудованием, в блоке передачи предусмотрена буферизация (запись) данных на внутреннюю память. В момент появления устойчивой связи с качественным сигналом буферизованные данные передаются на сервер в автоматическом режиме. [1]

Недостатком данного метода является невозможность обеспечить максимальную непрерывность передачи данных для предиктивного принятия решений на основе этих данных. Искусственно ограничивается сфера применения полученных данных. Например, на основе полученной информации уже нельзя предсказать и выдать определенные команды машинисту поезда или водителю грузовика для корректировки текущей скорости движения в зависимости от определенных обстоятельств.

Принципиально иным подходом является оптимизация доставки данных на основе сжатия и дедупликации передаваемых данных.

При разработке алгоритмов оптимизации передачи больших объемов телеметрии необходимо в первую очередь учитывать характер передаваемых данных. Как известно, по большей части передаваемая телеметрия является данными, полученными с различных датчиков транспортного средства, такими, как аналого-цифровые преобразователи, датчики частоты вращения, статическая информация о текущем режиме работы агрегатов, а также текущие координаты транспортного средства.

При обеспечении высокой частоты съема информации с датчиков и рассмотрении данной информации в некотором минимальном приближении, можно заметить, что в целом данные являются неизменными и непрерывными. На отклонение от данного правила влияют несколько факторов. Это в первую очередь шум, который присутствует в любой аналоговой системе. Также это могут быть моменты, когда в агрегатах происходят переходные процессы (например, процесс разгона или торможения). Также флуктуации данных могут быть повторяющимися с определенной частотой, что может быть вызвано неисправностями в агрегатах транспортного средства.

Принципиальное влияние на количество передаваемых данных может влиять непосредственно протокол передачи информации прикладного уровня. Для систем, которые не испытывают проблем с шириной канала передачи данных, вполне нормальным является использование текстовых форматов обмена данными, например, JSON. Непосредственная отправка данных от клиента к серверу может производиться по протоколу HTTP. Однако использование высокоуровневых протоколов передачи информации подразумевает наличие большого количества избыточных данных в передаваемом трафике, что абсолютно недопустимо в условиях узких каналов связи клиента с сервером.

В данном случае, выглядит целесообразным разработка бинарного протокола передачи больших данных, учитывающего специфику передаваемых данных, имеющего минимальное количество передаваемых избыточных данных, а также поддерживающего коррекцию ошибок, либо отбрасывание ложных данных.

Использование оптимально спроектированного протокола передачи данных с учетом оптимизации доставки и коррекции ошибок позволит использовать не только традиционные сети передачи информации (например, мобильные сети или спутниковый канал), но и низкоскоростную передачу пакетных данных в диапазоне коротких, средних или длинных волн, где скорость передачи данных ограничена естественными причинами, такими, как необходимость использования цифровой модуляции с узкой полосой излучаемого спектра. Таким образом, использование разработанных методов оптимизации доставки больших объемов телеметрической информации с транспортных средств позволит облегчить интеграцию технологий BigData и средств предиктивной аналитики в сферу транспорта.

Список использованных источников:

1. Пилецкий, И.И., Александров, А.А. Применение технологий Big Data в сфере транспорта // BIG DATA and Advanced

КЛАССИФИКАЦИЯ СОСТОЯНИЙ РАСТЕНИЙ С ПОМОЩЬЮ СВЁРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ

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

Барковский А.В.

Жвакина А.В. – к.т.н., доцент

Задачи, связанные с восприятием изображений и видео, называемые задачами компьютерного зрения, являются одними из самых актуальных среди задач машинного обучения. Одним из способов решения подобных задач является использование свёрточных нейронных сетей. В докладе рассматривается возможность использования свёрточных нейронных сетей для классификации растений в биологических исследованиях.

Автоматическая обработка изображений при помощи нейронных сетей может использоваться для классификации растений по виду или состоянию здоровья, обнаружения болезней, паразитов или сорных растений, автоматического отслеживания прогресса роста растения под воздействием различных факторов и многих других задач.

Задача является актуальной, так как сейчас на биологическом факультете Белорусского Государственного Университета данные мероприятия проводятся вручную, требуют больших затрат времени и человеческих ресурсов. Использование нейронной сети, способной самостоятельно отличать здоровые растения от умирающих, позволит автоматизировать процесс проведения биологических исследований. Это увеличит объём исследований, проводимых биологами, упростит наблюдение за экземплярами растений и снизит нагрузку на работников лабораторий.

Сравнительный анализ возможностей таких методов анализа изображений, как вручную подобранные свёртки[1], гистограммы цветов или ориентированных градиентов, а также самих свёрточных нейронных сетей показал, что нейронные сети наиболее эффективны для решения задачи классификации растений по состоянию. Они позволяют достичь необходимого уровня точности и обеспечивают скорость распознавания, позволяющую в некоторых условиях их использование в реальном времени.

Стоит отметить, что хотя нейронные сети-это относительно ресурсоёмкий метод обработки изображений, с развитием вычислительных компонентов (в частности, графических ускорителей) в последние годы даже они зачастую пригодны для использования в условиях ограниченных вычислительных ресурсов. К примеру, уже существует нейронная сеть, способная с высокой точностью разделять изображение на объекты, работающая со скоростью около 100 кадров в секунду даже на мобильном телефоне [2].

Также исследовался другой вид свёрточных сетей, особенно хорошо адаптированный под мобильные устройства[3]. Так как доступные в телефонах и стационарных компьютерах ресурсы продолжают расти, ресурсоёмкость нейронных сетей будет отходить на второй план, то в ближайшем будущем применимость и актуальность свёрточных сетей возрастет, особенно в рамках решаемой задачи.

Для разработки нейронной сети использована архитектура MobileNet. Среди её преимуществ:

- Низкая требовательность к ресурсам, что допускает возможность её использования в мобильных и других устройствах с низкой вычислительной мощностью.
- Уменьшенное время, необходимое на обучение сети.
- Адаптируемость под различные требования при помощи двух простых и хорошо изученных гиперпараметров.
- Относительная (в сравнении с некоторыми другими нейронными сетями) простота и небольшое количество параметров снижают тенденцию сети к переобучению, когда точность, достигнутая на тренировочных данных, не обобщается на другие входные данные.
- Точность, очень приближенная к другим намного более сложным видам свёрточных нейронных сетей.

Недостатком является то, что свёртки, разделяемые по глубине, несмотря на их эффективность, несколько снижают её точность. Также недостаточно изучена квантизация и в дальнейшей работе планируется исследовать возможность квантизации MobileNet, в частности её производительность и точность после этой процедуры.

В ходе разработки нейронной сети исследовались различные методы [4][5], позволяющие улучшить точность результатов нейронных сетей или достичь более быстрой сходимости при обучении.

При реализации модели на основе MobileNet использованы библиотеки TensorFlow и Keras, облегчающие описание, обучение и эксплуатацию нейронных сетей. Эти библиотеки появились относительно недавно и в них ещё присутствуют пробелы в обычно требуемом функционале. Поэтому были разработаны собственные компоненты для реализации необходимых возможностей.

Таким образом, нейронная свёрточная сеть, разработанная с использованием архитектуры MobileNet, облегчает задачу исследования состояния растений по внешним признакам, позволит выполнять широкомасштабные эксперименты и экономить при этом человеческие ресурсы и время.

Список использованных источников:

1. An Introduction to different Types of Convolutions in Deep Learning. [Электронный ресурс] – Режим доступа: <https://towardsdatascience.com/types-of-convolutions-in-deep-learning-717013397f4d>. – Дата доступа : 12.03.2017.
2. Mobile Real-time Video Segmentation. [Электронный ресурс] – Режим доступа: <https://research.googleblog.com/2018/03/mobile-real-time-video-segmentation.html>. – Дата доступа: 12.03.2017.
3. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. [Электронный ресурс] – Режим доступа: <https://arxiv.org/abs/1704.04861>. – Дата доступа: 09.03.2017.
4. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. [Электронный ресурс] – Режим доступа: <https://arxiv.org/abs/1502.03167>. – Дата доступа: 12.03.2017.
5. The Marginal Value of Adaptive Gradient Methods in Machine Learning. [Электронный ресурс] – Режим доступа: <https://arxiv.org/abs/1705.08292>. – Дата доступа: 09.03.2017.

МЕССЕНДЖЕР С ВАРИАТИВНОСТЬЮ ИСПОЛЬЗОВАНИЯ КРИПТОСИСТЕМ

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

*Матюшоняк А.Д., Валетко А.Н., Ковалёва Н.В., Ширяев Т.С.
Стройникова Е. Д. – ассистент кафедры информатики*

В современном мире большинство информации передаётся через интернет, где её без особого труда можно перехватить, в связи с этим всё больше развиваются и способы её защиты. Самым простым и популярным из них является внедрение различных криптосистем.

Криптосистема – это комплексная модель, состоящая из алгоритмов шифрования и дешифрования, текстов различного объёма и содержания.

В зависимости от типа использованных алгоритмов криптосистемы соответственно делятся на симметричные, асимметричные и смешанные, в которых используются алгоритмы обоих типов. Так, при использовании симметричных криптосистем сообщение шифруется и дешифруется одним ключом, поэтому собеседникам нужно заранее договориться об используемом ключе. В асимметричных криптосистемах используются закрытый и открытый ключи. Открытый ключ пересылается от получателя к отправителю, который с помощью открытого ключа зашифрует сообщение, а получатель дешифрует сообщение с помощью закрытого ключа, который известен лишь ему.

Наибольшая производительность достигается при использовании симметричных методов шифрования, их скорость на несколько порядков выше, длина используемого ключа также заметно меньше, однако зачастую возникают трудности с безопасной передачей ключа. В связи с этим более предпочтительным является использование смешанных криптосистем. В данных криптосистемах сообщение шифруется симметричным способом, отправляется получателю, после чего асимметричным методом отправляется ключ.

Ввиду существования риска перехвата и изменения сообщения после его отправки принято использовать цифровую подпись. Для её создания необходимо вычислить хеш-функцию текста или файла, после чего полученное значение зашифровать с использованием секретного асимметричного ключа отправителя и добавить полученную строку к исходному тексту. Для того чтобы удостовериться в подлинности полученного сообщения, необходимо расшифровать хеш-функцию с использованием открытого ключа отправителя и повторно вычислить хеш-функцию исходного текста. Если обе функции совпадают, делается вывод о сохранности исходного сообщения.

С целью достаточной защиты пользовательских сообщений было разработано приложение Safend, использующее при отправке сообщений смешанную криптосистему и цифровую подпись. Пользователь может комбинировать уже имеющиеся в программе методы симметричного шифрования, произвольно выбирая для них порядок и входные данные, таким образом создавая собственный метод шифрования.

Для реализации данной цели был выбран язык программирования C#, т. к. он располагает большим количеством библиотек асимметричного шифрования, удобен в сетевом использовании для отправки и получения зашифрованных сообщений, а также является кроссплатформенным, что позволит в будущем перенести программу и на мобильные системы.

Переписка может осуществляться как между двумя пользователями, так и в групповом виде. Для переписки между двумя абонентами требуется добавить пользователя в список контактов, указав его ip и дав имя контакту, после чего его можно будет выбрать и начать переписку. Для групповой переписки необходимо, чтобы один из пользователей стал сервером, вызвав в программе соответствующую функцию, после к нему можно будет подключиться, указав его ip.

При добавлении пользователя в контакты создаётся случайный симметричный шифр, который отправляется ему по алгоритму RSA. При получении данный шифр будет сохранён как у получателя, так и у отправителя. В дальнейшем при их переписке по умолчанию будет использоваться именно это шифрование. Однако при желании пользователь может зашифровать сообщение собственным симметричным ключом, который будет отправлен по алгоритму RSA.

Таким образом, при переписке может быть использовано вплоть до трёх уровней защиты информации, и для того, чтобы удостовериться в том, что сообщение не было перехвачено и изменено, может быть прикреплена цифровая подпись. Подобный подход обеспечивает необходимый уровень защиты информации, передаваемой посредством сети интернет.

Список использованных источников:

1. Шифрование [Электронный ресурс]. – Режим доступа: https://professorweb.ru/my/csharp/base_net/level2/2_3.php. – Дата доступа: 08.04.2018.

2. Криптографические методы и средства защиты информации [Электронный ресурс]. – Режим доступа: <http://itsphera.ru/it/cryptographic-methods-and-tools-for-information-protection.html>. – Дата доступа: 08.04.2018.

ПРИМЕНЕНИЕ РЕКУРСИВНЫХ НЕЙРОННЫХ СЕТЕЙ ДЛЯ АНАЛИЗА ТОНАЛЬНОСТИ ТЕКСТА

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

Витковский А.В.

Жвакина А. В. – к. т. н, доцент

Современные порталы в сети Интернет позволяют пользователям высказывать свое мнение о различных предметах, событиях, явлениях. Эти мнения могут быть полезными для различных исследований, необходимых аналитикам, SMM-специалистам, бренд-менеджерам, PR-агентам и иным специалистам, нуждающимся в получении агрегированной информации. Таким образом существует необходимость в инструментах для анализа отзывов пользователей. Пользователи оставляют свои комментарии в интернете на естественных языках, таких как английский, русский и др., что является проблемой для обработки программными средствами. В работе рассматривается такой метод обработки естественного языка, как анализ тональности текста.

Обработка естественного языка (Natural Language Processing, NLP) – общее направление искусственного интеллекта и математической лингвистики. Оно изучает проблемы компьютерного анализа и синтеза естественных языков. Одной из задач, решаемых в рамках обработки языка, является анализ тональности текста. Анализ тональности текстов – это класс методов анализа содержания, предназначенный для классификации автоматического распознавания в тексте лексики с эмоциональной окраской, а также мнений (эмоциональных оценок) автора об объектах, которые упоминаются в тексте.

Для решения задачи анализа тональности текста применяют нейронные сети. Искусственные нейронные сети (НС) — совокупность моделей биологических нейронных сетей. Представляют собой сеть элементов — искусственных нейронов — связанных между собой синаптическими соединениями. Сеть обрабатывает входную информацию и в процессе изменения своего состояния во времени формирует совокупность выходных сигналов. Работа сети состоит в преобразовании входных сигналов во времени, в результате чего меняется внутреннее состояние сети и формируются выходные воздействия. Обычно НС оперирует цифровыми, а не символьными величинами. Искусственные нейронные сети — набор математических и алгоритмических методов для решения широкого круга задач. Алгоритмы на основе машинного обучения показывают свою эффективность в задачах обработке естественных языков.

Большинство систем прогнозирования настроений работают по простому алгоритму, рассматривая слова в изоляции, давая положительные баллы для положительных слов и отрицательные баллы для отрицательных слов, а затем суммирует баллы. Таким образом, порядок слов игнорируется и теряется важная информация. Однако, модели на основе нейронных сетей фактически создают представления целых предложений, основанные на структуре предложения. Они вычисляют тональность, основанную на том, как слова влияют значение более длинных фраз. Таким образом, модель на основе нейронных сетей не так легко обмануть, как обычные алгоритмы.

Наиболее часто используемыми в исследованиях методами являются методы на основе машинного обучения с учителем. Сутью таких методов состоит в том, что сначала нейронная сеть получает коллекцию из данных и уже готовых точных решений, и в процессе машинного обучения сеть настраивается, и далее уже на других данных может выдавать нужные результаты.

Для решения задачи анализа тональности могут использоваться рекурсивные нейронные сети. Рекурсивные нейронные сети (англ. Recursive neural network; RvNN) – вид нейронных сетей, работающих с данными переменной длины. Модели рекурсивных сетей используют иерархические структуры образцов при обучении. Например, изображения, составленные из сцен, объединяющих подсцены, включающие много объектов. Выявление структуры сцены и её деконструкция – нетривиальная задача. При этом необходимо как идентифицировать отдельные объекты, так и всю структуру сцены. В рекурсивных сетях нейроны с одинаковыми весами активируются рекурсивно в соответствии со структурой сети. В процессе работы рекурсивной сети вырабатывается модель для предсказания для структур переменной размерности, так и скалярных структур через активацию структуры в соответствии с топологией. Рекурсивные нейронные сети успешно применяются при обучении последовательных структур и деревьев в задачах обработки естественного языка, при этом фразы и предложения моделируются через векторное представление слов. Рекурсивные сети первоначально появились для распределённого представления структур, используя

предикаты математической логики. Разработки рекурсивных сетей и первые модели появились в середине 1990-х.

Рекурсивные нейронные сети работают с векторными представлениями слов. Векторное представление слов – класс методов и подходов для обработки естественного языка, суть которых состоит в том, что слову словаря ставится в соответствие n-мерный вектор. Для получения векторных представлений используются нейронные сети, которые после обучения могут автоматически выполнять указанную задачу. Векторные представления позволяют программе работать со значениями слов, т.к. смысл векторных представлений состоит в том, чтобы для слов со схожими значениями получались близкие вектора, а для далеких понятий расстояние должно быть большое. На практике используются достаточно большие размерности, от 200 до 500. Таким образом для огромного количества слов из словаря можно получить векторы, которые могут обрабатываться на ЭВМ. С увеличением размерности результат анализа становится точней, а скорость работы и обучения замедляется.

Список использованных источников:

1. Заенцев, И. В. Нейронные сети: основные модели/ И. В. Заенцев. – Воронеж, 1999. – 76 с.
2. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank – Stanford University, 2013.
3. Википедия:Рекурсивные нейронные сети. [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Рекурсивные_нейронные_сети . – Дата доступа: 25.03.2018.
4. Википедия: Анализ тональности текста. [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Анализ_тональности_текста . – Дата доступа: 25.03.2017.

SQRT-ДЕРЕВО КАК НЕЗАСЛУЖЕННО ЗАБЫТАЯ СТРУКТУРА ДАННЫХ

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

Вишневецкий В.А.

Гербик А.И. – ассистент

В рамках данной работы будет освещена структура данных под названием sqrt-дерево и её преимущество перед другими открытыми структурами данных.

Для объяснения принципа работы sqrt-дерева будет проще рассмотреть следующую теоретическую задачу: “Дан набор из N чисел. На некоторых его отрезках требуется посчитать произведение чисел по заранее заданному модулю M”.

Описание sqrt-дерева лучше начать с идеи разбиения массива на блоки длины \sqrt{N} (здесь и далее под \sqrt{N} имеется в виду целая часть), за исключением последнего блока, который может иметь размер меньше, чем \sqrt{N} . Принцип разбиения указан на рисунке 1:

$$\left\{ a_1, \dots, a_{\sqrt{N}} \right\}, \left\{ a_{\sqrt{N}+1}, \dots, a_{2\sqrt{N}} \right\}, \dots, \left\{ a_{\left\lfloor \frac{N}{\sqrt{N}} \right\rfloor \cdot \sqrt{N} + 1}, \dots, a_N \right\}$$

Рис. 1 – принцип разбиения на блоки

Предпросчитаем для каждого блока: произведение всех чисел, принадлежащих ему, произведение чисел на каждом префиксе и суффиксе этого блока. А также для каждого отрезка блоков посчитаем произведение чисел. В итоге суммарное время работы предпросчета будет $O(N\sqrt{N})$. Используя эти данные, мы уже можем решать вышеприведенную задачу за $O(\sqrt{N})$ времени на запрос. Заметим, что мы можем отвечать за $O(\sqrt{N})$ на все запросы, кроме тех, которые затрагивают элементы ровно одного блока. Чтобы эффективно отвечать на такой тип запросов, мы можем построить в этом блоке такую же структуру данных, как была описана выше. Будем выполнять это действие рекурсивно для каждого блока, который имеет размер больше единицы. Таким образом мы имеем древообразную структуру, изображенную на рисунке 2:

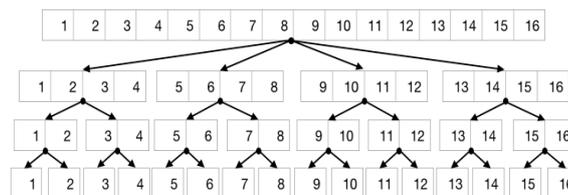


Рис. 2 – sqrt-дерево

На каждом слое находится ровно \sqrt{N} вершин, поэтому для оценки сложности работы нам достаточно найти количество слоёв. Давайте заметим, что новый слой создается только когда предыдущий слой имеет

размер больше одного, а значит нам нужно найти количество взятий операции корня к числу N для того, чтобы оно обратилось в единицу. Количество таких операций равно). Теперь у нас есть всё, чтобы оценить необходимые ресурсы для работы структуры данных:

- Время работы предпроцесса: .
- Необходимая память: .
- Время ответа на один запрос: или , при использовании битовых операций для определения нужного слоя.
- Время модификации элемента: .

В данной структуре без проблем можно использовать любую операцию, для которой выполняется свойство ассоциативности. Для примера, такими операциями являются: сложение, умножение, а также все битовые операции.

Немного о скорости относительно других известных структур данных на бенчмарке состоящем из запросов и элементов.

алгоритм\ количество модификаций	0	100		
Segment tree	10.11 сек.	10.28 сек.	11.78 сек.	12.54сек.
Sqrt tree	2.01 сек.	2.28сек.	4.92сек.	13.28сек.
Sqrt decomposition	138.17 сек.	138.75сек.	140.32сек.	162.83сек.

Как видно структура работает в несколько раз быстрее, чем другие структуры данных, пока дело не доходит до большого числа изменений.

Список использованных источников:

1. N Alon, B Schieber, Optimal preprocessing for answering on-line product queries
2. Т. Кормен, Алгоритмы. Построение и анализ
3. Sqrt Decomposition, e-maxx.ru

ГРАФОВЫЕ БАЗЫ ДАННЫХ

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

Чурин А.П., Владыко В.Д.

Теслюк В.Н. – к.ф-м.н., доцент

В настоящее время есть потребность для обработки разноордных данных, которые имеют непредсказуемую структуру. Которая может превратиться либо в BigData, либо в сложную семантическую сеть, и зачастую мы не можем предвидеть, какой она будет.

Как альтернатива базам данных SQL с 2000-х годов развивается направление NoSQL. В эту категорию попадают – от иерархических и сетевых БД до упрощённых БД, которые имеют подобие хэш-таблиц, и документарные БД. Причиной эволюции базы данных заключается в следующем: если программный продукт оперирует однотипными и примитивными данными, то можно использовать классический подход к разработке БД. Но если нужно обратиться к 10, 100, 1000 таблицам, чтобы получить данные, то реляционная база данных начинает работать медленно, а написание такого запроса займет довольно много времени.

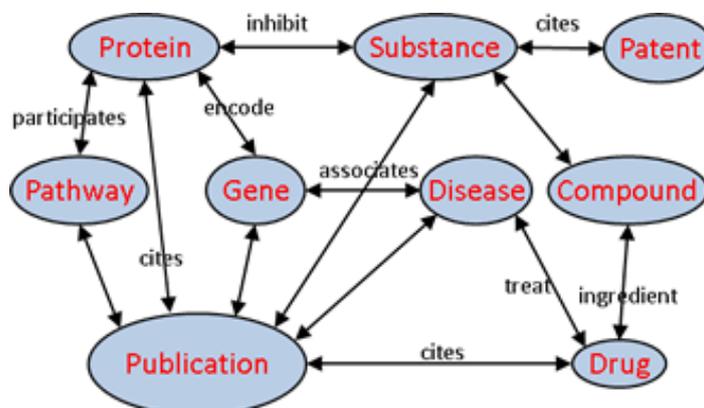


Рис. 1 – Иллюстрация графовой базы данных

Наконец отдельным классом стоят NoSQL – к ним относят графовые базы данных. Они располагают функционалом для естественного предоставления информации. Прямая альтернатива SQL – документарные базы данных, их главная особенность – это отсутствие схемы, присущие реляционным базам данных.

Преимущества графовых баз данных:

- База может сохранять любой объект, например объект с большим количеством полей.
- Гибкая система расширения и модернизация в любой момент времени.

Недостатки графовых баз данных:

- Проблемы с работой в параллельных архитектурах.
- Низкая производительность при большом количестве связей и больших объемах.

Графовые базы данных оказываются универсальным вариантом, которые помогают подстраховаться на случай изменения требований и увеличением либо уменьшением функционала. Добавление новых связей делает непригодной для использования документарную базу данных, а рост количества сырых объектов с несвязанной структурой снижает производительность реляционной БД. Сегодня ведется активная разработка и доработка RDF – основного стандарта, согласно которому работают графовые базы данных. Именно стандартизация SQL делала популярным реляционные БД. При этом ряд проектов демонстрируют поддержку OData, для создания классических веб-запросов через HTTP. Язык SPARKQL, обладающий возможностями для работы с вариативными видами запросов и данных. За счет развития архитектуры производительность графовых баз данных тоже растет, возможно в будущем за счет развития окажется выше реляционной даже при небольшом количестве связей.

Список использованных источников:

1. Ян Робинсон, Джим Вебер. Графовые базы данных. Новые возможности для работы со связанными данными. 2016. – 256с.

КЛАСТЕРИЗАЦИЯ В HADOOP

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

Владыко В.Д., Чурин А.П.

Теслюк В.Н. – к.ф.-м.н., доцент

Непрерывный рост данных и увеличение скорости их генерации порождают проблему их обработки и хранения. Неудивительно, что тема «больших данных» (Big Data) является одной из самых обсуждаемых. Одним из самых известных проектов в области распределенных вычислений является Hadoop — набор из утилит, библиотек и фреймворк для разработки и выполнения программ распределенных вычислений. Кластеризация один из важных моментов работы с большими данными.

Hadoop – это проект с открытым исходным кодом, находящийся под управлением Apache Software Foundation. Hadoop используется для надежных, масштабируемых и распределенных вычислений, но может также применяться и как хранилище файлов общего назначения, способное вместить петабайты данных.

В состав проекта Hadoop входят следующие подпроекты:

Common — набор компонентов и интерфейсов для распределенных файловых систем и общего ввода-вывода;

MapReduce — модель распределённых вычислений, предназначенная для параллельных вычислений над очень большими (до нескольких петабайт) объемами данных;

HDFS — распределенная файловая система, работающая на больших кластерах типовых машин.

Классическая конфигурация кластера Hadoop состоит из одного сервера имён (NameNode), одного мастера MapReduce (т.н. JobTracker) и набора рабочих машин, на каждой из которых одновременно крутится сервер данных (DataNode) и обработчик (TaskTracker).

namenode — сервер имён. Как правило, один узел на кластер. Хранит в себе все метаданные системы, сами файлы на этом узле не хранятся.

JobTracker — узел, который координирует параллельную обработку данных используя MapReduce.

TaskTracker — узел, который принимает задачи по обработке данных от JobTracker.

DataNode — таких узлов в кластере очень много. Они хранят непосредственно блоки файлов. Узел регулярно отправляет NameNode свой статус и ежечасно — информацию обо всех хранимых на этом узле блоках. Это необходимо для поддержания нужного уровня репликации.

При правильной архитектуре приложения, с помощью информации о том, на каких машинах (узлах) расположены блоки данных, позволяет запустить на них же вычислительные процессы и выполнить большую часть вычислений локально, т.е. не передавая данные по сети.

Кластеры в Hadoop позволяют ускорить анализ данных для приложений и улучшить их масштабируемость. Если при растущих объемах данных, кластер начинает не справляться, то можно добавить дополнительные узлы для увеличения пропускной способности. Также кластер обладает высокой отказоустойчивостью, поскольку каждый блок данных копируется на другие узлы, гарантируя, что данные не будут потеряны, даже если один из узлов выдаст ошибку.

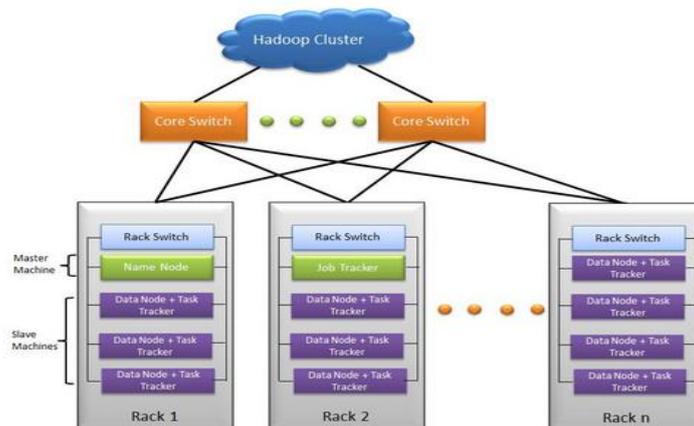


Рис. 1 - Структура кластера Hadoop

Список использованных источников:

1. Kevin T. Smith, Boris Lublinsky. Professional Hadoop Solutions. 2013.
2. Sammer E. Hadoop Operations. 2012 - 41с.
3. Tom White. Hadoop: The Definitive Guide: Storage and Analysis at Internet Scale. 2015 - 283с.

МЕТОДЫ УЛУЧШЕНИЯ АЛГОРИТМА Q-ОБУЧЕНИЯ

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

Волчек А.Ю., Соболев А.В.

Волорова Н.А. – к.т.н., доцент

В настоящее время применение машинного обучения помогает людям решать задачи в разных сферах жизни: медицина, физика, химия, системы безопасности, игровая индустрия и др. Уже сейчас есть прототипы самодвижущихся автомобилей, которые сами контролируют движение транспорта. Одним из главных методов для обучения такого рода искусственного интеллекта является метод обучения с подкреплением – самостоятельное и уже вполне сформировавшееся направление кибернетических исследований, одна из разновидностей которого – Q-обучение.

Q-обучение — метод, применяемый в искусственном интеллекте при агентном подходе. Относится к экспериментам вида обучения с подкреплением. На основе получаемого от среды вознаграждения агент формирует функцию полезности Q, что впоследствии дает ему возможность уже не случайно выбирать стратегию поведения, а учитывать опыт предыдущего взаимодействия со средой. Одно из преимуществ Q-обучения — то, что оно в состоянии сравнить ожидаемую полезность доступных действий, не формируя модели окружающей среды. Применяется для ситуаций, которые можно представить в виде марковского процесса принятия решений.

Q-обучение является важной вехой, однако известны некоторые ограничения этого алгоритма, для которых предложены несколько улучшений. В работе предложено несколько улучшений, в каждом из которых поясняется причина ограничений. Мы постарались выделить разноплановые улучшения, относящиеся к различным этапам алгоритма.

Двойное Q-обучение. Параметры нейросети в глубоком Q-обучении оптимизируются стохастическим градиентным спуском, минимизируя функцию потерь:

(1)

В работе (vanHasselt 2010) показано, что из-за применения операции максимизации в уравнении (1) оценка Q-функции почти всегда является смещённой. Предложенный в данной работе подход позволяет частично избавиться от этой проблемы. Предлагается поддерживать рядом ещё одну нейросеть (targetnetwork), параметры в которую копируются из основной раз в несколько итераций. Вместо максимизации в уравнении (1) мы сначала выбираем действие с максимальным значением Q-функции, основываясь на данных из основной сети, но дальше используем значения Q-функции для этого действия из дополнительной сети.

Приоритизированный буфер опыта. Буфер опыта (experienceplay) значительно ускоряет обучение и улучшает его стабильность, позволяя обучаться на ранее виденных ситуациях. В обычном его варианте мы выбираем обучающее множество равновероятно среди всех его элементов. Очевидно, что на практике существует много “простых” ситуаций, для которых сеть выучила хорошую аппроксимацию Q-функции и некоторое количество “сложных”. В приоритизированном буфере опыта мы выбираем каждый элемент в

обучающее множество с вероятностью, пропорциональной значению функции потерь на этом элементе во время предыдущей итерации обучения.

Многошаговое Q-обучение. Обычное Q-обучение обновляет значение Q-функции на основе награды на данном шаге и Q-значения на следующем шаге для жадно выбранного наилучшего действия. Альтернативой этому подходу является использование аккумулированной награды за несколько шагов.

Предложенные методы улучшения алгоритма Q-обучения позволяют ускорить обучение и увеличить стабильность получаемых результатов. Мы рассказали про некоторые улучшения алгоритма Q-обучения, которые могут быть внедрены в стандартный алгоритм для получения state-of-the-art производительности. Кроме них существует множество других возможных кандидатов, которые мы не рассмотрели в данной работе.

Список использованных источников:

1. vanHasselt, H. 2010, DoubleQ-learning. "Advances in Neural Information Processing Systems".
2. Sutton, R. S., and Barto, A. G. 1998. "Reinforcement Learning: An Introduction". The MIT press, Cambridge MA.
3. Schaul, T.; Quan, J.; Antonoglou, I.; and Silver, D. 2015. "Prioritized experience replay". In proc of ICLR.
4. van Hasselt, H; Guez, A.; and Silver, D. 2016. "Deep reinforcement learning with double Q-learning".
5. Sutton, R. S. 1988. "Learning to predict by the method of temporal differences".

АСПЕКТЫ СОЗДАНИЯ УНИВЕРСАЛЬНОЙ СИСТЕМЫ ОРГАНИЗАЦИИ УЧЁТА И ОТЧЁТНОСТИ ПРЕДПРИЯТИЯ В СФЕРЕ УСЛУГ

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

Воробей Ю.Г.

Черняевский Ю. А. – к. т. наук, доцент

На сегодняшний день лучшим решением при выборе типа приложения является универсальное приложение. Универсальное приложение может эксплуатироваться пользователями разных направлений одной сферы, где используются одни и те же функции. Например, в сфере услуг организацией работы с клиентами может заниматься как медицинские центры, так и салоны красоты. Данные для заполнения будут разные, однако функции, которые позволяют организовать запись клиентов, исследовать их поток и другие, одинаковы в обоих случаях.

Такие системы должны удовлетворять следующим критериям. Для начала они имеют глобальный характер, поэтому должны быть доступны для разных регионов мира. Для этого требуется вводить не только поддержку разных языков, но и поддержку разных способов представления информации, которая может отличаться в разных странах, таких как представление дробных чисел или валюты.

Одной из наиболее явных характеристик универсальных систем является их большая нагрузка из-за количества пользователей разных направлений. Поэтому важным критерием является устойчивость системы к сбоям и её быстродействие. Правильная архитектура расположение кэшей и создание пула соединений с базой данных поможет организовать быстродействующую систему, а различные инструменты по проверке качества кода и поиску возможных багов, а также разнообразное тестирование, поможет создать систему, достаточно устойчивую к возникновению ошибок.

Также для лучшего отклика системы необходимо установить так называемые "жадные таймауты". Это позволит пользователю не ждать долго ответа системы. При жадном таймауте пользователь просто получит сообщение, и запрос повторится. Главное поставить таймаут правильно, чтобы система не выкидывала его слишком часто, но и чтобы достаточно быстро реагировала.

Также важным критерием является быстрый поиск по запрашиваемым данным. Для этого как раз подходит организация системы кэшей и пула соединений с базой данных. Однако необходимо учитывать и правильность организации схемы базы данных, а также различные инструменты базы для организации данных и быстрого поиска.

Как ранее говорилось, один из наиболее важных критериев по созданию универсальной системы – её устойчивость к возникновению ошибок. Для этого систему тщательно тестируют и проводят через дополнительные инструменты по определению качества кода. В процессе выявляются важные аспекты, влияющие на её работу. Например, при функциональном тестировании можно определить, какие моменты не учтены при разработке системы, а при нагрузочном – сколько одновременных клиентов может выдержать система.

Таким образом, необходимо учесть множество критериев, которые влияют на создание универсальных систем. Поэтому создание таких систем является довольно трудоёмким процессом.

ИССЛЕДОВАНИЕ ПОГРЕШНОСТЕЙ ЧИСЛЕННЫХ МЕТОДОВ РЕШЕНИЙ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ В ЗАВИСИМОСТИ ОТ БАЗИСА РАЗЛОЖЕНИЯ ФУНКЦИИ

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

Литвинчук Д.В. Воротынцев Д.Д.

Калугина М.А. – к.т.н., доцент

Дифференциальные уравнения имеют многочисленные и самые разнообразные приложения в механике, физике, астрономии, технике и в других разделах высшей математики (например, в математической физике вариационном исчислении, теоретической механике, дифференциальной геометрии). При изучении некоторых явлений в физике, технических науках иногда бывает трудно найти соотношения между величинами, характеризующими данное явление, однако легко находится зависимость между этими величинами и их производными. Однако решения многих дифференциальных уравнений не выражаются в элементарных функциях или найти их очень сложно. В этих случаях пользуются приближенными методами решения дифференциальных уравнений. Одним из таких методов является представление решения уравнения в виде ряда.

В нашей работе были исследованы зависимость погрешности от выбранной ортогональной системы при решении дифференциальных уравнений высших порядков.

Одним из главных вопросов, возникающих при численном решении вычислительных задач, является оценка достоверности полученного результата. Под **погрешностью** понимается некоторая величина, характеризующая точность результата.

Определим последовательность многочленов степени формулой

(1)

и назовём их **многочленами Чебышева - Эрмита**. При $n = 0$ формула (1) теряет смысл, но мы положим просто

Полиномы Лежандра могут быть определены следующей дифференциальной формулой, называемой формулой Родрига: (2). Для полагаем по определению

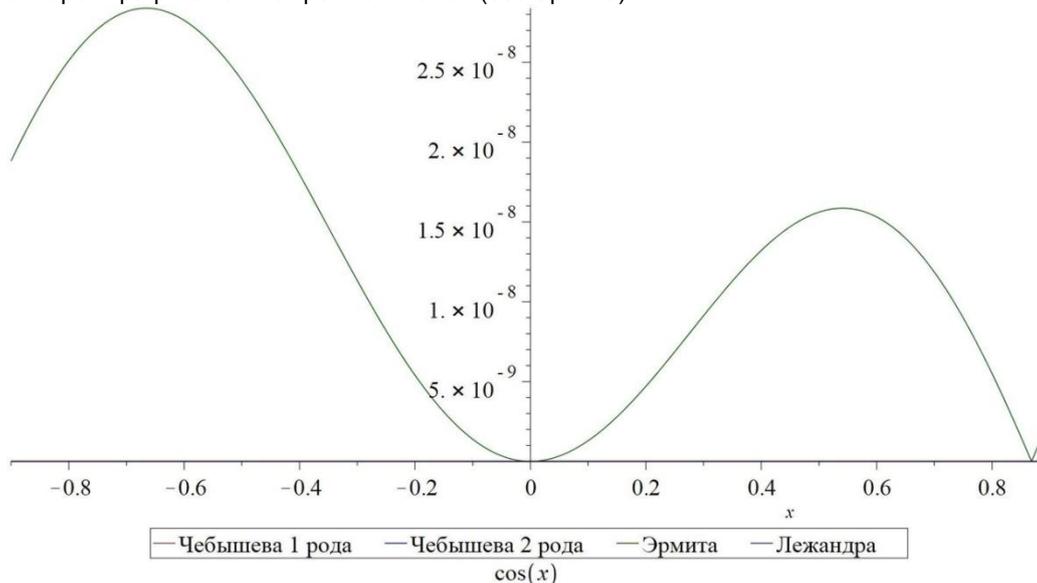
Полиномами Чебышева 1-го рода называются функции ,

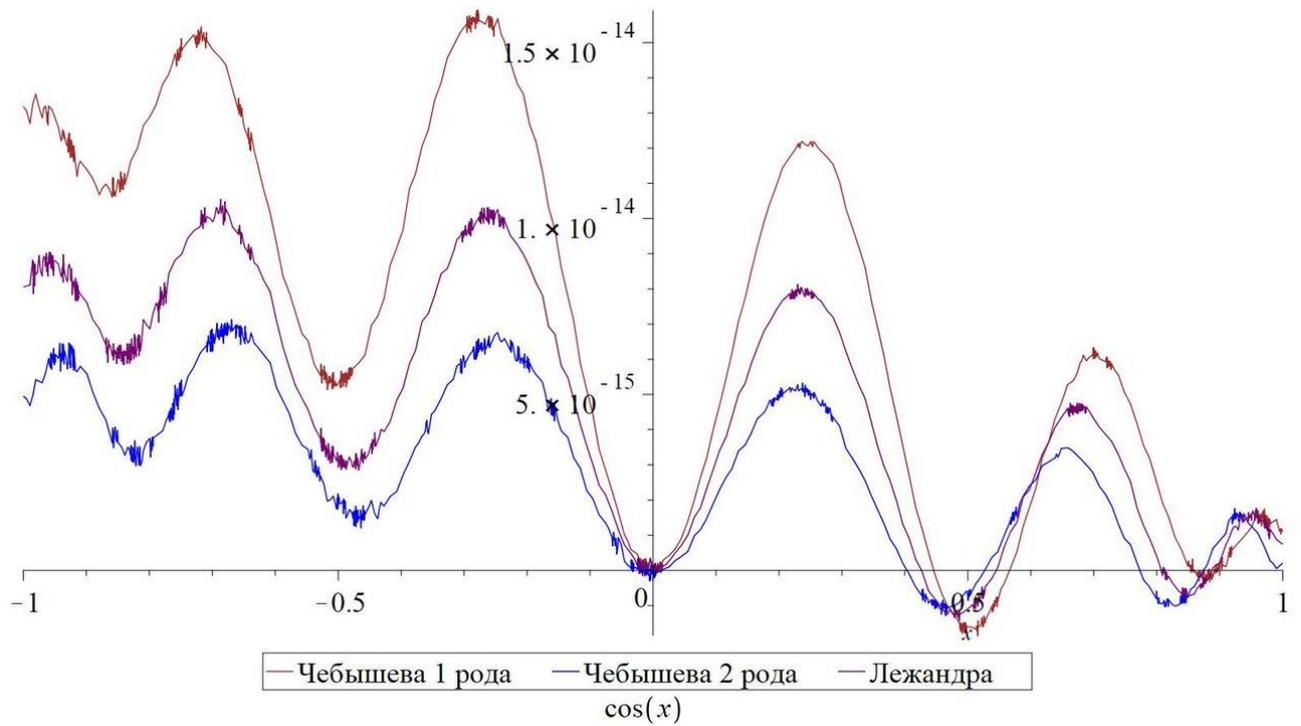
Полиномами Чебышева 2-го рода называются функции ,

В результате работы были рассмотрены 4 семейства функций: $\sin(ix)$, $\cos(ix)$, $ix\sin(x)$, $ix\cos(x)$. Все они дали приблизительно одинаковый результат (см. графики): на небольших значениях i решение ДУ с помощью полиномов Эрмита дает худший результат, с погрешностью порядка 10^{-7} , остальные - порядка 10^{-12} . При значениях i около 10-11, погрешности, даваемые всеми полиномами, выравниваются, далее полиномы Эрмита дают лучшее приближение. Остальные полиномы располагаются в следующем порядке (по ухудшению): полиномы Чебышева 2 рода, полиномы Лежандра, полиномы Чебышева 1 рода. Все они дают приблизительно одинаковые погрешности, но, тем не менее, изменение порядка эффективности на рассмотренных участках не наблюдалось.

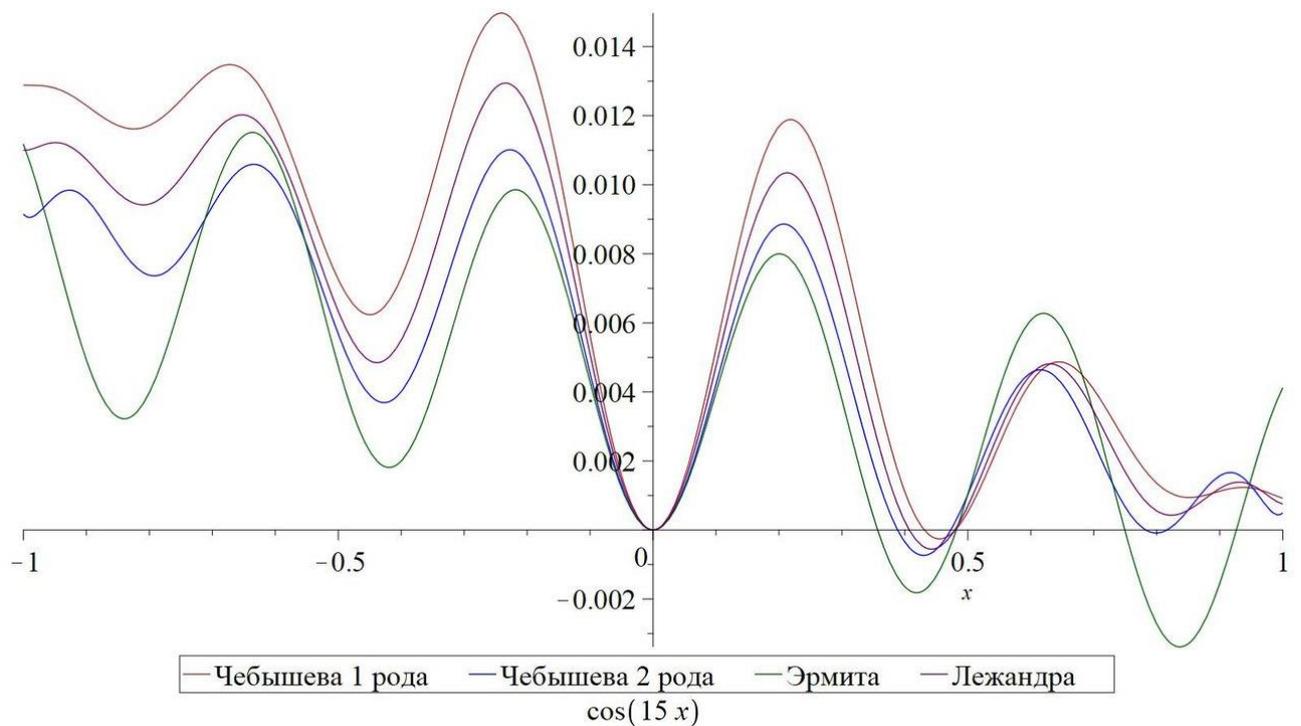
На графиках ниже представлена погрешность при разложении функции $\cos(x)$ по ортогональным полиномам. На первом графике представлены все четыре полинома: Эрмита, Лежандра, Чебышева 1 и 2 рода.

На втором графике только трех полиномов (без Эрмита)





На графике ниже представлена погрешность при разложении $\cos(15 \cdot x)$ по всем четырем ортогональным полиномам



- Список использованных источников:
1. Балакин А.Б. Классические ортогональные полиномы.
 2. Суевин П. С. Классические ортогональные многочлены
 3. Геронимус Я.Л. Теория ортогональных многочленов (Обзор достижений отечественной математики) - М.:Гостехиздат, 1950. 164 с.

ГРАФИЧЕСКОЕ ПОСТРОЕНИЕ ФРАКТАЛОВ ЖЮЛИА И МАНДЕЛЬБРОТА. ПРИМЕНЕНИЕ ФРАКТАЛОВ ДЛЯ ВЫЧИСЛЕНИЯ ФУНКЦИИ ВЕЙЕРШТРАССА

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

Григорьев К.П., Купчинов Л.А.
Стройникова Е. Д. – ассистент кафедры информатики

Задавшись вопросом, какой раздел математики появился только в XX веке и развивается с каждым годом, несложно найти ответ – теория хаоса. Фракталы всегда ассоциируются с теорией хаоса, они проявляют хаотическое поведение, благодаря которому они кажутся такими беспорядочными и случайными. Точное определение термина фрактал (лат. *fractus* – дроблённый, сломанный, разбитый) – множество, обладающее свойством самоподобия (объект, в точности или приближённо совпадающий с частью себя самого, т. е. целое имеет ту же форму, что и одна или более частей). В математике под фракталами понимают множества точек в евклидовом пространстве, имеющие дробную метрическую размерность (в смысле Минковского или Хаусдорфа) либо метрическую размерность, отличную от топологической, поэтому их следует отличать от прочих геометрических фигур, ограниченных конечным числом звеньев.

В настоящее время учёные определяют 3 вида фракталов:

- геометрические фракталы;
- алгебраические фракталы;
- стохастические фракталы.

С точки зрения математики наиболее интересными являются алгебраические фракталы. Наиболее известные из них – фракталы Жюлиа и Мандельброта. Они задаются точками на комплексной плоскости при помощи рекуррентной последовательности $z_n = F(z_{n-1})$, где $F(z) = z^2 + c$ и каждый член последовательности – комплексное число. Нас интересует поведение этой последовательности при стремлении n к бесконечности. Эта последовательность может:

- стремиться к бесконечности;
- стремиться к конечному пределу;
- демонстрировать в пределе циклическое поведение;
- вести себя хаотично, т. е. не демонстрировать ни один из трёх упомянутых типов поведения.

Во множество Жюлиа входят такие точки z_0 на комплексной плоскости, которые под действием рекуррентной последовательности будут резко менять поведение всей последовательности. А множество Мандельброта – это множество всех $c \in \mathbf{C}$, при которых z_0 для заданного z_0 не стремится к бесконечности. Была разработана программа построения фракталов Жюлиа и Мандельброта с использованием языка программирования C++. Выбор C++ был сделан в основном из-за производительных способностей этого языка и удобства графической библиотеки SFML, которая хорошо подходит для работы с точками или массивами точек. Программа следующим образом строит примеры множеств Жюлиа (рис. 1) и Мандельброта (рис. 2):

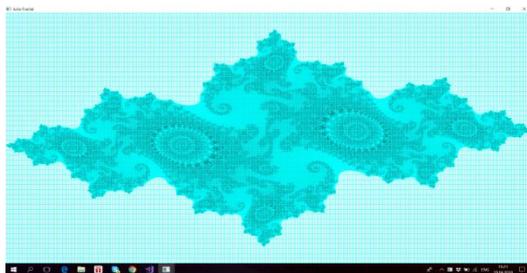


Рис. 1 – Пример фрактала Жюлиа

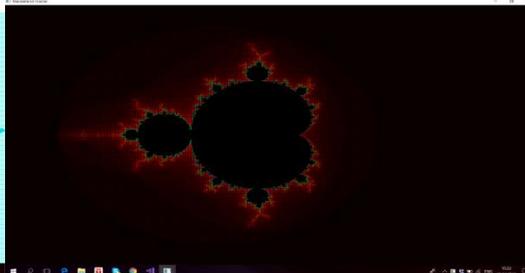


Рис. 2 – Пример фрактала Мандельброта

На вход программы поступают действительная и мнимая части начального комплексного числа c , полагается $z_0 = c$ для построения множества Жюлиа и $z_0 = 0$ для построения множества Мандельброта в рекуррентной последовательности $z_n = z_{n-1}^2 + c$ и итерационно проверяется каждый пиксель данного экрана (экран представляется, как комплексная плоскость) на уход комплексного числа z_n в бесконечность. Выход из итерации происходит при условии $|z_n| > 2$. Цвет каждого пикселя зависит от номера той итерации, когда число уходит в бесконечность.

Функция Вейерштрасса – пример непрерывной функции, нигде не имеющей производной, контрпример для гипотезы Ампера. Функция Вейерштрасса задается на всей вещественной прямой единым аналитическим выражением

$$w(x) = \sum_{n=0}^{\infty} b^n \cos(a^n \pi x),$$

где a – произвольное нечетное число, не равное единице, а b – положительное число, меньшее единицы. Этот функциональный ряд мажорируется сходящимся числовым рядом

$$\sum_{n=0}^{\infty} b^n .$$

Функция Вейерштрасса представляет собой пример функции, график которой имеет фрактальную структуру (рис. 3).

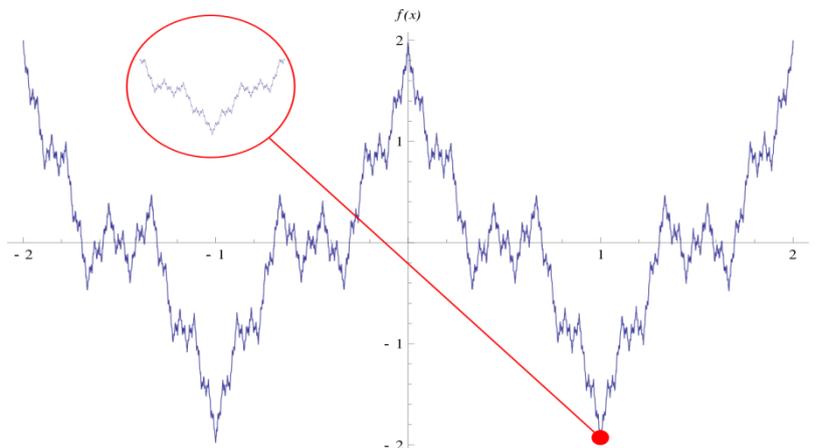


Рис. 3 – График функции Вейерштрасса с параметрами $a = 3$, $b = 1/2$

В настоящее время теория хаоса недостаточно хорошо изучена, ведь нахождение с первого взгляда случайных законов очень сложно. Фракталы являются примерами таких законов, поэтому, изучая их, мы можем продвигаться в изучении теории хаоса. Таким образом, тема фракталов может быть хорошим вариантом для знакомства студентов с теорией хаоса.

Список использованных источников:

1. Мандельброт, Б. Фрактальная геометрия / Б. Мандельброт. – Ижевск : Институт компьютерных исследований, 2002. – 656 с.
2. Свободная энциклопедия Википедия, статья «Фрактал» [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Фрактал>.
3. Свободная энциклопедия Википедия, статья «Функция Вейерштрасса» [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Функция_Вейерштрасса.

ОНЛАЙН-МАГАЗИН ПРОИЗВОДСТВЕННО-ТОРГОВОГО УНИТАРНОГО ПРЕДПРИЯТИЯ

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

Грузинов А.Г.

Жвакина А.В. – к.т.н., доцент

На определенном этапе развития предприятия, перед ним встает вопрос о расширении рынка сбыта и поиске новых клиентов. Существует множество методов решения данных задач, разного уровня сложности и эффективности. В информационном обществе можно найти новых клиентов быстро и эффективно с помощью WEB-сервисов.

В результате работы создан онлайн-магазин, который играет роль не только субъекта торговли, но и дает исчерпывающую информацию о деятельности предприятия, возможность быстрой связи с продавцом, а также определяет поле для возможного сотрудничества. Удобный для пользователя интерфейс и функциональность играют огромную роль при выборе товара. Сервис отвечает всем современным нормам, уникальный дизайн и функциональность добавляют популярности и обратят на себя внимание как можно большего числа лиц.

Создание данного сервиса дает возможность облегчить работу отдела сбыта продукции за счет автоматизации покупки клиентом товара. Также открывается возможность сократить количество работников занятых в отделе сбыта. Однако появляется необходимость поддерживать рабочее состояние сервиса. Проведенные экономические расчеты позволяют утверждать, что создание сервиса оправдано.

Проект делится на две условные части в зависимости от платформы, на которой он используется. Так для всех устройств с браузером доступен сайт, а для пользователей с ОС Андроид доступна дополнительно мобильная версия.

Основные функции, которые присутствуют в обеих версиях:

- Регистрация (упрощает дальнейшую работу с покупкой).
- Обратная связь (возможность написать/позвонить в реальном времени).

- Удобный просмотр объектов торговли (все систематизировано и в полной степени описано).
 - Оформление заказа.
- Достоинства данного сервиса:
- Удобный, быстрый, вариативный способ заказа.
 - Безопасность данных пользователя.
 - История покупок пользователя.
 - Удобный и информативный интерфейс.
 - Максимально полное и подробное описание торговых объектов.
 - Наличие специального Андроид приложения.
 - Гибкая и изменяемая серверная часть.
 - Возможность легкой модернизации за счет новейших технологий разработки.
- На рисунке 1 представлен скриншот интерфейса Web-сервиса.

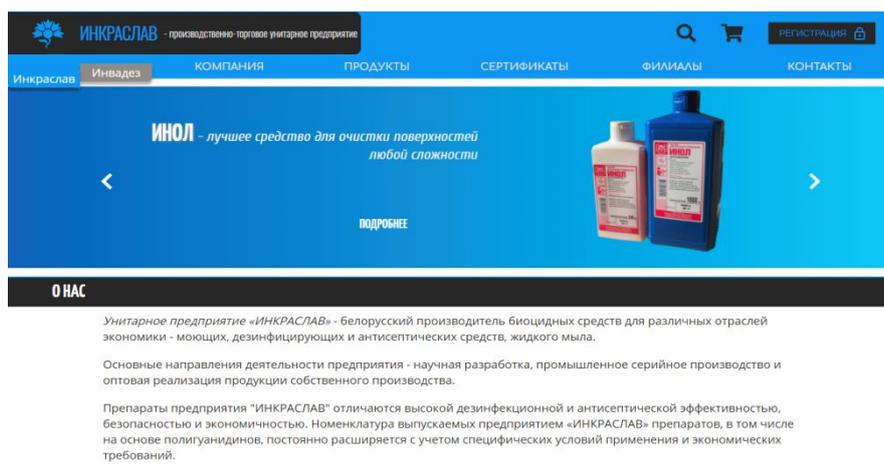


Рис. 1 – Интерфейс WEB-сервиса

Итогом работы является полностью рабочий Web-сервис и приложение, которые привлекут новых клиентов и обеспечат рост производства за счет качества товара и предоставления услуг на Web-сервисе.

Список использованных источников:

1. O'Reilly Media, Web Design in a Nutshell, 3rd Edition, A Desktop Quick Reference. O'Reilly Media. 2009, 832 P.
2. Thomas Connolly, Database Systems: A Practical Approach to Design, Implementation and Management with Learning SQL: A Step-By-Step Guide Using Oracle with Learning SQL. Addison Wesley. 2004.
3. O'Reilly Media. Learning React. Functional Web Development with React and Redux. O'ReillyMedia. 2017. 350 P.

РАСПОЗНАВАНИЕ ЖАНРА МУЗЫКАЛЬНЫХ КОМПОЗИЦИЙ С ПОМОЩЬЮ МАШИННОГО ОБУЧЕНИЯ

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

Дектярёв В.И.

Жвакина А. В. – к. т. наук, доцент

В настоящее время, в виду постоянно развивающихся веб-технологий и закономерно растущей численностью разнообразных сервисов, предоставляющих пользователям некоторую информацию, растёт конкуренция в различных сферах. В частности, появляется всё больше музыкальных сервисов. Для того, чтобы быть успешным на рынке, каждый из них должен выделиться некоторым функционалом, более особенным, чем возможность прослушивания композиций. Одним из вариантов такого функционала является анализ жанровых предпочтений пользователя по прослушанным композициям для последующего составления списка рекомендаций. Для анализа композиций на предмет различных характеристик зачастую используются нейронные сети. Они и были использованы в разработанном программном продукте для решения задачи.

Машинное обучение – класс методов искусственного интеллекта, характерной чертой которых является не прямое решение задачи, а обучение в процессе применения решений множества сходных задач. Для построения таких методов используются средства математической статистики, численных методов, методов оптимизации, теории вероятностей, теории графов, различные техники работы с данными в цифровой форме.

Искусственная нейронная сеть (ИНС) – математическая модель, а также её программное или

аппаратное воплощение, построенная по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма. Данное понятие сформировалось при попытке изучить и промоделировать процессы, являющиеся основой нашей мозговой активности. В контексте машинного обучения, нейронная сеть представляет собой частный случай методов распознавания образов. Данный механизм подходит для решения поставленной задачи: выделение характеристик в некотором наборе данных.

Рекуррентные нейронные сети – один из видов нейронных сетей, в котором связи между элементами образуют направленную последовательность. Благодаря такой архитектуре появляется возможность просчитать последовательные пространственные цепочки или серии событий во времени. В отличие от многослойных перцептронов, рекуррентные сети могут использовать свою внутреннюю память для обработки последовательностей произвольной длины. Долгая краткосрочная память (англ. *Longshort-term memory*; *LSTM*) – одна из разновидностей архитектуры рекуррентных нейронных сетей, предложенная в 1997 году Сеппом Хохрайтером и Юргеном Шмидхубером. Как и большинство рекуррентных нейронных сетей, LSTM-сеть является универсальной в том смысле, что при достаточном числе элементов сети она может выполнить любое вычисление, на которое способен стандартный компьютер. Для вычислений необходима соответствующая матрица весов: она может рассматриваться как программа. LSTM-сети выделяются среди прочих рекуррентных сетей благодаря приспособленности к обучению на задачах классификации, обработки и прогнозирования временных рядов в случаях, когда важные события разделены временными интервалами с неопределённой продолжительностью. В контексте поставленной задачи данное свойство особенно эффективно, так как жанр музыкальных композиций может быть опознан не только по всей композиции целиком, но также и по её ключевым фрагментам.

Прямой анализ звуковых сигналов во временной области может потенциально занять много времени в зависимости от длительности и качества записи, и он сам по себе не является наиболее эффективным методом, так как данный объём информации будет избыточным, если анализировать его на какие-либо общие характеристики. Составление спектрограмм и их дальнейший анализ будет быстрее, но всё же не достаточно производительным. На текущий момент одним из наиболее рациональных представлений записи для дальнейшего анализа является метод мел-частотных кепстральных коэффициентов, который широко применяется для составления характеристик речевых сигналов. Данные коэффициенты представляют лог мощности спектра в мел частотной области. Описывают мощность огибающей спектра, которая характеризует модель речевого тракта. Получаются путём преобразования Фурье исходного сигнала, отображения значений спектра на мел-шкалу и последующего дискретного косинусного преобразования значений на мел-шкале. Полученные значения амплитуд спектра и будут являться целевыми коэффициентами.

Для реализации системы распознавания жанра используется нейронная сеть, на вход которой подаётся преобразованный аудиофайл, а на выходе – один из 10 жанров музыки: блюз, классическая, кантри, диско, хип-хоп, джаз, металл, поп, регги, рок. То есть выходом нейронной сети является 10-мерный вектор со значениями вероятности принадлежности композиции к тому или иному жанру.

В качестве основной модели нейронной сети используется рекуррентная модель LSTM, так как она позволяет анализировать данные переменной длины и работает гораздо эффективнее в задачах распознавания длинных фрагментов, чем стандартный многослойный перцептрон: она способна запоминать контекстное состояние и состояние системы на предыдущем этапе, что позволит оценивать композицию в целом, а не только её части.

Перед передачей аудиозаписи в нейронную сеть, планируется её начальное преобразование. Механизм преобразования заключается в выделении мел-частотных кепстральных коэффициентов, применимых в распознавании речи: они способны обозначить ключевые особенности аудиозаписи.

Для обучения нейронной сети была найдена база из миллиона экземпляров метаданных аудиозаписей, находящихся в открытом доступе: <https://labrosa.ee.columbia.edu/millionsong/>, а также открытая база из 1000 реальных аудиозаписей GZTAN: <http://marsyas.info/downloads/datasets.html>. Сеть может обучаться на одной из данных баз в зависимости от формата целевых входных данных.

На данный момент различные крупные музыкальные сервисы уже имеют систему подбора композиций на основе прослушанных, однако их алгоритмы являются закрытыми и базируются на тегах аудиофайлов. Разработанный продукт базируется на анализе самой аудиозаписи и не требует наличия тегов у записи.

Список использованных источников:

1. И. Заенцев. Нейронные сети: основные модели – Воронеж, 1999. – 74 с.
2. Nielsen M. Neural Networks and deep learning. [Электронный ресурс]. – Режим доступа: <http://neuralnetworksanddeeplearning.com>. – Дата доступа: 15.02.2018.
3. das, Asutosh, Manas Ranjan Jena, Kalyan Kumar Barik. Mel-Frequency Cepstral Coefficient (MFCC) - a Novel Method for Speaker Recognition. – Digital Technologies, 2014 – 195 с.
4. Долгая краткосрочная память [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Долгая_краткосрочная_память – Дата доступа: 25.02.2018

ІНДЭКСАЦЫЯ БАЗ ДАДЗЕННЫХ

Беларускі дзяржаўны ўніверсітэт інфарматыкі і радыёэлектронікі
г. Мінск, Рэспубліка Беларусь

Дзікі А.А.

Стройнікава А. Д. – асістэнт кафедры інфарматыкі

Рана ці позна прыходзіць момант, калі прадуктыўнасць базы дадзеных больш не здавальняючая. Індэкс – гэта структура дадзеных, якая дазваляе аптымізаваць пошук і доступ да дадзеных. Калі пачынае расці база дадзеных, яе прадуктыўнасць будзе праблемай. Такім чынам, прыярытэтай задачай выкарыстання баз дадзеных з'яўляецца атрыманне непасрэдна канкрэтнага радку ў вялікай табліцы ў мінімальна магчымыя тэрміны.

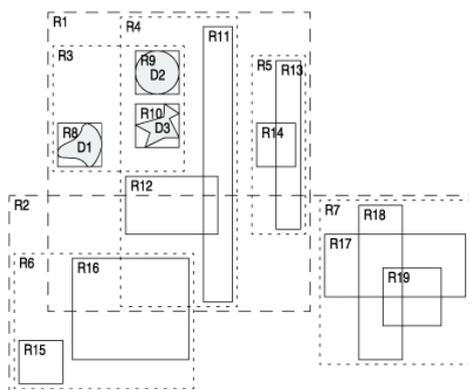
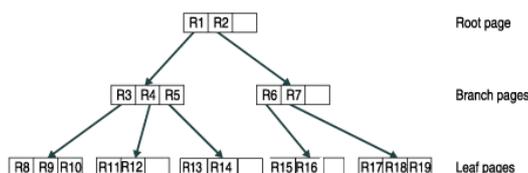
Індэксацыя ўяўляе сабой метады структуры дадзеных для эфектыўнага здабывання запісаў з файлаў базы дадзеных на аснове некаторых атрыбутаў, на якіх была зроблена індэксацыя. Індэкс можа быць кластарызаваны (радкі дадзеных сартуюцца па ключавым полі) і некластарызаваны (дадзеныя захоўваюцца ў кучы).

Найбольш папулярнай структурай дадзеных для аднамернага (іерархічнага) індэксу з'яўляецца В+-дрэва. Яго можна разглядаць як В-дрэва. Аднак замест таго, каб разглядаць усе вузлы роўнымі, дадзеная структура мае два тыпу вузлоў: ліставыя вузлы (утрымваюць фактычныя дадзеныя) і астатнія, уключаючы каранёвы вузел, якія ўтрымваюць толькі ключавыя значэнні і паказальнікі на наступныя вузлы. Дададзены тып аптымальны для мноства з добрым размеркаваннем значэнняў і высокай магнутнасцю.

Для арганізацыі хуткага пошуку ў прасторавых дадзеных, напрыклад геаграфічныя дадзеныя з двухмернымі каардынатамі (шырынёй і даўжынёй), выкарыстоўваюцца індэксы прасторавай сеткі (Spatialgrid) і R*-дрэвы.

Індэкс прасторавай сеткі дзеліць вобласць на лагічныя квадратныя сеткі з фіксаваным памерам, якія задаюцца пры стварэнні індэкса. Можна вызначыць да трох роўняў прасторавага індэкса (роўні сеткі). Выкарыстанне некалькіх роўняў сеткі карыснае, бо дазваляе аптымізаваць індэкс для розных памераў прасторавых дадзеных. Прасторавы індэкс на аснове сеткі мае тую перавагу, што структура індэкса можа быць створана першай, а дадзеныя дадаюцца на сталай падставе, не патрабуючы якіх-небудзь змен у структуры індэкса.

R*-дрэва (мал. 1) таксама выкарыстоўваецца для геапрасторавых дадзеных. Яно арганізавана падобна да Spatial grid, але, ў адрозненне ад Spatial Grid, не абавязана цалкам пакрываць бацькоўскія ячэйкі, і яны могуць перасякацца. Індэкс выкарыстоўвае абмежавальны прамавугольнік, які з'яўляецца прамалінейнай формай, якая цалкам утрымвае абмежаваны аб'ект ці аб'екты. Абмежавальныя палі могуць складаць аб'екты дадзеных ці іншыя абмежавальныя палі. Абмежавальныя палі звычайна захоўваюцца як набор каардынат аднолькавай памернасці, як абмежаваны аб'ект. Хоць з меркаванняў прадукцыйнасці карысна выбраць абмежавальны прамавугольнік, які як мага менш. Напрыклад, мінімальны абмежавальны прамавугольнік для двухмернай акружнасці – гэта квадрат, бок якога роўны дыяметру круга. Мінімальным абмежавальным прамавугольнікам для трохмернай сферы з'яўляецца куб, край якога роўны дыяметру сферы.



Мал. 1 – Структура індэкса R-дрэва

Перавагай R-дрэва з'яўляецца тое, што яно лягчэй ў рэалізацыі, патрабуе менш дыскавай прасторы, можа выкарыстоўвацца для шматмерных прастораў (Spatialgrid толькі для двухмерных) і пры ўзросце памеру

адсутнічае пагаршэнне прадукцыйнасці. Яго нястачай з'яўляецца праблема апраксімацыі палігонаў і неабходнасць стварэння індэкса для ўсёй Зямлі.

Для параўнання або пабудовы індэксаў для радковых і двайковых дадзеных можна выкарыстоўваць хэш-індэкс. Яны прадугледжваюць захоўванне не саміх значэнняў, а іх хэшаў, дзякуючы чаму змяняецца памер (а, адпаведна, і павялічваецца хуткасць апрацоўкі) індэксаў з вялікіх палёў. Такі тып індэксаў карысны пры аперацыях сартавання і параўнання дадзеных, аднак, выкарыстоўваючы яго, нельга вызначыць, колькі прыкладна радкоў ёсць паміж двума значэннямі, і для яго выкарыстання патрабуюцца толькі цэлыя значэнні ключоў.

Там, дзе ёсць вялікія мноства з нізкай магутнасцю і добрай кластарызацыяй па іх значэннях, выкарыстоўваюцца бітавыя індэксy (bitmap). Бітавыя індэксy выкарыстоўваюць бітавыя масівы і выконваюць пабітавыя лагічныя аперацыі на гэтых масівах. Гэты індэкс эфектыўны пры вялікіх табліцах і калі слупкі часта ўдзельнічаюць у аперацыях ўстаўкі / абнаўлення / выдалення. Аднак, пры аднаўленні мноства бітаў спатрэбіцца час, каб выканаць DML-аперацыю і аднавіць індэксy.

Для манатонна нарастальных значэнняў (напрыклад аўтаінкараментны ідэнтыфікатар) выкарыстоўваюць індэкс зваротнага ключа. Індэкс зваротнага ключа, па параўнанні са стандартным індэксам, звяртае кожны байт слупка, захоўваючы пры гэтым парадак слупкоў. Калі слупок індэксуецца ў зваротным рэжыме, значэнні слупка будуць захоўвацца ў індэксе ў розных блоках па меры таго, як зыходнае значэнне адрозніваецца. Такая кампануюка можа дапамагчы пазбегнуць пагаршэння прадукцыйнасці ў індэксах, дзе змены індэкса сканцэнтраваны на невялікіх наборах блокаў. Індэксy зваротнага ключа змяняюць «гарачыя пункты» ў індэксах, асабліва індэксy першасных ключоў, шляхам змены байтаў блокаў ліста. І, такім чынам, знішчаюць канкурэнцыю за ліставыя блокі па ўсіх асобніках.

Пры выбары тыпу індэкса трэба ўлічваць структуру дадзеных, якія індэксуюцца, а таксама характар працы з імі. Кожны тып па-свойму унікальны, моцны і карысны. І няма ўніверсальнага алгарытму і структуры дадзеных, якія былі б карысны ўсім. Важна памятаць, што выбар тыпу індэксавання можа як дапамагчы вырашыць праблему, гэтак і нашкодзіць, дадаўшы дадатковы расход памяці і замаруджванне часу апрацоўкі запытаў.

Спіс выкарыстаных крыніц:

1. Рассел, Д. Индекс (базы данных) / Д. Рассел, Р. Кон. – М. : Оникс, 2013. – 520 с.
2. Цэнтр ведаў IBM [Электронны рэсурс]. – 2018. – Рэжым доступу: <https://www.ibm.com/support/knowledgecenter>.

СИЛЬНЫЕ СТОРОНЫ ФРЕЙМВОРКА ANGULAR

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

Харитонов Н.В., Дроздов А.С.

Стержанов М.В. – к.т.н., доцент

В современном мире, где разработка программного обеспечения развивается быстрыми темпами, наибольшую популярность приобретают веб-приложения. Основными плюсами использования веб-приложений являются простота, удобство, доступность и скорость. Для реализации таких приложений существует множество различных фреймворков разработки, среди которых одним из наиболее популярных в последние три года является Angular.

Angular – Javascript-фреймворк, созданный на основе TypeScript, который позволяет разрабатывать сложные SPA-приложения (Single Page Application). Написание приложения состоит из создания шаблонов с помощью языка разметки HTML, написания классов-компонентов для работы с данными шаблонами, а также выделения общей логики приложения в сервисы. Все вместе это можно объединять в отдельные модули. Angular 2 написан в соответствии с шаблоном проектирования «модуль», что позволяет изолировать части логики приложения от глобального контекста, и содержит собственную систему модулей (NgModules) с тем, чтобы приложение имело понятную структуру и позволяло переиспользовать необходимые части без циклических зависимостей. Разработанный и поддерживаемый компанией Google, он описывается как JavaScript MVW-фреймворк. Angular (он же – Angular 2+, он же – Angular 2 или ng2) является переписанным преемником AngularJS (он же – Angular.js или AngularJS 1.x). Несмотря на то, что AngularJS (ранняя версия) был выпущен в октябре 2010 года, его создатели до сих пор устраняют недоработки данного фреймворка. Новый же Angular (без окончания «JS») был выпущен в свет в сентябре 2016 года в качестве версии №2 своего предка. На данный момент самой последней версией является 5-я. Фреймворк Angular используется такими компаниями, как Google, Wix, weather.com, healthcare.gov и Forbes.

Традиционно, веб-приложение на Angular 2 состоит из набора компонентов (виджетов), которые образуют древовидную структуру. В ней, в свою очередь, родительский компонент имеет ссылку на все вложенные в него дочерние. Обмен данными реализован следующим способом. При получении данных родитель отправляет их часть дочерним компонентам. Они, в свою очередь, либо передают их часть дальше по дереву компонентов, либо отображают эти данные с использованием различных элементов интерфейса. Также, в Angular 2 дочерние компоненты имеют возможность оповестить родительский о различных пользовательских событиях: клик мыши или нажатие клавиши. Также возможно внедрение так называемых сервисов (services) - специальных объектов, предоставляющих компоненту возможность получить данные в

любой момент, вне зависимости от его расположения в дереве компонентов. Стоит упомянуть, что в Angular 2 реализован прием, позволяющий любому объекту веб-приложения получить доступ к другому, предварительно зарегистрированному объекту (provider) при необходимости.

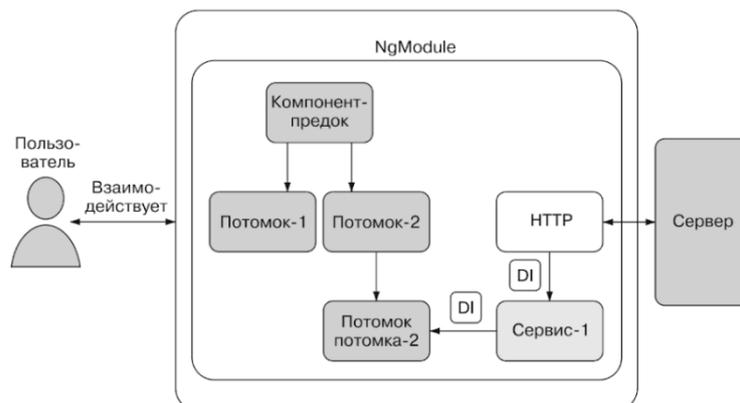


Рис. 1. Пример архитектуры Angular-приложений

Так же хотелось бы отметить, что Angular 2 разработан с нуля для мобильных приложений и оптимизирован с точки зрения эффективности памяти и меньшего количества циклов центрального процессора. Здесь объектами первого класса являются touch-события и жесты, работающие на всех устройствах. Все тесты производительности, проводимые командой Angular, находятся в открытом доступе в Github. Так что абсолютно любой может на них взглянуть и даже провести их.

Таким образом, фреймворк Angular.js является полноценным инструментом, полноценно реализующим компонентный подход, позволяя разработчикам создавать отдельные компоненты, а из компонентов создавать полноценные приложения. Созданные компоненты можно использовать в других приложениях, так как они независимы.

Список использованных источников:

1. Angular 2 [Электронный ресурс]. – Режим доступа: <https://angular.io/>.
2. Fain Y., Moiseev A. Angular 2 Development with TypeScript. – Manning Publications, 2016

ПРИМЕНЕНИЕ СКРЫТЫХ МАРКОВСКИХ МОДЕЛЕЙ ДЛЯ СОПОСТАВЛЕНИЯ КАРТ

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

Иванин Н.С.

Волорова Н.А. – к.т.н., доцент

В последнее время было разработано множество подходов, позволяющих улучшить навигацию и управление транспортным средством на основе данных, собираемых его различными датчиками. Одним из таких датчиков является датчик Глобальной Системы Позиционирования (GPS). Зачастую требуется определить по какой дороге движется транспортное средство на основе данных датчика GPS – эта проблема известна как проблема сопоставления карт. В настоящей работе предлагается подход на основе использования Скрытых Марковских Моделей (СММ) для решения задачи сопоставления карт. Этот подход позволяет бороться с такими проблемами, как шумы в GPS данных и разреженность данных.

Сопоставление карт – это процесс определения дороги, по которой совершалось движение транспортного средства на основе данных, собранных специальными датчиками. Как правило, датчики собирают данные GPS, поскольку система GPS доступна повсеместно. Например, сопоставление карт используется навигационными системами. В последнее время сопоставление карт также используется для измерения скорости движения по дороге, а также построения статистических моделей пробок. В дальнейшем такие модели могут быть использованы для нахождения оптимального пути движения транспортного средства в обход пробок [1].

В работе [2] приводятся основные элементы структуры дорожной сети:

Узел. Дорога, являющаяся линейным объектом, при цифровом представлении хранится как последовательность точек (узлов). Узлами являются также точки пересечения дорог, начала и концы дорог, по которым транспортные средства могут двигаться.

Сегменты дорог. При существовании прямого пути между двумя соседними узлами такой путь будет называться сегментом дороги. Начальный и конечный узел такого пути будут называться начальными и конечными узлами сегмента.

Дуга. При существовании направленного пути между двумя соседними узлами такой путь будет называться дугой, а узлы соответственно началом и концом дуги. Дуга может принадлежать одному или нескольким сегментам.

При использовании Скрытых Марковских Моделей(СММ) для проблемы сопоставления карт движение объекта моделируется как Марковский процесс между узлами дорожной сети. Такие узлы дорожной сети не являются явно видимыми и рассматриваются как скрытые состояния. Координаты GPS в таком случае являются выходами скрытых состояний, чье распределение вероятностей зависит только от скрытых состояний.

Основными компонентами СММ являются вероятность эмиссии, вероятности изменения состояний, вероятности начальных состояний. Вероятности эмиссии определяют вероятность присвоения точки GPS (описывает позицию движущегося объекта) данной точке на сегменте дороги и определяется как $P(z_t | point_t)$. Вероятность эмиссии показывает вероятность наблюдения z_t , если движущийся объект находится в точке $point_t$ в момент времени t . В работе [3] вероятность эмиссии моделируется с помощью функции Гаусса с центром в точке $point_t$:

$$P(z_t | point_t) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{\|z_t - point_t\|_{gc}^2}{2\sigma}\right)$$

где $\|z_t - point_t\|_{gc}$ – это расстояние между реальным положением движущегося объекта и точкой дорожного сегмента

σ – погрешность измерений датчика GPS.

Вероятности изменения состояний определяют условную вероятность движения объекта из одного узла в другой узел дорожной сети. В работе [4] предлагается в качестве этих вероятностей принимать значения, зависящие от расстояния между узлами дорожной сети, следующим образом:

$$P(s_t | s_{t-1}) \approx \lambda \exp\{-\lambda(\|z_t - z_{t-1}\| - \|s_t - s_{t-1}\|)\},$$

где λ - параметр, который необходимо подбирать для каждой конкретной выборки (последовательности позиций движущегося объекта).

Вероятности начальных состояний показывают вероятность того, что начальной точкой движущегося объекта является $point_t$. В работе[5] начальные вероятности аппроксимируются из эмиссионных вероятностей относительно первых наблюдений $P(z_1|point_1)$.

При реализации сопоставления карт было принято решение использовать готовую библиотеку [5] для языка python. Для поиска состояний СММ был использован алгоритм Витерби [7]. Алгоритм Витерби- это алгоритм динамического программирования для поиска наиболее подходящего списка состояний, называемым путем Витерби. В результате алгоритм находит последовательность наблюдаемых событий в контексте Марковских моделей. В случае сопоставления карт алгоритм для каждой GPS точки найдет наиболее подходящие узлы на карте. Сложность алгоритма составляет $O(T*S)$, где S-размер пространства событий, а T-число выходов.

Среди всех точек выбираются такие точки, которые имеют минимальную дистанцию до GPS-точки. Для каждой такой точки-кандидата и GPS-точки рассчитываются эмиссионные вероятности. После этого с помощью алгоритма Витерби для поиска наиболее подходящего списка состояний для текущих состояний (GPS точек). Итерации алгоритма продолжают до тех пор, пока существуют GPS точки, не сопоставленные точкам на дороге.

Список использованных источников:

1. Goh C., Dauwels J. и др., Online map-matching based on hidden Markov model for real-time traffic sensing applications //15th International IEEE Conference. – 2012.
2. Kibal J., Hidden Markov Model based map matching.
3. Newson P., Krumm J., Hidden Markov map matching through noise and sparseness // 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. – 2009.
4. Mattheis S., Al-Zahid K. K. и др., Putting the car on the map: A scalable map matching system for the Open Source Community // INFORMATIK 2014: Workshop Automotive Software Engineering. – 2014.
5. Raymond R., Morimura T. и др., Map matching with hidden Markov model on sampled road network // 21st International Conference on Pattern Recognition (ICPR '12). – 2012.
6. HiddenMarkovpython [Электронный ресурс]. — Режим доступа: https://pypi.org/project/hidden_markov/
7. Forney G., The Viterbi algorithm // IEEE. – 1973.

ПОСТРОЕНИЕ РЕКОМЕНДАТЕЛЬНОЙ СИСТЕМЫ НА ОСНОВЕ АЛГОРИТМОВ ОБРАБОТКИ ЕСТЕСТВЕННОГО ЯЗЫКА И ГРАФОВЫХ БАЗ ДАННЫХ

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

Козуб В.Н., Пилецкий И.И.

Пилецкий И.И. – к.ф.-м.н., доцент

В данном докладе рассматривается система рекомендаций на основе схожести контента и с учётом реакций других пользователей (лайки, репосты и т.д.). Такая система является более эффективной, чем традиционный подход (фильтрация), благодаря использованию дополнительных метрик при формировании рекомендации. Применение такой системы позволит пользователям находить релевантные материалы, хранящиеся в социальных сетях. Предлагается реализация рекомендательной системы с применением графовых баз данных. Рассматриваются векторы схожести материалов.

1. Разработка модели данных

В последние годы рост популярности социальных сетей породил огромное количество материалов, сгенерированных пользователями и хранящихся в социальных сетях. Важной задачей является реализация рекомендательной системы, которая позволит пользователям быстрее находить релевантные материалы. Одним из вариантов реализации такой системы может быть рекомендация на основе схожести контента с учётом реакций других пользователей (лайки, репосты и т.д.). В то же время социальные взаимодействия могут быть удобно описаны в виде графовой модели данных.

Под рекомендательной системой понимается система для поиска и предсказания материалов, которые могут быть интересны пользователю. Предсказание даётся с определённой точностью и основывается на ряде факторов, рассматриваемых далее в разделе 2.

Под схожестью контента подразумевается некоторая оценка подобия двух материалов, основанная на ряде критериев.

В среде социальных сетей под материалом может подразумеваться сообщение, твит, пост в блоге и т.д. Любой материал может быть охарактеризован в основном тремя элементами:

Внутреннее содержимое материала и внутренние тэги;

Тэги, назначенные пользователем;

Пользовательские взаимодействия с документом.

Под пользовательским взаимодействием подразумевается любое действие, которое пользователь может совершить с материалом, например, просмотр, комментирование, лайк и т.д.

При традиционном подходе индексируется только внутреннее содержимое документа, и этот индекс затем используется для помощи в нахождении документов, релевантных поисковому запросу пользователя. Этот подход до сих пор пользуется популярностью во многих поисковых системах [1].

В данной работе предлагается использовать комбинированный подход при подсчёте схожести материалов, который включает в себя содержимое материала, его тэги, а также все пользовательские взаимодействия с материалом. Эти три фактора рассматриваются как три измерения документа в социальном пространстве (назовём их «Контент», «Тэг», «Взаимодействие»). Каждое измерение несёт в себе различный взгляд на материал.

В «Контенте» смысл материала задаётся его автором. А «Тэг» отражает то, как материал воспринимают пользователи соцсети. Каждый пользователь может предоставить свой, отличный от других взгляд на материал простым действием: установкой тэга. Во «Взаимодействии» смысл материала задаётся активностью пользователей соцсети, их действиями по отношению к данному материалу.

При таком подходе могут быть использованы семантические алгоритмы для извлечения иерархий из концептов. Это позволяет отыскивать связи между тэгами, и таким образом обнаруживать скрытые отношения между на первый взгляд несвязанными материалами.

Схема данных для такой модели может выглядеть следующим образом (см. рис. 1):

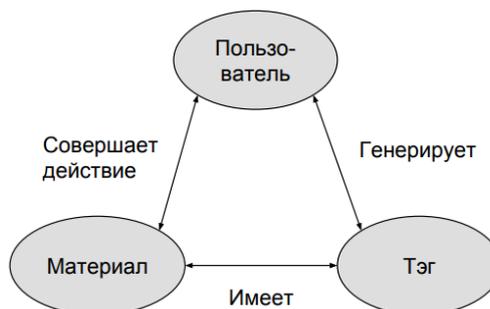


Рис. 1 – Схема данных

На схеме видно, как такая сложная система может быть легко представлена и в дальнейшем расширена при использовании графовых баз данных. Например, база данных Neo4j содержит инструменты для работы с графовым представлением данных [2].

2. Определение схожести контента

Используя собранную о материале информацию, могут быть созданы три различных вектора: вектор контента, вектор социальных тэгов, вектор пользователя.

Вектор контента можно представить следующей формулой:

$$C_i = \{wc(i,1), wc(i,2), \dots, wc(i,n)\},$$

где n – общее количество тэгов в базе данных, $wc(i,k)$ – вес k -того тэга в материале или в иерархии тэгов.

$wc(i,k)$ рассчитывается по следующей формуле:

$$a * T(f) - i * df(i,k),$$

где a – вес в иерархии, он равен единице, если тэгом является сам материал, иначе – нулю.

Вектор контента представляет собой оценку схожести материалов и учитывает «статичный контент» (информацию в самом материале и внутренние тэги).

Вектор социальных тэгов выглядит следующим образом:

$$T_i = \{wt(i,1), wt(i,2), \dots, wt(i,p)\},$$

где p – общее количество тэгов в базе данных, $wt(i,k)$ – вес k -того тэга материала. Таким образом, $wt(i,k)$ также является частотой k -того тэга в i -том документе.

Вектор социальных тэгов представляет собой оценку схожести материалов, основанную на сравнении социальных тэгов материалов, то есть, специальных меток, которые были добавлены потребителями контента, а не его автором.

Вектор пользователя:

$$U_i = \{wu(i,1), wu(i,2), \dots, wu(i,q)\},$$

где q – общее количество пользователей в базе данных, $wu(i,k)$ – вес k -того пользователя материала. Вес может быть рассчитан различными способами в зависимости от уровня интереса различных пользователей к материалу.

Вектор пользователя представляет собой оценку схожести материалов, основанную на интересе пользователя (его действиях по отношению к материалу).

Также возможно использование более чем одного пользовательского вектора, если необходимо использовать различные веса для различных компонентов (например, один вектор для «лайков», второй для «репостов» и т.д.).

Используя все эти векторы, можно рассчитать различные компоненты схожести, а затем сложить их для получения итогового значения схожести:

$$CombinedSimilarity(i, j) = aCosSim(C_i, C_j) + bCosSim(T_i, T_j) + c * CosSim(U_i, U_j),$$

где $a+b+c=1$.

Стоит отметить, что вычисленная схожесть представляет собой новую информацию, извлечённую из данных в графовой базе данных. Она хранится как модель в рекомендательной системе и может быть использована для предоставления рекомендаций пользователю.

3. Заключение

В данной работе были рассмотрены принципы построения рекомендательной системы для социального контента. Разработаны коэффициенты схожести, основанные на контенте и социальных взаимодействиях. Предлагается их использовать для определения релевантного контента, а также комбинировать их вместе с традиционным подходом (фильтрацией), чтобы получить более релевантные для пользователя результаты. В качестве системы хранения предлагается использовать графовые базы данных, так как модель рассмотренной рекомендательной системы может быть адекватно представлена в виде графа и запросы к данным могут быть выполнены в режиме, близком к реальному времени.

Список использованных источников:

1. Tran Vu Pham, Le Nguyen Thach, "Social-Aware Document Similarity Computation for Recommender Systems", vol. 00, pp. 872-878, 2011.
2. Neo4jDocumentation [Электронный ресурс] / Neo4j.com – 2018. – Режим доступа: <https://neo4j.com/docs/>. – Дата доступа: 10.03.2018.

КОПУЛЫ КАК ИНСТРУМЕНТ МАТЕМАТИЧЕСКОЙ СТАТИСТИКИ

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

Литвинчук Д.В. Королев К.Ф.

Волорова Н.А. – к.т.н., доцент

В настоящее время все более актуальным становится вопрос о получении, подготовке и анализе данных. Копула-функция представляет собой инструмент для выполнения последнего. Метод копула-функции определяет характер взаимодействия между случайными величинами, что позволяет сформировать более адекватную модель описания совместного распределения нескольких переменных, чем многомерный нормальный закон. Важным преимуществом функции является то, что при моделировании многомерного распределения, помимо частных распределений она учитывает характер их взаимодействия. Копула-функция может быть применена для описания совместного распределения переменных имеющих разные частные распределения.

Общеизвестно, что лишь по маргинальным распределениям невозможно достоверно восстановить n -мерное совместное распределение. Всё дело в том, что в таком случае нам не хватает информации о взаимозависимости одномерных распределений. Копула, в данном случае, является недостающим звеном, содержащим всю информацию о взаимосвязи маргиналов.

Копула — это распределение, определённое на n -мерном единичном кубе, все маргиналы которого распределены равномерно на $[0;1]$.

Теорема Склара (Sklar's theorem) утверждает, что какими бы ни были n -мерное распределение $F(x) = F(x_1, x_2, \dots, x_n)$ и его маргинальные распределения $F_k(x_k), k = 1 \dots n$ существует такая копула $C(u_1, u_2, \dots, u_n)$, что для всех $x \in R^n$ выполняется равенство:

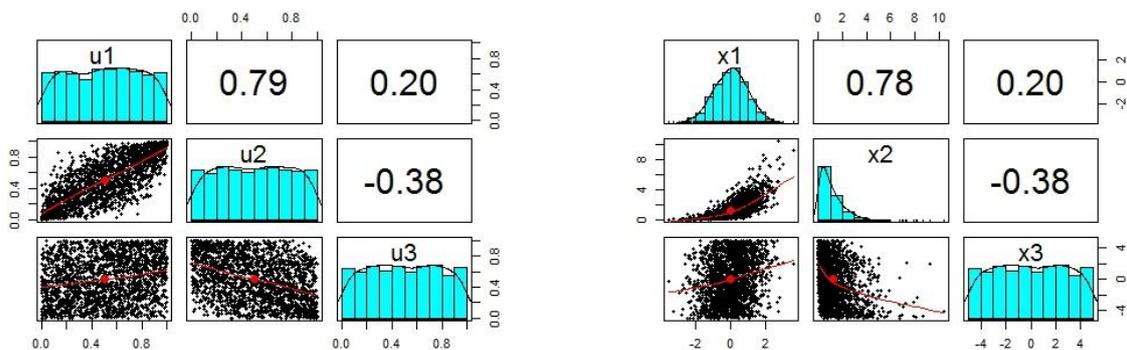
$$F(x_1, x_2, \dots, x_n) = C(F_1(x_1), F_2(x_2), \dots, F_n(x_n))$$

Копула является обыкновенной функцией распределения, потому для копул можно определить плотность как неотрицательную функцию $c(F_1(x_1), F_2(x_2), \dots, F_n(x_n))$, для которой выполняется:

$$C(u_1, u_2, \dots, u_n) = \int_0^{u_1} \int_0^{u_2} \dots \int_0^{u_n} c(s_1, s_2, \dots, s_n) ds_1 ds_2 \dots ds_n$$

Следствие Теоремы Шкляра устанавливает связь между плотностями совместного и маргинальных распределений с плотностью копулы, описывающей данное распределение.

$$c(F_1(x_1), F_2(x_2), \dots, F_n(x_n)) = \frac{f(x_1, x_2, \dots, x_n)}{\prod_{i=1}^n f_i(x_i)}$$



Выборка на основе копулы

Результирующая выборка

Копулы предлагают элегантный способ моделирования сложных распределений. Например, с помощью копулы Гаусса можно построить n -мерное распределение по заданной корреляционной матрице и маргинальным распределениям.

$$C_{GA}(u_1, u_2, \dots, u_n) = \Phi_n(\Phi^{-1}(u_1), \Phi^{-1}(u_2), \dots, \Phi^{-1}(u_n))$$

Где Φ — функция нормального распределения. Благодаря копулам, данный способ позволяет отдельно задавать корреляционные зависимости маргиналов и сами маргиналы. Для примера, сгенерируем выборку на 2000 элементов, средствами языка R, из распределения с маргиналами: нормальным, экспоненциальным, равномерным; и попарными коэффициентами корреляции равными 0,8, 0,2, -0,4. Для этого сгенерируем выборку из трёхмерного нормального распределения с соответствующей корреляционной

матрицей. Затем, к полученным данным, по координатам применим сначала Φ , получив тем самым выборку на основе копулы, а затем F_i^{-1} — обратную функцию для данной координаты.

Список использованных источников:

1. Благовещенский Ю.Н. Основные элементы теории копул. Прикладная эконометрика, 2012.
2. Фантаццини Д. Моделирование многомерных распределений с использованием копула-функций. Прикладная эконометрика 2011.

ПРОГРАММНАЯ СИСТЕМА СБОРА НОВОСТЕЙ ДЛЯ ГОРОДСКИХ ИНФОРМАЦИОННЫХ ПОРТАЛОВ

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

Королёва П.Б.

Жвакина А.В. – к.т. тех. наук, доцент

Информационный портал представляет собой универсальный сайт, основной контент которого – информационные тематические статьи. Существует множество классификаций таких сайтов. Блоги, словари и справочники, объявления и вакансии, социальные сети – это лишь некоторые из них. Данная работа связана с ещё одним типом – с информационными сайтами городов и районов.

Можно с уверенностью сказать, что почти каждый город имеет собственный сайт. Обычно он содержит не только историю города, основную информацию о существующих организациях и различные справочные данные, но и новости. Новости группируются по их содержанию, и почти всегда на сайте выделяются такие разделы, как «В Беларуси» или «В мире».

Проработав пару месяцев с контентом одного городского информационно-новостного сайта, я узнала, что содержание вышеупомянутых разделов формируется с помощью более крупных новостных сайтов-источников. Это происходит вручную путём изучения контента стороннего сайта, отбора новостей и стандартной их публикации с ссылками на оригинальные источники.

Данный процесс требует достаточно много усилий от контент-менеджера. Это приводит к необходимости решения задачи оптимизации рабочего времени и упрощения агрегации информации. Ручной «копи-паст» новостей в настоящее время, когда появляется всё больше и больше различных агрегаторов и грабберов является проявлением нерационального подхода. В таких случаях целесообразно использовать автоматизированную настраиваемую систему сбора новостей.

Разработанная мною программа представляет собой именно такую систему. В перспективе её можно интегрировать на любой сайт для агрегации любых видов данных, но в данный момент акцент ставится именно на сбор новостей. Принцип работы указан на рисунке 1:

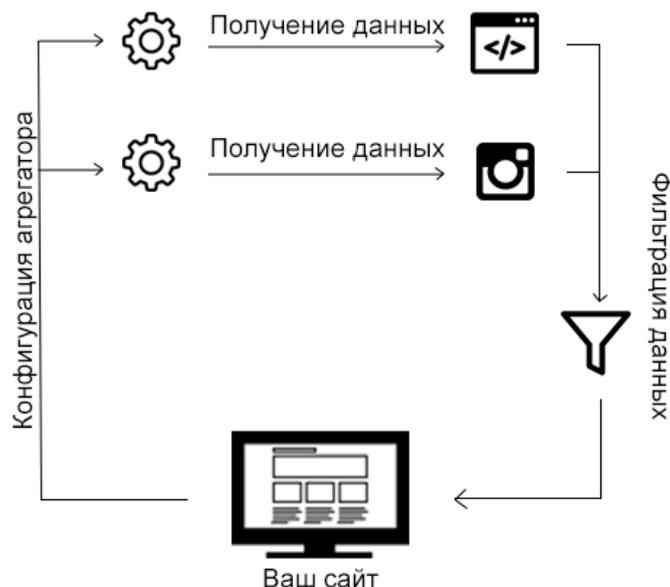


Рис. 1– Схема работы системы сбора новостей

Данный программный продукт обеспечивает возможность выбора имеющихся источников информации или создания и настройки новых.

При агрегировании исключаются повторяющиеся новости, полученные из разных источников, в зависимости от их приоритета.

Фильтр новостей по категориям, разделам, темам или тегам позволяет ограничить сбор данных по установленным критериям, подгрузив только необходимую информацию.

Последним шагом настройки данной системы является установление связи между структурой полученных данных и структурой новостей на сайте – так называемый маппинг.

Полученная система – довольна гибкая и легко дополняемая. В целом можно выделить следующие преимущества:

- значительная экономия времени в процессе сбора новостей;
- уменьшение загруженности команды контент-менеджеров путём автоматизации части их рабочего процесса или исключение потребности в наёме отдельных работников, чьей обязанностью был бы сбор данных из внешних новостных источников;
- расширяемость, возможность доработки функционала, добавления сбора информации из другого типа источников или внедрение анализа получаемого контента.

Таким образом, разработанная система позволяет значительно облегчить работу с контентом информационно-новостных порталов городов. Она может использоваться для сбора информации другого рода с целью публикации полученных данных на любом портале. Также актуально внедрение данной программы в виде отдельного модуля для систем управления содержимым сайта.

Список использованных источников:

1. Информационный онлайн-портал: цели и особенности – [Электронный ресурс] – <http://jnetwork.kz/blog/informacionnyy-onlayn-portal-celi-i-osobennosti>
2. Новостные агрегаторы как инструмент оптимизации – [Электронный ресурс] – <http://xn--h1aieep.xn--p1ai/novostnye-agregatory-v-pomoshh-optimizatoru>

АРХИТЕКТУРА СОБЫТИЙНОГО ПРЕДСТАВЛЕНИЯ ДАННЫХ

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

Кравцов Д.В.

Волорова Н.А. – к.т.н., доцент

В настоящее время большинство приложений работают с данными, и типичный подход заключается в том, чтобы приложение хранило текущее состояние данных, обновляя его тогда, когда пользователи взаимодействуют с ним. Например, в традиционной модели создания, чтения, обновления и удаления (CRUD – Create, Read, Update, Delete) типичным процессом обработки данных является чтение данных из хранилища, внесение некоторых изменений и обновление текущего состояния данных новыми значениями – часто используя транзакции, которые блокируют данные. Такой подход имеет ряд ограничений:

- системы CRUD выполняют операции обновления непосредственно в хранилище данных, что может замедлить производительность и оперативность, а также ограничить масштабируемость из-за требуемых затрат на обработку;
- при совместной работе со многими параллельными пользователями конфликты обновления данных более вероятны, поскольку операции обновления выполняются на одном элементе данных;
- если нет дополнительного механизма аудита, который записывает детали каждой операции в отдельный журнал, история теряется.

Архитектура Event Sourcing определяет подход к обработке операций над данными, которые являются результатом последовательности событий, каждое из которых записывается в хранилище. Приложение отправляет ряд событий, которые однозначно описывают каждое действие, которое произошло в системе, в хранилище событий, где они сохраняются [1, 2].

Типичное использование событий заключается в том, чтобы генерировать представления сущностей одновременно с изменением их в системе, а также для интеграции с внешними системами.

Архитектура Event Sourcing предоставляет следующие преимущества:

- события неизменяемы и могут только добавляться в хранилище, что позволяет значительно повысить производительность и масштабируемость приложений, особенно для уровня представления или пользовательского интерфейса;
- события – это простые объекты вместе с любыми связанными данными, необходимыми для описания действия, представленного событием. События не обновляются непосредственно в хранилище данных, а просто записываются для обработки в соответствующее время, что может упростить внедрение и управление;
- события обычно имеют смысл для эксперта домена, тогда как несоответствие объектно-реляционного представления может затруднить понимание сложных таблиц базы данных;
- хранилище событий обеспечивает журнал, который может использоваться для мониторинга действий, выполненных в хранилище данных, восстановления текущего состояния в виде

материализованных представлений или прогнозов путем повторного воспроизведения событий в любое время и оказания помощи в тестировании и отладке системы. Список событий также может использоваться для анализа производительности приложений и выявления тенденций поведения пользователей или для получения другой полезной бизнес-информации.

Данный подход обычно используется тогда, когда необходимо минимизировать или полностью избежать конфликтов при обновлении данных, а также когда особую важность играет возможность воспроизводить происходящие в системе события, чтобы восстановить состояние системы, отменить изменения или сохранить журнал истории и аудита.

Очень часто EventSourcing используется вместе с шаблоном CQRS (Command-QueryResponsibilitySegregation – разделение ответственности на команды и запросы), что позволяет разделить процессы записи и чтения данных, увеличивая производительность и надежность системы [3].

Список использованных источников:

1. Martin Fowler, Development of Further Patterns of Enterprise Application Architecture
2. Event Sourcing Pattern [Электронный ресурс] / Christopher Bennis. – 23 июня 2017. – Режим доступа: <https://docs.microsoft.com/en-us/azure/architecture/patterns/event-sourcing>.
3. Betts, D. Exploring CQRS and Event Sourcing: A journey into high scalability, availability, and maintainability with Windows Azure. / D. Betts. – Microsoft, 2013. – 213 p.

ПРОГРАММНОЕ СРЕДСТВО ОБФУСКАЦИИ VHDL-КОДА

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

Крагель Н.А.

Иванюк А.А. – профессор, д.т.н., доцент

Одной из важных проблем, возникающих при проектировании цифровых устройств является защита интеллектуальной собственности. Существуют различные способы ее обеспечения и доказательства авторских прав: лексические и функциональные обфускации проектных описаний, внедрение водяных знаков и др.

Обфускация - приведение исходного текста проектных описаний к виду, сохраняющему функциональность, но затрудняющему анализ, понимание алгоритмов работы и модификацию.

Цели обфускации:

Затруднение изучения и модификации проектных описаний;

Усложнение реверс инжиниринга проектных описаний.

Целью создания программы была автоматизация обфускации VHDL-кода с поддержкой основных конструкций языка.

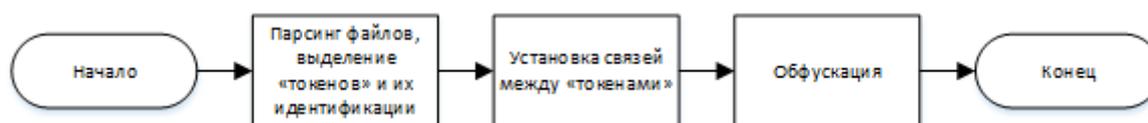


Рис. 1. Основные этапы работы программы.

Процесс работы программы представлен на рисунке 1. На первом этапе исходные файлы разбиваются на «токены». На втором этапе происходит инициализация «токенов» и установление связей между ними. Таким образом после этого этапа имеется полное программное представление загруженных исходных файлов. Рассмотрим подробнее шаг обфускации с внедрением водяного знака. Для обфускации генерируются новые имена для переменных, содержащие только знаки «1», «0», «O», «I» и «L». Можно выделить 2 группы элементов:

Элементы визуально похожие на «1»: «L», «I»;

Элементы визуально похожие на «0»: «O».

Распределение данных групп в рамках генерируемых названий нормальное. Однако можно управлять распределением элементов в рамках группы, увеличивая частоту использования одного символа и уменьшая частоту другого.

В рамках реализованной программы обфускации подлежат все переменные, сигналы, сущности и порты. Перед процессом обфускации были выделены все связи, что теперь позволяет пройти по всем элементам, объявленным в загруженных файлах. Такой подход позволяет избежать проблем, возникающих, когда были загружены не все исходные файлы и некоторая часть структурных элементов была объявлена в отсутствующих файлах.

Ниже представлен пример работы программы.

Исходный код:	Обфусцированный код:
<pre> library IEEE; use IEEE.STD_LOGIC_1164.ALL; entity D_trigger is Port(mD, mC, mR, mS : in std_logic; mQ, mnQ : out std_logic); end D_trigger; architecture Behavioral of D_methoda is begin process(mC, mR, mS) begin if(mr = '0') then if(ms = '0') then mq <= 'X'; mnq <= 'X'; else mq <= '0'; mnq <= '1'; end if; elsif(ms = '0') then mq <= '1'; mnq <= '0'; elsif rising_edge(mc) then mq <= md; mnq <= not(md); end if; end process; end Behavioral; </pre>	<pre> library ieee; use ieee.std_logic_1164.all; entity OOIOIIIOIO10II0OIO1100IOOIO1 is port(IIIO100110IO10110IO010IOIOIOI,IIIO100110IOIO110IO0 1010IOIOI,IIIO100110IO10110IO010IOIOIOI,IIIO100110I OIO10IO010IOIOIOI:in std_logic; IIIO100110IOIO110IO010IOIOIOI,IIIO100110IOIO110IO010I 0IOIOI:out std_logic); end OOIOIIIOIO10II0OIO1100IOOIO1; architecture behavioral of OOIOIIIOIO10II0OIO1100IOOIO1 is begin process(IIIO100110IOIO110IO010IOIOIOI,IIIO100110IO1 0110IO010IOIOIOI,IIIO100110IOIO110IO010IOIOIOI) begin if(IIIO100110IO10110IO010IOIOIOI='0')then if(IIIO100110IOIO110IO010IOIOIOI='0')then IIIO100110IOIO110IO010IOIOIOI<='X'; IIIO100110IOIO110IO010IOIOIOI<='X'; else IIIO100110IOIO110IO010IOIOIOI<='0'; IIIO100110IOIO110IO010IOIOIOI<='1'; end if; elsif(IIIO100110IOIO110IO010IOIOIOI='0')then IIIO100110IOIO110IO010IOIOIOI<='1'; IIIO100110IOIO110IO010IOIOIOI<='0'; elsif rising_edge(IIIO100110IOIO110IO010IOIOIOI)then IIIO100110IOIO110IO010IOIOIOI<=IIIO100110IO10110IO 010IOIOIOI; IIIO100110IOIO110IO010IOIOIOI<=not(IIIO100110IO1011 0IO010IOIOIOI); end if; end process; end behavioral; </pre>



Рис. 2. Упрощенное представление архитектуры программы

На рисунке 2 представлена упрощенная архитектура программы. Ключевой особенностью данной архитектуры является модульность обфускаторов, каждый из которых производит свою, независимую от других обфускаторов, операцию. Например, один производит лексическую обфускацию сигналов, второй – лексическую обфускацию переменных и т.д. Основным недостатком лексической обфускации является то, что процесс синтеза нивелирует ее. Другими словами, схема, полученная синтезом описания с внедрением лексической обфускации будет идентична схеме, полученной синтезом описания, которое не было подвержено лексической обфускации. В свою очередь это требует добавления схмотехнической обфускации. Однако архитектура разработанного приложения позволяет добавлять любое количество новых независимых обфускаторов, тем самым позволяя решить это проблему.

Список использованных источников:

1. Иванюк, А.А. Проектирование встраиваемых цифровых устройств и систем: монография / А. А. Иванюк – Минск: Бестпринт, 2012. – 337 с.
2. Surreptitious Software: Obfuscation, Watermarking, and Tamperproofing for Software Protection: Obfuscation, Watermarking, and Tamperproofing for Software Protection. By Christian Collberg, Jasvir Nagra. Published Jul 24, 2009 by Addison-Wesley Professional.

ДОСТАТОЧНЫЕ УСЛОВИЯ ЭКСТРЕМУМА ФНП

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

Краснов И.А.

Анисимов В.Я. – канд. физ.-мат. наук, доцент

В учебных пособиях по высшей математике часто ограничивается исследованием второго дифференциала. В данном случае рассматриваются ситуации, когда дифференциалы 1-ого, 2-ого, ... $k-1$ -ого порядка равны нулю.

Пусть дана функция $f()$ от n переменных, определенная и дифференцируемая в точке M_0 и некоторой ее окрестности, имеет непрерывные частные производные $2m+1$ -ого порядка, $f'_1 = 0, f'_2 = 0, \dots, f'_n = 0$, а $f''_{ij} > 0$ ($m > 0$), то экстремум в точке M_0 отсутствует.

Пусть дана функция $f()$ от n переменных, определенная и дифференцируемая в точке M_0 и некоторой ее окрестности, имеет непрерывные частные производные $2m$ -ого порядка, $f'_1 = 0, f'_2 = 0, \dots, f'_n = 0$, а $f''_{ij} > 0$ ($m > 0$), то в точке M_0 точка максимума, если квадратичная форма A положительно знакоопределена, или точка минимума, если отрицательно знакоопределена.

Квадратичная форма A - квадратичная форма от m переменных вида $A(x, y, z)$. Ее матрица состоит из частных производных $2m$ -ого порядка.

Например, дана $f(x, y, z)$ определенная в M_0 , дифференцируемая и имеющая непрерывные частные производные 4-ого порядка, $f'_1 = 0, f'_2 = 0, f'_3 = 0$. Рассмотрим 4-ый дифференциал как квадратичную форму от переменных x, y, z , и матрица A будет выглядеть так:

Если квадратичная форма A неотрицательно или неположительно знакоопределена, то экстремум в точке M_0 может как быть, так и не быть.

Список использованных источников:

1. Альсевич Л.А., Булатов В.И., Красовский С.Г. Экстремум функции нескольких переменных: учеб. материалы для студентов факультета прикладной математики и информатики. – Минск: БГУ, 2016. – 39 с.
2. Ильин В.А., Садовничий В.А., Сендов Бл.Х. Математический анализ. Начальный курс. – Москва: МГУ, 1985. – 662 с.
3. Фихтенгольц Г. М. Курс дифференциального и интегрального исчисления. В 3 т. Т. I — 8-е изд. — Москва: ФИЗМАТЛИТ, 2003. - 680 с.

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ КОММУНИКАЦИИ МЕЖДУ СТУДЕНТАМИ И САМОКОНТРОЛЯ УЧЕБНОЙ ДЕЯТЕЛЬНОСТИ

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

Кулик Г.В.

Жвакина А.В. – к.т.н., доцент

Для того чтобы стать высококвалифицированным специалистом необходимо много практиковаться. Основной вид учебной деятельности студентов, обучающихся на технических специальностях – лабораторные работы. Университет предоставляет такую возможность на лабораторных работах, большое количество которых затрудняет контроль текущей успеваемости. У многих студентов возникает проблема с отслеживанием своей успеваемости. Также, во время выполнения работы или чтения условия могут возникать вопросы, на которые не всегда можно легко найти ответ. Это приводит к тому, что студенты тянут до конца семестра, потом наконец-то начинают что-то делать, видят, что не успевают и начинают паниковать и выкручиваться из данной ситуации. В итоге КПД обучения страдает. Таким образом, студенты нуждаются в автоматизированной системе контроля своей успеваемости и удобном обмене информацией в процессе обучения.

Разработанное мобильное приложение DailyLabs – многофункциональная система контроля успеваемости. Такая система включает в себя функции:

- Объединение и систематизация предметов и лабораторных в удобном для пользователя интерфейсе.
- Предоставление информации о лабораторной.
- Расчет учебного прогресса.
- Коммуникация студентов внутри потока специальности и группы.
- Возможность задать анонимный вопрос по лабораторной работе или предмету.

Работа приложения DailyLabs представлена на рисунке 1:

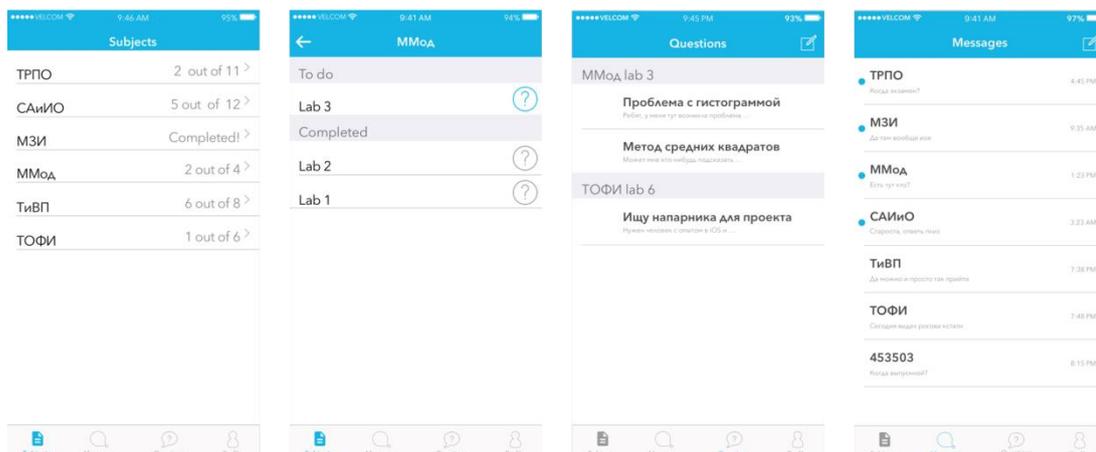


Рисунок 1 – Пример работы DailyLabs

Достоинства данного сервиса:

- Отказ от старомодных записей на бумагу и переход на новый технический и современный уровень.
- Удобный и понятный интерфейс.
- Возможность доработки сервиса для любого учебного заведения.
- Своевременная сдача лабораторных работ студентом при контроле приложением дедлайнов.
- Уведомления помогут выполнить лабораторную работу в срок.
- Наглядное планирование вектора выполнения лабораторных работ.

Использование таких средств, как excel-подобные таблички, приложений «to-do лист», записи результатов лабораторных работ на бумаге плохой вариант контроля успеваемости. Файл может быть поврежден, лист с отмеченными сданными лабораторными может быть утерян и студенту вместе с преподавателем придется восстанавливать эти данные, что занимает драгоценное время, которое могло бы быть потрачено на что-то более полезное.

Коммуникация студентов очень важна в процессе обучения. Один студент знает больше, другой меньше. Общий чат группы позволит студентам обсуждать процесс обучения. Так же видя в чате, что кто-то сдал лабораторных больше, чем ты, человеку захочется нагнать одногруппника. Элемент соревнования может способствовать мотивации студента к скорейшему выполнению лабораторных работ. Кто-то может стесняться задать вопрос преподавателю лично по тем или иным причинам. Но в нашем приложении он сможет анонимно или не анонимно задать вопрос напрямую преподавателю и обсудить все какие-либо непонятные моменты в выполнении работы.

В начале каждого семестра у студента возникает вопрос: а что это за преподаватель будет вести у меня лабораторные работы и что это за предмет такой? Он может спросить у старшекурсников, но это не всегда так просто и легко. Наше же приложение предоставляет информацию о преподавателе, дисциплине, которую он преподает и самих лабораторных.

Важнейшим критерием успешного обучения являются вовремя сданные лабораторные работы. Но не всегда удобно контролировать, когда следующий дедлайн. Наше приложение следит за выставленными преподавателем сроками для выполнения лабораторных работ и контролирует, что бы сдавали их вовремя, напоминая о дедлайне. Так же присутствует наглядная демонстрация процесса выполнения лабораторных работ.

Представленные возможности показывают, что для функционирования данного приложения нужен лишь сервер, который может позволить себе каждое учебное заведение. Другим вариантом является использование уже имеющегося сервера с достаточной вычислительной мощностью. Переход учебного заведения на электронную систему контроля успеваемости предоставит обучающимся великолепную возможность следить за процессом своего обучения и обмениваться учебной информацией. В результате быстро и без особых затрат повысится общий уровень технического оснащения университета, увеличится престиж и статус учебного заведения.

Список использованных источников:

1. Методы и формы контроля знаний студентов. [Электронный ресурс] – Режим доступа: <http://www.smt74.e-stile.ru/page17/>. – Дата доступа: 28.03.2017.

ПРИЛОЖЕНИЕ UVENT ДЛЯ РАЦИОНАЛЬНОГО ПОИСКА МЕСТ ОТДЫХА И ОРГАНИЗАЦИИ МЕРОПРИЯТИЙ

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

Одним из факторов, способствующих развитию рынка мобильных приложений, является увеличение количества «мобильных» пользователей. Сегодня перед разработчиками стоит ряд задач, связанных с поиском новых решений, способных разнообразить жизнь пользователей. В работе представлено приложение для рационального поиска мест отдыха и организации мероприятий.

Мобильные приложения – это вид IT продуктов, которые разработаны специально для мобильных устройств, смартфонов, планшетных компьютеров или других устройств, призванные решать прикладные задачи, облегчить и разнообразить жизнь их пользователей [3]. Формирование рынка мобильных приложений традиционно связывается с запуском первого специализированного магазина приложений Apple App Store для пользователей операционной системы iOS в 2008 году. Стоит отметить, что в настоящее время мобильные приложения выступают одним из главных трендов в развитии информационных технологий.

Клаус Шваб отмечает, что цифровые технологии, которые основаны на аппаратном и программном обеспечении и сетях, не являются новшеством, однако они останутся более усовершенствованными и интегрированными. Это уводит нас от третьей промышленной революции и вызывает трансформацию глобальной экономики и общества [4]. Сегодня мы говорим о четвертой промышленной революции, которая кардинальным образом изменит способ взаимодействия между людьми, организациями и учреждениями. Повсеместное увеличение числа интернет-пользователей, ежегодное увеличение мобильного трафика свидетельствует об актуальности развития рынка мобильных приложений. Согласно отчёту Global Digital Snapshot: в среднем пользователь смартфона в 2017 г. потребляет в месяц на 70% больше мобильного трафика, чем в 2016 году [1]. На сегодняшний день рынок мобильных приложений достаточно разнообразен: Google Play, Windows Phone Store, BlackBerry App World и др.

И. А. Карпюк выделяет следующие типы приложений [2]:

- приложения-события (предназначены для трансляции спортивных или иных событий);
- приложения службы (аналоги сайтов);
- игры;
- интернет-магазины;
- промо-приложения;
- бизнес-приложения (позволяют оптимизировать процесс работы организации);
- системные приложения, использующие дополнительные настройки и опции телефона и его программного обеспечения;
- навигационные и поисковые сервисы;
- мультимедийные приложения, расширяющие возможности телефона при работе с видео и аудио информацией;
- социальные сети (представляют собой онлайн-сервисы для общения, распространения информации и организации социальных взаимоотношений);
- контентные приложения и др.

Предлагаемое мобильное приложение Uvent можно отнести к мобильным социальным сетям. Этапу разработки мобильного приложения предшествовал онлайн опрос потенциальной целевой аудитории в отношении проблем, с которыми они сталкиваются при выборе места досуга. 46% молодых людей ответили, что они сталкиваемся с проблемой выбора, особенно, если идея встретиться пришла кому-нибудь из них спонтанно. В результате достаточно часто их встречи отменялись, или они посещали те места, в которых они неоднократно бывали.

Нельзя не учитывать тот факт, что такие порталы как relax.by, afisha.tut.by размещают информацию о мероприятиях города. Однако, если вы захотели сами что-то организовать, например, музыкальный вечер, танцевальный батл и т.д., разместить информацию о своем мероприятии и собрать аудиторию с помощью данных порталов проблематично. Предлагаемое мобильное приложение предназначено для рационального поиска мест отдыха и организации мероприятий. Приложение обслуживает потребности индивидуальных и корпоративных пользователей (например, бар, кафе и т.д.). При разработке мобильного приложения Uvent использовался язык программирования Swift, доступно для мобильных устройств на базе iOS.

Приложение позволяет:

- производить поиск мест отдыха по заданным критериям (бюджет, тип встречи, день недели);
- приглашать друзей;
- создавать свое мероприятие и индивидуально его настроить (с указанием места проведения, стоимости),
- добавить дополнительные ссылки, для получения информации о мероприятии;
- экономить время для принятия решения о месте и форме проведения досуга с друзьями.

Список использованных источников:

1. Итоги 2017 года в мобильном маркетинге. [Электронный ресурс]. – Режим доступа: <https://www.cossa.ru/trends/189900/>. – Дата доступа: 03.04.2018
2. Карпюк И. А., Куляшова Н. М. Сравнительный анализ мобильных приложений и инструментальных средств их разработки // Научно-методический электронный журнал «Концепт». – 2017. – Т. 31. – С. 826–830. – [Электронный ресурс]. – Режим доступа: <http://e-koncept.ru/2017/970180.htm>. – Дата доступа: 03.04.2018
3. Рынок магазинов приложений и контента, 2010-2015 // J'son & Partners Management Consultancy [Электронный ресурс]. – Режим доступа: http://www.json.ru/poleznye_materialy/free_market_watches/analytics/rynok_magazinov_prilozhenij_i_kontenta_2010-2015/. – Дата

доступа: 03.04.2018

4. Шваб К. Четвертая промышленная революция / К. Шваб – Эксмо», 2016 [Электронный ресурс]. – Режим доступа: <http://www.eurasiancommission.org/ru/> . – Дата доступа: 03.04.2018

СЕРВИС FOLLOW THE ART ДЛЯ ПРОДАЖИ И ЗАКАЗА КАРТИН

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

Лазорина К.И.

Жвакина А.В. – к.т.н., доцент

Картина, висящая на стене, – только на первый взгляд привычный аксессуар. На самом деле, картинный декор – сложное искусство, требующее внимательного и тщательного отношения. Представленный в данной работе сервис предназначен для повышения эффективности продажи и упрощения процесса покупки картин.

В настоящее время найти оригинальное произведение искусства в Беларуси не так просто: нужно знать, где искать. Разумеется, существуют картинные галереи. Но не все талантливые художники могут себе это позволить, так как там они могут выставить свои картины на продажу, жертвуя почти третью ее стоимости. Мест, где можно продавать картины самостоятельно, в Минске очень мало и о них почти никто не знает. Остается искать картины в интернете на различных сайтах о продаже.

Разработанное мобильное приложение FollowTheArt – многофункциональная система для продажи и заказа картин в режиме online. Такая система включает в себя функции:

- Авторизация пользователей.
- Каталог картин белорусских художников.
- Совершение сделок по продаже картин.
- Заказ картин у художников.
- Подбор картин по критериям.

На фоне похожих приложений и FollowTheArt обладает рядом преимуществ, делающих его уникальным:

- Множество работ художников со всей Беларуси, собранных в одном приложении.
- Максимальный доход художников от продажи своих работ.
- Возможность прямой связи покупателя с художником.
- При загрузке нового изображения, оно автоматически отображается в разделе "Feed" на домашней странице, с максимальной аудиторией.
- Возможность покупателей с помощью различных фильтров подобрать для себя картину, идеально вписывающуюся в интерьер.
- Пользователь может сделать заказ картины по фотографии.

На рис.1 представлены форма авторизации пользователей и обновляющаяся домашняя страница «Feed».

В приложении имеется четыре основные вкладки:

- Каталог (место, куда попадают картины, которые выставили на продажу),
- поиск с фильтрацией картин по определенным критериям,
- заказ картины
- профиль пользователя.

Использование разработанного приложения позволит упростить процесс приобретения картин и будет способствовать популяризации белорусской живописи.

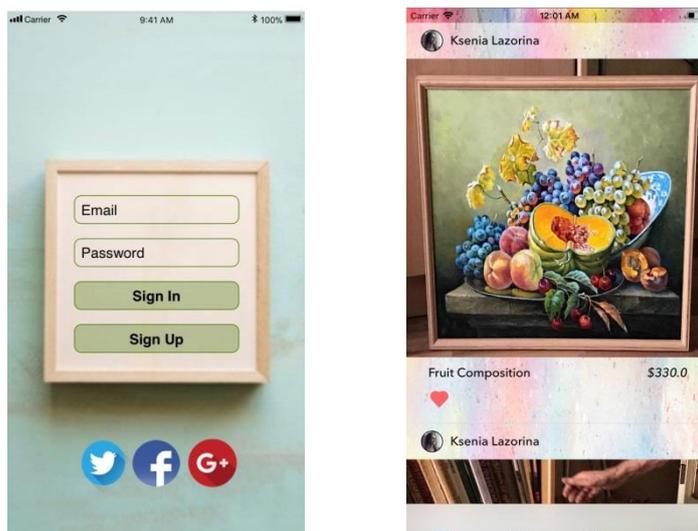


Рисунок 1 - Форма авторизации пользователей и обновляющаяся домашняя страница «Feed»

Список использованных источников:

1. Художественной галерее "Беларт". [Электронный ресурс] – Режим доступа: <http://bel-art.by/index.php> – Дата доступа: 03.04.2018.
2. Торговая площадка "Artmajeur". [Электронный ресурс] – Режим доступа: <https://www.artmajeur.com> – Дата доступа: 01.04.2018.

АППРОКСИМАЦИЯ ФУНКЦИЙ В ВИДЕ ЭКСПОНЕНЦИАЛЬНОЙ СУММЫ

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

Мизеев В.Д., Лесун А.И.

Анисимов В.Я. – к.ф.-м.н., доцент

В мире, окружающем нас, все взаимосвязано, поэтому одной из часто встречающихся задач является нахождение характера зависимости между различными величинами, что позволяет по значению одной величины определить значение другой. Математической моделью зависимости одной величины от другой является понятие функции. При математическом моделировании зачастую требуется представить некую зависимость, заданную отдельными точками, в виде гладкой функции. Исходные точки могут быть заданы с ошибками. В данном случае целесообразно применить аппроксимацию исходных данных методом наименьших квадратов.

Пусть требуется установить зависимость между двумя переменными x и y по результатам экспериментальных измерений. Для этого необходимо построить кривую, которая воспроизводила бы график исходной экспериментальной закономерности, т.е. была бы максимально близка к экспериментальным точкам, но в то же время была бы невосприимчива к случайным отклонениям измеряемой величины. Выбор функции зависит от теоретических исследований и характера расположения на плоскости экспериментальных точек. Часто в практических работах ограничиваются линейной функцией, где требуется определить коэффициенты.

Однако в большинстве случаев данный вид функции недостаточно точно описывает зависимость. Поэтому целесообразней представить функцию в другом виде. Один из вариантов – это представление функции в виде экспоненциальной суммы:

Данный вид позволяет получить достаточно хороший уровень приближения. Но если попробовать представить функцию в таком виде, то в результате стандартного метода МНК мы придём к системе нелинейных уравнений, которые зачастую решаются довольно сложно.

Для того, чтобы упростить эту аппроксимацию рассмотрим функцию, которая соответствует аппроксимации в сумму 2 экспонент:

Для того, чтобы найти коэффициенты воспользуемся МНК:

Если в качестве функции мы возьмём , то в результате преобразований мы получим трёхчлен .

Его решением являются коэффициенты, находящиеся в показателе степени экспоненты. Оставшиеся коэффициенты можно получить применив МНК для функции с уже посчитанными коэффициентами .

Аналогичным образом можно представить функцию в виде суммы большего числа экспонент.

Список использованных источников:

1. Жевняк Р.М., Карпук А.А. Высшая математика: Функции многих переменных. Интегральное исчисление функций одной и многих переменных. Векторный анализ: Учеб. – Мн., Выш. Шк., 1993. – 411с.:ил.
2. Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. Численные методы – М.:Лаборатория Базовых знаний, 2001 г. – 632с.:ил.
3. Минченко Л.И. Краткий курс численного анализа. Учебное пособие по курсу “Методы численного анализа” для студ. Спец. “Информатика” для всех форм обучения. –Мн.:БГУИР, 2006. –92с.:ил.

ПРИЛОЖЕНИЕ ДЛЯ ПОМОЩИ ПЕРВОКУРСНИКАМ В УЧЁБЕ

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

Таланец А.В., Лесун А.И., Мизеев В.Д.

Бережнов Д.Е. – асс. к.

При поступлении в университет коренным образом меняется стиль жизни, условия труда и отдыха, режим дня первокурсника в целом. Но кроме проблемы с адаптацией, поначалу тяжело усваивать новые знания, которые кардинально отличаются и объёмом, и сложностью от тех, что дают в школе, поэтому возникла идея создать приложение, которое могло бы облегчить учёбу первокурсника.

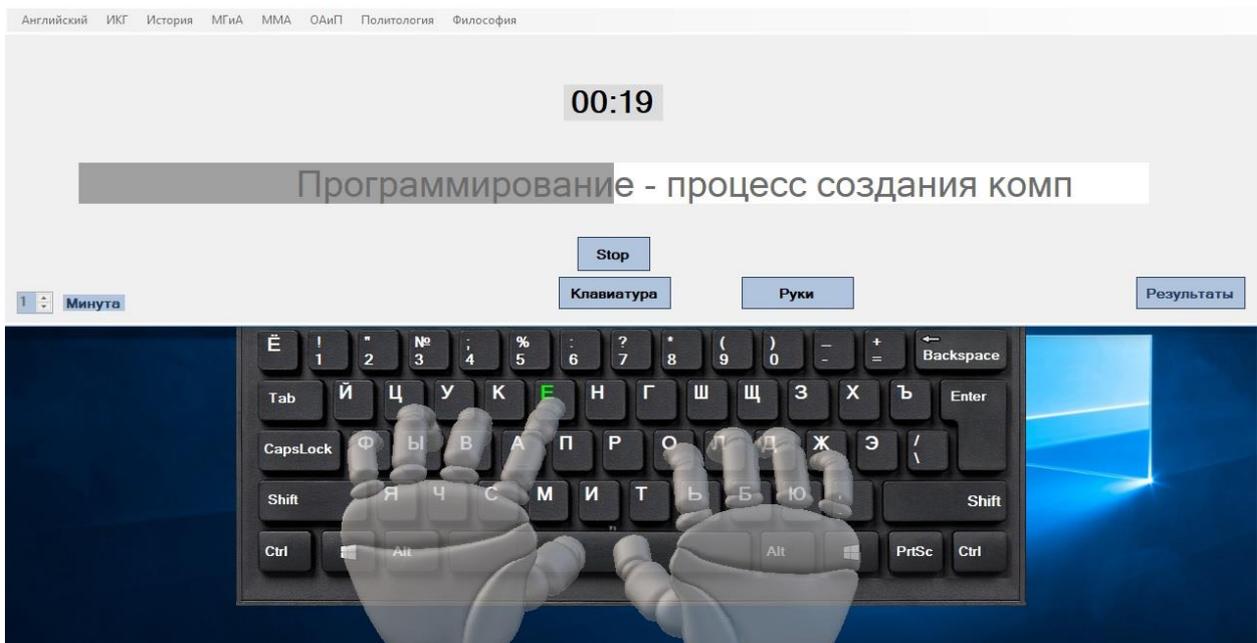
Приложение «Helper первокурсника» предназначено для помощи первокурсникам в обучении некоторым дисциплинам, для улучшения навыков набора текста и проверки собственных знаний. Также в программе содержится информация, которая поможет первокурснику адаптироваться на новом месте.

Одной из главных идей проекта является тренировка быстрого набора текста на русском и английском языках. Особенностью тренажёра является наличие графических элементов, которые показывают наиболее удобное положение рук при нажатии на клавиши. Кроме того, тексты представляют собой материалы дисциплин, изучаемых на первом курсе, что позволяет во время тренировки улучшить свои знания в соответствующей дисциплине.

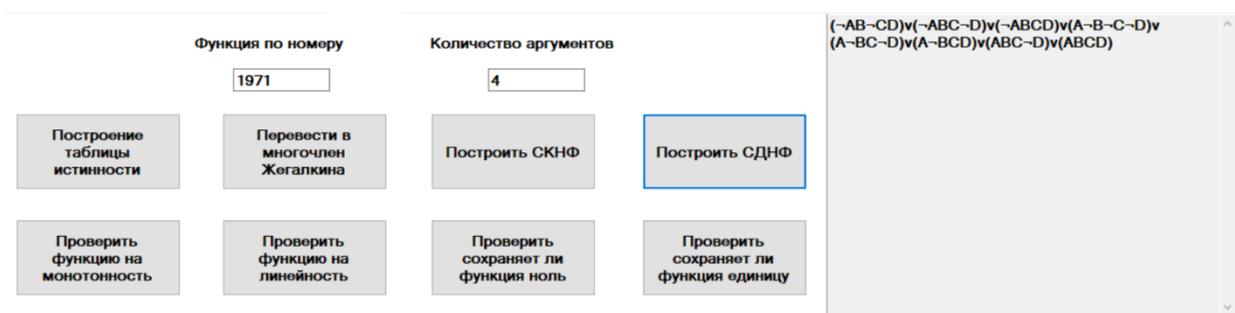
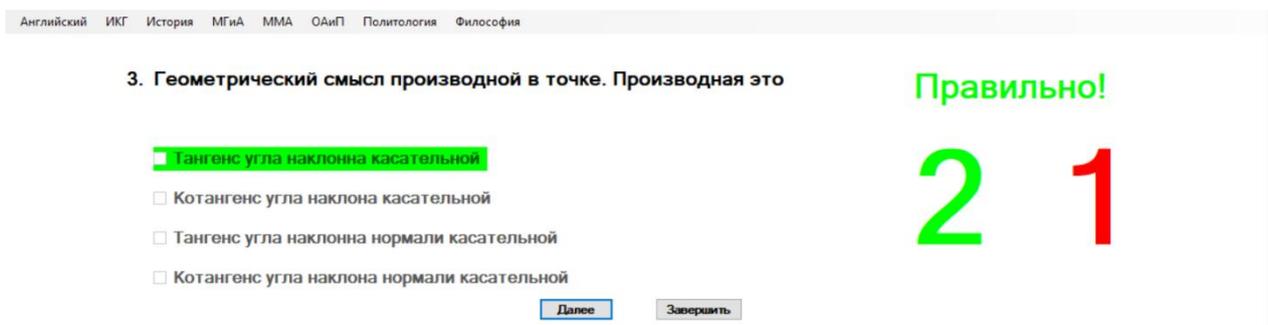
В приложение встроен специальный полигон для тестирования программ, в том числе и лабораторных, что может помочь при создании качественных программ, в которых учтены всевозможные входные данные.

Также в приложении есть калькулятор с полезными в учёбе и выполнении различных расчётов функциями. К примеру, для матлогики составлены функции:

- построение таблиц истинности
- расчёт СДНФ
- расчёт СКНФ
- построение многочлена Жегалкина
- проверка на принадлежность к определённому классу функций
- и другие



Проверка знаний осуществляется в виде теста по интересующей теме.



Основные преимущества:

- 1) Приложение полезно для многих студентов 1го курса технических университетов
- 2) Множество полезных возможностей собраны в одном приложении и не придётся тратить время на поиск нужной информации вручную
- 3) Не требуется доступ к интернету
- 4) Имеет понятный интерфейс и не требует специальных знаний для использования

Список использованных источников:

1. Руководство по языку C# <https://docs.microsoft.com/ru-ru/dotnet/csharp/index>

ИСКУССТВЕННЫЕ НЕЙРОННЫЕ СЕТИ В МЕДИЦИНСКОЙ ДИАГНОСТИКЕ

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

Луговская А.М.

Жвакина А.В. – к.т.н., доцент

Диагностика играет в медицине важнейшую роль. Своевременно поставленный точный диагноз облегчает выбор метода лечения и значительно повышает вероятность выздоровления больного. Применение нейронных сетей является одним из способов повышения эффективности медицинской диагностики.

Точность диагноза и быстрота, с которой его можно поставить, зависят, разумеется, от очень многих факторов: от состояния больного, от имеющихся данных о симптомах и признаках заболевания и результатах лабораторных анализов, от общего объема медицинской информации о наблюдении таких симптомов при самых различных заболеваниях и, наконец, от квалификации самого врача. Большую роль в этом процессе играет человеческий фактор, что нередко ведёт к ошибкам.

Ниже перечислены некоторые определенные трудности медицинского диагноза, которые необходимо учитывать:

Основа для достоверного диагноза - богатый практический опыт - достигается только к середине карьеры врача и, естественно, отсутствует в конце академического образования.

Особенно это актуально для редких или новых заболеваний, где опытные врачи находятся в той же ситуации, что и новички.

Качество диагностики напрямую зависит от мастерства, знаний и интуиции врача.

Эмоциональные проблемы и усталость негативно сказываются на работе врача.

Обучение специалистов является длительной и дорогостоящей процедурой, в связи с чем во многих, даже в развитых, странах мы можем ощущать недостаток мастеров.

Медицина - одна из наиболее быстро растущих и развивающихся областей науки. Новые результаты дисквалифицируют прежние, новые лекарства появляются день ото дня. Это же относится и к самим заболеваниям, которые принимают новые формы.

Данные факторы обуславливают необходимость поиска новых решений и средств, например, использовать искусственные нейросети (ИНС). ИНС — это программная реализация нейронных структур нашего мозга. Биологический нейрон — это специальная клетка, которая структурно состоит из ядра, тела клетки и отростков. Одной из ключевых задач нейрона является передача электрохимического импульса по всей нейронной сети через доступные связи с другими нейронами. При этом, каждая связь характеризуется некоторой величиной, называемой силой синаптической связи. Эта величина определяет, что произойдет с электрохимическим импульсом при передаче его другому нейрону: либо он усилится, либо он ослабится, либо останется неизменным. Нейрон активируется тогда, когда суммарный уровень сигналов, пришедших в его ядро, превысит определенный уровень (порог активации).

Искусственный нейрон — упрощенная модель биологического нейрона. ИНС представляет собой набор узлов (нейронов) и связей (синапсов) между ними. Сила связи определяется величиной, называемой весом. Связи с положительным весом называются возбуждающими, а с отрицательным — тормозящими.

Одной из областей, где ИНС проявляют себя лучше всего — задачи классификации, к которым и относится диагностика. Существует множество различных типов нейронных сетей, но наиболее часто для задач классификации вообще, и в медицинской диагностике в частности, используется многослойный персептрон.

Многослойный персептрон является сетью с прямым распространением сигнала (без обратных связей), обучаемой с учителем. Для обучения в большинстве случаев используется алгоритм обратного распространения ошибки, который минимизирует среднеквадратичную ошибку нейронной сети. Обучение построено на простой идее, что, зная эталонный ответ, мы можем вычислить разницу между полученным и требуемым результатами, т.н. ошибку. Эту ошибку можно отправить обратно ко всем входам нейрона и определить, какой вход насколько сильно повлиял на эту ошибку, и соответственно, подкорректировать вес на этом входе так, чтобы ошибку уменьшить. В качестве активационной функции чаще всего используют сигмоидальную функцию, разновидностями которой являются логистическая функция и гиперболический тангенс.

Нейронные сети — очень универсальный инструмент, т.к. они способны обрабатывать данные, полученные из самых различных источников. Однако нужно понимать, что нейронная сеть может работать только с числовыми данными, поэтому один из ключевых этапов в обучении сети — предварительная подготовка данных: кодирование и нормирование.

Все данные можно разделить два типа - численные и категориальные. Численные данные — это данные, которые непрерывно изменяются в некотором диапазоне. Категориальные данные являются дискретными. Частным случаем категориальных данных являются бинарные данные, кодирование которых заключается в закреплении за одной категорией значения 1, а за другой — 0. В выбранной для исследования области в роли объектов выступают пациенты. Данные представляют собой результаты обследований, симптомы заболевания и применявшиеся методы лечения. Примеры бинарных признаков: пол, наличие головной боли, слабости. Категориальный признак — тяжесть состояния (удовлетворительное, средней

тяжести, тяжёлое), группа крови. Количественные (численные) признаки — возраст, пульс, артериальное давление, уровень холестерина.

Нормирование представляет собой процесс приведения данных к фиксированному диапазону. Одной из причин, обуславливающих необходимость этой процедуры, являются ситуации, когда одна переменная изменяется в широком диапазоне (например, от 1000 до 5000), а другая – в узком (например, от 0,1 до 0,4). Очевидно, что ошибки, обусловленные влиянием первой переменной будут сильнее влиять на обучение, чем ошибки, обусловленные второй. Приведение каждой переменной к определённому фиксированному диапазону обеспечивает равное влияние каждой из них на изменение весов в процессе обучения.

Один из методов сжатия данных в заданный диапазон заключается в выделении максимальных и минимальных значений. Если в качестве требуемого диапазона выбран диапазон от нуля до единицы, тогда каждую величину t поделим на диапазон ее изменения, чтобы получить новую величину S . Преимущество этого метода заключается в том, что он сохраняет соотношения между величинами.

Линейное нормирование величины t в переменную s , распределенную в диапазоне от 0 до 1 можно осуществить с помощью формулы:

С целью разобраться в архитектуре нейронных сетей и оценить преимущество использования многослойного персептрона для диагностики заболеваний, мною разработана и обучена нейронная сеть, определяющая, имеются ли у пациента болезни сердца, на основе данных обследования таких как кровяное давление, частота сердечбиения, уровень холестерина и т.д., а также некоторые специфичные показатели (смещение ST сегмента), учитывая возраст и пол пациента (в общей сложности 13 признаков).

Для программной реализации выбран язык C#. Нейронная сеть была написана с нуля без использования готовых программных пакетов для проектирования ИС. Сеть состоит из трех слоев: входного, скрытого и выходного. Для обучения и тестирования использовалась база данных, представленная в репозитории UCIMachineLearningRepository. Входные выборки автоматически делятся на обучающие и тестовые наборы. Данные приводятся к диапазону $[0, 1]$ с помощью линейного нормирования.

Практические исследования показали, что нейронная сеть, имеющая такую структуру, действительно хорошо справляется с поставленной задачей и при наличии соответствующей базы данных может быть обучена для диагностирования многих болезней.

ИНС представляют собой мощный инструмент, помогающий врачам проводить диагностику. В связи с этим ИНС имеют ряд преимуществ, в том числе:

- Возможность обрабатывать большой объем данных
- Снижение вероятности недоучёта соответствующей информации
- Сокращение времени диагностики

Кроме того, их использование делает диагноз более надежным. Методы обобщения и разработки информационных и интеллектуальных данных постоянно совершенствуются и могут в значительной степени способствовать эффективной, точной и быстрой медицинской диагностике. Однако, несмотря на все преимущества использования нейросетей, их следует рассматривать только как вспомогательный инструмент, помогающий медику принять окончательное решение, т.к. в конечном итоге ответственность лежит на нём.

Список использованных источников:

1. Importance of Artificial Neural Network in Medical Diagnosis disease like acute nephritis disease and heart disease Irfan Y. Khan, P.H. Zope, S.R. Suralkar Dept. of Ele. & Tele. SSBT's college of Engg. & Tech, Bambhori, Jalgaon, India
2. Dua, D. and Karra Taniskidou, E. (2017). UCI Machine Learning Repository [<http://archive.ics.uci.edu/ml>]. Irvine, CA: University of California, School of Information and Computer Science.
3. Artificial Neural Networks in Medical Diagnosis Qeethara Kadhim Al-Shayea MIS Department, Al-Zaytoonah University of Jordan Amman, Jordan
4. Neural Network Using C# by James McCaffrey – 128 с.
5. С. Хайкинг Нейронные сети. Полный курс – 1101с.
6. Kevin Swingler "Applying Neural Networks. A practical Guide"

ПРОГРАММНОЕ СРЕДСТВО МОДЕЛИРОВАНИЯ СТРУКТУРНО-СЛОЖНЫХ СИСТЕМ

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

Лукашик Р.В.

Волорова Н.А. – к.т.н., доцент

Целью работы является разработка программного средства моделирования структурно-сложных систем. К структурно-сложным системам относятся многие современные промышленные предприятия, такие, как объекты нефтегазовой отрасли, химической промышленности, энергетики и многие другие.

Первым делом для реализации поставленной цели необходимо было определиться с методом. Выбор стоял между методом деревьев отказов, методом блок-схем работоспособности и общим логико-вероятностным методом. После тщательного сравнительного анализа вышеперечисленных методов было принято решение в пользу общего логико-вероятностного метода. Данный метод позволяет рассчитать показатели надежности, живучести, безопасности, отказа сложных систем.

Следующим этапом реализации поставленной цели являлось непосредственно разработка программного средства. Разработка предусматривала автоматизацию следующих этапов:

- На основе структурной модели (схемы функциональной целостности) построить систему логических уравнений.

- Построить логическую функцию работоспособности системы. Данная функция строится на основе деревьев решений универсального графоаналитического метода. Результат выполнения данного этапа представляется в виде всевозможных комбинаций безопасной или аварийной работы системы в виде дизъюнктивной нормальной формы.

- Построить вероятностную функцию. Данная функцию строится комбинированным методом.

- Рассчитать вероятности выполнения заданных критериев исследуемой системы.

Пользователю остается лишь построить схему функциональной целостности (рис. 1), выбрать логический критерий функционирования, по которому будут производиться расчеты, и задать вероятностные параметры элементов системы. Все остальное будет рассчитано автоматически.

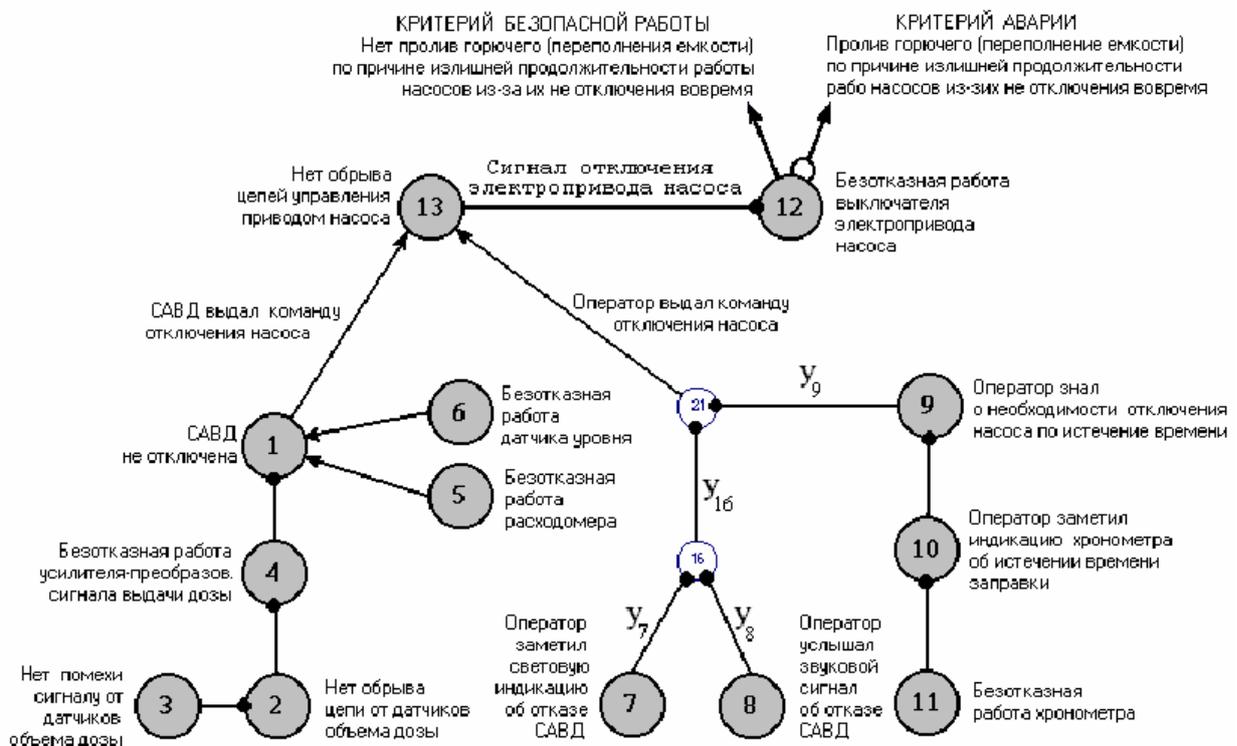


Рис. 1 – Прием схемы функциональной целостности

Программное средство разрабатывалось как веб-приложение, написанное на языке программирования JavaScript.

В настоящее время существуют аналоги разработанного программного средства. Например, АРБИТР, RiskSpectrum, Cris, RastrWin. Перечисленные программные средства являются платными и большинство не имеют даже бесплатной пробной версии. В связи с этим, разработанное мною программное средство, являясь веб-приложением и находясь в свободном доступе, идеально подходит для студентов и магистрантов для выполнения лабораторных и курсовых работ.

Список использованных источников:

1. Рябинин И. А., Можаяев А. С., Свиринов С. К., Поленин В. М. Технология автоматизированного моделирования структурно-сложных систем. // «Морская радиоэлектроника», № 3 (21), сентябрь 2007, с. 58-63.
2. Можаяева И. А. Методики структурно-логического моделирования сложных систем с сетевой структурной: Диссертация, 2015. 170с.

АЛГОРИТМЫ И МЕТОДЫ ОПТИМАЛЬНОГО КОДИРОВАНИЯ АППАРАТУРЫ САМОТЕСТИРОВАНИЯ ДЛЯ ОПЕРАТИВНЫХ ЗАПОМИНАЮЩИХ УСТРОЙСТВ

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

Лукьянов А.А.

Иванюк А.А. – д.т.н., доцент

Оперативные запоминающие устройства (ОЗУ) - энергозависимая часть системы компьютерной памяти, в которой во время работы устройства хранится исполняемый машинный код (программы), а также входные, выходные и промежуточные данные, обрабатываемые процессором. Современные операционные системы, активно используют оперативную память, для хранения и обработки в ней важных и часто используемых данных. Если бы в электронных устройствах не использовалась оперативная память, то все операции происходили бы гораздо медленней и для считывания с постоянного источника памяти (ПЗУ), требовалось бы значительно больше времени. Из за высокой нагрузки на ОЗУ, важным вопросом является тестирование памяти для обеспечения надежности ее работы.

С увеличением сложности цифровых устройств наиболее актуальным и распространенным способом тестирования стала встроенная аппаратура самотестирования (BIST — Built-In Self-Test). Ядро BIST передает тестовые последовательности тестируемому устройству, считывает результаты и сравнивает их с ожидаемыми, и таким образом определяет, исправно ли тестируемое устройство. Существуют различные подходы к проектированию BIST. BIST может реализовывать один определенный алгоритм тестирования. В таком случае аппаратные затраты на BIST минимальны и скорость применения теста является максимальной, но данный подход является наименее гибким, так как он определяет единственный алгоритм тестирования, способный обнаруживать фиксированный набор неисправностей. Изменение алгоритма тестирования невозможно без необходимости перепроектирования BIST. Альтернативным подходом является BIST с микропрограммным управлением P-MBIST (Programmable Memory BIST), в которой алгоритм тестирования определяется микрокодом, хранящимся в памяти микропрограмм. P-MBIST позволяет изменять алгоритм тестирования во время жизненного цикла тестируемого устройства.

Было разработано много алгоритмов тестирования памяти для покрытия FFM (functional fault model), большинство из которых имели теоретическое происхождение. Традиционные ad-hoc тесты использовались в прошлом для неисправных устройств. Широкоизвестные тесты проход 1/0, GALPAT, Butterfly, Zero-one тест и Checkerboard. Однако, временная сложность первых двух тестов совершенно неприемлема в данный момент; в то время как покрытие ошибок последних трех тестов неприемлемы в промышленности. Маршевые тесты были введены для обнаружения Inversion Coupling Faults (CFin), Idempotent Coupling Faults CFid (CFid), а также SFs [1]. В таблице 1 представлено покрытие дефектов маршевыми тестами. В таблице, например, «a/b» означает, что тест обнаруживает «а», из «b» FP (fault primitive) соответствующего FFM. Например, March C обнаруживает оба FP TF, в то время как MATS+ обнаруживает только одну из них. CFin не включен в таблицу, т.к. имеет теоретическое происхождение и никогда не проявлялся в реальных конструкциях.

FFM	March Tests							
	MATS+	March C-	March B	PMOVI	March U	March LR	March Sr	March SS
SF	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2
TF	1/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2
WDF	0/2	0/2	0/2	0/2	0/2	0/2	0/2	2/2
RDF	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2
DRDF	0/2	0/2	0/2	2/2	0/2	0/2	2/2	2/2
IRF	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2
CFst	4/8	8/8	6/8	8/8	8/8	8/8	8/8	8/8
CFid	3/8	8/8	8/8	7/8	8/8	8/8	8/8	8/8

Табл. 1

Один из возможных способов реализации маршевых тестов на P-MBIST - микропрограммный подход. В отличие от BIST в нем используется дополнительный запоминающий модуль, предназначенный для хранения тестовых наборов и микрокодов маршевых операций. Набор микроопераций также должен включать в себя операции сравнения, команды переходов, автодекремента и инкремента, остановки тестирования [2]. Маршевый тест можно представить как совокупность predetermined компонентов, представляющих собой целую фазу теста [3]. В маршевых тестах из табл. 1 можно выделить следующие компоненты: SM0 (wd), SM1 ($\bar{r}\bar{d}$, wd), SM2 ($\bar{r}\bar{d}$, wd, rd, $\bar{w}\bar{d}$), SM3 ($\bar{r}\bar{d}$, wd, $\bar{w}\bar{d}$), SM4 ($\bar{r}\bar{d}$, $\bar{r}\bar{d}$, $\bar{r}\bar{d}$), SM5 ($\bar{r}\bar{d}$), SM6 ($\bar{r}\bar{d}$, wd, $\bar{w}\bar{d}$, wd), SM7 ($\bar{r}\bar{d}$, wd, rd). С учетом программной реализации можно составить микропрограмму (пример на рис. 1), которая воспроизведет маршевый тест для аппаратуры встроенного самотестирования ОЗУ.

Одним из важных вопросов программирования аппаратуры самотестирования является оптимизация микрокода. В работе [3] предложен способ кодирования, при котором каждый элемент может быть представлен 5 значениями: AO - порядок обхода, LO - флаг, показывающий является ли операция последней

в текущем SMx компоненте, OT - тип операции (r/w), dt - эталонное значение, NOE - количество SMx компонентов в маршевом тесте. Каждый маршевый тест может быть закодирован:

Таким образом, большинство маршевых тестов будет занимать не более 42 бит.

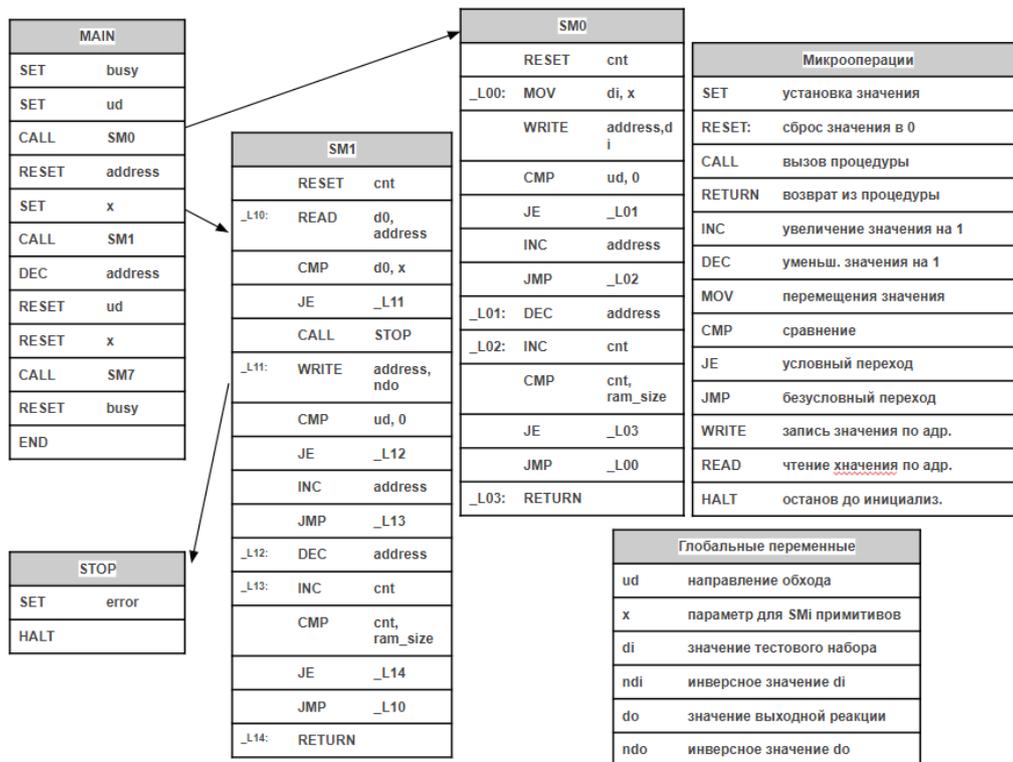


Рис. 1

Альтернативой может являться метод программируемого конечного автомата. Этот метод заключается в реализации маршевых примитивов как последовательностей состояний управляющего автомата. Входные параметры и порядок следования могут храниться в небольшой памяти. Это решение обладает меньшими аппаратными затратами, но в то же время менее гибкое.

Развитие средств встроенного самотестирования ОЗУ связано с появлением так называемых неразрушающих маршевых тестов, идея применения которых базируется на использовании сигнатурного анализа. Маршевый тест может быть использован для проведения процедуры неразрушающего тестирования, которая состоит из трех частей [4]:

- Вычисление значения эталонной сигнатуры SL;
- Применение неразрушающего теста с вычислением значения рабочей сигнатуры SW;
- Анализ значений двух сигнатур SL и SW, по результату которого можно судить о состоянии ОЗУ;

В работе рассмотрены некоторые методы и алгоритмы, применимые для тестирования ОЗУ. Так как одна из проблем тестирования ОЗУ заключается в небольшом количестве памяти, доступной для хранения, в дальнейшем планируется рассмотрение и создание методов оптимизации микрокода встроенной аппаратуры самотестирования ОЗУ.

Список использованных источников:

1. Said Hamdioui, Ad J. van de Goor "March SS: A Test for All Static Simple RAM Faults" In proc IEEE International Workshop on Memory Technology, Design and Testing, 2002
2. Иванюк А.А., Ярмолик В.Н. // Проектирование контролепригодных цифровых устройств. 2006. С. 193–205
3. Ivaniuk A. // Optimal Memory Tests Coding for Programmable BIST Architecture. 2008.
4. Zarineh K. Upadhyaya S.J. // Design, Automation and Test in Europe (DATE'99): proc. of IEEE Int. Conf. Munich, Germany. Mar. 9–12, 1999. P. 709–713.

УПРАВЛЕНИЕ ДОСТУПОМ В ПРИЛОЖЕНИЯХ НА БАЗЕ ФРЕЙМВОРКА RUBY ON RAILS

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

Медунецкий М.А., Хоронько М.П.

Стержанов М.В. – к.т.н., доцент

В настоящее время на многих сайтах присутствует система управления доступом и разделения прав на выполнение действий. Наибольшую популярность в веб-программировании получила система на основе ролей, которая и будет описана в этой работе.

Управление доступом на основе ролей — развитие политики избирательного управления доступом, при этом права доступа субъектов системы на объекты группируются с учётом специфики их применения, образуя роли[1]. Например, существует несколько типов пользователей сайта со своими правами. Гости могут только смотреть информацию, зарегистрированные пользователи оставлять комментарии, модераторы удалять их, а администраторы изменять содержимое веб-ресурса.

Рассмотрим пользователя сайта, который представлен моделью User. Модель пользователя должна содержать следующие поля: логин или email, пароль (обычно в целях безопасности хранится хэш пароля) и роль. Зачастую роль представлена перечислением и в базу данных сохраняется номер элемента в этом перечислении. Ruby On Rails предоставляет удобный метод взаимодействия с базой данных через класс ActiveRecord. Модель наследуется от этого класса и получает удобные методы для выполнения CRUD (создание, чтение, изменение и удаление) операций.

Для аутентификации и работы с ролями удобно использовать gem Devise [2], который содержит методы для проведения этих действий.

Аутентификация — процедура проверки подлинности. Эта процедура выполняется следующим образом. Если посетитель сайта не вошёл в систему, то отправляет запрос, который содержит логин и пароль. Сервер ищет такого пользователя в базе данных и в случае успеха генерирует и возвращает токен доступа. В дальнейшем этот токен отправляется в каждом запросе, обычно заголовком. Сервер получает запрос и сопоставляет полученный токен с пользователем, которого можно получить методом `current_user`, если пользователь не идентифицирован, то метод возвращает `nil`. Devise содержит и другие методы для проверки аутентификации посетителя.

Авторизация — предоставление определённому лицу или группе лиц прав на выполнение определённых действий. Для установки ограничений на доступ описываются методы, которые возвращают булево значение. Затем они прикрепляются к контроллеру, либо к его методам. Это позволяет запретить пользователям получение или изменение информации, чья группа не уполномочена на такие действия.

Таким образом, фреймворк Ruby on Rails предоставляет все необходимые инструменты для создания и поддержки приложения с управлением доступом на основе ролей, а gem Devise значительно упрощает эту работу.

Список использованных источников:

1. Ferraiolo D. F., Kuhn D. R. (October 1992). "Role Based Access Control". 15th National Computer Security Conference: 554—563.
2. <https://github.com/plataformatec/devise> - Flexible authentication solution for Rails on github

РЕКУРРЕНТНАЯ НЕЙРОННАЯ СЕТЬ LSTM ДЛЯ ПРОГНОЗИРОВАНИЯ РЕЗУЛЬТАТОВ СОРЕВНОВАНИЙ ПО КИБЕРСПОРТУ

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

Минчук С.Г.

Жвакина А. В. – канд. техн. наук, доцент

Возрастающая популярность киберспорта, большое количество соревнований по всему миру, в том числе международных, рост призового фонда, а также признание нескольких киберспортивных дисциплин настоящим видом спорта во многих странах заставляют относиться к данным мероприятиям не как обычным развлечениям.

Прогнозирование направлено на определение тенденций развития конкретного объекта или события на основе предыдущих данных, иными словами, анализа его состояния в прошлом и настоящем [7]. Результат игры базируется не только на данных в определенный момент, а также данных, собранных на протяжении всей игры до этого момента. Рекуррентные нейронные сети — это сети, основанные на использовании предыдущих состояний для вычисления текущего, поэтому необходимо рассмотреть возможность

использования для прогнозирования рекуррентной сети LSTM, которая способна к обучению долгосрочным зависимостям.

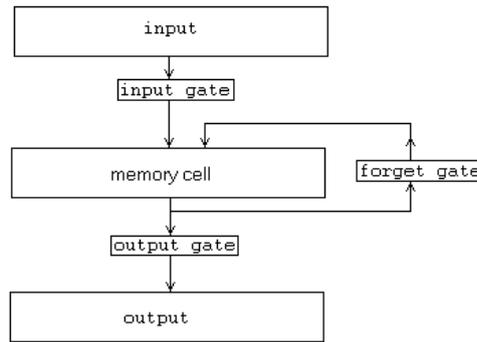


Рис 1. Схема нейронной сети LSTM

LSTM состоит из следующих частей [2] (рис.1): вход (*input*) нейронной сети; выход (*output*) нейронной сети; внутреннее состояние нейронной сети или запоминающая ячейка (*memorycell*); фильтр очистки памяти или фильтр забывания (*forgetgate*); входной фильтр или фильтр обновления памяти (*inputgate*); выходной фильтр или фильтр выдачи результата (*outputgate*).

Рассмотрим подробнее структуру рекуррентной сети LSTM [2]. Центральным понятием здесь является запоминающая ячейка, которая, наряду с состоянием сети, вычисляется на каждом шаге, используя текущее входное значение и значение ячейки на предыдущем шаге, где t это шаг обучения.

Входной фильтр определяет, насколько значение запоминающей ячейки на текущем шаге должно влиять на результат. Значения входного фильтра лежат в интервале $[0, 1]$, что обеспечивается областью значений функции активации:

- где w – веса входа сети для входного фильтра,
 x_t – вход сети на текущем шаге,
 y_{t-1} – выход сети на предыдущем шаге,
 w – веса выхода сети для входного фильтра,
 z – состояние памяти сети,
 w – веса состояния памяти для входного фильтра,
 b – сдвиг значений входного фильтра,
 σ – функция активации значения входного фильтра.

Фильтр очистки памяти удаляет часть информации запоминающей ячейки, вычисленной на предыдущем шаге:

- где w – веса входа сети для фильтра очистки памяти,
 x_t – вход сети на текущем шаге,
 y_{t-1} – выход сети на предыдущем шаге,
 w – веса выхода сети для фильтра очистки памяти,
 z – состояние памяти сети,
 w – веса состояния памяти для фильтра очистки памяти,
 b – сдвиг значений фильтра очистки памяти,
 σ – функция активации для значения фильтра очистки памяти.

На основе данных входного фильтра и фильтра очистки памяти (1) и (2), поступающих на текущем шаге, вычисляется состояние запоминающей ячейки:

- где z – состояние запоминающей ячейки в текущий момент времени,
 Δz – изменение памяти,
 x_t – значение входного фильтра,
 z – значение фильтра очистки памяти.

Изменение памяти вычисляется по следующей формуле:

- где x_t – вход сети на текущем шаге,
 w – веса входа сети для запоминающей ячейки,
 y_{t-1} – выход сети на предыдущем шаге,
 w – веса выхода сети для запоминающей ячейки,
 z – сдвиг значения запоминающей ячейки,
 σ – функция активации для запоминающей ячейки.

Далее рассчитывается значение выходного фильтра. Выходной фильтр аналогичен двум предыдущим и имеет вид:

- где w – веса входа сети для выходного фильтра,
 x_t – вход сети на текущем шаге,

- выход сети на предыдущем шаге,
- веса выхода сети для выходного фильтра,
- состояние памяти сети,
- веса состояния памяти для выходного фильтра,
- сдвиг значений выходного фильтра,
- функция активации для значения выходного фильтра.

Итоговое значение сети определяется выходным фильтром (5) и нелинейной трансформацией над состоянием запоминающей ячейки (3):

(6)

где – состояние памяти сети, – значение выходного фильтра, – функция активации для выходного значения сети [2].

Повышенная невосприимчивость к продолжительности разрывов во времени является преимуществом LSTM над обычными рекуррентными нейронными сетями, марковскими моделями и другими методами обучения для последовательностей во многих областях применения.

Суть прогнозирования игры заключается в том, что на основе данных нейронной сети, собранных до текущего момента времени, она определит вероятность победы одной из команд.

На вход в нейронную сеть поступают следующие параметры: количество добытых ресурсов каждой из команд; номер команды, вероятность выигрыша которой мы хотим узнать; текущие предметы каждого из персонажей; количество убийств и использованных вспомогательных предметов, разрушенных зданий противника и оставшихся собственных зданий; опыт каждого из персонажей; распределение способностей у персонажей.

Все время одной партии игры t разбито на промежутки Δt . Набор игровых параметров представлен вектором \mathbf{x} в конкретный момент времени игры t . Номер команды, вероятность выигрыша которой мы хотим узнать (может принимать значения 0 или 1) обозначен c . Входным набором параметров в нейронную сеть будет номер команды c и матрица:

(10)

Выходным параметром нейронной сети является вероятность выигрыша команды \hat{c} .

После обучения нейронной сети на большом объеме выборки, ее можно использовать не только для прогнозирования результата игры, но также в качестве помощника. Dota2 – это командная игра и в начале происходит стадия выбора героев. Данную нейронную сеть можно применить для выбора наилучшего героя в зависимости от уже выбранных. Также ее можно будет применить для выбора наилучшего предмета для героя в зависимости от стадии игры.

Нейронные сети будут эффективны при обучении аналитиков, участников соревнований, 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. Рекуррентная нейронная сеть. [Электронный ресурс] – Режим доступа: <http://mechanoid.kiev.ua/neural-net-lstm.html>. – Дата доступа: 14.03.2018.

ПОСТРОЕНИЕ КРИПТОСТОЙКОГО ГЕНЕРАТОРА ПСЕВДОСЛУЧАЙНЫХ ЧИСЕЛ

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

Молчанов И.В.

Калугина М.А. – канд. физ.-мат. наук, доцент

Источники настоящих случайных чисел найти крайне трудно. Ими могут быть, например, физические шумы, такие, как детекторы событий ионизирующей радиации, дробовой шум в резисторе, или космическое излучение. Однако реальное применение таких чисел порождает ряд проблем (время- и трудозатраты при установке и настройке, невозможность воспроизведения ранее сгенерированной последовательности и другие). Поэтому на практике чаще используются генераторы псевдослучайных чисел.

Генератор псевдослучайных чисел (ГПСЧ) — это алгоритм, порождающий последовательность чисел, элементы которой почти независимы друг от друга и подчиняются заданному распределению (обычно равномерному). Во многих задачах современной информатики качество получаемых результатов напрямую зависит от качества используемых ГПСЧ.

Вихрь Мерсенна основывается на свойствах простых чисел Мерсенна и обеспечивает быструю генерацию высококачественных псевдослучайных чисел. Вихрь Мерсенна лишён многих недостатков, присущих другим ГПСЧ, таких как малый период и предсказуемость. Тем не менее, этот генератор не является криптостойким, что ограничивает его использование в криптографии.

Числа Мерсенна в широком смысле — числа вида $2^n - 1$, где n — натуральное число. Иногда числами Мерсенна называют только числа вида $2^p - 1$ с простым показателем p . Вне зависимости от выбранного определения, простое число Мерсенна — число вида $2^n - 1$, являющееся простым (для всех простых чисел вида $2^n - 1$ показатель степени n также является простым числом). С использованием простых чисел Мерсенна связана длина цикла Вихря Мерсенна: $2^{19937} - 1$ (24-е простое число Мерсенна), что позволяет говорить о практической невозможности зациклить данный генератор.

В данной работе алгоритм Вихря Мерсенна реализован в классе `Generator`. Несмотря на то, что в среде .NET существует несколько генераторов псевдослучайных чисел и, возможно, их использование было бы логичным, подобная работа проведена по следующим причинам:

- + чтобы показать отсутствие зависимостей и возможность перевести код на любой язык;
- + для дальнейшей разработки может понадобиться прямое обращение к коду;
- + чтобы убедиться в возможности получить последовательность, имея некоторое количество сгенерированных чисел.

Чтобы показать некриптостойкость Вихря Мерсенна, был также реализован класс `Extractor`, который по 624 последовательно сгенерированным числам получает исходную последовательность. Алгоритм получения исходной последовательности основан на инвертировании процесса «заковки» в Вихре Мерсенна.

Так как цель данной работы состоит в построении криптостойкого генератора псевдослучайных чисел, то рассмотрим несколько вариантов решения:

Алгоритм `Blum-Blum-Shub` (генератор псевдослучайных чисел, основанный на вычислительной сложности задачи факторизации больших чисел);

Алгоритм `Blum-Micali` (генератор псевдослучайных чисел, основанный на вычислительной сложности задачи дискретного логарифмирования);

Композиция любого генератора псевдослучайных чисел и стойкого блочного шифра либо криптографического алгоритма хеширования (сведение задачи о взломе генератора к задаче о взломе стойкого шифра).

Первые два варианта хороши, если речь идет о задачах криптографии, выполняемых на машинах с большой вычислительной мощностью. Оба алгоритма основаны на вычислительной сложности некоторых задач, и, как следствие, имеют низкую скорость работы, что порождает большое количество проблем (например, при генерации ключей — длинных простых чисел). Именно из-за этих возможных проблем был выбран последний вариант построения такого генератора. Учитывая, что мы имеем уже реализованный алгоритм Вихря Мерсенна (класс `Generator`), то остается только реализовать шифрование его регистра сдвига.

Построим на основе Вихря Мерсенна криптостойкий генератор с помощью стандартных средств шифрования `C#`. Например, создадим класс, инкапсулирующий объект класса `Generator` и использующий класс `AesManaged` для шифрования регистра. Заметим, что при реализации такого алгоритма мы не ограничены языком программирования `C#` и шифром AES и сможем без труда перевести код на любой язык с C-подобным синтаксисом, возможно, используя любой другой блочный шифр. В то же время генерация действительно случайных чисел требует источника энтропии, обращение к которому различается в зависимости от языка программирования, что порождает зависимости генератора.

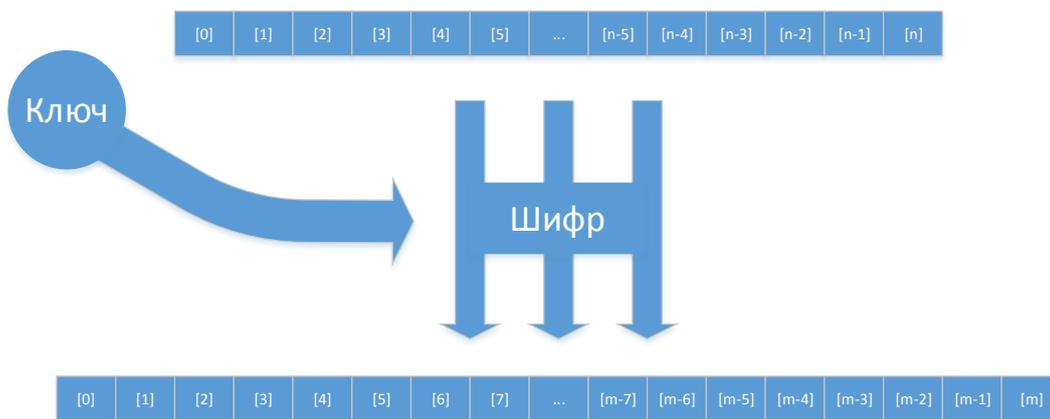
Реализация выбранного решения — это класс `CryptoGenerator`, инкапсулирующий объект `Generator` и объект, реализующий интерфейс `ICryptoTransform`, предназначенный для шифрования регистра сдвига. Для простоты возьмем размер регистра сдвига равным размеру массива `uint[260]`, что представляет собой длину `uint[256] + 32` бит (превышение является следствием наложения шифра AES). Стоит заметить, что длина может не меняться или изменяться по-другому вследствие использования другого блочного шифра, что не влияет на работу генератора.

Чтобы получить данные об эффективности построенного генератора, необходимо сравнить его со схожим стандартным средством. В сборке `System.Security.Cryptography` платформы .NET находится класс `RNGCryptoServiceProvider`, который предоставляет криптостойкие случайные числа. Стоит заметить, что информации о работе этого генератора крайне мало, что препятствует детальному сравнению с разработанным генератором. Сравнение будем строить на сопоставлении скорости работы и распределении значений генераторов.

При сравнении скорости работы генераторов была замечена следующая особенность: при многочисленных обращениях к стандартному генератору `RNGCryptoServiceProvider` (1 число — 1 обращение) скорость работы была значительно меньшей, чем при более редких обращениях с заполнением большего по объему буфера. Таким образом, для дальнейших исследований можно поставить дополнительную проблему — исследовать зависимость скорости работы `RNGCryptoServiceProvider` от внешних факторов, а также исследовать возможный принцип его работы. В общем, можно сказать, что скорость работы обоих генераторов удовлетворяет требованиям задач, в которых они используются (генерация 10^6 чисел происходит за 10 - 500мс).

Что касается исследования распределения, стоит опираться на следующее утверждение: значения, полученные с помощью Вихря Мерсенна без доработок, равномерно распределены. Таким образом, остается проверить, повлияет ли использование шифрования на распределение конечного генератора. Чтобы сравнить распределения генераторов, построим таблицу на основе теоретических расчетов для равномерного распределения и данных, полученных нашей программой (в качестве параметров распределения возьмем математическое ожидание и дисперсию).

Выходная последовательность Вихря Мерсенна



Полученная последовательность криптогенератора

Отрезок	Теоретические		CryptoGenerator		RNGCryptoServiceProvider	
	Матожидание	Дисперсия	Матожидание	Дисперсия	Матожидание	Дисперсия
0..99	49	816	49	833	49	833
0..999	499	83166	499	83326	499	83303
0..9999	4999	8331666	4999	8330739	4999	8333683
0..99999	49999	833316666	49996	833291139	50007	833501899
0..999999	499999	83333166666	499997	83328154282	499961	83330635124

Таким образом, распределение обоих генераторов можно считать равномерным (на основании схожести данных, полученных опытным путем, и теоретических значений).

Кроме криптостойкости и равномерного распределения генерируемых значений, написанный нами генератор имеет и некоторые другие преимущества:

- + функциональность для генерации стандартных типов данных;
- + возможность использовать различные шифры;
- + высокая скорость работы;
- + открытый код и возможность доработки.

Планируется исследовать возможности оптимизации построенного генератора, а также исследовать проблемы, поставленные в ходе данной работы: зависимость характеристик работы генератора от вида шифрования, скорость обращения к объекту RNGCryptoServiceProvider, реализовать возможность генерации различных типов данных. Также планируется построить генераторы с распределением, отличным от равномерного (моделировать заданное распределение различными методами).

Список использованных источников:

1. <https://habrahabr.ru/post/196442/>
2. <https://habrahabr.ru/company/mailru/blog/274253/>
3. https://ru.wikipedia.org/wiki/Числа_Мерсенна
4. https://ru.wikipedia.org/wiki/Вихрь_Мерсенна
5. https://ru.wikipedia.org/wiki/Генератор_псевдослучайных_чисел

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ ДИСТАНЦИОННОГО УПРАВЛЕНИЯ ПЕРСОНАЛЬНЫМ КОМПЬЮТЕРОМ

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

Неведомский Д.А.

Алексеев Ю.И., м.т.н.

Системы дистанционного управления используются человеком повсеместно, начиная от домашней бытовой техники и заканчивая сложными системами, такими как беспилотные самолёты. Решений же для дистанционного управления персональным компьютером существует немного, и все они имеют ограниченный функционал.

Данное мобильное приложение для платформы Android позволяет управлять некоторыми возможностями персонального компьютера. В такой функционал входит:

- использование мобильного устройства в качестве манипулятора ввода персонального компьютера;
- перемещение курсора;

- осуществление нажатий клавиш мыши;
- набор текста с помощью виртуальной клавиатуры;
- управление медиа возможностями персонального компьютера:
- воспроизведение;
- остановка;
- переключение композиций;
- изменение уровня громкости;
- переключение слайдов в офисном программном обеспечении;
- приглушение звука при поступлении входящего или начале исходящего звонка на мобильном устройстве;
- получение информации о поступивших уведомлениях на мобильное устройство и ответ на них, если он подразумевается;
- информация об уровне заряда мобильного устройства;
- приём и передача файлов с персонального компьютера на мобильное устройство и наоборот;

Исходя из вышесказанного, стоит отличать данное программное обеспечение дистанционного управления от систем удалённого доступа, таких как программа TeamViewer или протокол передачи VNC. Данное мобильное приложение подразумевает тесную интеграцию полноразмерных и мобильных устройств в единый комплекс, нежели удалённый доступ с дублированием экрана.

Данное мобильное приложение поставляется в паре с серверным приложением для персонального компьютера для обеспечения работы протокола клиент-сервер между устройствами. Серверная часть разрабатывается с использованием языка программирования C# на платформе .NETCore, предоставляемой и поддерживаемой Microsoft. Эта современная платформа позволяет в будущем переиспользовать большую часть кода для переноса серверного кода на GNU/Linux и, возможно, для более закрытой с точки зрения разработчика программного обеспечения macOS.

На операционной системе Microsoft Windows для управления возможностями персонального компьютера используется функционал, предоставляемый подсистемой Win32 (WINAPI), состоящей из неуправляемого кода.

Клиентская часть реализована для платформы Android, самой популярной мобильной платформы в мире, с использованием языка программирования Kotlin и языка разметки XML. Обмен данными между устройствами происходит с помощью локальной сети по протоколу UDP. С UDP компьютерные приложения могут посылать сообщения (в данном случае называемые датаграммами) другим хостам по IP-сети без необходимости предварительного сообщения для установки специальных каналов передачи или путей данных, что позволяет ускорить обработку поступающей информации. В дальнейшем существует возможность использовать помимо локальной сети прямое подключение при помощи протоколов Wi-FiDirect и Bluetooth, что позволит использовать данную систему автономно. Протокол взаимодействия между клиентом и сервером будет стандартизирован, документирован и открыт для доработки, что позволит для каждой индивидуальной платформы реализовать свой независимый функционал.

Уникальность системы заключается в возможности использовать её во многих направлениях. Например, превратить в любой персональный компьютер в локальной сети в медиацентр, подключив его, например, к большому экрану. Или же использовать экран мобильного устройства в качестве графического планшета.

На данный момент реализована функция манипуляции курсором и нажатия, но изучена возможность реализации всего остального функционала на платформах Microsoft Windows и GNU/Linux и будет производиться доработка и дальнейшее развитие как самого протокола связи, так и их реализаций для конкретных платформ.

Таким образом, данный программный комплекс поможет более тесно интегрировать устройства между собой в единую систему для более удобной и эффективной работы с каждым из них.

СЕРВИС ДЛЯ КОНТРОЛЯ ВЫПОЛНЕНИЯ ЗАДАЧ

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

Неволько Н.Д.

Жвакина А.В. – к.т.н., доцент

В современном мире информационных технологий, при работе над новым проектом с использованием гибких методологий разработки программного обеспечения, нередко встает вопрос: как организовать эффективное разделение и отслеживание выполнения задач между различными группами специалистов (программистами, тестировщиками, бизнес-аналитиками и др.)? Разработанный сервис для контроля выполнения задач предназначен для решения данной задачи.

При использовании Agile методологий, как например Scrum или Kanban, задачи делятся на подзадачи и выносятся на зонированную общую доску. Но как быть, если члены команды работают удаленно, да еще и в разных часовых поясах? А как управлять проектом и задачами, если команда состоит из более чем ста

человек? Здесь на помощь придет решение в виде веб-приложения по управлению задачами проекта.



Рис. 1 – Пример Agile доски с задачами

Применение данного приложения поможет руководителям организовать работу в командах, создавать свой рабочий процесс для каждого вида задач, а интуитивно понятный интерфейс позволяет представить статистику и текущие статусы выполнения задач в удобном виде. Система обеспечивает доступ в любое время и в любом месте. Благодаря выбранному современному стеку технологий данные пользователей надежно защищены.

В основе принципа работы лежат задачи (Task, Ticket) – атомарные единицы работы над проектом, которые по мере своего выполнения изменяют свой текущий статус, двигаясь по рабочему процессу. Основной функционал включает в себя:

- создание задач и подзадач;
- организацию своего рабочего процесса для каждого вида задач;
- возможность учета запланированного и фактически затраченного времени;
- комментирование задач с целью лучшей коммуникации между исполнителями задач.

Достоинствами предоставляемого сервиса являются:

– отказ от представления задач и подзадач на старомодных «физических» досках, перенос данных на сервер, доступный круглосуточно;

- удобный и понятный интерфейс;
- возможность использования любыми рабочими группами в любых проектах;
- учет затраченного времени над задачами каждого сотрудника;
- наглядность текущего статуса исполнения всех задач на общей доске задач;
- безопасность данных пользователей;
- использование современных технологий в разработке.

Итогом работы является полноценное веб-приложение, которое поможет оптимизировать контроль выполнения задач, улучшить коммуникацию между сотрудниками, а значит, и увеличить скорость и качество создания программных продуктов.

Список использованных источников:

1. Гибкая методология разработки. [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Гибкая_методология_разработки. – Дата доступа : 08.04.2018.

БАЗИСЫ ГРЁБНЕРА. АЛГОРИТМ БУХБЕРГЕРА

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

Новак Д.С.

Калугина М.А.–канд. физ.-мат. наук, доцент

В течение всей истории математики стоял вопрос о решении систем алгебраических уравнений с большими целочисленными коэффициентами. Австрийский математик Вольфганг Грёбнер еще в 1930-х годах разрабатывал теорию стандартных базисов, которая позволяла разрешать различные САУ, однако его работа не нашла практического применения из-за больших практических подсчетов, которые занимали много времени, а иногда были вовсе неразрешимы.

В начале 60-х годов прошлого века был достигнут прогресс в области систем алгебраических уравнений. Австрийский математик Бруно Бухбергер, ученик Грёбнера ввел такие понятия как базис Грёбнера, а также разработал эффективный алгоритм, позволяющий находить базис Грёбнера для систем алгебраических уравнений с большими коэффициентами, и, как следствие, находить корни этих уравнений.

Пусть дана нелинейная САУ. Для решения системы нам необходимо перейти к эквивалентной системе. Введем понятие идеал системы:

Пусть \mathbb{C} - множество всех многочленов от переменных с комплексными коэффициентами:

Определение: Идеал, порожденный данным набором многочленов, - это множество всех комбинаций вида $\sum w_i f_i$ где w_i - произвольные многочлены. Идеал I .

Также важно понимать если $I = J$ (идеалы двух систем совпадают, тогда системы многочленов от :

эквивалентны. Т.е. имеют одно и то же множество решений (обратное утверждение неверно).

Вывод: Множество решений системы однозначно определяется идеалом системы. Различные базисы одного идеала отвечают эквивалентным системам, следовательно, каждая САУ эквивалентна конечной системе.

Для перехода к другой эквивалентной системе нам необходимы лишь идеалы САУ, для этого нам нужно выбрать другой базис, например, базис Гребнера.

Определение: пусть $B = \{f_1, \dots, f_n\}$ - набор многочленов из \mathbb{C} . Тогда они называются базисом Гребнера, если для любого многочлена f его старший член делится на старший член хотя бы одного из многочленов f_i .

Определение: Многочлены f_1, \dots, f_n имеют зацепление, если их старшие члены делятся одновременно на некоторый многочлен w , отличный от констант.

Определение: Полином f редуцирован относительно g , если старший моном полинома f не делится на старшие члены полиномов из g .

Лексикографический метод упорядочивания многочленов от нескольких переменных: известно, что введение в исходном алфавите полного упорядочивания приводит к полному упорядочиванию составленных из него слов. В каждом выражении с коммутативной операцией производится упорядочивание составляющих его слов в соответствии с указанным отношением, и в качестве представителя класса эквивалентных выражений выбирается старшее из них.

В системе Maple существует пакет Groebner с командой Groebner[Basis](), которая позволяет быстро находить базис Грёбнера для заданной системы.

```
> G1 := Groebner[Basis]( B1, plex(x,y,z) );
      G1 := [z^4 - 3z^2 + 3, y^2 + z^2 - 3, -yz^3 + 3x]
> G2 := Groebner[Basis]( B2, plex(x,y,z) );
      G2 := [1]
```

Также мы можем поэтапно найти базис Грёбнера в системе Maple с помощью реализации алгоритма Бухбергера.

Для этого нам понадобятся команды:

LeadingMonomial() - которая позволяет найти моном высшей степени из данного многочлена с помощью лексикографического упорядочивания.

SPolynomial() - позволяет находить зацепление (S(f_i, f_j)).

NormalForm() - редуцирование.

Рассмотрим пример: пусть дана САУ

$$\begin{aligned} f_1 &:= a \cdot b - c^2 - c; & f_1 &:= a b - c^2 - c \\ f_2 &:= a^2 - a - b \cdot c; & f_2 &:= a^2 - b c - a \\ f_3 &:= (a \cdot c) - b^2 - b; & f_3 &:= a c - b^2 - b \end{aligned}$$

Найдем старшие члены с помощью лексикографического упорядочивания командой LeadingMonomial()

```
> LeadingMonomial(f1, plex(a, b, c));
      a b
> LeadingMonomial(f2, plex(a, b, c));
      a^2
> LeadingMonomial(f3, plex(a, b, c));
      a c
```

Найдем зацепление с помощью команды SPolynomial().

```
> f4 := SPolynomial(f1, f2, plex(a, b, c));
      f4 := -a^2 + b^2 c + a b - a c
```

Редуцируем относительно

```
> f4 := NormalForm(f4, G0, plex(a, b, c));
      f4 := -b^2 - b c + c^2 - b + c
```

Если остаток не равен 0, добавляем =

```
> G1 := [f1, f2, f3, f4];
```

Аналогично предыдущим действиям находим - зацепление, - зацепление, редуцируем, добавляем к, если остаток не равен нулю. В итоге получаем базис Грёбнера

$$\begin{aligned} &> GI; \\ &[ab - c^2 - c, a^2 - bc - a, ac - b^2 - b, -b^2 \\ &\quad - bc + c^2 - b + c, 2bc, -2c^2 - 2c^2] \end{aligned}$$

Теорема (о количестве конечных решений):

Число решений системы конечно тогда и только тогда, когда базис Гребнера идеала содержит старшие члены которых являются степенями переменных соответственно.

За конечное число шагов мы получим набор \dots , где каждое зацепление разрешимо. Это и есть базис Гребнера идеала

Тогда базис Гребнера для данной САУ: Найдем решения из \dots . При \dots . Наборы (1,0,-1) – не подходят, (0,0,1) – подходят. При \dots . Наборы (0,0,0), (0,-1,0), (1,0,0)- подходят. (1,-1,0) - не подходят.

Ответ: {(0, 0, 0); (1, 0, 0); (0, -1, 0); (0, 0, -1)}.

Список использованных источников:

1. И.В.Аржанцев Базисы Гребнера и системы алгебраических уравнений
2. <https://habrahabr.ru/post/177237/>
3. http://halgebra.math.msu.su/wiki/lib/exe/fetch.php/specialcourses:ind_popovsky.pdf

ПРОГРАММА СБОРА ДАННЫХ О СТРУКТУРЕ ВЕБ-САЙТОВ

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

Потехин А.С.

Стержанов М.В. – к.т.н. доцент

В настоящее время всеобщие глобальные тенденции приближаются к тому, что все операции и торговые сделки будут проходить с использованием веб-ресурсов. Для того, чтобы успешно вести бизнес очень важно получать актуальные данные о движении рынка (динамика цен и товаров) и локальные новости, которые порой всецело влияют на формирование спроса, своевременно. Но необходимые данные не всегда легко доступны пользователю и чаще всего они неструктурированы. Рассматривается приложение, которое будет обладать необходимым функционалом для сбора и структурирования данных с различных веб-ресурсов.

Целью исследования, для которого необходим сбор данных из Сети Интернет, является сентимент-анализ данных с различных новостных сайтов. Данные должны содержать полную информацию о новости, включая заголовок, текст, дату и автора новости. Для того, чтобы обеспечить сбор указанной информации, необходимо реализовать инструмент - web-scraпер.

В широком понимании web scraping — это сбор данных с различных интернет-ресурсов. Общий принцип его работы можно объяснить следующим образом: автоматизированный код выполняет запросы на целевой сайт и получая ответ, парсит HTML-документ, ищет данные и преобразует их в заданный формат. Т.е. инструменты веб-скрапинга позволяют вручную или автоматически извлекать новые или обновленные данные и сохранять их для последующего использования.

Для того чтобы выполнять эту задачу, инструмент должен поддерживать работу со следующими данными:

HTML, JavaScript, так как большинство сайтов построены с использованием этих технологий;
 Plaintext, PDF и другие форматы представления текстовых данных;
 URLs, с возможностью построения на их основе графа веб-ресурсов.
 Также инструмент должен обладать требованиями [1],[2],[3]:

- Надежность – Веб содержит ресурсы, которые могут вводить скрапер в бесконечный цикл или недоступные сервисы, ожидать выполнения которых он не должен. Скрапер должен быть устойчивым к таким ловушкам;
- Вежливость – интернет-ресурсы имеют явные и неявные политики, регулирующие частоту, с которой скрапер может посетить их. Они описаны в файле robots.txt и эти политики должны соблюдаться;
- Распределенность – скрапер должен иметь возможность выполняться в распределенном режиме на нескольких машинах;
- Масштабируемость – скрапер должен поддерживать возможность увеличения производительности за счет добавления дополнительных вычислительных узлов, на которых он исполняется;
- Производительность и эффективность – скрапер должен обеспечивать эффективное использование системных ресурсов, включая процессор, память и полосу пропускания сети;
- Качество – скрапер должен уметь отделять спам-страницы от полезных и извлекать последние;
- Актуальность – скрапер должен поддерживать обновление собранных данных;
- Расширяемость – скрапер должен быть модульным, т.е. позволять добавлять новую функциональность, для анализа новых форматов данных, протоколов и т.д.

Помимо описанных общих требований для скраперов, можно обозначить основные требования, для конкретной задачи исследования:

Скрапер должен быть кроссплатформенным, чтобы его можно было одинаково настраивать и конфигурировать на вычислительных узлах с разными операционными системами;

Скрапер должен обеспечивать производительность обработки порядка 100 стр/сек, чтобы время сбора описанного выше объема данных составляло часы, а не дни. В том случае если окажется, что данных для сбора и анализа больше предполагаемого, скрапер должен предоставлять возможность легко увеличить его производительность путем выделения ему для работы большего числа потоков или добавления дополнительных вычислительных узлов;

Скрапер должен быть интегрирован с базой данных для хранения собранной информации и полнотекстовым индексом, позволяющим быстро извлекать данные для последующего анализа, отвечающие указанным условиям;

Требуется скрапер для сбора данных в ширину и вертикального поиска, так как в указанной задаче необходимо извлечь информацию о конкретной предметной области, а не узкое множество фактов;

В настоящее время существует множество готовых решений веб-скраперов, но готового решения для данной задачи исследования нет, поэтому, для реализации поставленной задачи был разработан собственный веб-скрапер.

Созданный скрапер является эффективным инструментом для поиска в Вебе, ядро написано на C++ с которым взаимодействует Ruby-оболочка., поддерживает граф связей узлов, различные парсеры, фильтры и нормализаторы URL. Он позволяет использовать различные хранилища данных, такие как Cassandra, Hbase и др. Скрапер также является масштабируемым (до 100 узлов в кластере и легко настраивается и расширяется, в полной мере является “вежливым”).

Список использованных источников:

1. PAPAVALASSIOLIOU V., PROKOPIDIS P., THURMAIR G. A modular open-source focused crawler for mining monolingual and bilingual corpora from the web // Proceedings of the 6th Workshop on Building and Using Comparable Corpora. — 2013.
2. AHUJA M.S., BAL J.S., VARNICA Web Crawler: Extracting the Web Data // International Journal of Computer Trends and Technology. — 2014.
3. YADAV M., GOYAL N. Comparison of Open Source Crawlers-A Review// International Journal of Scientific & Engineering Research. — 2015.

КРИПТОГРАФИЧЕСКИЕ АЛГОРИТМЫ ШИФРОВ ЗАМЕНЫ

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

Сидоренко К.А., Приловский Е.В., Усенко Д.В.

Стройникова Е. Д. – ассистент кафедры информатики

Шифры замены являются наиболее часто используемыми шифрами на сегодняшний день. Они характеризуются тем, что отдельные части сообщения (слова, буквы) заменяются на другие буквы, числа, символы и т. д. Но, при этом замена осуществляется так, чтобы через зашифрованное сообщение можно было восстановить передаваемое сообщение. Несмотря на вытеснение шифров подстановки блочными шифрами одноразовые блокноты ещё остаются применимыми на государственном уровне. Они используются для обеспечения сверхсекретных каналов связи. Так, по некоторым данным, телефонная линия между главами США и СССР шифровалась при помощи одноразового блокнота и вполне возможно, что подобные линии существуют до сих пор. Одноразовые блокноты применяются шпионами в различных государствах для сокрытия важной информации. Такие сообщения невозможно расшифровать, если отсутствует ключ, записанный в блокноте, независимо от вычислительной мощности ЭВМ.

Шифр Цезаря

Шифр Цезаря является шифром подстановки, который работает следующим образом: все символы циклически заменяются символами, которые расположены на определенном числе позиций в любом направлении от них в алфавите. Рассмотрим следующий пример: в шифре со сдвигом вправо на 3 происходит замена А на D, В на Е, ..., Z на С и т. д. (рис.1). Способ шифрования с помощью шифра Цезаря есть составляющая часть более сложных шифров, например такого, как шифр Виженера. Но, т. к. шифр Цезаря является моноалфавитным, его легко разгадать и он не практичен в использовании.

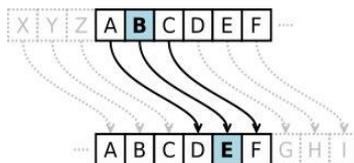


Рис. 1 – Шифр Цезаря с величиной циклического сдвига 3

Шифр Атбаш

Атбаш – простой шифр подстановки для алфавитного письма. Правило шифрования состоит в замене i -й буквы алфавита буквой с номером $n - i + 1$, где n – число букв в алфавите. Ниже, на рис. 2, дан пример для английского алфавита:

Исходная буква	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
Зашифрованная буква	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A

Рис. 2 – Ключевая таблица шифра Атбаш для английского алфавита

Шифр Виженера

Для реализации шифра воспользуемся следующей таблицей для английского алфавита (рис. 3):

	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
A	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
B	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	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	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

Рис. 3 – Таблица Виженера для английского алфавита

Как видим, таблица Виженера составляется из строк по 26 символов, причём каждая строка циклически сдвинута относительно строки, находящейся над ней, на одну букву влево. Таким образом, в таблице получается 26 различных шифров Цезаря. На каждом этапе шифрования используются разные алфавиты, их выбирают в зависимости от символа ключевого слова. Например, пусть дан исходный текст «ATTACKATDAWN». Человек, посылающий сообщение, записывает ключевое слово «LEMON» циклически до того момента, пока не достигнет конца исходного текста. В нашем случае получится последовательность «LEMONLEMONLE». Первый символ исходного текста «А» зашифрован с помощью символа «L», который является первым символом ключа. Первый символ «L» зашифрованного текста находится на пересечении строки, соответствующей «L», и столбца, соответствующего «А», в квадрате Виженера. Аналогично для второго символа исходного текста используется второй символ ключа; т. е. второй символ зашифрованного текста «Х» получается на пересечении строки, соответствующей «Е», и столбца, соответствующего «Т». Аналогично шифруется остальная часть исходного текста. Таким образом, имеем:

ATTACKATDAWN – исходный текст;

LEMONLEMONLE – ключевая последовательность;

LXFOPVEFRNHR – зашифрованный текст.

На основе алгоритма шифра Виженера разработана учебная программа на языке программирования C#. Сначала требуется ввести текст, который подлежит шифрованию. Затем высчитываем символ шифра по формуле $c[i] = (\text{alphabet}[i] + \text{keyword}[i]) \% N$, где $i = \{0..n\}$, n – количество символов, $c[i]$ – символ в строке, $\text{alphabet}[i]$ – символ в алфавите, $\text{keyword}[i]$ – ключ. Получаем зашифрованный текст и выводим его на экран. Расшифровка производится по аналогичному принципу.

Шифровальные устройства

1. Энигма – переносная шифровальная машина, которая применялась для шифрования и дешифрования. Шифрование производилось за счет комбинаций нажатия клавиш.

2. Шифровальное устройство М-94 – принцип работы основывался на механизме крутящихся дисков, на которых были написаны буквы и цифры.

3. Шифровальное колесо Болтона – работало по принципу простой замены одной буквы на другую.

4. Шифровальная машина Конвертер М-209 – зашифрованное сообщение распечатывалось на бумаге в виде пятизначных групп.

5. Шифровальная машина Лоренц – принцип работы был основан на поточном шифре Вернама.

Подготовлен доклад с целью ознакомления студентов с основами криптографии. Данная информация может быть полезна студентам, обучающимся по специальности “Защита информации”, а также тем, кто умеет или учится профессионально программировать, интересуется сжатием данных или занимается исследованием современных средств шифрования. В наши дни криптография используется практически во всех сферах, работающих с приёмом и передачей информации, обеспечивает работу сверхсекретных каналов связи, а также эта наука успешно применяется в банковской деятельности.

Список использованных источников:

1. Василенко, О. Н. Теоретико-числовые алгоритмы в криптографии / О. Н. Василенко. – М. : МЦНМО, 2003. – 326 с.
2. Введение в криптографию / В. В. Яценко [и др.] ; под общ. ред. В. В. Яценко. – 3-е изд., перераб. – М. : МЦНМО, 2003. – 400 с.
3. Математические и компьютерные основы криптологии : учеб. пособие / Ю. С. Харин [и др.]. – Минск : Новое знание, 2003. – 382 с.
4. Стройникова, Е. Д. Основы прикладной алгебры : учеб.-метод. пособие / Е. Д. Стройникова. – Минск : БГУИР, 2010. – 120 с.
5. Шифровальные устройства [Электронный ресурс]. – Режим доступа: <http://kryptography.narod.ru/mashiny.html>.

ДИАГНОСТИКА НА РАК ПРИ ПОМОЩИ НЕЙРОННЫХ СЕТЕЙ

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

Пунько В.В., Приходько В.С.

Волорова Н. А. – к.т.н., доцент

Популяция людей на планете с каждым днем увеличивается, как и увеличивается число болезней, не подлежащих лечению и очень трудных в диагностике. Одним из таких заболеваний является рак. Около 13 % всех смертей в мире происходит из-за онкологических заболеваний, которые сегодня считаются самой распространенной патологией после инсульта и ишемии.

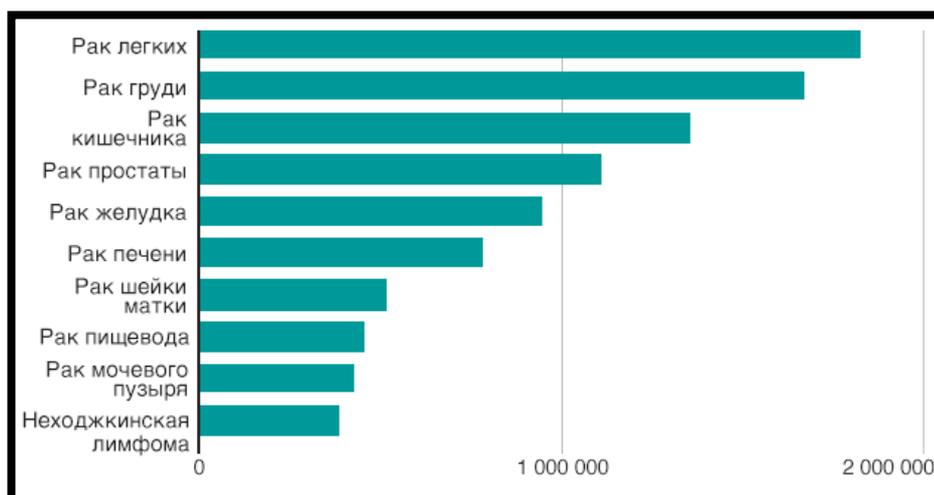


Рис. 1 – Статистика раковых заболеваний на 2012 год

Дело в том, что онкологических заболеваний (того самого рака) — огромное количество разновидностей. Например, раков молочной железы существует более 20 видов, и, кроме этого, у каждого вида рака молочной железы есть характеристики, влияющие на стратегию лечения. Заболеваний, которые называют лимфомы — тысячи видов и подвидов. И, опять же, существует принципиальная разница в их лечении. От правильности постановки диагноза в итоге зависит успешность или не успешность лечения. Пациента можно лечить сколь угодно хорошо, но если его лечат от другого вида рака — лечение не имеет существенного эффекта.

Про онкологическую диагностику много говорят, и, несмотря на это, очень мало знают в среде непрофессионалов. Во-первых, скрининг (ранняя диагностика) и диагностика — это совершенно разные вещи. Во-вторых, от этапа, когда пациент впервые попадает к онкологу, и онколог подозревает у пациента профильное заболевание, должно пройти несколько принципиально важных этапов для того, чтобы начать лечение. Все эти этапы и называются онкологической диагностикой.

Человек не прозрачен, и даже если обнаружено новообразование, говорить о том, что это рак или не рак, и, тем более, какой это рак, в большинстве случаев очень преждевременно. Роль диагностики в онкологии очень высока. Именно поэтому, после того как есть основания предполагать онкологическое заболевание, пациента направляют на такой этап диагностики, как морфологический.

Он состоит из хирургической процедуры забора материала (биопсии [5]) и, собственно, самого морфологического исследования (гистологический/иммуногистохимический/молекулярный анализ). Именно на основании этих анализов человеку или подтверждают, или корректируют поставленный клинический диагноз. Именно на основании морфологического (патоморфологического) заключения пациента будут потом лечить. И именно от точности этого этапа онкологической диагностики будет зависеть насколько подходит лечение данному пациенту с его заболеванием, насколько оно будет эффективно.

Но человек может ошибиться, и тогда рак может быть не выявлен. Анализ данных биопсии можно передать компьютеру, т.к. машина точнее человека, и многие задачи может решать быстрее и лучше. Цель этого исследования – создание нейронной сети для распознавания раковых заболеваний на ранней стадии по анализу биопсии.

Изначально, перед работой с изображением, его нужно подготовить, в этом случае делается его препроцессинг. Сперва мы применим к изображению фильтры для уменьшения шумов и сглаживания. Далее мы представляем каждый пиксель полученного изображения как точку в трёхмерном пространстве (первые 2 координаты – координаты пикселя в изображении, а третья – его цвет в RGB) (Рис. 2).

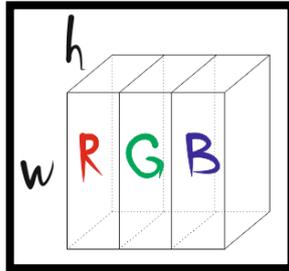


Рис. 2 – Представление картинки в качестве трехмерного куба

Теперь это представление надо нормализовать. Для каждого цветового спектра делим значение цвета в этом пикселе на среднее арифметическое значений всех пикселей в этом цвете. Для более мощной нормализации можно найти среднее значение не на данном изображении, а на всей выборке изображений. Пример нормализации изображения 2 на 2 пикселя:

Таким образом у нас все значения сместятся ближе к нулю и с ними будет проще подсчитывать веса для нейронной сети (Рис. 3).

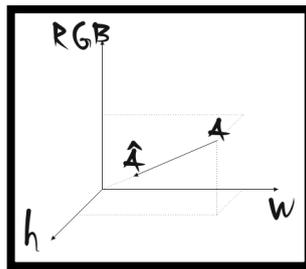


Рис. 3 – Смещение пикселей ближе к нулю в трехмерном пространстве

На изображении биопсии присутствуют не только клетки, но ещё и достаточно большое количество соединительной ткани, которая нам не нужна, и соответственно было бы неплохо её убрать. Для этого мы после препроцессинга сегментируем изображение оставляя только клетки. Сегментирование производится в данной работе при помощи нейронной сети архитектуры UNet [1]. Сети данной архитектуры могут быстро и качественно сегментировать изображения. Недостатком этой сети является долгое обучение. Принцип работы довольно простой: изображение, поданное на вход, проходит несколько слоев свертки после чего результат разворачивается, учитывая результаты предыдущих слоев свертки. Таким образом на выходе у нас получается изображение с наложенной маской, в данном случае – клетка без соединительной ткани.

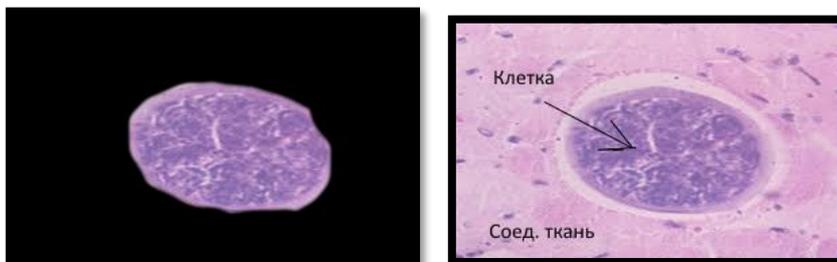


Рис. 4 – Не сегментированная клетка и клетка после сегментации соответственно

Далее мы формируем объект, состоящий из изображения биопсии и тега (кодового названия органа, откуда она была взята). После этого механизм принятия решений по тегу выбирает на какой из ансамблей нейронных сетей передать данную биопсию для анализа. Изображение из этого объекта подаётся на

ансамбль из пяти нейронных сверточных сетей [4], которые учились вне зависимости друг от друга. Т.к. они обучались независимо результаты у них могут быть разные. Из полученных результатов считается средний и это среднее значение и есть вероятность нахождения рака на данном органе. Полная схема архитектуры программы приставлена на рисунке 5.

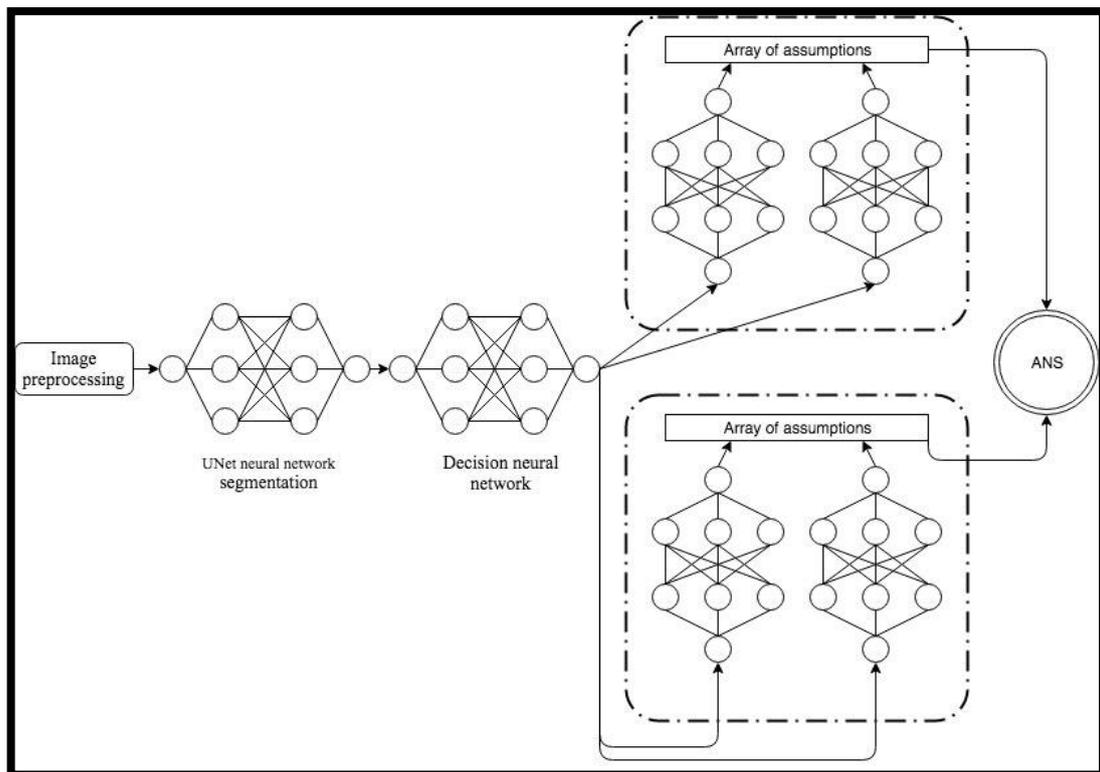


Рис. 5 – Схема взаимодействия между нейронными сетями для анализа картинки

Так как данных, по которым можно обучать нейронную сеть у нас не много (в открытом доступе достаточно мало изображений биопсий пациентов больных раком) сделаем небольшие модернизации всех сетей. Воспользуемся сеткой под названием автокодирование [2], предназначенной для выявления важной информации на изображении. Дело в том, что после каждого пропуска через данную сеть одного и того же изображения мы всегда будем получать нужную информацию с некоторыми отклонениями (например, будут появляться искажения на фото). Тем самым мы усилим тренировочную выборку и обучим сеть более точно. Принцип автокодировщика прост – это однослойный персептрон с одинаковым кол-вом входных нейронов и классов, а единственный скрытый слой имеет размер в 2 раза больше, чем входной. Вставляя такую сетку через каждый слой в нашей сверточной сети, и внося прием Dropout [3] (выбивание нейронов на каждой эпохе для увеличения вариативности сети) мы можем уменьшить тренируемый сет. Для еще большего улучшения используем написанную мной библиотеку `neural_fitbox`, которая позволяет во время обучения подменять и изменять данные постоянно перемешивая их перед подачей. Таким образом мы имеем модель, в которой каждый слой имеет автокодирование, и мощного учителя, который может во время тренировки менять данные. Имея такую архитектуру, мы можем учить наши сети имея очень малое кол-во данных. Вплоть до 1000 изображений на тренировку и проверку.

После того, как ансамбль сетей делает вывод, составляем отчет и сохраняем. Если все 5 сверток дают не ясные результаты, то мы пробуем подать изображение еще раз. После чего можно сделать вывод. Стоит учитывать, что это – вероятность, полученная на маленьких данных, и для использования в промышленных масштабах потребуется переобучить сеть на больших данных. Это можно сделать достаточно просто, учитывая, что сеть принятия решения уже обучена, а сегментация никак не влияет на результат. Для переобучения на тот или иной рак, нужно переобучить ансамбль нейронных сетей, после чего включить блок в схему и продолжить работу. Данный подход может легко распознавать раковые заболевания точнее и быстрее человека.

Список использованных источников:

1. UNetpaper - <https://arxiv.org/abs/1505.04597>
2. Autodecoder - <https://tensorchiefs.github.io/bbs/files/vae.pdf>
3. Dropout - <http://jmlr.org/papers/volume15/srivastava14a.old/srivastava14a.pdf>
4. CNN - http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture5.pdf
5. Biopsy - <https://www.euroonco.ru/glossary-a-z/biopsy>

ПРИМЕНЕНИЕ РЕЖИМА Пониженного Энергопотребления ДОЗУ В Идентификации Цифровых Систем

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

Пучков А.В.

Иванюк А. А. – д-р. техн. наук, доцент

Неотъемлемой частью любой современной цифровой системы является оперативное запоминающее устройство. Таковым нередко выступает динамическое запоминающее устройство (ДОЗУ), которое может использоваться не только по основному назначению, но и, в дополнение к этому, в качестве криптографического примитива для решения задачи уникальной неклонированной идентификации. Классическим подходом является анализ состояния ДОЗУ после включения, но не меньший интерес представляет состояние ДОЗУ после выхода из режима пониженного энергопотребления.

Структурная сложность физических, в частности, электронных систем рассматривается в физической криптографии как основа для построения криптографических примитивов. Основным понятием физической криптографии является физически неклонированная функция (ФНФ), которую можно понимать как устройство, генерирующее значения ответов на некоторые входные воздействия (запросы), причем, пары запрос-ответ обладают уникальностью, непредсказуемостью и неклонированностью на других экземплярах интегральных схем, выпущенных в рамках заданного технологического процесса. Процесс производства цифровых устройств и систем предполагает, что значениями отдельных их параметров принципиально невозможно управлять, задавая для них конкретные значения. Такие параметры принимают случайные, уникальные для конкретного экземпляра цифровой системы, значения, а задачей ФНФ является их извлечение и усиление. Классическим примером такого рода параметров являются задержки распространения сигналов [1]. Уникальность пар запрос-ответ означает для ФНФ возможность использовать их в рамках решения задачи неклонированной идентификации цифровых систем, а также более сложных протоколах аутентификации на их основе. С другой стороны, случайный характер рассмотренных выше параметров позволяет использовать ФНФ для построения генераторов истинно случайных последовательностей, при этом такие генераторы сами будут являться неклонированными.

Для оперативных запоминающих устройств, как статических, так и динамических, характерно то, что при включении напряжения питания некоторая часть запоминающих ячеек оказывается в состоянии 1, оставшаяся часть – в состоянии 0. В самом общем случае данный процесс является случайным, а полученное распределение – уникальным со статистической точки зрения. Это даёт возможность говорить об ОЗУ как о ФНФ с ответом, представляющем собой весь массив запоминающих элементов, запросом же выступает включение питания. Поскольку чаще всего в процессе работы цифровой системы ОЗУ непрерывно используется, такой подход сопряжен со сложностями в практической реализации, т.к. использование такого криптографического примитива возможно только непосредственно после включения питающего напряжения ОЗУ, что практически всегда совпадает с включением всей цифровой системы [2]. В этой связи можно выделить ДОЗУ, для которых представляется возможным альтернативный подход – отключение регенерации части запоминающих ячеек. В отличие от предыдущего метода, это может выполняться многократно во время работы цифровой системы. В частном случае, отключение регенерации массива запоминающих элементов выполняется при переходе в режим пониженного энергопотребления, что особенно актуально для мобильных устройств. Несмотря на некоторые недостатки, из-за низкой стоимости ДОЗУ получают широкое распространение в цифровых системах различного назначения.

Для экспериментального исследования были использованы ДОЗУ Micron M45W8MW16, которыми комплектовались имеющиеся в наличии 10 плат быстрого прототипирования Digilent Nexys 4 на основе FPGA Xilinx Artix-7. Данные схемы памяти имеют объем 16 Мбайт и обладают интерфейсом, практически идентичным статическим запоминающим устройствам. При этом регенерация запоминающих ячеек осуществляется автоматически средствами специальной логики, которой можно управлять посредством набора регистров. Был разработан аппаратно-программный комплекс, который позволяет производить операции записи и чтения из ОЗУ, устанавливать различные режимы его работы, в том числе вводить и выводить его из режима пониженного энергопотребления. Задачей экспериментов было оценить принципиальную возможность использования режима энергопотребления как альтернативного подхода для реализации ФНФ на основе ДОЗУ.

В рамках эксперимента ОЗУ было записано константными значениями '1', затем оно было введено в режим пониженного энергопотребления. Через 5 минут ОЗУ выводилось в нормальный режим работы, и его содержимое вычитывалось для анализа. Значение '1' было выбрано исходя из характера процессов, которые протекают в запоминающей ячейке: данное значение соответствует наличию заряда, который будет утекать из ячейки при отключении регенерации с течением времени. Чтобы выполнить первичную оценку случайности полученных данных, были построены графики частоты встречаемости для всех возможных значений, которые может принимать байт, один из которых представлен на рисунке 1. Очевидно, что распределение имеет неравномерный характер с преобладанием длинных последовательностей единичных бит.

Несмотря на это, попарное сравнение данных разных экземпляров ОЗУ показало, что они полностью различимы. Так, доля различий составила в среднем 46%, а часть карты различий для 10 устройств показана на рисунке 2.

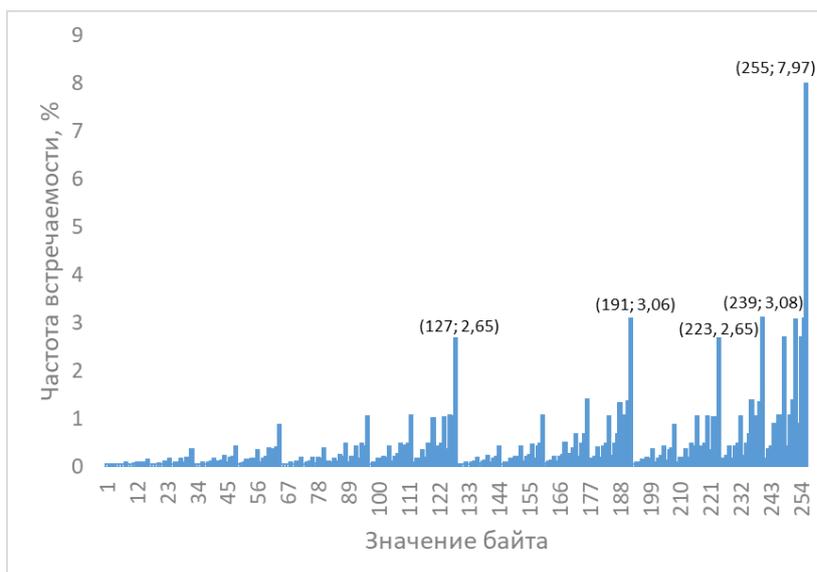


Рис. 1 – Частота встречаемости всех возможных значений байта в ДОЗУ после выхода из режима пониженного энергопотребления

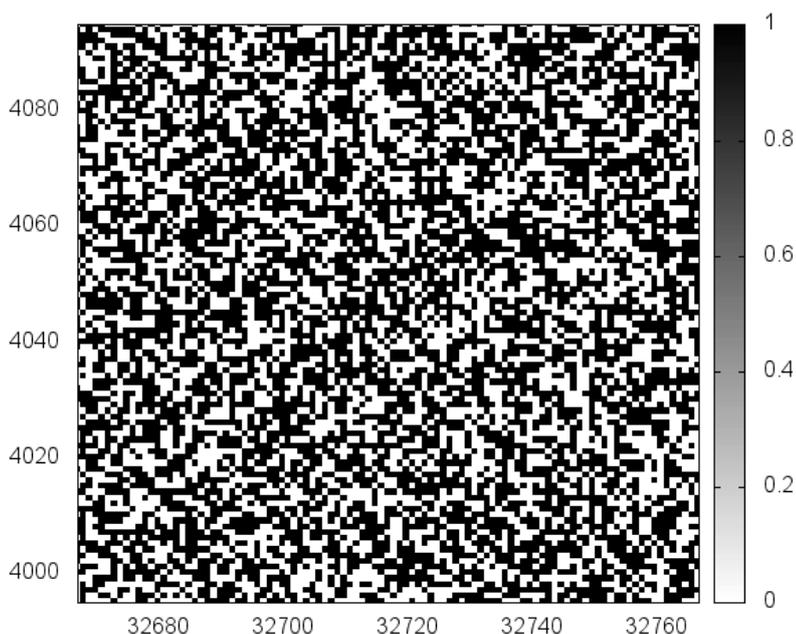


Рис. 2 – Часть карты различий состояний запоминающих ячеек для 10 экземпляров ДОЗУ

Таким образом, в ходе первичных экспериментов было показано, что с использованием режима пониженного энергопотребления можно осуществлять извлечение уникальных характеристик ДОЗУ, что является альтернативным подходом, не требующим выключения и включения всей цифровой системы. Кроме того, использование режима пониженного энергопотребления представляет интерес для мобильных устройств. Однако первичные данные говорят о низкой случайности данных, полученных данным методом, что не дает использовать его для построения генераторов истинно случайных последовательностей.

Список использованных источников:

1. Tao, S. Reliable low-overhead arbiter-based physical unclonable functions for resource-constrained IoT devices / S. Tao, E. Dubrova // Proc. of the 4th Workshop on Cryptography and Security in Computing Systems. – Stockholm, Sweden, 2017. – P. 1–6.
2. Rosenblatt, S. A Self-Authenticating Chip Architecture Using an Intrinsic Fingerprint of Embedded DRAM / S. Rosenblatt, S. Chellappa, A. Cestero, N. Robson, T. Kirihata, S. S. Iyer // IEEE Journal of Solid-State Circuits. – 2013. – Vol. 48, No. 11. – P. 2934–2943.

KUBERNETES КЛАСТЕР ДЛЯ ПОЛНОТЕКСТОВОГО ПОИСКА

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

Сенькович Д.С.

Жвакина А. В. – к. т. н, доцент

На сегодняшний день в разработке программного обеспечения используется все больше независимых взаимодействующих между собой приложений. Особенно остро проблема проявляется при разработке приложений по обработке больших данных. Целью проводимых исследований является создание кластера Docker контейнеров с Apache Hadoop и Apache Solr под управлением Kubernetes. Решение полезно как для обучения технологии полнотекстового поиска, так и при развертывании кластера для тестирования и разработки.

Задача поиска по тексту предполагает работу со сложными индексами и большими объемами данных. Поэтому для ее решения образовалась целая группа отдельных инструментов.

Многие базы данных имеют встроенную возможность поиска по тексту, однако обычно это очень ограниченная реализация. В большинстве случаев следует использовать более подходящие технологии. Как следствие, получили широкое распространение технологии полнотекстового поиска, например, Apache Solr и Elasticsearch.

Solr – платформа полнотекстового поиска с открытым исходным кодом, основанная на проекте Apache Lucene. Ее основные возможности: полнотекстовый поиск, подсветка результатов, фасетный поиск, динамическая кластеризация, интеграция с базами данных, обработка документов со сложным форматом (например, Word, PDF). Так как в Solr есть возможность распределенного поиска и репликации, Solr хорошо масштабируем. По состоянию на май 2016 года Solr является вторым по популярности поисковым движком, уступая Elasticsearch.

Solr и Elasticsearch являются лидерами на рынке полнотекстового поиска. Более популярным решением является Elasticsearch ввиду простоты начала работы с ним. Solr в то же время имеет большую историю и является решением с открытым исходным кодом, также как и его экосистема. Обе системы богаты своими возможностями и не уступают друг другу по скорости, используя один движок для полнотекстового поиска Apache Lucene. Принимая во внимание, что и движок, и Apache Solr разрабатываются одной командой, можно предполагать лучшую интеграцию и использование возможностей движка. Из различий также можно отметить более долговременную поддержку существующих возможностей в Apache Solr и возможность использовать встроенное машинное обучение, что возможно только в платной версии Elasticsearch.

Solr использует Apache HDFS для хранения документов и индексов, что обеспечивает сохранность и быстрый доступ.

HDFS (Hadoop Distributed File System) -- файловая система, предназначенная для хранения файлов больших размеров, поблочко распределённых между узлами вычислительного кластера. Все блоки в HDFS (кроме последнего блока файла) имеют одинаковый размер, и каждый блок может быть размещён на нескольких узлах, размер блока и коэффициент репликации (количество узлов, на которых должен быть размещён каждый блок) определяются в настройках на уровне файла. Благодаря репликации обеспечивается устойчивость распределённой системы к отказам отдельных узлов. Файлы в HDFS могут быть записаны лишь однажды (модификация не поддерживается), а запись в файл может вести только один процесс. Организация файлов в пространстве имён – традиционная иерархическая: есть корневой каталог, поддерживается вложение каталогов, в одном каталоге могут располагаться и файлы, и другие каталоги.

В связи с увеличением количества компонентов (программ) при разработке программного обеспечения установка и мониторинг, подготовка среды для предложения становятся сложнее. Поэтому активно развиваются технологии виртуализации, в частности, контейнеризации - виртуализации уровня ОС. Наиболее популярным решением является Docker.

Docker – это инструмент, предназначенный для упрощения создания, развертывания и запуска приложений с помощью контейнеров. Контейнеры позволяют разработчику упаковать приложение со всеми необходимыми ему компонентами, такими как библиотеки и другие зависимости, и поставлять все это как один пакет. Таким образом, благодаря Docker, разработчик может быть уверен, что приложение будет работать на любой другой машине Linux, независимо от каких-либо пользовательских настроек, которые могут быть у машины, которые могут отличаться от машины, используемой для написания и тестирования кода.

Docker помогает в решении унификации использования программного обеспечения, но остается необходимость мониторинга и поддержки самих контейнеров Docker, в чем помогает Kubernetes.

Kubernetes, или k8s, или "kube"- открытое программное обеспечение для автоматизации развёртывания, масштабирования и управления контейнеризированными приложениями. Оригинальная версия разработана компанией Google. Впоследствии Kubernetes был передан под управление Cloud Native Computing Foundation. Предназначение Kubernetes – предоставить «платформу для автоматического развёртывания, масштабирования, управления приложениями на кластерах или отдельных хостах». Kubernetes группирует контейнеры, составляющие приложение, в логические блоки для упрощения управления и обнаружения. Kubernetes устраняет многие ручные процессы, связанные с развертыванием и масштабированием контейнерных приложений. Другими словами, Kubernetes позволяет объединять группы

хостов, работающих под управлением контейнеров Linux, и легко и эффективно управлять этими кластерами. Такие кластеры могут охватывать узлы в общедоступных, частных или гибридных облаках.

Kubernetes поддерживает различные технологии контейнеризации, включая Docker, VMWare и ряд других. Kubernetes является наиболее популярным решением по сравнению с Docker Swarm и Nomad ввиду своего функционала и расширяемости, несмотря на сложности и нужный уровень технической подготовки для работы с ним. Целью исследования является построение масштабируемого отказоустойчивого кластера контейнеров Docker, поэтому использоваться будет Kubernetes.

В работе Apache Solr и Apache Hadoop (в частности, используемая его часть HDFS) помещены в Docker контейнеры для унификации и упрощения работы с ними. Docker контейнеры сгруппированы в Kubernetes кластер для упрощения управления и мониторинга. Таким образом, решение представляет собой конфигурируемый кластер полнотекстового поиска, удобный для обучения и разработки.

Список использованных источников:

1. Хайлоад. Полнотекстовый Поиск. [Электронный ресурс]. – Режим доступа: <https://ruhighload.com/%D0%9F%D0%BE%D0%BB%D0%BD%D0%BE%D1%82%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B9+%D0%BF%D0%BE%D0%B8%D1%81%D0%BA> - Дата доступа: 24.03.2018.
2. Apache Solr. [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Apache_Solr. – Дата доступа: 25.03.2017.
3. Docker Cookbook – O'Reilly Media, 2015. [Электронный ресурс]. – Режим доступа: <http://shop.oreilly.com/product/0636920036791.do> – Дата доступа: 25.03.2017.
4. Scaling Big Data with Hadoop and Solr - Second Edition – Packt, 2015. [Электронный ресурс]. – Режим доступа: <http://shop.oreilly.com/product/0636920036791.do> – Дата доступа: 25.03.2017.

УЛУЧШЕНИЕ КАЧЕСТВА ОЦЕНКИ АРХИТЕКТУРЫ ПО С ПОМОЩЬЮ ИСПОЛЬЗОВАНИЯ SAAM МЕТОДОЛОГИИ

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

Сорокина А.Г.

Волорова Н.А. – к.т.н., доцент

В настоящее время архитектура программного обеспечения является объектом повышенного внимания, как в академических кругах, так и в промышленной разработке. Но, несмотря на популярность этой темы, достаточно мало внимания уделяется методам анализа разработанной архитектуры, которые могут показать, что она удовлетворяет определенным свойствам.

На большинстве крупных проектов особенно из сферы финансов, обороны и медицины проводят рецензирование разработанного дизайна, которое является важным компонентом процесса создания программного обеспечения. Тем не менее, такие проверки часто проводятся не постоянно и многие участники процесса считают их напрасной тратой времени, так как они не выявляют всех проблем и дефектов, которые впоследствии выявятся только в процессе разработки, что уже слишком поздно.

Эта проблема появляется вследствие того, что в большинстве случаев сложно объективно оценить заявления разработчика о качествах, присущих созданной программной архитектуре. Примерами таких заявлений являются фразы на подобие [1]:

- Мы разработали компоненты, которые можно перенастроить с минимальными усилиями.
- Этот метод поощряет такое разделение приложения на части, которое уменьшит стоимость последующих модификаций системы.
- Этот подход лучше, чем традиционные подходы с точки зрения модульности и повторного использования кода.

Трудность в оценке этих требований возникает по двум причинам. Во-первых, различные архитектурные решения не используют общий словарь. А когда разрабатываются новые архитектуры, то они обычно разрабатывают новые термины для описания или используют старые термины по-новому. Во-вторых, часто сложно связать архитектурные абстракции с проблемами развития системы. Разработчики, как правило, концентрируются на функциональных особенностях своих архитектур и редко решают, как их архитектуры решают проблемы в рамках развития определенной системы.

Как одно из решений описанных выше проблем в рамках этой работы рассмотрим процесс рецензирования программного обеспечения, основанный на методе анализа архитектуры ПО (Softwarearchitectureanalysismethod - SAAM). SAAM является первым задокументированным методом анализа архитектуры приложений и служит основой для многих более современных методов анализа архитектуры. Например, он является предшественником для метода анализа альтернатив (Architecturetradeoffanalysismethod - ATAM).

Процесс анализа архитектуры SAAM представляет собой метод, который призван помочь в оценке и понимании архитектуры программного обеспечения и в решении проблем качества, таких как переносимость, сопровождаемость, расширяемость, модульность, повторное использование и т.д. Согласно результатам исследований, использование специализированных методов анализа программной архитектуры, а в

частности SAAM, дает потенциальные преимущества по сравнению с традиционным процессом рассмотрения проектов при выявлении и уточнении требований.

Основной целью методологии SAAM является выяснение того как были достигнуты конкретные атрибуты качества приложения и как возможные изменения в будущем повлияют на эти атрибуты. Поэтому главным этапом SAAM процесса является идентификация сценариев, которые показывают то, как система может использоваться или модифицироваться. Для оценки и сравнения сценариев рассчитывается их стоимость путем подсчета количества необходимых изменений. К тому же в процессе оценки возникает понимание того на какие компоненты влияет каждый сценарий, что позволяет выявить компоненты, которые учувствуют в слишком большом количестве сценариев и как следствие являются показателем плохого разделения ответственностей в архитектуре.

Процесс анализа состоит из 6 активностей [2]:

- 1) Разработка сценариев;
- 2) Описание архитектуры программного обеспечения;
- 3) Классификация сценариев и задание им приоритетов;
- 4) Оценка индивидуальных сценариев;
- 5) Взаимодействие сценариев;
- 6) Общая оценка;

Часто дополнительно выделяют этапы подготовки и беглого обзора архитектуры, на которых подробно изучается вся предоставленная документация. При наличии нескольких вариантов архитектур добавляется этап сравнения этих архитектур, используя выбранные сценарии.

Как можно заметить из вышеперечисленного, SAAM предоставляет нам много преимуществ. А именно:

- Дает возможность обнаружения проблем на ранних стадиях;
- Помогает улучшить документацию и наше понимание системы и путей ее развития;
- Привлекает к процессу оценки архитекторы дополнительных заинтересованных людей, таких как акционеры, архитекторы, команды сопровождения и разработки;
- Предоставляет методы и рекомендации по повышению качества и классификации различных сценариев;
- Позволяет проводить объективное сравнение нескольких архитектур благодаря унифицированному словарю и системе оценок и с учетом потребностей конкретной системы.

Список использованных источников:

1. Rick Kazman, Len Bass, Gregory Abowd, Mike Webb, SAAM: A Method for Analyzing the Properties of Software Architectures
2. Ali Athar, A Comparative Analysis of Software Architecture Evaluation Methods

ГРАФИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ ДЕЙСТВИТЕЛЬНЫХ И КОМПЛЕКСНЫХ КОРНЕЙ ТРИГОНОМЕТРИЧЕСКИХ И ГИПЕРБОЛИЧЕСКИХ УРАВНЕНИЙ

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

Таланец А.В.

Стройникова Е. Д. – ассистент кафедры информатики

Многие процессы окружающей среды описываются с помощью уравнений. Уравнения широко применяются в различных областях науки и техники. Решение уравнений – составная часть описания процессов, позволяющая изучить их. Одним из способов решения уравнений является графический способ. Графическое представление корней уравнения даёт полное представление о количестве корней уравнения, позволяет лучше понять взаимосвязь коэффициентов уравнения и расположения корней, даёт глубокое понимание связи между уравнениями и функциями. В данном докладе приведён пример графического представления корней тригонометрических и гиперболических функций.

Рассмотрим уравнение

$$f(z) = 0. \quad (1)$$

Абсциссы точек пересечения графика функции $y = f(z)$ с осью Ox являются корнями уравнения (1). Эти точки определяют геометрический смысл корней уравнения. Однако если уравнение имеет не только действительные корни или не имеет действительных корней, то графическое представление всех корней уравнения может дать только исследование функции $f(z)$ в трёхмерном пространстве.

Для представления корней уравнения $f(z) = 0$ достаточно рассмотреть лишь действительные значения, которые принимает функция $f(z)$.

Зададим трёхмерное пространство:

- 1) плоскость xOy – плоскость комплексного аргумента $z = x + iy$, Ox – действительная ось, Oy – мнимая ось;
- 2) перпендикулярно данной плоскости отложим ось Or – ось действительных значений функции

$$r = f(z).$$

Графическим представлением корней уравнения (1) являются точки, в которых график функции пересекает плоскость xOy .

С помощью формул $\sin x = \frac{e^{ix} - e^{-ix}}{2i}$, $\cos x = \frac{e^{ix} + e^{-ix}}{2}$, $\text{sh } x = \frac{e^x - e^{-x}}{2}$, $\text{ch } x = \frac{e^x + e^{-x}}{2}$ тригонометрические функции от мнимого аргумента выражаются через гиперболические функции:

$$\sin(ix) = i \text{sh}x, \quad (2.1)$$

$$\cos(ix) = \text{ch}x. \quad (2.2)$$

Рассмотрим функцию $r(z) = \cos(z)$.

Пользуясь формулами (2.1) и (2.2), можем записать

$$\cos(z) = \cos(x + iy) = \cos(x) \cos(iy) - \sin(x) \sin(iy) = \cos(x) \text{ch}(y) - i \sin(x) \text{sh}(y),$$

$$\cos(z) = 0 \Rightarrow \text{Im}(r) = \sin(x) \text{sh}(y) = 0, \text{ т. е. } \text{sh}(y) = 0 \text{ или } \sin(x) = 0.$$

1. Если $\text{sh}(y) = 0$, то $y = 0$ и $r = \cos(x) \text{ch}(y) = \cos(x)$.

2. Если $\sin(x) = 0$, то $x = \pi k$, $k = 2n$, $n \in \mathbf{Z}$ и $r = \text{ch}(y)$ или $x = \pi k$, $k = 2n + 1$, $n \in \mathbf{Z}$ и $r = -\text{ch}(y)$.

Построив график функции $r = \cos(z)$ (рис. 1.1, 1.2), можно заметить, что уравнение $\cos(z) = r$ при $|r| \leq 1$ имеет бесконечно много действительных корней и не имеет комплексных корней, при $|r| > 1$ – бесконечно много комплексных корней и не имеет действительных корней.

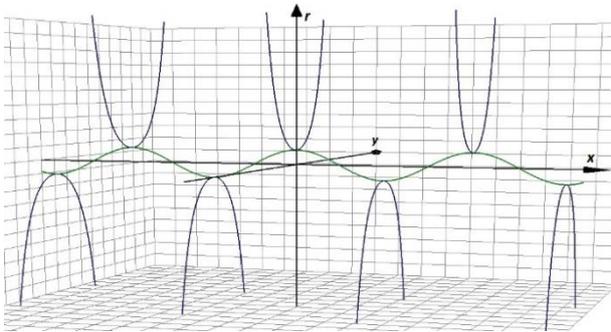


Рис. 1.1 – График функции $r = \cos(z)$

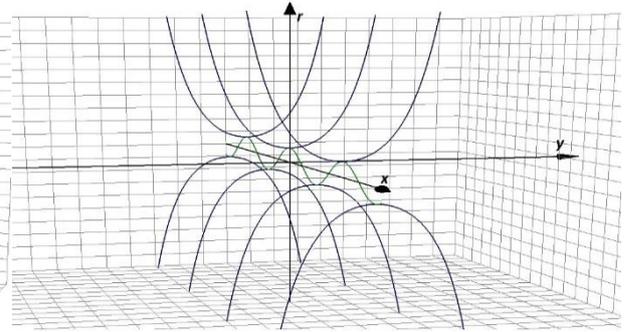


Рис. 1.2 – График функции $r = \cos(z)$

График функции $r(z) = \sin(z)$ можно получить аналогичными рассуждениями или сдвигом графика функции $r(z) = \cos(z)$ по оси Ox вправо на $\pi/2$.

Нетрудно убедиться, что уравнения $\text{tg}(z) = r$ и $\text{ctg}(z) = r$ при $r \in \mathbf{R}$ имеют бесконечно много действительных корней и не имеют комплексных.

Рассмотрим функцию $r(z) = \text{sh}(z)$.

Пользуясь формулами (2.1) и (2.2), можем записать

$$\text{sh}(z) = \text{sh}(x + iy) = \text{sh}(x) \text{ch}(iy) + \text{ch}(x) \text{sh}(iy) = \text{sh}(x) \cos(y) + i \text{ch}(x) \sin(y),$$

$$\text{sh}(z) = 0 \Rightarrow \text{Im}(r) = \text{ch}(x) \sin(y) = 0, \text{ т. е. } \sin(y) = 0 \text{ (т. к. } \text{ch}(x) \geq 1, x \in \mathbf{R}).$$

Значит, $x = \pi k$, $k = 2n$, $n \in \mathbf{Z}$ и $r = \text{sh}(x)$ или $x = \pi k$, $k = 2n + 1$, $n \in \mathbf{Z}$ и $r = -\text{sh}(x)$.

Построив график функции $r = \text{sh}(z)$ (рис. 2.1, 2.2), можно заметить, что уравнение $\text{sh}(z) = r$ при любом $r \in \mathbf{R}$ имеет один действительный корень и бесконечно много комплексных.

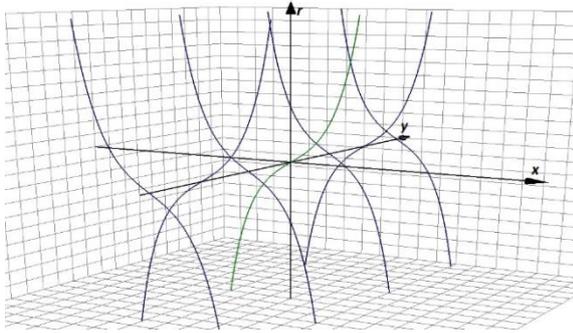


Рис. 2.1 – График функции $r = sh(z)$

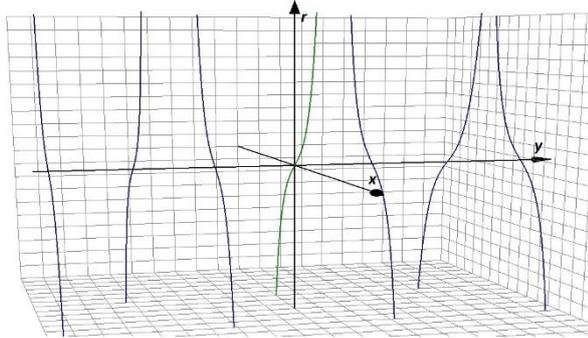


Рис. 2.2 – График функции $r = sh(z)$

Рассмотрим функцию $r(z) = ch(z)$.

Пользуясь формулами (2.1) и (2.2), можем записать

$$ch(z) = ch(x + iy) = ch(x) ch(iy) + sh(x) sh(iy) = ch(x) \cos(y) + ish(x) \sin(y),$$

$$ch(z) = 0 \Rightarrow \text{Im}(r) = sh(x) \sin(y) = 0, \text{ т. е. } sh(x) = 0 \text{ или } \sin(y) = 0.$$

1. Если $sh(x) = 0$, то $x = 0$ и $r = ch(x) \cos(y) = \cos(y)$.

2. Если $\sin(y) = 0$, то $y = \pi k$, $k = 2n$, $n \in \mathbf{Z}$ и $r = ch(x)$ или $y = \pi k$, $k = 2n + 1$, $n \in \mathbf{Z}$ и $r = -ch(x)$.

Построив график функции $r = ch(z)$ (рис. 3.1, 3.2), можно заметить, что уравнение $ch(z) = \pi r$ при $r > 1$ имеет два действительных корня и бесконечно много комплексных, при $r = 1$ – один действительный корень и бесконечно много комплексных, при $r < 1$ – бесконечно много комплексных корней и не имеет действительных.

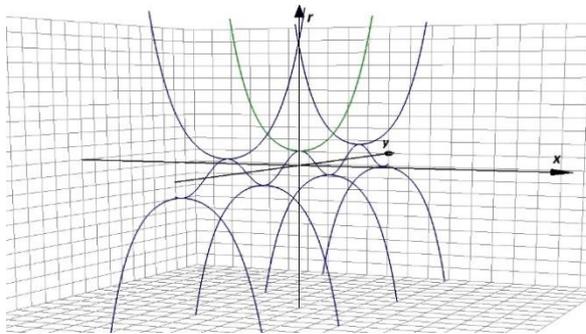


Рис. 3.1 – График функции $r = ch(z)$

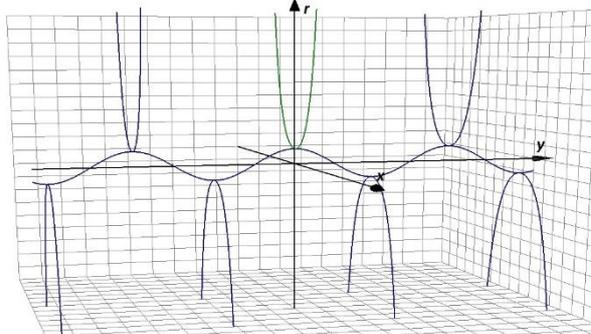


Рис. 3.2 – График функции $r = ch(z)$

Заметим, что график функции $ch(z)$ можно получить поворотом графика функции $\cos(z)$ на 90° относительно оси Or . Таким образом, в силу сделанного замечания и симметрии графиков относительно плоскостей xOr и yOr можем получить формулу

$$\cos(x + iy) = ch(y + ix).$$

Важно понимать, что полученная формула верна лишь для действительных значений функций, т. е. при $y = 0$ или $x = \pi k$, $k \in \mathbf{Z}$.

Уравнения $th(z) = r$ и $cth(z) = r$ так же, как и уравнения $tg(z) = r$ и $ctg(z) = r$, при $r \in \mathbf{R}$ не имеют комплексных корней.

Заключение

В ходе исследования было получено графическое представление корней тригонометрических и гиперболических уравнений. Была продемонстрирована возможность доказательства формул графическим способом. Полученные результаты позволяют решать некоторые уравнения с помощью графической интерпретации, не прибегая к сложным алгебраическим выражениям. Аналогичным способом возможно рассмотрение других функций, графическое представление корней которых поможет решать некоторые уравнения и понимать зависимость корней от коэффициентов.

Список использованных источников:

1. Федосов Б. Т. О представлении корней алгебраических полиномов в трехмерном пространстве (этикет о комплексных числах) [Электронный ресурс]. – Режим доступа: http://model.exponenta.ru/bt/bt_001141.html.
2. Свободная энциклопедия Википедия, статья «Гиперболические функции» [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Гиперболические_функции.
3. Свободная энциклопедия Википедия, статья «Формула Эйлера» [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Формула_Эйлера.

ОБРАБОТКА БОЛЬШИХ ОБЪЕМОВ ИНФОРМАЦИИ С ИСПОЛЬЗОВАНИЕМ ПЛАТФОРМЫ HADOOP И СЛУЖБЫ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ MICROSOFT AZURE

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

Хадасевич А.И., Швец В.И.

Хотеев А.Л. – к.ф.-м.н., доцент

С каждым годом задачи обработки больших объемов данных все чаще становятся перед разработчиками. В настоящее время генерируются данные о деятельности людей и объектов в огромном количестве и растущем масштабе. Для изучения, поиска закономерностей и анализа этих данных используются специальные инструменты и методы.

Платформа Hadoop представляет собой набор утилит, библиотек и фреймворков для разработки и выполнения распределённых программ, работающих на кластерах из сотен и тысяч узлов с использованием простых моделей программирования. Он также масштабируемый от отдельных серверов до тысяч компьютеров, каждый из которых предлагает локальные вычисления и хранение. Данная платформа является своего рода стандартом для решения задач, связанных с полу- и неструктурированными большими данными. Однако для начала работы с платформой Hadoop, требуется много времени для изучения основ платформы. Также достаточно велики начальные финансовые затраты на аппаратное обеспечение такого рода кластера. Эти причины так же усложняют обучение для работы с данной платформой. Одно из возможных решений приведенных проблем, является объединение облачных технологий и платформы Hadoop. Это достаточно многообещающий способ решения проблемы «Больших данных», имеющий низкий уровень входа (квалификация разработчика и стоимость на запуск).

Компания Microsoft предлагает Windows Azure HDInsight — это полностью управляемая облачная служба, которая позволяет быстро, просто и без лишних затрат обрабатывать большие объемы данных. Данное облачное решение поддерживает различные сценарии:

- 1) хранение данных;
- 2) обработка, анализ данных и машинное обучение;
- 3) извлечение, преобразование и загрузка;
- 4) "интернет вещей";
- 5) приложения потоковой передачи.

Для разработчиков Windows Azure HDInsight имеет плагин для Visual Studio, который поддерживает создание приложений. Для разработчиков Linux или Windows у HDInsight есть плагины для IntelliJ IDEA и Eclipse, двух очень популярных платформ Java IDE с открытым исходным кодом. HDInsight также поддерживает команды PowerShell, Bash и Windows, позволяющие создавать сценарии рабочих процессов. Также отличительной особенностью является возможность использования не JVM языков с платформой Hadoop. Таким образом увеличивается количество разработчиков, которые могут работать с Hadoop. JavaScript API позволяет разработчикам определять задачи Hadoop в более естественном стиле. Разработчики .NET также могут работать с Hadoop, используя преимущества языковых функций C# и F#, таких как LINQ.

Для ученых-исследователей, HDInsight включает Jupyter. Jupyter- это среда разработки на базе обычного текстового редактора, которая позволяет интегрировать код и контент. Когда код и контент объединяются, они создают живой документ, который обновляется данными.

Также компания Microsoft предоставляет удобную бесплатную среду разработки WindowsAzureHDInsightEmulator. Это реализация HDInsight для Windows, предоставляющая набор служб на базе Apache Hadoop в Windows используя только одноуровневое развертывание. Сервер HDInsight обеспечивает локальную среду разработки для службы Windows Azure HDInsight. Эта технология разрабатывается для обеспечения программной среды, предназначенной для управления, анализа и отчетности по большим данным. Подобно службе Windows Azure HDInsight, эта локальная среда разработки для HDInsight упрощает настройку, выполнение и пост-обработку заданий Hadoop, предоставляя библиотеку PowerShell с командлетами HDInsight для управления кластером и выполняемыми на нем заданиями.

Windows Azure HDInsight является гибким решением, которое позволяет программистам за небольшое время развернуть Hadoop кластер или выбрать требуемую услугу в облаке. Windows Azure HDInsight снижает уровень входа для разработчиков, которые будут работать с приложениями, связанными с обработкой больших объемов данных. Разработчики, которые ранее не работали с большими данными, получают большой выбор языков программирования и знакомые инструменты разработки.

Список использованных источников:

1. AzureHDInsightDocumentation. [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://docs.microsoft.com/en-us/azure/hdinsight/>
2. HDInsight - Fully managed, full spectrum open-source analytics service for enterprises. [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://azure.microsoft.com/en-us/services/hdinsight/>

ОБРАБОТКА БОЛЬШИХ ОБЪЕМОВ ДАННЫХ В АВТОМАТИЗИРОВАННЫХ СИСТЕМАХ УПРАВЛЕНИЯ АВТОМОБИЛЬНЫМ ПОТОКОМ

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

Швец В.И., Хадасевич А.И.

Теслюк В.Н. – к.ф.-м.н., доцент

Ввиду постоянного роста автомобилизации на дорогах применяются новейшие технологии, призванные повысить уровень безопасности. Одно из приоритетных направлений – развитие автоматизированных систем управления дорожным движением, способных в режиме реального или близком к реальному времени считывать, анализировать и управлять обстановкой на дорогах. Скорость работы этих систем обеспечивается не только аппаратной составляющей, но и применением алгоритмов обработки больших объемов информации.

Автоматизированная система управления автомобильным дорожным движением представляет собой комплекс программно-аппаратных средств, способных в режиме реального или близком к реальному времени считывать постоянно изменяющуюся обстановку на дорогах, оценивать её по заданным параметрам (интенсивность, средняя скорость, заполненность полосы или дороги и т.д.), распознавать возникающие нештатные ситуации (заторы, сложные метеорологические условия, дорожно-транспортные происшествия), а затем принимать меры по регулированию потока, недопущению аварийных ситуаций или минимизации их негативных последствий.

Сбор данных о дорожной обстановке осуществляется с помощью специально разработанной аппаратуры: камер скорости, видеокамер, дорожных метеостанций, детекторов транспортного потока, датчиков и радаров. Регулирование потока выполняется с использованием электронных информационных панелей, указателей и табло, светофорных объектов, шлагбаумов. Вся вышеперечисленная аппаратура располагается непосредственно на участках дорог, обмен информацией между ней и центром управления дорожным движением осуществляется по проводным и беспроводным каналам связи.

Наиболее сложным процессом является автоматизация анализа данных и принятия решений. Ещё в недалёком прошлом данные, получаемые в центре управления дорожным движением, обрабатывались операторами - специально обученными людьми. Современные же системы, применяемые в крупных городах, могут насчитывать десятки и сотни тысяч объектов взаимодействия с водителями автомобилей. Данные от них поступают регулярно через короткие промежутки времени, и обработать их человеку или даже группе людей не под силу. В результате чего необходимо применять алгоритмы обработки больших объемов данных, способные за приемлемое время выдавать наиболее оптимальный результат. В качестве примера можно привести ситуацию, когда в случае распознавания затора система с помощью светофорных объектов и информационных табло перенаправляет водителей на соседние улицы, если те свободны. Другой пример – временное ограничение максимально допустимой скорости движения в сложных погодных условиях (снегопад, гололёд).

При использовании автоматизированных систем управления нельзя не учитывать специфику конкретного города и его районов, зависимость транспортных потоков от времени суток и периода года. Ввод в систему ограничений и параметров требует длительного наблюдения за дорожной обстановкой со стороны человека. Более предпочтительным в данном вопросе видится применение алгоритмов машинного обучения. Система с течением времени сама накапливает данные, производит статистический анализ и предлагает оператору системы наиболее оптимальные варианты поведения для конкретного участка дороги в распространённых ситуациях. Например, если в зимний период года система фиксирует снижение средней скорости движения, в ручном или автоматическом режиме могут быть изменены интервалы светофорного регулирования. Необходимость принятия решения оператором в таких ситуациях сводится к минимуму.

Следует отметить, что алгоритмы обработки больших объемов данных могут быть применены не только для анализа дорожной обстановки в центрах обработки данных. К вышеуказанной категории также относятся алгоритмы распознавания изображений. В результате распознавания картинки с дорожных камер можно определить, например, тип транспортного средства, автомобильный номер. Алгоритмы анализа изображений применяются также для распознавания правонарушений: движение с непристёгнутым ремнём безопасности, выезд за стоп-линию и т.д. Эта информация при необходимости может быть передана взаимодействующим компьютерным системам.

Для поддержания высокой скорости обработки данных, стабильности и бесперебойности работы подобные системы создаются распределёнными, т.е. работают на кластерах из сотен и тысяч узлов. Обрабатываемые данные дублируются и в случае отказа одного узла/кластера не теряются, процесс обработки не замедляется. Горизонтальная масштабируемость систем позволяет использовать необходимое количество недорогих узлов для увеличения производительности, не прибегая к покупке мощных серверов и дорогих сетей хранения данных. Классическим примером фреймворка для распределённого хранения и обработки данных является проект ApacheHadoop.

Хранение огромного массива данных в таких системах требует комплексного использования как классических SQL баз данных, так и СУБД с подходом NoSQL. Для решения определённых задач с высокой скоростью доступа может быть полезным использование графовых баз данных.

Таким образом, применение методов хранения и алгоритмов обработки данных больших объёмов в современных автоматизированных системах управления дорожным движением является не просто необходимым, но и закономерным шагом для их эволюции. Благодаря этому поддерживается высокая скорость доступа и обработки данных, получение результатов в режиме реального времени, автоматическая адаптация под изменяющиеся условия на базе алгоритмов машинного обучения, преобразование форматов данных с помощью алгоритмов распознавания изображений. Ко всему этому добавляется высокая отказоустойчивость системы в результате распределённой обработки данных.

Список использованных источников:

1. Развитие автоматизированной системы управления дорожным движением Минска как части интеллектуальной транспортной системы города. [Электронный ресурс]. – Электронные данные. – Режим доступа: https://rep.bntu.by/bitstream/handle/data/27593/Razvitiie_avtomatizirovannoj_sistemy_upravleniya_dorozhnym_dvizheniem_Minska.pdf
2. Автоматизированная система управления дорожным движением «АГАТ» [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://agat.by/wp-content/uploads/2015/05/АСУДД-АГАТ-презентация.pdf>
3. АСУДД: Что висит над дорогой? [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://habr.com/post/124249/>

ПОСТРОЕНИЕ РЕКОМЕНДАТЕЛЬНОЙ СИСТЕМЫ НА ОСНОВЕ АЛГОРИТМОВ ОБРАБОТКИ ЕСТЕСТВЕННОГО ЯЗЫКА

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

Хопова А.А., Волорова Н.А.

Волорова Н.А. – к.т.н., доцент

В данном докладе рассматриваются методы тематического моделирования текстов, а также оценки качества получаемых результатов. При этом в качестве исходных данных используются тексты микроблогов, которые существенно отличаются от традиционных текстов книг, статей и пр. Так же рассматриваются система рекомендаций на основе схожести контента и с учётом реакций других пользователей (лайки, репосты и т.д.). Такая система является более эффективной, чем традиционный подход (фильтрация), благодаря использованию дополнительных метрик при формировании рекомендации. Применение такой системы позволит пользователям находить релевантные материалы, хранящиеся в социальных сетях.

Введение

В наше время количество информации постоянно растёт, большая её часть представляет собой неупорядоченные текстовые данные, например, различные WEB-ресурсы, социальные сети, блоги, форумы, новостные сайты и тому подобное. Человеку сложно самостоятельно их обрабатывать. Более того, ручной анализ неэффективен для больших объёмов текста, так как он ограничен скоростью, погрешностями и ошибками, обусловленными человеческим фактором. Следовательно, требуются методы, способные автоматически обрабатывать такие данные. Методы могут основываться на извлечении определенной информации, например, списках тем, содержащихся в текстах.

Для того чтобы выделить из текста основные темы, человеку достаточно его прочитать. В условиях постоянно увеличивающегося количества информации, в частности, текстовой (так называемый, информационный бум), приходится анализировать данные такого объема, которые человек не в силах обработать. Поэтому необходимы методы, позволяющие автоматически извлекать темы из большого набора данных. Теоретически обоснованным и активно развивающимся направлением в анализе текстов на естественном языке, является тематическое моделирование коллекций текстовых документов.

Построение тематической модели может рассматриваться как задача одновременной кластеризации документов и слов по одному и тому же множеству кластеров, называемых темами. В терминах кластерного анализа тема – это результат би-кластеризации, то есть одновременной кластеризации и слов, и документов по их семантической близости. Как правило, выполняется нечёткая кластеризация, то есть документ может принадлежать нескольким темам в различной степени. Таким образом, сжатое семантическое описание слова или документа можно представить в виде распределения на множестве тем. Процесс нахождения этих распределений и называется тематическим моделированием.

В 2003 году Д.Блей предложил модель скрытого размещения Дирихле (Latent Dirichlet Allocation, LDA). Это одна из первых и широко используемых вероятностных тематических моделей. Основной идеей таких моделей является наличие генеративного процесса – процесса, порождающего документы с использованием предопределённых тем. Задача заключается в том, чтобы подобрать темы таким образом, чтобы вероятность сгенерировать данный набор документов была максимальной.

В вероятностных тематических моделях темы представляются в виде распределений над словами. Оценить качество полученных тем можно вручную: можно выбрать слова с наибольшей вероятностью и понять, что они вместе означают. При большом количестве тем требуется много времени, чтобы оценить, насколько понятными для человека они получились.

Алгоритмы поиска наиболее правдоподобных скрытых параметров делятся на две категории: на основе сэмплирования и вариационные методы. Алгоритмы первой группы пытаются собрать конечную выборку переменных, на которой ищется максимум.

Как правило, алгоритм принадлежит классу методов Монте-Карло для марковских цепей (MarkovChainMonteCarlo, MCMC). Примером такого алгоритма является сэмплирование по Гиббсу, которое состоит в том, чтобы на каждом шаге фиксировать все переменные, кроме одной, и выбирать оставшуюся переменную согласно распределению вероятности этой переменной при условии всех остальных. Методы второй группы вариационные алгоритмы. В них сначала задается параметризованное семейство распределений над скрытыми переменными, а затем с помощью EM-алгоритма ищется распределение из этого семейства, наиболее близкое к исходному апостериорному распределению.

В качестве языка программирования, на котором выполнялась реализация практических исследований, был выбран язык Java. Это объектно-ориентированный язык, который хорошо подходит для прикладных задач. Кроме того, программы, написанные и скомпилированные на Java можно запускать на любой операционной системе, где поддерживается запуск виртуальной машины Java. При проверке орфографии на этапе предобработки данных использовалась библиотека Snowball и MyStem. Для оценки интерпретируемости с помощью Google использовалась разрабатываемая в ИСП РАН утилита для скачивания веб-страниц из сети Интернет. Обе библиотеки реализованы на Java, что также является доводом в пользу данного языка программирования.

Использовались готовые реализации тематических моделей на языке C. Выбор этих реализаций обусловлен тем, что они принадлежат авторам исследуемых методов тематического моделирования. Кроме того, язык C хорошо подходит для таких задач, где производится большое количество вычислений.

Заключение

В процессе выполнения работы были исследованы методы определения тематической направленности текстового содержимого микроблогов и реализован алгоритм автоматической оценки интерпретируемости результатов тематического моделирования текстов микроблогов. Так же были исследованы существующие методы тематического моделирования и способы оценки их качества. Выполнена экспериментальная оценка интерпретируемости методов тематического моделирования текстов микроблогов с использованием разработанных методов. Разработаны и реализованы методы автоматической оценки интерпретируемости результатов тематического моделирования по ключевым словам тем. А также были проведены расчёты популярности сообщений (твиттов) при помощи метрик, на основании которых был осуществлён прогноз тенденции популярности.

ПРИНЯТИЕ РЕШЕНИЯ О ДЕНОРМАЛИЗАЦИИ БАЗЫ ДАННЫХ

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

Чочиева А.С.

Калугина М. А. - к.ф.-м.н., доцент

В ходе проектирования баз данных может возникнуть вопрос: “Необходима ли её большая нормализация и если да, то до какой степени?” Большинство источников указывают, что всегда нужно стремиться к нормализации базы данных - “больше таблиц, меньше столбцов”, к большему разделению на сущности. Нормализация является важным процессом проектирования базы данных. Но могут ли у неё иметься недостатки?

Нормализация базы данных — это процесс реструктурирования реляционной базы данных в соответствии с нормальными формами отношения с целью снижения избыточности данных и обеспечения их целостности. Денормализация — обратный этому процесс.

Сравним эти два процесса для выделения очевидных аргументов в пользу денормализации.

Нормализация экономит место, но это место стоит дешевле [1]. В денормализованной базе данных может находиться 10000 записей, например, названия страны, что занимает гораздо больше места, чем просто внешний целочисленный ключ на одну строку в другой таблице. Но терабайтные носители сейчас дешевле, чем ранее.

Нормализация упрощает изменение строки, но чтение обычно происходит чаще, а нормализация замедляет чтение. Таким образом, если данные редко меняются, имеется смысл задуматься о денормализации.

Расчетные данные [3]. Существуют ситуации, когда есть запросы с долгими, сложными вычислениями, потребляющими много ресурсов. В таких случаях можно выделить столбцы, в которых будут храниться часто используемые расчётные значения, чтобы избежать потери ресурсов и времени на их повторное вычисление.

Нормализация снижает производительность. Это и есть наиболее часто упоминаемая причина денормализации. Сильная нормализация может привести к большому количеству соединений таблиц, что выливается в объёмные составные SELECT-запросы с большим количеством JOIN операций- это будет замедлять работу базы данных. В источниках также упоминается [1], что нормализация часто становится причиной неполадок в работе программного обеспечения из-за ошибок в сложных запросах. В некоторых случаях целесообразно хранить редко изменяемые данные просто в текстовом файле, загружая их по мере необходимости.

С учетом проведенного анализа, денормализацию не рекомендуется применять в следующих случаях:

- когда присутствуют отношения “один-ко-многим” (например, у пользователя есть возможность ввести несколько адресов, поэтому имеет смысл отделить адреса в отдельную таблицу);
- когда есть частая необходимость создать уникальный список из этих данных (например, для выпадающего меню, из которого будет выбирать пользователь);
- когда данные часто обновляются.

С целью оптимизации работы с данными принимать решение о денормализации рекомендуется лишь после проектирования нормализованной базы данных [2,3]. Необходимо проанализировать характер и частоту запросов и только после этого, при необходимости, аккуратно, с учётом сохранения целостности данных, произвести денормализацию [2, 3]. В таком случае её побочные эффекты снижаются до минимума, при этом обеспечивая увеличение быстродействия базы данных.

К недостаткам денормализации можно отнести:

1. большее потребление места на диске (из-за дублирования данных);
2. возможное замедление операций вставки, обновления и удаления данных;
3. аномалии данных (нарушение целостности данных): в какой-то момент данные могут измениться в нескольких местах, в таком случае нужно обеспечить корректное обновление их копий [3] (или в случае с расчётными данными – пересчитать расчётные значения, полученные из изменённых данных);
4. документация: денормализованную базу данных сложнее поддерживать, поэтому денормализацию базы данных следует подробно документировать (если возникнет необходимость поменять структуру базы данных, то нужно будет учитывать все предыдущие изменения) [3];
5. денормализация может потребовать написания большего количества кода [2], например, триггеров и процедур, обеспечивающих целостность данных, а если она проводится на существующей рабочей базе данных, то может потребоваться изменение существующих запросов и написание кода для обновления уже имеющихся записей [3].

Таким образом, залог быстрой и исправно работающей базы данных – это нахождение оптимального равновесия между нормализацией и денормализацией. Но началом и основой проектирования базы данных остаётся нормализация.

Список использованных источников:

- 1.Scott Selikoff , «Why too much Database Normalization can be a Bad Thing», November 19th, 2008.
2. Michelle A. Poole, «Responsible Denormalization: How to break the rules and get away with it», 2002 Penton Media, Inc.
- 3.Блог компании Латера Софтвр, «Зачем нужна денормализация баз данных, и когда ее использовать», 9 апреля

2016

CHLORINE ENGINE И МИНИ-ИГРА НА НЕМ

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

Шпаков Н.И.

Жвакина А.В. – к.т.н., доцент

Игры все больше и больше заполняют наш мир. Кто-то считает это лишь простой забавой, кто-то бизнесом, а для кого-то это жизнь или искусство. В любом случае, нельзя отрицать большого влияния этой отрасли на IT в целом. Развиваются технологии для обработки графики, звука, появляются новые технологии анимации и рендера, увеличивается мощность железа. Так или иначе, игры являются двигателем прогресса в мире компьютерных технологий.

В данной работе представлена игра, написанная на движке Clorineengine и сам движок, написанный с использованием библиотек SDL2 (окно и обработка событий), GLEW (для вытягивания функций OpenGL), OpenAL (библиотека для стереозвука), GLM (математическая библиотека), picoPNG (декодирование PNG файлов), freetype (преобразование текста в битмапы). В качестве языка программирования выбран C++ 11. Для сборки проекта используется технология CMake. Готовый проект с открытым кодом можно найти по ссылке на github: [https://github.com/ShpakovNikita/Chlorine-5].

Движок умеет обрабатывать пользовательский ввод, рисовать квадраты на поверхности (под углом также), проигрывать звуки с учетом модели дистанции, делать рэйкасты, проверять коллизии разными способами, отрисовывать простейший свет, рисовать GUI, динамический текст по шрифту, работать с анимациями спрайтов и многое другое. Отрисовка происходит посредством передачи данных о каждом полигоне в вершинный шейдер, а затем передачи его во фрагментный шейдер для дальнейшего вывода цвета каждого пикселя на экран. Звуки реализованы через модель источников и слушателя, где для каждого динамического объекта создается свой источник и для слушателя, «привязанного» к камере, идет просчет доходящего звука от источников. Игра же, по сути, в игровом цикле делает все эти действия последовательно, с учетом ООП модели.

Игра является относительно простым примером работы с движком, задействуя абсолютно все его возможности и реализуя множество алгоритмов совместно с функциями движка. Окружение в игре генерируется случайным образом с помощью алгоритма процедурной генерации подземелья, и динамически подбирает тайлы для создания изометрического эффекта. Для изометрического эффекта используется также zbuffer. Реализован также искусственный интеллект, который представлен системой конечных автоматов и

использует систему анимации и звука движка для реалистичного представления врагов.

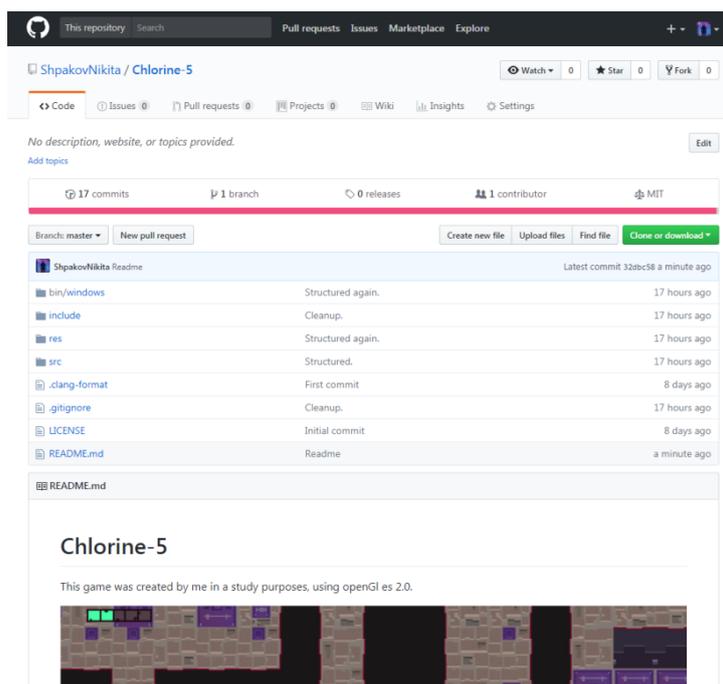


Рис. 1 – Проект на github

Основным преимуществом игрового движка является его легковесность. В нем присутствуют только самые основные функции, которых достаточно для создания полноценной 2д игры. Также движок может быть легко кастомизирован из-за подробного описания каждой функции и своего небольшого размера. Другим преимуществом является полная открытость исходного кода движка, который большинство других движков до сих пор скрывают. Однако у движка имеются недостатки, связанные с архитектурой, отсутствует возможность автоматического портирования проекта на другие платформы и пользователю придется делать это вручную. Устранение перечисленных проблем позволит игре стать более универсальной и конкурентоспособной.

Список использованных источников:

1. Материалы для обучения OpenGL. [Электронный ресурс] – Режим доступа: <https://learnopengl.com>. – Дата доступа: 13.03.2018.
2. Документация OpenGL. [Электронный ресурс] – Режим доступа: <http://www.khronos.org>. – Дата доступа: 13.03.2018.
3. Официальный сайт OpenAL с доступной pdf документацией. [Электронный ресурс] – Режим доступа: <http://www.openal.org>. – Дата доступа: 13.03.2018.
4. Документация SDL1, SDL2. [Электронный ресурс] – Режим доступа: <http://wiki.libsdl.org/FrontPage>. – Дата доступа: 13.03.2018.

СЕКЦИЯ «ФИЛОСОФИЯ И ЖИЗНЬ»

ПРОБЛЕМЫ ГЛОБАЛИЗАЦИИ ИНФОРМАЦИОННОГО ОБЩЕСТВА

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

Вильчук Ю.В.

Лосик Г.В. д. — д. псих.н., профессор

Пройдя ступени дошкольного и школьного образования, человек оказывается на пороге самостоятельного выбора. Предполагается, что он созрел для этого не только физически, но и духовно: профессиональное образование того или иного типа и уровня, непосредственное включение в продуктивную деятельность – всё это смысл жизни человека, как духовной единицы в условиях глобального информационного общества [1].

Актуальность данного исследования обусловлена изучением проблем глобализации, возникающих в процессе социализации личности в условиях существующего глобального информационного общества.

Мировые тенденции современного мира далеки от совершенства. Бог – путеводная звезда жизни нашей. Возникающие проблемы построения единого информационного общества связаны с разными мировоззренческими взглядами на мир, которые сформировались у молодых людей в процессе воспитания. В свою очередь, с одной стороны, это усложняет процессы духовной глобализации информационного общества, порождает новые проблемы, а с другой выступает многомерным инструментом по консолидации противоположных сил в единую гармоничную систему для развития новой культуры глобального информационного общества, основанного на единых ценностных ориентирах.

Наш мир является самоорганизующейся системой. От выполнения необходимых задач, записанных в виде программных кодов на генетическом уровне души человека, зависит успешность саморазвития единой системы глобального информационного общества, а также успешность построения нового информационного общества, где каждый находится на своём посту, и выполняет свой долг перед обществом, трудится по миссии с чувством собственного достоинства.

Данная концепция глобального информационного общества, на мой взгляд, отражает наиболее оптимальную основную идею становления и развития духовного глобального информационного общества в настоящих условиях.

Глобальное информационное общество должно постоянно эволюционировать, находиться в динамике времени, своевременно переключаться на новые по спирали витки развития. Но как показывает практика последних лет, мир меняется не столько эволюционно, сколько революционно.

Важной характеристикой глобального информационного общества является культура. Культура – это коллективный эгрегор развития душ человеческих. Однако сейчас существует большое множество разделённых культур и течений, которые могут существовать определённое время самостоятельно до тех пор, пока не возникнет единая бифуркационная точка, объединяющая их. По сути, это та точка, из которой данные направления культур и течений появились, перенесённая во времени, но уже являющейся синтезом культур с накопленным опытом, который, при правильном понимании, способствует укрупнению культуры для зарождения новых культур и течений более высокого порядка мерности в будущем.

Формирующийся глобальный мир является сложным и нелинейным. Глобализация означает резкое расширение того социального пространства, на котором реализуются разнородные связи людей и детерминация когнитивной и мотивационно-ценностной сфер их сознания. Она стимулирует распространение не только новых типов экономической деятельности, технологии, информации, образа жизни, но и все более разнородных культурных моделей, жизненных слоев, мотиваций и ценностных ориентаций личности. Изучение механизмов влияния глобализации на духовную и интеллектуальную составляющую культуры показало, что трансформация интеллектуальной компоненты культуры происходит в сторону ее межкультурного нивелирования, а духовной, наоборот, в сторону развития локальных своеобразий с учетом современной реальности [2].

Устойчивый мир творится в диалоге цивилизаций как общее пространство многогранной духовности. Если человечество когда-нибудь построит глобальную устойчивую цивилизацию на планете Земля, то она будет результатом синергии технологических достижений и духовного опыта [3].

Концепция глобальной цивилизации – это стратегия человечества, осознающего себя как единое целое. Она задает новый гуманистический импульс, качественно отличный от технократической стратегии информационного общества, и вектор этого импульса совпадает с духовным императивом устойчивого развития [4].

Список использованных источников:

1. Гор Эл. Земля на чаше весов. Экология и человеческий дух. М.: ППП., 1993.
2. Аванесова Е.А. Синергетические аспекты межкультурного взаимодействия в условиях глобализации // Синергетическая парадигма: Человек и общество в условиях нестабильности. М.: Прогресс – Традиция, 2003.
3. Моисеев Н.Н. Информационное общество: возможности и реальность // Полис. 1993. № 3.
4. Мантатова Л.В. Философские перспективы устойчивого развития информационного общества. Улан-Удэ: Изд-во БГУ, 2002.

СВОБОДА ВОЛИ КАК ПРЕДМЕТ ФИЛОСОФСКОЙ РЕФЛЕКСИИ

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

Матюшонак А.Д.

Ратникова И. М. – преподаватель

Свобода воли – понятие, не имеющее однозначного определения в современном социально-гуманитарном знании. К основным подходам интерпретации данного понятия можно отнести следующие: детерминизм, компатибилизм и либертарианство. Так, ключевая идея детерминизма состоит в том, что все подчиняется законам природы, а, значит, всё будущее предопределено предшествующими причинами. Одним из аргументов, предназначенным подтвердить утверждение о предопределённости наших решений, является эксперимент, проведённый исследователем Калифорнийского университета Б. Либетом. Перед испытуемыми ставилась задача сгибать кисть руки в любой момент, когда они этого захотят. По результатам эксперимента, выяснилось, что активность мозга появилась раньше, чем испытуемые зафиксировали принятие решения. Однако этот аргумент вызывает определенную долю сомнения, поскольку и осознанность наших действий влияет на общую нейронную активность мозга. В тоже время, наше сознание есть функция мозга, а, согласно детерминизму, всё материальное жестко подчинено внешней причинности. Так являются ли свобода воли и детерминизм взаимоисключающими противоположностями?

Позиция компатибилизма обосновывает совместимость свободы воли и каузальности. По мнению шотландского философа Д. Юма, причиной всякой активности человека выступают его мотивы. Мотивы позволяют объяснить, рационализировать действие. Рациональное действие, в свою очередь, всегда обусловлено чем-либо, т.е. детерминировано. Важно то, чем обусловлено всякое действие – нашими мотивами. Однако не всегда мотивы совпадают с нашими желаниями. Согласно великому немецкому философу И. Канту, свобода воли это возможность действовать в соответствии с тем, что наш разум признаёт практически необходимым, подавляя при этом наши эгоистические желания. Заядлый курильщик, может иметь множество мотивов для того, чтобы бросить курить, но при этом он имеет желание выкурить сигарету. То есть мы обладаем свободой воли лишь до тех пор, пока наши мотивы способны подавить наши желания.

Концепция либертарианства утверждает, что детерминизм в корне неверен. Эта позиция хорошо представлена концепцией свободы воли американского философа Р. Кейна. Для опровержения детерминизма он опирается на законы квантовой механики, где события, происходящие на микроуровне можно предсказать лишь с определённой долей вероятности. А свобода воли, по его мнению, сопряжена, в первую очередь, со способностью человека к самоидентификации и самореализации в соответствии со своими внутренними мотивами. Во время ситуаций трудного выбора, когда мы не можем определиться, как поступить, у нас возникают противоборствующие мотивы, и неизвестно какая из частей нашей личности победит. Нашу личность, в конечном счёте, определяют наши поступки, а личность нашу волю, т.е. мы сами себя определяем.

Если человек совершает какое-то действие не по своей воле, то тогда вероятно его нельзя осуждать за это. Моральная же ответственность полностью определяется свободой воли. Социобиологом Р. Докинзом убедительно представлена концепция «взаимного альтруизма», согласно которой эта связь имеет обратный характер. Для того, чтобы повысить вероятность своего выживания, людям нужно было объединяться. В целях сохранения этих объединений формировались различные нормы и правила, которые были направлены на поддержание всеобщего блага. Каждый должен был нести ответственность за свои действия. Соответственно, чтобы винить человека за неверные поступки необходимо предположить наличие возможности поступить иначе, а, значит, свободы воли. Согласно этой позиции, именно свобода воли есть необходимое условие моральной ответственности. Другими словами, свобода воли есть исторически сложившееся понятие, обусловленное эволюционными процессами.

Таким образом, ответ на вопрос о существовании свободы воли зависит от того, какого подхода к интерпретации данного понятия мы придерживаемся. Наше сознание является функцией мозга, а, значит, идеальные по своей природе процессы действительны. Именно свобода воли наполняет наши поступки моральным содержанием, обеспечивая тем самым возможность мирного сосуществования людей в обществе.

ПОНЯТИЕ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА И ПРОБЛЕМА ЕГО ОСМЫСЛЕНИЯ

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

Смирнова З.Д.

Дисько-Шуман М.Р. – к.ф.н., доцент

Ещё в 1950-м году Алан Тьюринг в своей работе задал вопрос, «может ли машина мыслить», тем самым обозначив проблему «умных машин», или искусственного интеллекта (ИИ). И по сей день ведутся исследования и споры, можно ли говорить о воссоздании человеческого разума в искусственной системе.

К середине XX века, считающегося моментом зарождением научного направления, связанного с искусственным интеллектом, уже появилось достаточно предпосылок: в философской среде поднимался вопрос о природе человека и процессе познания мира, в медицинском кругу выдвинули ряд предположений относительно работы мозга человека, математики искали способ оптимизации вычислений и формализации знаний; наконец, зародилась теория алгоритмов и были созданы первые компьютеры [1]. Вычислительные возможности компьютера оказались больше человеческих, поэтому в учёном сообществе был поднят вопрос: где лежит граница возможностей машин и смогут ли они сравняться по уровню развития с людьми?

В 1950 году Алан Тьюринг, пишет статью под названием «Может ли машина мыслить?», где описывает процедуру определения факта, что машина сравнялась по разумности с человеком. Позже её назвали тестом Тьюринга или «игрой в имитацию». Во время теста, экспериментатор-человек задает письменные вопросы двум собеседникам, которых он не видит и не знает, и пытается определить, кто из них человек, а кто – машина [2]. Позже Дж. Серл предложил мысленный эксперимент «Китайская комната», призванный доказать, что прохождение теста Тьюринга не означает возникновение у машины разума. Суть эксперимента заключается в следующем: в комнате находится человек, он берёт плитки с китайскими иероглифами и размещает их напротив других иероглифов в соответствии с инструкцией, тем самым отвечая на заданные ему наблюдателем вопросы, совершенно не зная смысла каждого отдельного иероглифа и не понимая сути заданных вопросов [3]. Несмотря на то, что ответы и вопросы такого диалога верно соответствуют друг другу, в «китайской комнате» не возникает никакого понимания.

Существует несколько подходов к определению понятия искусственного интеллекта. Самый общий из них предполагает, что ИИ будет «вести себя по-человечески». Эту идею можно считать обобщением теста Тьюринга. Писатели научной фантастики часто предлагают ещё один подход, связанный с наличием у машины чувств и способности творить. Так, в «Двухсотлетнем человеке» обладатель андроида Эндрю начинает воспринимать его как человека, когда тот придумывает и создаёт игрушку. В «StarTrek» Дейта помимо способностей к обучению и коммуникации, хочет однажды научиться выражать эмоции [2].

Можно выделить несколько тенденций в исследованиях ИИ. Бесспорно, важным свойством предполагаемого искусственного интеллекта является способность к обучению. Моделирование рассуждений, как одно из направлений, подразумевает создание такой системы, которой будет поставлена некая задача, а после машинного анализа требуется получить её решение. Стоит выделить из этого направления две подобласти: машинное обучение, связанное с самостоятельным получением знаний из имеющихся фактов, и экспертные системы, оперирующие широкой предустановленной базой данных. К подобласти машинного обучения можно отнести задачи на распознавание образов: символов, рукописного текста, речи, то есть обработка естественного (человеческого) языка. Она может использоваться для машинного перевода текстов и поиска информации. Есть и другие примеры использования искусственного интеллекта: программирование ИИ в играх, системы информационной безопасности [1].

В научной фантастике часто затрагивается тема искусственного интеллекта. Взаимодействие человека и ИИ и связанные с этим с социально-психологические проблемы рассматриваются во многих работах Ф. К. Дика, в том числе книге «Снятся ли андроидам электроовцы?». В произведениях «Ghost in the Shell» поднимается вопрос значимости души у человека и роботов. Предполагается, что люди обладают «призраком», и даже если некоторые части их тел заменить механическими (в том числе и мозг), они не перестают быть людьми. Однако робот, никогда не бывший человеком, то есть не имевший «призрака», человеком не станет. Страх людей уступить другой разумной форме жизни реализуется в фильмах «Терминатор», где повествуется об искусственном интеллекте SkyNet, запустившем в людей их же ядерное оружие. Впрочем, здесь показан и взаимный страх машины быть отключенной в случае неповиновения. В книге «Недетские игры» военному суперкомпьютеру, решившему запустить ядерные ракеты, предлагают сыграть в «крестики-нолики» с самим собой. Поскольку он сам для себя стал идеальным соперником и может предсказать результат каждой партии, он приходит к выводу, что в ядерной войне проиграют все [2].

Список использованных источников:

1. Г. С. Осипов. Искусственный интеллект: состояние исследований и взгляд в будущее // Российская ассоциация искусственного интеллекта [Электронный ресурс] – 2016. – Режим доступа: <http://www.raai.org> – Дата доступа: 22.03.2018.

2. Хель И. Алан Тьюринг и философские проблемы искусственного интеллекта // Новости информационных технологий [Электронный ресурс] – 2013. – Режим доступа: <https://hi-news.ru> – Дата доступа: 23.03.2018.

3. Сычев А. А. «Гильотина Юма» в контексте институционального подхода Дж. Р. Сёрля // Этическая мысль. 2012. № 12. С. 143—156.

ЭТИЧЕСКАЯ ПРОБЛЕМАТИКА НЕЙРОМАРКЕТИНГА

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

Дершень В.В.

Ермолович Д.В. – к.ф.н., доцент

В последние годы многие науки начали применять техники нейровизуализации в своих исследованиях и таким образом сделали поразительные успехи в понимании работы мозга и сознания. К их числу можно отнести и маркетинг, несмотря на то, что использование методов нейровизуализации в маркетинговых исследованиях вызывает множество вопросов и споров.

Нейромаркетинг – новое направление маркетинговых исследований, предметом которого является изучение неосознанных сенсомоторных, когнитивных и эмоциональных реакций человека на определенные стимулы [1]. В исследованиях используются такие инструменты, как измерение гальванической реакции кожи, функциональная магнитно-резонансная томография, электроэнцефалография и др. Такие исследования позволяют маркетологам понять, что движет потребителями и как повлиять на их решения.

Важнейшей проблемой исследований в области нейромаркетинга является этическая проблема нейровизуализации в целях повышения коммерческой выгоды. С научной точки зрения, нейромаркетинг не может позволить исследователям разработать такую маркетинговую кампанию, которая ограничивала бы свободу индивидуума. Однако многие высказывают опасения в связи с данной проблематикой [2]. Так некоторые ученые-нейробиологи отстаивают позицию, что «визуализация мозга будет использоваться таким образом, чтобы ущемлять личную неприкосновенность частной жизни в совершенно неприемлемой степени» [3]. В Соединенных Штатах Америки группа защиты потребителей, Consumer Alert, подала жалобы в университеты, федеральное правительство США, а также в сенатский комитет США, высказывая свое мнение о неэтичности исследований подобного рода.

К этическим проблемам нейромаркетинга можно отнести:

- умалчивание основных целей эксперимента (исследователи вмешиваются в частные мысли потребителя, которые он не может скрыть, и используют их в своих целях);
- дискриминацию отдельных лиц посредством влияния на психофизиологические уязвимые точки потребителя;
- использование отдаленных образов в рекламе или дизайне товара, которые не соответствуют его сущности;
- отсутствие средств надзора и контроля нейромаркетинговых исследований;
- основную нацеленность производителя на реализацию своей продукции через использование данных, полученных недобросовестным путем при нейромаркетинговом исследовании [4].

Для решения этих проблем необходимо определить круг возможных действий, связанных с разработкой этической составляющей новейших нейротехнологий в области маркетинга. Эта деятельность должна препятствовать возможности нанесения потенциального вреда респонденту, а также формировать стремление профессионального маркетингового сообщества к интеграционным взаимодействиям с нейронауками и укреплению доверия потребителей к подобного рода исследованиям. Создание своеобразного «этического кодекса» позволит предупредить распространение неэтичного отношения к потребителям и предотвратит возможность бесконтрольно распоряжаться информацией, полученной в ходе маркетинговых исследований [5].

Таковыми вопросами сегодня занимается «нейроэтика» – сравнительно новая междисциплинарная область, в которой обсуждаются разнообразнейшие вопросы, возникшие вместе с современными исследованиями мозга. В ее рамках рассматриваются основные принципы нормативной этики нейрофизиологии, а также таких понятий классической этики, как свобода воли, моральная ответственность, автономия личности и индивидуума [4].

«Нейроэтика» ищет не только свое место в теории, но и находит применение на практике. Так, чтобы убедить клиентов и публику в этичности своих методов и отказать от использования техники подсознательного внушения, ряд американских компаний, занимающихся нейромаркетинговыми исследованиями, принял в 2010 году резолюцию о «нейростандартах». Ее поддержали Gallup & Robinson, Innerscope, Mindlab и некоторые другие участники рынка. В рамках этой программы участники назначают группу экспертов, которая независимо оценивает используемые в исследованиях методы и их этичность [6].

Список использованных источников:

1. Нейромаркетинг >> Маркетинг [Электронный ресурс]. – 2010. – Режим доступа: <http://marketopedia.ru/107-nejromarketing.html>.
2. Madan, C. R. Neuromarketing: the next step in market research? //Eureka. – 2010. – Т. 1. – №. 1. – С. 34-42.
3. Brammer, M. Brainscam? //NatureNeuroscience. – 2004. – Т. 7. – №. 10. – С. 1015.
4. Этические проблемы нейромаркетинговых исследований в процессе управления потребительским спросом [Электронный ресурс]. – 2010. – Режим доступа: http://www.economyandsociety.in.ua/journal/13_ukr/55.pdf.
5. Чернова, М. А., Клепиков, О. Е. Нейромаркетинг: к вопросу об этической составляющей // Национальный психологический журнал. – 2012. – №1.
6. Коновалов, А. Секретная кнопка // Бизнес-журнал. – 2014. – №3 (216).

ПРАБЛЕМА ІНТЭРПРЭТАЦЫІ ТЭРМІНА “УСТОЙЛІВАЕ РАЗВІЦЦЁ” У СУЧАСНЫМ БЕЛАРУСКІМ МЫСЛЕННІ

*Беларускі дзяржаўны ўніверсітэт інфарматыкі і радыёэлектронікі
г. Мінск, Рэспубліка Беларусь*

Лось Г.А.

Ермаловіч Дз.В. – к.ф.н., дацэнт

У сучасным беларускім мысленні ўслед за міжнароднай супольнасцю распаўсюджваюцца ідэі ўстойлівага развіцця. На дзяржаўным узроўні актыўна прапагандуецца нацыянальная стратэгія ўстойлівага развіцця, але само паняцце для шырокіх пластоў насельніцтва не раскрыта. Без разумення самаго тэрміна ўстойлівага развіцця і зацвярджэння яго трактоўкі, зразумелай усім, казаць пра пошук шляхоў рашэння міжнародных глабальных крызісаў не прыходзіцца.

На дадзены момант паняцце ўстойлівага развіцця ў беларускім асяроддзі ў большай ступені ўспрымаецца з пункту гледжання тэрміна ўстойлівасці, а згодна тлумачальнаму слоўніку “ўстойлівы” – пастаянны, які не падаецца ваганням. Адпаведна гэтаму з фармуліроўкі тэрміна на рускай і беларускай мовах можна палічыць, што ўстойлівае развіццё – імкненне да пастаянства.

Існуе шэраг тэорый для разумення канцэпцыі і паняцця ўстойлівага развіцця.

Згодна Міжнароднай камісіі па навакольнаму асяроддзю і развіццю пад устойлівым развіццём разумеецца такое развіццё, якое задавальняла бы патрэбы сучаснага перыяду часу, але не ставіла бы пад пагрозу здольнасць будучага часу задавальняць свае патрэбы [1, с.110].

Паняцце ўстойлівае развіццё пайшло ад англійскага “sustainable”, што можа перакладацца яшчэ і як “сбалансаванае”, “якое падтрымліваецца самастойна”. Такой інтэрпрытацыяй скарысталіся ў Польшчы і “sustainable” пераклалі як “zrównoważony”, пазіцыянучы ўстойлівае развіццё з пункту гледжання сбалансаванага развіцця.

Адзін з падыходаў да разумення ўстойлівага развіцця заснаваны на дыялектычным законе пераходу колькасных характарыстык у якасныя і наадварот і выкарыстоўваецца ў цыклах Кандрацьева, Глазьева; у дадзеным кантэксце цыклічнасць – форма прагрэсіўнага развіцця, не стаяння на месцы.

У яшчэ адной канцэпцыі постіндустрыялізм і ўстойлівае развіццё разглядаюцца як крыніцы масавага спажывання. На першы план выходзяць манапольная, планавая, таварна-размеркавальныя эканомікі.

Член-карэспандэнт НАН Беларусі П. А. Вадап’янаў гаворыць пра тое, што ўстойлівае развіццё павінна прадугледжваць магчымасць кіраваць цяперашнім з будучыні, неабходна нораўнае кіраванне будучыняй, а тэрмін ўстойлівае развіццё разглядае з пазіцыяй “дастатковасці” і “сбалансаванасці”.

Доктар філасофскіх навук Э. М. Сарока сцвярджае, што ўстойлівае развіццё па азначэнню немагчыма, любое развіццё ёсць вагальны працэс, і адбываецца па экспаненце. У яго артыкулах сустракаецца тэрмін дынамічнае ўстойлівае развіццё.

Аналіз літаратурных крыніц паказвае, што праблема інтэрпрэтацыі ўстойлівага развіцця на сённяшні дзень знаходзіцца ў цэнтры ўвагі вучоных і дасюль з’яўляецца неадназначнай, недастаткова распрацаванай і актуальнай як на міжнароднай арэне, так і ў беларускім навуковым асяроддзі.

Ва ўмовах нарастаючай патрэбы разумення ўстойлівага развіцця беларусы маглі бы звярнуцца да магчымасцяў беларускай мовы і па аналогіі з палякамі выбраць больш адпаведны тэрмін – “ураўнаважае развіццё”, што будзе сінанімічна замежнаму слову “сбалансаваны”. Такім чынам, можна пазбавіцца праблемы неадназначнасці трактоўкі слова “ўстойлівы”.

Актуальным з’яўляецца і пытанне пераадолення крызісаў пры дапамозе ўстойлівага (ураўнаважанага) развіцця. Згодна В. С. Стёпіну крызісы бываюць двух відаў: антрапалагічны і экалагічны, астатнія з’яўляюцца вытворнымі ад іх. Рэзкае абвастрэнне гэтых крызісаў ставіць пытанне ад пошуку новых стратэгий развіцця. Такія стратэгіі могуць прывесці да новага тыпу цывілізацыйнага развіцця [2, с.353]. А гэта ў сваю чаргу прадугледжвае фарміраванне новых каштоўнасцяў, якія адпавядаюць ідэалу захавання і сбалансаванага развіцця біясферы і чалавецтва.

У такім выпадку канцэпцыю ўраўнаважанага развіцця можна было бы трактаваць як дасягненне разумнай сбалансаванасці сацыяльна-эканамічнага развіцця чалавецтва і захавання навакольнага асяроддзя.

З гэтага пункту гледжання можна зрабіць выснову, што гэты напрамак сацыяльна-экалагічнай палітыкі павінен быць выкарыстаны для распрацоўкі мадэлі ўстойлівага развіцця Рэспублікі Беларусь.

Спіс выкарыстаных крыніц:

1. Коряков, А. Г. Методологические вопросы устойчивого развития предприятий / А. Г. Коряков // Вопросы экономики и права. – 2012. – №4. – С.110-114.

2. Стёпин, В. С. Кризис современной цивилизации и проблема объединяющих ценностей / В. С. Стёпин // **Национальная философия в глобальном мире** : тезисы Первого белорусского философского конгресса / Национальная академия наук Беларуси, Институт философии ; редкол.: В. Г. Гусаков (пред.) [и др.]. – Минск : Беларуская навука, 2017. – 765 с.

СОЦИАЛЬНО-ЭКОЛОГИЧЕСКАЯ КУЛЬТУРА БЕЛАРУСИ ПОСЛЕ АВАРИИ НА ЧЕРНОБЫЛЬСКОЙ АЭС

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

Гурская И.А.

Дисько-Шуман М.Р., к.ф.н., доцент

Последствия аварии на Чернобыльской атомной электростанции (ЧАЭС) с 1986 года по настоящий момент ставят перед белорусским обществом серьёзные задачи: актуализируются проблемы грамотного статистического учёта данных и их корректной интерпретации, проблемы влияния Чернобыля на здоровье белорусов, различные проблемы социального и экологического характера. Феномен Чернобыля присутствует во многих отраслях современной науки и постоянно предпринимаются попытки дать грамотную оценку последствиям данной аварии: социальным, правовым, медицинским, философским, политическим. При попытке дать оценку (культурную, социальную, экологическую, правовую и др.) по влиянию Чернобыля для различных социальных групп необходимо учитывать различные факторы, которые привели к формированию уникального информационного пространства вокруг темы Чернобыля.

Концепция национальной стратегии устойчивого социально-экономического развития Республики Беларусь до 2030 года в числе главных приоритетов определяет развитие человеческого потенциала и рост качества жизни населения страны. Решающими условиями успешного осуществления данной стратегии являются: поддержание надлежащего качества природных систем, расширение сферы «зеленой экономики», обеспечение экологической безопасности, системная работа по улучшению материальных и нематериальных компонентов благосостояния личности и общества.

Главными современными проблемами в белорусской философии, касающиеся социально-экологических исследований и исследований, посвящённых Чернобыльской катастрофе и оценке её последствий являются следующие:

- 1) проблема идеалов после Чернобыля (в самом широком смысле: идеал современного человека, идеал чистой природы, идеал безопасного предприятия и др.);
- 2) проблема критериев оценки «устойчивого развития» после Чернобыля (какое именно развитие Чернобыльских территорий считать устойчивым в рамках концепции Устойчивого развития);
- 3) проблема формирования экологической культуры, как необходимого атрибута современного человека.

Следует также отметить, что в современном белорусском обществе существуют программы реабилитации районов, пострадавших от аварии на ЧАЭС. В основном реабилитация касается сельскохозяйственной деятельности: внесения удобрений в почвы, внесение кормовых добавок в корм сельскохозяйственным животным, радиологический контроль продукции, выращиваемой на загрязнённых территориях и т.п. Однако многие социальные проблемы, проблемы психологического здоровья населения на пострадавших территориях остаются за пределами данных программ реабилитации. Программы же социальной реабилитации пострадавших от аварии на ЧАЭС районов в Беларуси строятся в большей степени на идеологической, нежели на научной основе – и эта тенденция должна быть изменена.

Список использованных источников:

1. Экологические, медико-биологические и социально-экономические последствия катастрофы на ЧАЭС в Беларуси под ред. акад. Е. Ф. Конопки, проф. И. В. Ролевича. – Минск: Министерство по чрезвычайным ситуациям и защите населения от последствий катастрофы на Чернобыльской АЭС Республики Беларусь, Институт радиобиологии Академии наук Беларуси, 1996. – 280 с.
2. Чернобыльский форум. Наследие Чернобыля: медицинские, экологические и социальные последствия – доклад МАГАТЭ при содействии ВОЗ, ПРООН, ФАО, ЮНЕП и др. Радиация и риск. Спецвыпуск 2, 2005.
3. Новости Совета Министров Республики Беларусь «Об отнесении земельных участков радиационно-опасных земель отчуждения к радиационно-опасным землям ограниченного хозяйственного использования» Режим доступа: <http://www.government.by/ru/content/7770> Дата доступа – 15.01.2018
5. Яблоков А.В. Миф о безопасности малых доз радиации. Москва 2002
6. Переслегин С.Б. Мифы Чернобыля 2006 г.
8. Шевцов Ю. В. Объединённая нация. Феномен Белоруссии 2005 г.
9. Борисевич Н.Я информационно-психологическая безопасность населения в условиях радиоактивного загрязнения территорий. / Борисевич Н.Я, Горанская Е.И., статья от 20.10.2014., Режим доступа в базе статей cyberleninka.ru Дата доступа – 20.03.2018.
10. Водопьянов П.А. Социально-экологическая проблематика в концептуальной структуре отечественной философии / П. А. Водопьянов, Н. Е. Захарова, А. С. Червинский – Философские исследования. Сборник научных трудов. Выпуск 4. Минск, 2017.
11. В.А. Ремизов Гуманитарный и антигуманитарный след Чернобыля/ В.А. Ремизов, В.С. Садовская – Вестник МГУИКИ 2016 №1 (69).

**СЕКЦИЯ «ВСТРАИВАЕМЫЕ ВЫЧИСЛИТЕЛЬНЫЕ
СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ»**

АЛГОРИТМ ОБУЧЕНИЯ МНОГОСЛОЙНОГО НЕЙРОСЕТЕВОГО КВАНТОВАТЕЛЯ АУДИОКОДЕРА

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

Аврамов В.В.

Петровский А.А. – д.т.н., профессор

Сжатие звуковой информации является актуальной задачей в современном мире, поскольку активное развитие и внедрение получают такие технологии как передача аудиоинформации по коммуникационным каналам (VoIP, VoLTE), потоковое вещание мультимедиа (Streaming Media), цифровое радиовещание (DAB). В каждой из перечисленных технологий одной из наиболее важных задач является компактное представление цифрового звукового сигнала. В любом алгоритме сжатия звука важнейшим шагом процесса кодирования является алгоритм квантования данных.

Результатом работы аудиокодера [1] является набор наиболее перцептуально важных для восприятия человеком параметров (атомов). Структура каждого атома представляется как его вес (вещественное число), и два целочисленных параметра характеризующие позицию атома в дереве реконструкции пакетного дискретного вейвлет преобразования декодера. Позиции атомов эффективно кодируются с использованием энтропийного кодирования Хаффмана, в то время как вес каждого атома должен быть квантован для компактного представления и передачи декодеру.

Нейросетевое квантование (NNQ – NeuralNetworkQuantization) представляет собой совместное квантование вектора параметров, представленных вещественными числами, в некоторый дискретный набор. Процесс NNQ устраняет избыточность благодаря эффективному использованию взаимосвязанных свойств векторных параметров. Таким образом, данный подход предлагает несколько уникальных преимуществ по сравнению с скалярным квантованием, включая возможность использования линейных и нелинейных зависимостей между векторными компонентами и гибкость в выборе многомерных форм ячеек квантователя. Определение NNQ формулируется следующим образом: NNQ размера M является отображением в N -мерный кодовый вектор Y содержащий K дискретных выходных значений.

Исходя из вышеприведенного определения, естественной архитектурой искусственной нейронной сети (ИНС) для реализации квантователя является симметричная сеть прямого распространения (АЕ - autoencoder), содержащая входной, кодовый и выходной слои. Основной задачей обучения АЕ является получение на выходе вектора с минимальным отклонением от входного. Описанная однослойная сеть весьма ограничена по своим вычислительным возможностям, особенно в задаче квантования, предполагающей дискретные ограничения на выходы кодового слоя. Исходя из этого, очевидно, что одного скрытого слоя будет недостаточно, и выбор следует сделать в пользу многослойной архитектуры (DAE – deepautoencoder).

Архитектура DAENNQ была экспериментально выбрана как показано на рисунке 1. Весовые коэффициенты DAE были предварительно обучены тремя однослойными АЕ: $\{200 \times 100\}$, $\{100 \times 50\}$, $\{50 \times 20\}$, в соответствии со стратегией жадного послойного обучения [2]. Первый АЕ был обучен на подготовленном тренировочном наборе. Каждый последующий АЕ был обучен с использованием выходов скрытого слоя предыдущего АЕ в качестве входных тренировочных данных. Затем, обученные слои были объединены чтобы сформировать DAE, и дальнейшей тонкой подстройки весовых коэффициентов. Параметры модели обновлялись после каждого набора из случайно выбранных 1000 обучающих примеров с использованием алгоритма стохастической оптимизации Adam [3].

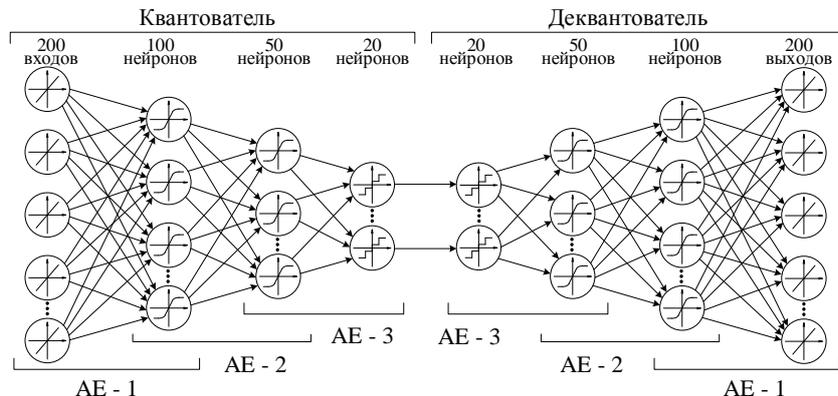


Рис. 1 – Нейросетевой квантователь

Функцией активации первых двух АЕ является гиперболический тангенс. Для получения дискретного представления в центральном кодовом слое DAE в третьем АЕ функцией активации выступает ступенчатая функция активации [4], образованная гиперболическими тангенсами. Данная функция, естественным образом, в процессе обучения, отображает входные векторы с вещественными компонентами в дискретный набор координат и описывается следующим выражением:

где – контролирует угол наклона ступеней, – количество ступеней.

В процессе проведения экспериментов, нейросетевой квантователь был обучен на тренировочной последовательности длительностью около 36 минут, включающей речевые образцы, музыку и другие звуковые сигналы. В качестве тестовой последовательности выступали образцы, описанные в таблице 1.

Табл. 1 – Описание тестовых образцов

Образец	Описание	Образец	Описание
es01	Вокал (Suzan Vega)	si01	Клавесин
es02	Речь на немецком	si02	Кастаньеты
es03	Речь на английском	si03	Камертон
sc01	Соло на трубе	sm01	Волынка
sc02	Оркестр	sm02	Металлофон
sc03	Поп-музыка	sm03	Струнные

Средний бюджет бит для квантованных весов атомов и кодирования положения оценивается следующим образом. Кодовый слой для варианта 200 атомов состоит из 20 нейронов со ступенчатой функцией активации, состоящей из 32 ступеней, что соответствует 5 битам на каждую ступень. Умножение количества нейронов на количество бит на каждый нейрон даст 100-битный бюджет для представления 200 атомов. Таким образом, суммарный средний битрейт составляет около 10 кбит/с, что соответствует эквивалентной степени сжатия около 70. Для экспериментов с другими вариантами числа атомов структура слоев изменяется пропорционально (например, для 300 атомов, кодовый слой состоит из 30 нейронов).

Для оценки качества реконструируемого аудио-сигнала была использована метрика ITU-R Recommendation BS.1387-1 PEAQ (Perceptual Evaluation of Audio Quality). Это метрика используется для оценки перцептуального искажения на основе слуховой модели человеческого уха. Оценка PEAQ- это показатель объективной разности (ODG). Шкала ODG определяется следующим образом: 0 - незаметное ухудшение; -1 - ощутимое, но не раздражающий; -2 - немного раздражает; -3 - раздражает; -4 - очень раздражающее ухудшение. На рисунке 2 показаны результаты оценки качества универсального аудио / речевого кодера в составе с представленным методом квантования.

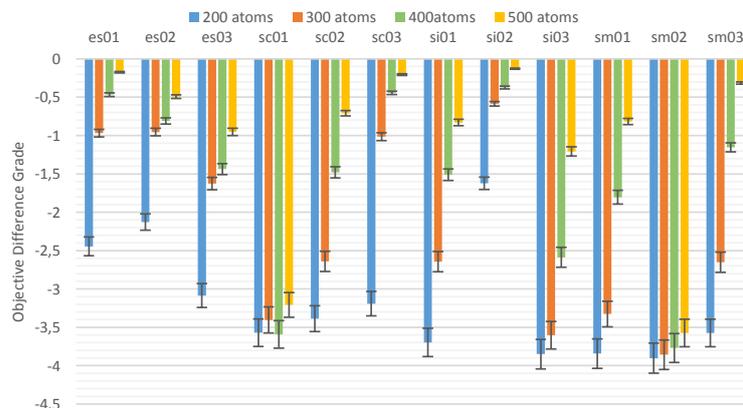


Рис. 2 – Результаты оценки качества реконструированных тестовых образцов

Список использованных источников:

1. Петровский Ал.А., Петровский А.А. Масштабируемые аудиоречевые кодеры на основе адаптивного частотно-временного анализа звуковых сигналов //Труды СПИИРАН, 1(50). – 2017. – с. 56-92.
2. Bengio Y. Popovici, Greedy layerwise training of deep networks / Y. Bengio, P. Lamblin, D. Popovici // NIPS. – 2006. – pp.153–160.
3. Diederik K. Adam: a method for stochastic optimization / K. Diederik, J. Ba // Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015). – 2015.– arXiv preprint: arXiv:1412.6980.
4. Hecht-Nielsen R. Replicator Neural Networks for Universal Optimal Source Coding / R. Hecht-Nielsen // Science, vol. 269, no. 5232. – 1995. – pp. 1860-1863.

ПЕРСОНИФИКАЦИЯ ЗВУЧАНИЯ ПРИ ПОМОЩИ ЦИФРОВОЙ ТОНКОМПЕНСАЦИИ

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

Андреев И.Д.

Азаров И.С. – д.т.н., доцент

В работе реализован эффект цифровой тонкомпенсации. Целью эффекта является повышение разборчивости звука при воспроизведении. Эффект достигается путем обработки сигнала таким образом, чтобы при изменении громкости не происходило искажение «звуковой картины» т.е. сохранялась относительная громкость частотных составляющих. Эффект тонкомпенсации реализован путем декомпозиции сигнала на частотные составляющие и коррекции их интенсивности согласно функции громкости. Функция коррекции интенсивности получена в виде аналитического выражения. Приводятся результаты практических экспериментов.

Известно, что чувствительность слуха различна для звуков разной частоты. Одинаковой субъективной громкости звука на разных частотах соответствуют разные уровни звукового давления. В связи с этим громкость звука оценивают, сравнивая её с громкостью стандартного чистого тона (обычно частотой 1 кГц). Для этого применяется такая величина, как фон. 1 фон (единица уровня громкости звука) – разность уровней громкости двух звуков данной частоты, для которых равные по громкости звуки с частотой 1 кГц отличаются по интенсивности (уровню звукового давления) на 1 дБ.

Любая аудиозапись подразумевает определенный «идеальный» уровень громкости при воспроизведении. Если громкость отличается от этого уровня, то «звуковая картина» искажается и человек слышит больше либо меньше звуков, чем должен. При повышении громкости все частотные составляющие звука становятся громче в том числе шумы и искажения, при понижении часть звука пропадает из восприятия что приводит к ухудшению разборчивости. В обоих случаях меняется субъективная относительная громкость между частотными составляющими звука что влияет на общее восприятие звука, искажается «звуковая картина».

В данной работе мы предлагаем исследовать изменение относительной громкости частотных составляющих звука при изменении громкости и реализовать способ цифровой тонкомпенсации, позволяющий увеличить разборчивость. Основной целью является возможность воспроизведения звука на низкой громкости с сохранением высокого уровня разборчивости без заметных для слуха искажений. Основная идея заключается в сохранении относительной субъективной громкости между различными частотными составляющими звука при уменьшении громкости.

Регулировка громкости в звуковоспроизводящей аппаратуре осуществляется изменением интенсивности всех частотных составляющих на один и тот же коэффициент. Такая регулировка не учитывает особенностей слуха человека и приводит к появлению искажений либо потере разборчивости. Более наглядно проблема показана на рисунке 1.

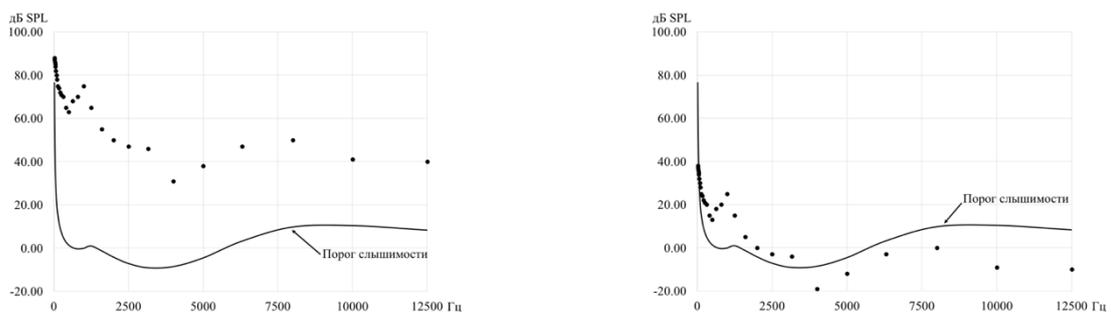


Рисунок 1 – Частотные составляющие фрагмента звукового сигнала при «идеальной» громкости воспроизведения (слева) и при пониженной на 50 дБ громкости (справа)

Точками показаны интенсивности частотных составляющих звукового фрагмента. Черной линией показана кривая порога абсолютной слышимости. Все точки, расположенные ниже этой кривой, не фиксируются человеческим ухом. После уменьшения громкости на 50 дБ некоторые частотные составляющие исчезли из восприятия, т.к. амплитуды этих составляющих оказались ниже порога. Таким образом потерялась часть звуковой информации. С другой стороны, изменилась субъективная относительная громкость между составляющими. Например, разница в субъективной громкости между составляющими с частотой 1 кГц и 5 кГц до понижения составляет 37 дБ, а после понижения разница стала бесконечной, поскольку звуки оказались по разные стороны порога слышимости.

Для решения поставленной задачи необходимо перейти в субъективные единицы измерения громкости. Для оценки громкости международной организацией по стандартизации принята единица сон (sone). 1 сон соответствует громкости чистого тона частотой 1 кГц и уровнем звукового давления 40 дБ. При принятии данной величины был проведен ряд экспериментов на предмет зависимости ощущения громкости

человеком от уровня звукового давления [1, с. 205]. При этом осуществлялся поиск такого инкремента интенсивности звука, при котором ощущение изменяется в 2 раза. Измерения такого рода показали, что уровень звукового давления тона с частотой 1 кГц должен увеличиться на 10 дБ, чтобы увеличить ощущение громкости в два раза. Таким образом, уровень звукового давления 40 дБ должен быть увеличен до 50 дБ, чтобы удвоить громкость, которая при этом станет равна двум сонам. Функция нарастания громкости построена во всем диапазоне уровней на основе экспериментальных измерений.

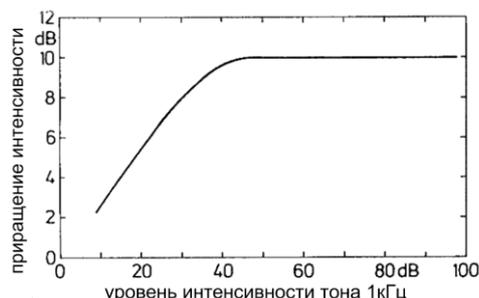


Рисунок 2 – Изменение уровня звукового давления необходимое для изменения ощущения громкости тона частотой 1 кГц вдвое как функция интенсивности

Данные, изображенные на рисунке 2, показывают увеличение уровня, необходимое для создания ощущения вдвое более громкого, чем ощущение от начального уровня. Уменьшение такого же размера будет уменьшать ощущение громкости вдвое. Используя в качестве опорной точки тон в 1 кГц и интенсивность 40 дБ SPL (соответствующую 1 сон) была получена функции коррекции интенсивности (тонкомпенсации):

(1)

где F – громкость в фонах, L – уровень условной субъективной громкости в дБ.

Таким образом, идея реализуемой тонкомпенсации заключается в том, чтобы изменять громкость звука используя одинаковое значение в единицах условной субъективной громкости для каждой частотной составляющей. Изменение уровня звукового давления для каждой частоты в таком случае окажется разным.

Результат обработки формируется на основании применения к каждой частотной составляющей коэффициента, рассчитываемого следующим образом:

Расчет коэффициентов коррекции производится по формуле:

(2)

где K – коэффициент усиления j -й полосы в дБ, ΔL – изменение громкости в дБ.

Для проверки работоспособности полученной модели был проведен следующий эксперимент. Для звуковоспроизводящего оборудования, использованного в эксперименте, был произведен замер уровня порога слышимости. Затем был выбран эталонный уровень громкости (комфортный для слушателя), относительно которого производилось изменение громкости. После этого производилось понижение уровня громкости с коррекцией звука и без нее. Полученные оценки приведены в таблице 1.

Таблица 1 – Оценка разборчивости (MOS)

Изменение громкости, дБ	-3	-6	-9	-15	-20
Разборчивость без коррекции	4,57	4	3,29	2,57	1,86
Разборчивость с коррекцией	4,72	4,43	4,14	3,57	2,86

Под разборчивостью в рамках эксперимента подразумевались полнота ощущения при прослушивании музыкальной композиции, наличие всех музыкальных инструментов в «звуковой картине». Эксперимент показал, что при понижении громкости уменьшение разборчивости без применения коррекции происходит быстрее.

Список использованных источников:

1. Zwicker, E. *Psychoacoustics Facts and Models, 3rd edition* / E. Zwicker, H. Fastl // Springer, 2007. – 463 p.

МЕТОД ШУМООЧИСТКИ РЕЧИ НА ОСНОВЕ ВЫЧИСЛЕНИЯ АВТОКОРРЕЛЯЦИОННОЙ ФУНКЦИИ, СОГЛАСОВАННОЙ С ПЕРИОДОМ ОСНОВНОГО ТОНА

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

Бабок Е.И.

Вашкевич М.И. – к.т.н., доцент

Шум-это любой нежелательный звук, который мешает человеку нормально воспринимать полезные акустические сигналы. Всемирная организация здравоохранения в своих докладах определяет шум, как один из трех основных загрязнителей окружающей среды человека, как воду и воздух. Примерно 20% населения ЕС (около 80 миллионов человек), страдает от повышенного уровня шума, который ученые и эксперты в области здравоохранения считают неприемлемым. Из-за этого условия жизни многих людей изменились, причем последствия варьировались, вплоть до потери слуха.

В соответствии с рассматриваемым подходом к шумоочистке, зашумленный сигнал должен обрабатываться в следующей последовательности:

1. Расчет кратковременной автокорреляционной функции для входного фрейма.
2. Вычисление длины нового фрейма.
3. Выборка, согласованная с периодом основного тона.

Используя свойства кратковременной автокорреляционной функции, мы можем определить, имеет ли взятый фрейм речь. Обработка фрейма выполняется только в том случае, если фрейм сигнала содержит вокализованную речь, что позволяет увеличить скорость вычислений и позволяет пропускать 2-й и 3-й этап обработки.

Определив, что фрейм содержит речь, следующим шагом, мы определяем максимум в рамках периода основного тона, для вычисления расстояния графика кратковременной автокорреляционной функции от начала координат до первого максимума. Период является длиной следующего фрейма обрабатываемого сигнала. В процессе обработки в качестве обработанного сигнала выдается участок автокорреляционной функции. Выборка начинается с точки, в которой график кратковременной автокорреляционной функции примерно равен нулю при переходе из отрицательной части оси в положительную и сдвигаем период до этой точки.

Принцип работы алгоритма, показан на рисунке 1.

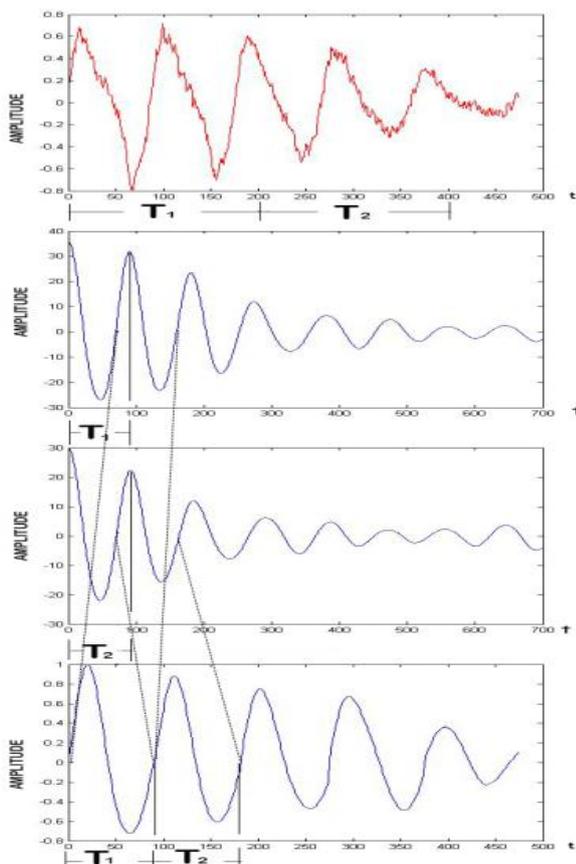


Рис. 1 – Пример работы метода шумоочистки

Автокорреляционная функция показывает взаимосвязь между сигналом и его сдвинутой версией от величины временного сдвига.

Кратковременную автокорреляционную функцию можно рассчитать по выражению представленному ниже:

где n – -ый отсчет фрейма;
 τ – задержка, временной лаг.

Для моделирования выбраны следующие значения параметров: α и β , соответственно.

Для проведения шумочистки, речь была предварительно зашумлена стационарным шумом. Стационарный шум — шум, который характеризуется постоянством средних параметров: мощности, спектральная плотность, автокорреляционной функции. В результате работы модуля, были получены спектрограммы, представленные на рисунке 2.

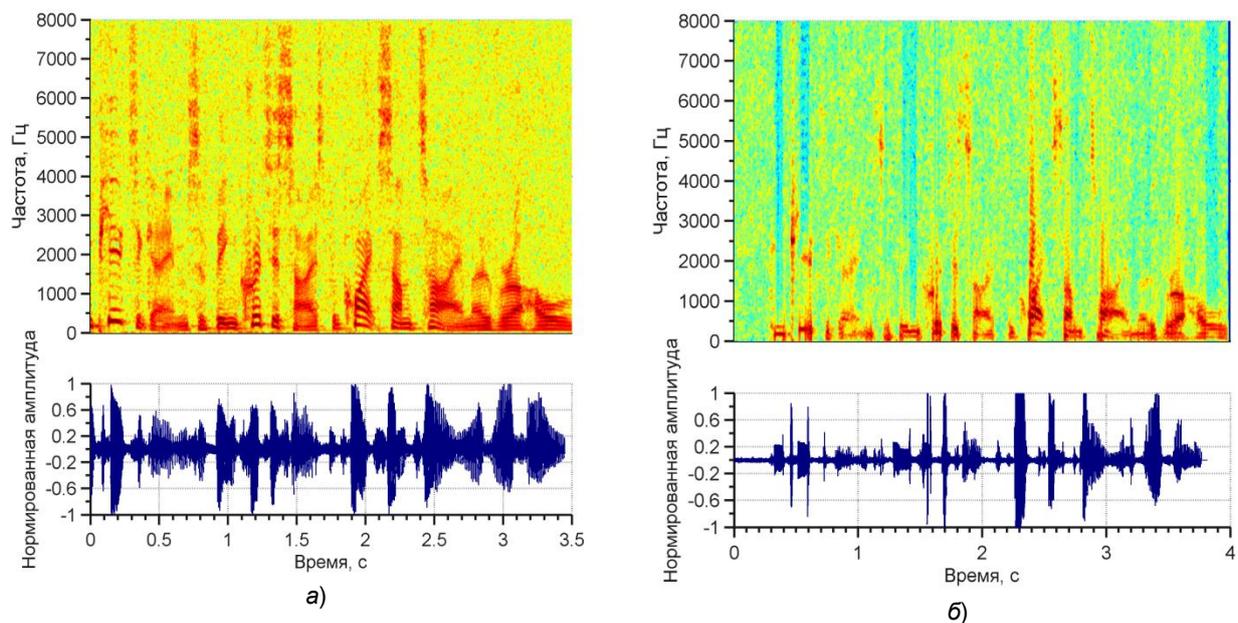


Рис.2 – Пример работы метода шумочистки: а) зашумленный сигнал до обработки; б) результат обработки сигнала

Основным преимуществом данного метода является то, что данный алгоритм очень прост для возможной последующей аппаратной реализации.

Список использованных источников:

1. Suzuki J., Speech processing by splicing of autocorrelation function// Radio Research Laboratories Koganei, Tokyo 184, Japan.
2. R. Fano, Short-time autocorrelation functions and power spectra//The Journal of the Acoustical Society of America 22, 1950.
3. T. Stetzler, N. Magotra, P. Gelabert, P. Kasthuri, S. Bangalore, Low-Power Real-Time Programmable DSP Development Platform for Digital Hearing Aids // Texas Instruments Application report (SPRA657 – April 2000).
4. H.Suzuki, J.Igarashi, Y.Ishii, Eliminating of noise by autocorrelation vocoder// 1972 Autumn Meeting, Acoust. Soc. Japan.

ДЕТЕКТОР РЕЧЕВОЙ АКТИВНОСТИ НА ОСНОВЕ СВЁРТОЧНОЙ НЕЙРОННОЙ СЕТИ С РЕКУРРЕНТНЫМИ СВЯЗЯМИ

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

Вашкевич Г.С.

Азаров И.С. – д.т.н., доцент

В данной работе исследуется задача обнаружения речевой активности в зашумленном звуковом сигнале. Классические подходы к решению данной задачи используют MFCC в качестве базовых признаков, и вычислительно-сложные модели для их классификации, что плохо сказывается на производительности. Мы представляем модель свёрточной нейронной сети с рекуррентными связями использующую информацию о гармонической природе речевого сигнала в качестве базовых признаков. В сравнении с другими нейросетевыми моделями, предложенная модель имеет на несколько порядков меньше настраиваемых параметров, что позволяет ей быстрее обучаться и обладать низкими вычислительными затратами.

В работе [1] нами была предложена модель детектирования речевой активности на основе свёрточной нейронной сети. Текущая работа развивает идею выделения гармонических компонент речевого сигнала при помощи свёрточной нейронной сети. Мы добавляем еще один свёрточный слой с рекуррентными связями, который должен учитывать небольшие изменения тона речевого сигнала во времени.

Как видно на рисунке 1, частота гармонических компонент речевого сигнала (слева) медленно изменяется во времени, в то же время частотные пики шумового сигнала (справа) почти хаотично разбросаны как вдоль частотной, так и вдоль временной оси. Задача нового слоя заключается в том, чтобы сеть смогла обнаружить именно плавные изменения частотных пиков во времени.

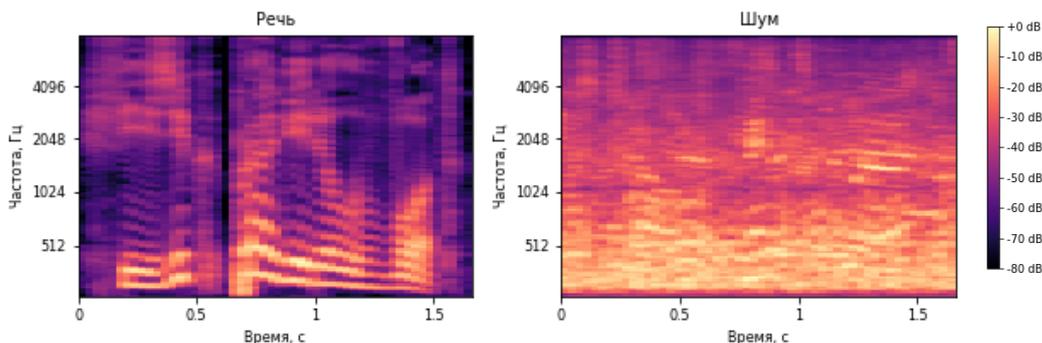


Рис.1 –Спектрограмма речевого и шумового сигналов

На рисунке 2 представлена архитектура усовершенствованной модели нейронной сети. Данная модель отличается лишь тем, что после второго свёрточного слоя базовой модели добавлен еще 1 слой. Данный слой работает следующим образом:

- 1) каждая из SEQ_LEN матриц, размером $N_CAND \times 1$ сворачивается с обучаемым ядром размером $N_CH \times 1$;
- 2) полученная матрица умножается на настраиваемый коэффициент обратной связи и результат умножения суммируется с аккумулятором, который хранит сумму всех предыдущих фреймов;
- 3) к аккумулятору прибавляется настраиваемый параметр смещения, результат пропускается через логистическую функцию активации.

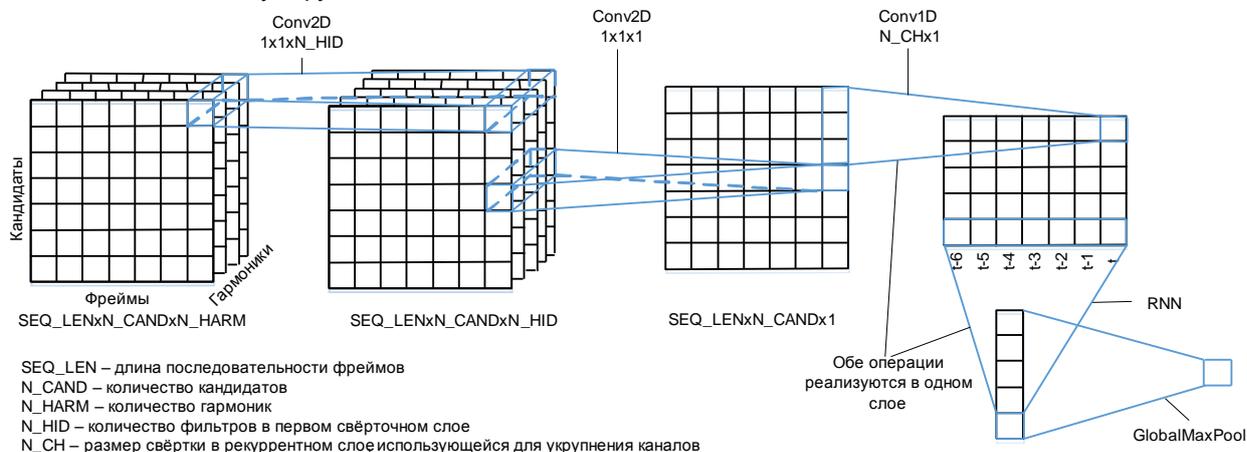


Рис.2 – Архитектура предложенной модели

Результатом работы данного слоя является матрица, размером $N_{CH} \times 1$, максимальное значение которой и будет выходом всей нейронной сети. Т.к. значения выхода сети представляют собой активации логистической функции – их можно интерпретировать как вероятность наличия шума в заданном входном примере. Высокое значение активации свидетельствует о высокой вероятности наличия во входном примере голосовой активности, низкое значение активации, в свою очередь, свидетельствует о наличии во входном примере шума.

Обучение модели проходило на открытом наборе данных Musan [2]. Данный набор данных содержит 3 типа звуковых записей – речь, шум, и музыка. В своих экспериментах мы использовали лишь записи речи и шума. Шумовые записи подмешивались в записи с речью с определенным коэффициентом, который определялся требуемым соотношением сигнал/шум (SNR). Значения SNR (в дБ) выбирались случайным образом из ряда [-10, -5, 0, 5, 10, 15, 20]. Коэффициент k , на который необходимо было умножить шумовой сигнал рассчитывается следующим образом:

$$SNR = 20 * \log_{10} \left(\frac{A}{k * A_n} \right) \Rightarrow k = 10^{\log_{10} \left(\frac{A}{A_n} \right) - \frac{SNR}{20}},$$

где A – мощность речевого сигнала;
 A_n – мощность сигнала с шумом.

Т.к. мощность сигнала – это квадратичная величина, то окончательное выражение для расчёта коэффициента k имеет вид:

$$k = 10^{\frac{\log_{10} \left(\frac{A}{A_n} \right) - \frac{SNR}{20}}{2}}.$$

Значение параметра N_{CH} было принято равным 32. Все остальные значения гиперпараметров по сравнению с базовой работой остались неизменными.

Обучение модели производится в 2 этапа. Необходимо отдельно обучать свёрточную и рекуррентную части всей сети. Свёрточная часть обучается аналогичным образом, как это происходило в базовой работе. При обучении рекуррентной части обучаются только веса рекуррентного слоя, который включает в себя параметры свёртки, укрупняющей каналы, коэффициент обратной связи, и коэффициент смещения.

Общее количество настраиваемых параметров предложенной сети составляет 2083 коэффициента, что является очень малым значением по сравнению с современными архитектурами нейронных сетей.

Результаты тестирования предложенной и базовой модели на одном и том же наборе данных представлен в виде матриц запутанности на рисунке 3.

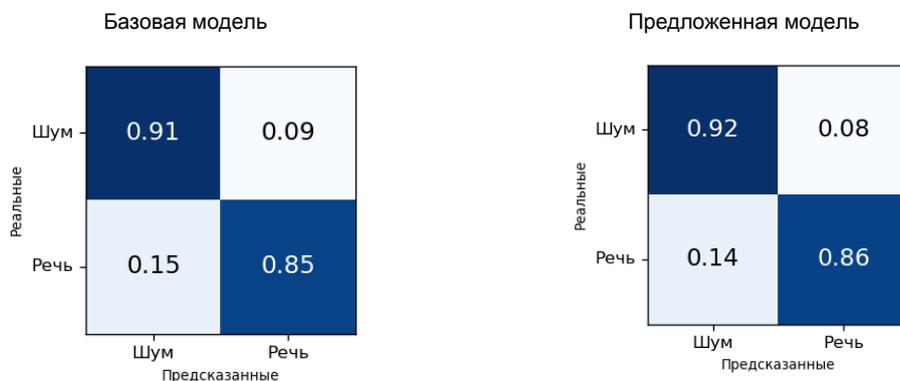


Рис.3 – Результат тестирования моделей

Из матриц запутанности на рисунке 3 видно, что предложенная в данной работе модель обладает немного лучшей точностью детектирования речевого сигнала. При этом по сравнению с базовой, новая модель стала сложнее лишь на 34 настраиваемых параметра, что почти не сказывается на вычислительной сложности.

Список использованных источников:

1. Вашкевич, Г. Свёрточная нейронная сеть с семантически-значимыми активациями для анализа речи / Г. Вашкевич, И. Азаров // Открытые семантические технологии проектирования интеллектуальных систем: материалы международной научно-технической конференции, Минск, 15-17 февраля 2018 г. / Белорус. гос. ун-т информатики и радиоэлектроники ; редкол.: В. В. Голенков (гл. ред.) [и др.]. – Минск, 2018. – С.227.
2. Snyder, D. MUSAN: A Music, Speech, and Noise Corpus // arxiv preprint, arXiv:1510.08484v1. – 2015.

СИСТЕМА ДИАГНОСТИКИ И ВЫЯВЛЕНИЯ НЕВРОЛОГИЧЕСКИХ ЗАБОЛЕВАНИЙ НА ОСНОВЕ АНАЛИЗА РЕЧЕВОГО СИГНАЛА

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

Гвоздович А.Д.

Вашкевич М.И. – к.т.н., доцент

Неврологические заболевания – одна из самых распространённых категорий болезней в мире. Условием эффективного лечения любого неврологического заболевания является его ранняя диагностика. В работе рассматривается вопрос построения системы диагностики и выявления неврологических заболеваний на основе анализа речевого сигнала.

В неврологии существует несколько способов получения информации о болезни:

- сбор анамнеза и регистрация объективных данных;
- неврологические тесты на наличие рефлексов;
- аппаратные исследования (МРТ, ЭЭГ, УЗИ, ЭКГ);
- лабораторные исследования: крови, мазков, люмбальной жидкости, цитология тканей.

Но далеко не у каждого есть возможность посетить специалиста для диагностики неврологического заболевания. В связи с этим было решено разработать алгоритм, который сможет диагностировать и выявить заболевание в домашних условиях.

В ходе изучения различных возможностей диагностики заболевания замечен самый доступный для обычного человека способ: диагностика и выявление неврологических заболеваний на основе анализа речевого сигнала.

Неврологические заболевания влияют на работу органов, отвечающих за формирование речи. В зависимости от недуга, его степени и возраста пациента возникают различные речевые патологии. Во всех неврологических заболеваниях есть такая патология, как невозможность выдержать одинаковую паузу между словами в речи. Также есть такие патологии, как слабость в голосе, хрипкость, дисфония и т.д. Данные патологии можно выявить путем исследования речевого сигнала и проанализировав найденные патологии можно сделать вывод о наличии или отсутствии неврологического заболевания.

Большинство подходов к автоматической диагностике по речевому сигналу основаны на использовании большого набора признаков, извлеченных из речевого сигнала (дискретное преобразование Фурье, автокорреляция, джиттер, шиммер, отношение "гармоники/шум" и т.д.). На основе этих данных обучается нейронная сеть, которая учится классифицировать речевой сигнал как нормальный или патологический. Однако такой подход имеет два недостатка. Во-первых, существует риск, что система будет слишком сильно оптимизирована для учебных данных, что ограничит её способность к обобщению. Во-вторых, данный подход требует больших вычислительных ресурсов, что не позволяет его использовать в портативных устройствах [2].

Цель данной работы состоит в том, чтобы сделать достаточно точную систему, которая не будет требовать больших вычислительных ресурсов.

Неврологические заболевания влияют на тембр голоса, разборчивость и темп речи, то для их выявления целесообразно анализировать спектральные огибающие гласных звуков и паузы между словами. Для анализа брались речевые сигналы с записью счета от 1 до 10, которые обрабатывались по следующей схеме (рис. 1).

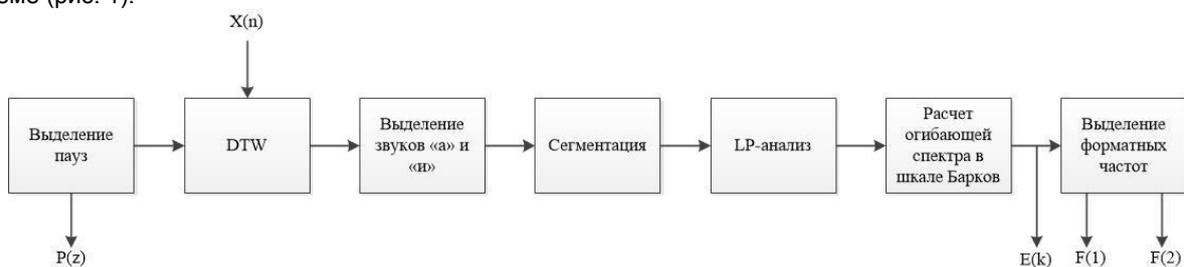


Рисунок 1– Схема анализа речевого сигнала

В блоке DTW (dynamic timewarping – динамическая трансформация временной шкалы) речевой сигнал сравнивается с размеченным эталоном [3]. Далее уже из размеченного речевого сигнала выделяются паузы $P(z)$ и звуки «а» и «и». В блоке сегментации речевой сигнал разбивается на кадры длительностью 27 мс с перекрытием в 1/8 от длительности кадра. Каждый кадр сигнала анализируется методом линейного предсказания (англ. LP – Linear Prediction), а коэффициенты линейного предсказания используются для получения огибающей спектра $E(f)$, которая затем переводится в психоакустическую частотную шкалу барков $E(k)$. В последнем блоке выделяются формантные частоты $F(1)$ и $F(2)$ [4].

Для количественной оценки степени нарушения взаимной формантной структуры звуков /а/ и /и/ предлагается использовать меру

Помимо изменений во взаимной формантной структуре звуков «а» и «и» у пациентов наблюдалось увеличение сходства между формами огибающих этих звуков. Для количественной оценки различия между огибающими звуков «а» и «и» предлагается использовать l_1 -норму расстояния:

где огибающая звука «и», огибающая звука «а», – число точек в частотной области барков, в которых определена огибающая.

Для оценки пауз предлагается анализировать среднюю паузу суммируя между максимальной и минимальной паузой:

где максимальная по длине пауза, минимальная пауза.

Так как здоровый человек может выдержать паузу при медленном счете, то значение у здорового будет стремиться к 2. В то время как у больного это значение необязательно будет близко к 2.

Ниже представлены распределения плотности вероятности для вычисляемых признаков.

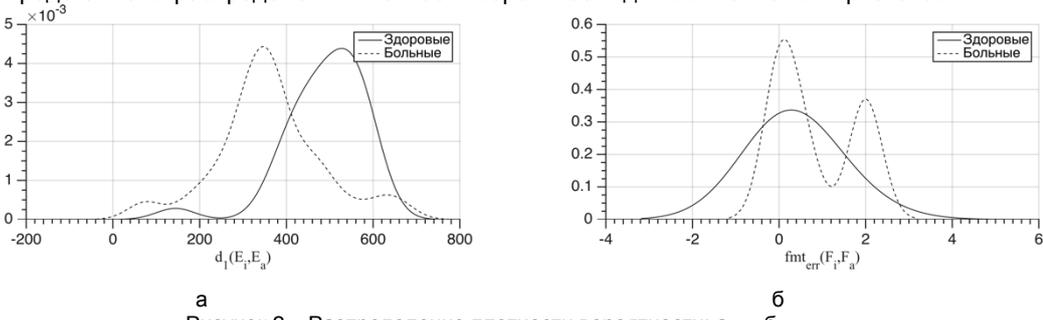


Рисунок 2 – Распределение плотности вероятности: а – , б –

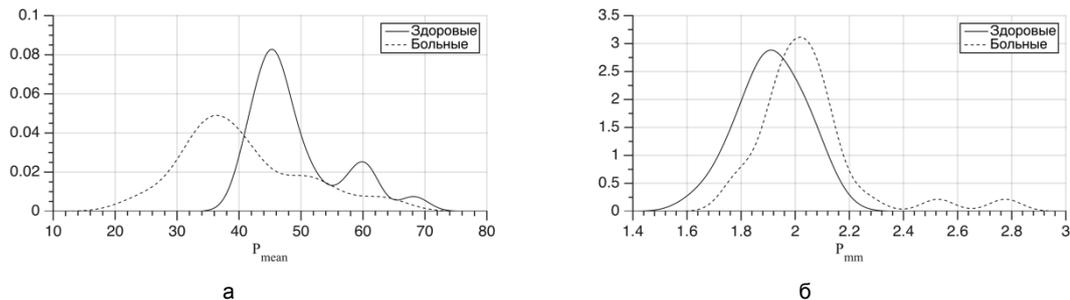


Рисунок 3 – Распределение плотности вероятности: а – , б –

В работе рассмотрена задача выявления и диагностики неврологических заболеваний на основе анализа речевого сигнала. Предложено для решения данной задачи выполнять совместный анализ огибающих различных гласных звуков и анализ пауз. Анализ графиков показывает предлагаемые признаки позволяют выявлять патологические изменения в речи.

Список использованных источников:

1. M. Little Nonlinear, biophysically-informed speech pathology detection / M. Little, P. McSharry, I. Moroz, S. Roberts // IEEE International Conference on Acoustics Speech and Signal Processing Proceedings, May 16 2006, pp. 1080-1083.
2. Thomas Prätzlich, Jonathan Driedger, and Meinard Müller (2016). Memory-Restricted Multiscale Dynamic Time Warping. Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 569–573.
3. Вашкевич М.И. Выделение признаков неврологических заболеваний из речевого сигнала / М.И. Вашкевич, Ю.Н. Рушкевич, И.С. Азаров, А.А. Петровский // Цифровая обработка сигналов и ее применение: труды 20-й междунар. конф., Россия, Москва, 28-30 марта, 2018 г. – Москва, 2018. – Т. 1. – С. 179–184.

ЗАЩИЩЕННЫЙ РАДИОКАНАЛ ДЛЯ ДИСТАНЦИОННОГО УПРАВЛЕНИЯ АВТОМАТИЗИРОВАННЫМИ ОБЪЕКТАМИ

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

Григоришин В.А.

Лихачев Д.С. – к.т.н., доцент

На сегодняшний момент существует множество объектов, как промышленного, так и бытового назначения, в которых присутствует в той или иной степени автоматизация различных процессов. При удаленном контроле и управлении такими автоматизированными объектами используется пакетная передача данных, по радиоканалу или другим системам телекоммуникации, непосредственно с самого автоматизированного объекта на централизованный пульт оператора.

Применение системы автоматизированного контроля и управления объектами с зашифрованным радиоканалом особенно актуально в нефтегазовой промышленности, в системах газораспределения, в газовых котельных, на газовых распределительных подстанциях, в системах контроля и учета электроэнергии, в системах дистанционного управления стратегически важными объектами, узлами, оборудованием, т.е. там, где скорость и последствия принятия решения имеют существенное значение. По этой причине необходимо предусмотреть надежную защиту радиоканала от несанкционированного доступа на любом этапе взаимодействия оператора централизованного пульта управления и удаленного автоматизированного объекта.

Отталкиваясь от сферы применения, радиоканал должен иметь следующие характеристики защиты передаваемой информации:

- надежную защиту от несанкционированного подключения и доступа к удаленному объекту, а также к централизованному пульту оператора [3];
- надежную защиту от несанкционированного перехвата передаваемой информации и дальнейшего использования перехваченной информации для управления и контроля автоматизированным объектом;
- зашифрованный радиоканал должен иметь устойчивую и надежную криптозащиту информации, передаваемой на большие расстояния [2].

В настоящее время распространены следующие методы защиты информации, передаваемой по радиоканалу:

1. Препятствие. Этот метод не удовлетворяет поставленным задачам, так как на физическом уровне ограничить доступ к радиоканалу невозможно.
2. Метод управления доступом. Опираясь на этот метод, можно реализовать одно из предъявленных требований для защищенного радиоканала – обеспечить защиту от несанкционированного подключения, но передаваемая информация вовсе не будет защищена.
3. Маскировка, т.е. защита информации посредством ее криптографического закрытия. Используя этот метод можно реализовать все ранее перечисленные характеристики, которыми должен обладать защищенный радиоканал. Используя секретные алгоритмы шифрования и дешифрования информации, можно обеспечить минимальный риск несанкционированного подключения и перехвата информации, передаваемой по радиоканалу, и сделать невозможным дальнейшее использование перехваченной информации.
4. Регламентация. Этот метод защиты информации не применим к цифровой технике, осуществляющей автоматизированное управление объектами различного назначения.
5. Принуждение. Метод может быть применим к персоналу и сотрудникам обслуживающих систему.
6. Побуждение. Этот метод защиты информации не применим к цифровой технике, осуществляющей автоматизированное управление объектами различного назначения.

Для обеспечения безопасности информации от несанкционированного доступа, передаваемой по радиоканалу, предлагается использовать метод маскирования, так как он в наиболее полном объеме удовлетворяет требованиям поставленной задачи [1].

В методе маскирования предлагается использовать симметричные и асимметричные алгоритмы шифрования информации. В асимметричных системах необходимо применять длинные ключи (512 битов и больше). Длинный ключ хотя и резко увеличивает время шифрования, но позволяет распределять ключи по незащищенным каналам. В симметричных алгоритмах, в которых используются более короткие ключи, шифрование происходит быстрее, но в таких системах сложно обеспечить надежное распределение ключей.

Совместное использование симметричного и асимметричного алгоритмов позволит с минимальными аппаратными затратами реализовать защиту радиоканала и передавать информацию по защищенному радиоканалу в режиме реального времени.

Протокол обмена информацией между автоматизированным объектом и центральным пультом оператора можно описать следующим образом:

- получатель вычисляет открытый и секретный ключи, секретный ключ хранит в тайне, открытый же делает доступным;
- отправитель, используя открытый ключ получателя, зашифровывает сеансовый ключ, который пересылается получателю по незащищенному каналу;
- получатель получает сеансовый ключ и расшифровывает его, используя свой секретный ключ; отправитель зашифровывает сообщение сеансовым ключом и пересылает получателю;

– получатель получает сообщение и расшифровывает его.

Список использованных источников:

1. Водлазский В. Коммерческие системы шифрования: основные алгоритмы и их реализация. Часть 1. – М.: Монитор, 1992. – 14с.
2. Ковалевский В., Максимов В. Криптографические методы. – СПб.: КомпьютерПресс, 1993. – 31с.
3. Мафтик С. Механизмы защиты в сетях ЭВМ. – М.: Мир, 1993.

СИСТЕМА КОРРЕКЦИИ РЕЧИ

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

Демидович В.С.

Лихачев Д.С. – к.т.н., доцент

Важнейшим достижением человека, позволяющее ему использовать общечеловеческий опыт, является речевое общение, которое развивалось на основе трудовой деятельности. Речь – один из важнейших элементов человеческой деятельности, позволяющий человеку узнавать мир, делиться знанием и опытом с другими людьми.

В настоящее время большое количество людей страдает теми или иными расстройствами нарушения речи (нарушение звукопроизношения, темпа речи и др.), и их количество постепенно увеличивается. В логопедии и медицине на сегодняшний день существует большое количество разнообразных методик по коррекции дефектов речи: от лекарственных препаратов, физических и дыхательных упражнений до хирургических операций.

У людей с дефектами устной речи может наблюдаться как одно расстройство речи, так и несколько одновременно, поэтому к данным пациентам необходимо применять комплексное лечение. В настоящее время существуют тренажеры для коррекции речи. Например, АКР «Монолог» для коррекции при заикании. В основном данные тренажеры узконаправленные и используются для коррекции одного расстройства и под контролем специалистов. Поэтому актуальным является вопрос создания комплексной системы коррекции речи.

К нарушениям устной речи относятся:

- 1) *брадилалия* и *тахилалия* (нарушение темпа речи);
- 2) *заикание*;
- 3) *дислалия* (нарушение звукопроизношения);
- 4) *ринолалия* (нарушение тембра голоса);
- 5) *алалия* (отсутствие или недоразвитие речи вследствие органического поражения речевых зон коры головного мозга);
- 6) *афазия* (полная или частичная утрата речи, обусловленная локальными поражениями головного мозга);

При нарушении темпа речи коррекционная работа строится на различных речевых упражнениях. Основные упражнения: *произношение речевого материала различной сложности* (слов, слов, коротких фраз, скороговорок и т. п.), *чтение под отбиваемый рукой такт, под метроном с постепенным ускорением темпа говорения и чтения; прослушивание и воспроизведение речевого материала, записанного в ускоренном темпе; запись слов, слов и т. п.*

При заикании используются следующие виды коррекции: *прямое торможение речевого центра* (замедление речи, ритмизация речи, длительное молчание), *включение двигательной сферы в процесс речеобразования* (синхронизация речи с движением пальцев рук, артикуляционный контроль).

Основные этапы коррекции при дислалии: 1) *формирование первичных произносительных умений и навыков* (подражание, с механической помощью, смешанные); 2) *автоматизация звука и включение его в речь*.

При ринолалии осуществляются следующие виды корректировок: активизация работы артикуляционного аппарата (способы активизации зависят от состояния дефекта); развитие артикуляции звуков; разделение звуков с целью предотвращения нарушения звукового анализа; устранение назальных звуков; нормализация просодики речи.

При алалии упор в корректировке направляется в первую очередь на создание речевой системы: формирование коммуникативного намерения, формирование внутренней программы высказывания, лексическая развёрстка, отбор и организация системы лексико-грамматических средств, грамматическое структурирование.

При афазии методика коррекционной работы подбирается, учитывая, какие участки головного мозга пострадали, а какие функционируют в нормальном режиме. Необходимо учитывать, что восстановление речевого механизма сильно отличается от его формирования, поскольку высшие корковые функции говорящего и пишущего человека организованы иначе, чем у человека, начинающего говорить.

Проанализировав данные заболевания и логопедические методы коррекции речи, для включения в систему коррекции речи были выбраны следующие дефекты речи: нарушение темпа речи, заикание, дислалия. Выбор данных дефектов был обусловлен возможностью автоматизации некоторых методов коррекционной работы.

При нарушениях темпа речи в тренажере планируется использовать речевые упражнения с использованием корректоров темпа. Пользователю будет предложено прослушать фрагмент звукового

сообщения под определенный ритм. После прослушивания пациенту нужно будет повторить данный фрагмент под соответствующие ритмичные удары.

Навязывание внешнего ритма с частотой 80 - 100 ударов в минуту способствует нормализации речи, а использование на этом фоне слога, усиливает процесс нормализации речеобразования.

При заикании предлагается использовать метод *задержки акустической обратной связи (DAF, Delayed auditory feedback)* и *изменение частоты обратной акустической связи (FAF, Frequency-shifted auditory feedback)*.

При использовании DAF голос пациента выводится на наушники с задержкой на доли секунды. Исследования выявили, что задержка в диапазоне 50-75 мс позволяет уменьшить заикание на 60-80%.

Основная идея FAF заключается в сдвиге частоты тона голоса пациента в наушниках по сравнению с его нормальным голосом. Подобно DAF, этот метод немедленно уменьшает заикание на 60-80% при нормальной и быстрой речи, без какой-либо предварительной подготовки и сколь-нибудь значимых усилий. Основное преимущество FAF состоит в том, что этот метод не затрудняет речь в процессе его использования, так как пациент слышит свою речь в реальном времени и без шума.

При нарушении звукопроизношения предлагается в тренажере использовать метод автоматизации звука и постановку его в речь. Процедура по автоматизации звука состоит в голосовых упражнениях со специальными словами (простыми по фонетическому составу), которые не содержат другие нарушенные звуки. Во время проведения тренировок требуется произносить слова, в которых звук расположен в разных частях слова. Первым отработывают звук, который находится в начале слова (перед гласным), следующим в конце (для глухих звуков) и в самом конце — в середине, потому что такое расположение звука является самым сложным в произношении. После отработки звука в разных частях слова простых по своей структуре переходят к следующему этапу. Данный этап заключается в произношении данного звука в словах, которые содержат сочетание нарушенного звука с согласными. После этого рекомендуется перейти к произношению словосочетаний с ними и коротких предложений.

Для оценки правильности выполнения упражнений предполагается использование метода обратной визуальной связи. Предлагается определять параметры речи пациента и отобразить их в виде визуальной информации на экране. Цель — научить пациента управлять голосом путём достижения заданных целевых параметров (предполагается, что пациент в процессе произношения видит на экране визуальное отображение как текущих параметров, так и целевых).

Основные преимущества системы коррекции речи над другими подобными тренажерами:

- комплексный подход (коррекция нескольких дефектов речи);
- проведение самостоятельных сеансов коррекции без контроля специалиста;
- автоматическая оценка правильности выполнения упражнений.

Таким образом, система коррекции речи — тренажер, предназначенный для коррекции нескольких дефектов речи таких как: дислалия (нарушение звукопроизношения), заикание, нарушение темпа речи. Целью создания данного тренажера является возможность проведения самостоятельных сеансов коррекции речи без участия специалистов, путем автоматической оценки правильности выполнения упражнений.

В связи с ростом людей с нарушением устной речи актуальным является вопрос создания такой комбинированной системы для коррекции речи, которая позволит самостоятельно проводить сеансы терапии выбранных дефектов.

Список использованных источников:

1. Chesters J, Baghai-Ravary L, Möttönen R. The effects of delayed auditory and visual feedback on speech production. The Journal of the Acoustical Society of America. 2015;137(2):873-883. doi:10.1121/1.4906266.
2. Волкова Л.С. Логопедия: Учебник для студентов дефектолог. фак. пед. Вузов / Под ред. Л.С. Волковой, С.Н. Шаховской. — М.: Гуманит. изд. центр ВЛАДОС, 1998. — 680 с.
3. Лохов М.И., Миссуловин Л.Я., Фесенко Ю.А. Заикание: возможности педагогической коррекции // Вестник, 2013, № 3, 48-58.

ОПТИКО-ЭЛЕКТРОННАЯ СИСТЕМА ВИДЕОНАБЛЮДЕНИЯ

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

Егоров А.В.

Качинский М.В. — к.т.н., доцент

От современных систем видеонаблюдения требуется возможность обнаружения движущихся объектов, их классификация и распознавание. Это позволяет уменьшить итоговую цену системы видеонаблюдения путем автоматизации камер и снизить нагрузку на охранные комплексы, путем передачи видео непосредственно до и после сигнала тревоги.

Обнаружение движения, слежение, распознавание и классификация объектов являются одними из основных направлений видеоаналитики. Применение видеоаналитики дает возможность в автоматическом режиме, в процессе видеонаблюдения, решать задачи, которые обычно под силу только человеческому зрению. Данная технология используется как для обеспечения безопасности, так и для повышения эффективности бизнеса в торговле, финансовом секторе и на транспорте.

Разработка оптико-электронной системы видеонаблюдения с возможностью обнаружения движения и последующей классификацией объектов и стала основной темой этого доклада.

Оптико-электронная система видеонаблюдения в общем случае представляет собой ip-камеру, работающую на базе связки FPGA и основного процессора с ARM архитектурой. Предобработка и фильтрация входных данных будет выполняться на FPGA, сокращая время последующего анализа и обработки видеопотока на ARM. Перед системой видео наблюдения стоит проблема решения следующих задач видеоаналитики:

1. Детектирование движения в кадре

Это несложная с точки зрения вычислений задача, заключающаяся в анализе двух соседних кадров, полученных в разное время, и нахождении между ними разницы свыше заданных параметров. Такими параметрами в простейших случаях являются:

- размер движущегося объекта.

Который задается в процентах от размера всего кадра и позволяет обнаружить тот размер объекта, при движении которого необходимо начинать запись.

- чувствительность.

Контраст движущегося объекта. То как движущийся объект выделяется на общем фоне. Фильтрация неконтрастных объектов позволяет исключить ложные срабатывания детектора, таких как при раскачивании на ветру деревьев, веток, травы.

2. Классификация и распознавание движущихся объектов

Является более сложной задачей, требующей значительных затрат вычислительной мощности. Для оптимизации этих процессов в настоящее время все чаще используются различные нейросети, которые обучаются для распознавания конкретных объектов.

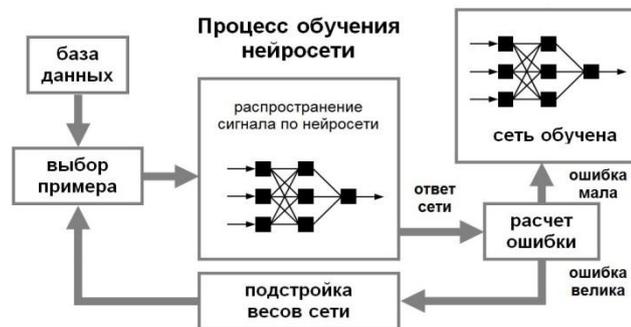


Рис. 1 – Обучение нейронной сети

Для реализации этого проекта была выбрана нейронная сеть GoogleNet. Она обучалась на заранее сформированной выборке, подобранной специально чтобы обучить нейросеть на классификацию объектов на людей и транспортные средства. Общий принцип обучения нейронной сети представлен на рисунке 1

Использование таких систем видеонаблюдения позволит с применением ограниченных ресурсов основного процессора и FPGA получать требуемый результат и обеспечивать безопасность и наблюдение за объектом. Использование нейронных сетей делает эту систему весьма гибкой и настраиваемой, т.к. переобучение нейросети для обнаружения объектов другого типа не займет много сил и времени.

Список использованных источников:

1. OpenCV [Электронный ресурс]. – Электронные данные – Режим доступа: <https://opencv.org>
2. Медведев В. С., Потемкин В. Г. Нейронные сети. MATLAB 6. М.: ДИАЛОГ-МИФИ, 2001. 630 с.
3. Demuth H., Beale M. Neural Network Toolbox. For Use with MATLAB. The MathWorks Inc. 1992-2000.

КЛАССИФИКАЦИЯ ЗВУКОВ С ПОМОЩЬЮ SEMI-SUPERVISED ПОДХОДА

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

Жук И.Н.

Петровский А.А. – д.т.н., профессор

Данные-это ключ для успешного обучения глубоких нейронных сетей, однако для классического обучения с учителем данные должны быть размечены. Для того, чтобы использовать неразмеченные данные были разработаны различные методы semi-supervised обучения. Один из таких методов называется лестничная сеть [2]. Если применить данный подход к задаче классификации звуков окружающей среды, можно добиться значительного повышения результатов классификации.

Лестничная архитектура состоит из трёх нейронных сетей: “чистого” кодировщика, “зашумленного” кодировщика и декодировщика. При этом оба кодировщика по сути являются одной сетью, т.к. они разделяют весовые коэффициенты, но в “зашумленной” версии кодировщика после каждого слоя добавляется нормально-распределенный шум $N(0, \sigma^2)$. Декодировщик преобразует информацию таким же образом, как и кодировщик, но в другую сторону для того, чтобы на выходе получились предсказания данных, которые были поданы на кодировщик.

Лестничная архитектура одновременно обучается двум задачам. Первая задача – это обучение классификации размеченного набора данных. В ней принимает участие только “зашумленный” кодировщик и обучение происходит по классической схеме обучения классификатора. Вторая задача – это подавление шума. Именно благодаря обучению этой задачи выполняется semi-supervised подход, потому что нам не нужны метки для процесса обучения. Соответственно для этой задачи используются неразмеченные данные. В этой задаче используются все три сети, «зашумленный» кодировщик и декодировщик для расчета предсказания входных данных, а «чистый» кодировщик для расчета функции потерь при подавлении шума.

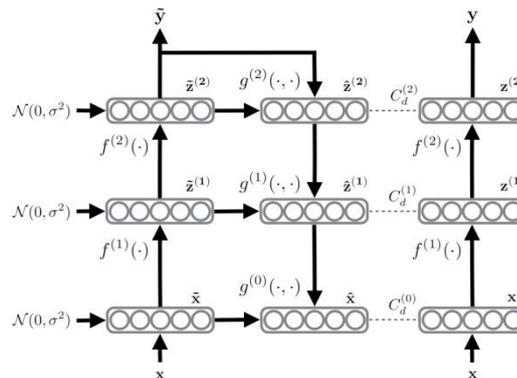


Рис. 1 – Архитектура лестничной сети для классификации

Где x – входные данные, y – предсказания классов, \tilde{x} – зашумленные входные данные, \tilde{y} – зашумленные предсказания классов, $C_d^{(n)}$ – ошибка подавления шума, $f^{(n)}(\cdot)$ – батч-нормализация и функция активации (ReLU), $g^{(n)}(\cdot, \cdot)$ – функция шумодав с обучаемыми параметрами.

Исследования строились на основе двух наборов данных. Набор данных ESC 10 [1] использовался как основной, размеченный набор для классификации. Набор данных UrbanSound8K использовался как неразмеченный, второстепенный набор. Для унификации входных данных все записи были приведены к длине 5с и затем были рассчитаны логарифмические мел-частотные спектрограммы с перекрытием 50%. Итоговое разрешение входных данных 128x431.

Экспериментальные результаты:

Характеристический вектор	Классификатор	Доля правильных ответов
Мел-кепстральные коэффициенты	“случайный лес”	0.73 ± 0.078
спектрограмма	Сверточная нейронная сеть	0.81 ± 0.044
спектрограмма	лестничная архитектура	0.91 ± 0.025
	человек	0.96 ± 0.014

Список использованных источников:

1. K. J. Piczak. ESC: Dataset for Environmental Sound Classification // In Proceedings of the 23rd ACM international conference on Multimedia, pp. 1015-1018, ACM, 2015.
2. Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks // NIPS, 2015.

СИСТЕМА ТЕХНИЧЕСКОГО ЗРЕНИЯ МОБИЛЬНОГО КОМПЛЕКСА ОБНАРУЖЕНИЯ БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ

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

Конопелько Я.Д.

Качинский М.В. - доцент кафедры ЭВС, к.т.н.

В последнее время в развитии беспилотных летательных аппаратов произошел значительный прогресс. Подобная техника находит применение в различных условиях и для решения разных задач. В результате чего расширился и спектр противоправных действий, осуществляемых с их применением. В связи с этим появилась необходимость создания систем противодействия беспилотным летательным аппаратам, способных обнаружить, распознать и нейтрализовать их. Возникла потребность в защите определенных объектов и территорий от проникновения беспилотных летательных аппаратов. По этой причине в последнее время большое внимание уделяется противодействию беспилотным системам.

Система технического зрения мобильного комплекса обнаружения беспилотных летательных аппаратов – это система, входящая в состав комплекса обнаружения и борьбы с беспилотными летательными аппаратами. В состав современных комплексов входят радиолокационные станции с разными характеристиками, вероятность обнаружения воздушной цели зависит от эффективной площади рассеивания. В случае с малогабаритными летательными аппаратами ЭПР уменьшается, что значительно увеличивает сложность обнаружения. Применение технического зрения в системах такого класса совместно с радиолокационными системами позволяет увеличить вероятность обнаружения цели.

Основной функцией рассматриваемой в докладе системы является обнаружение беспилотных летательных аппаратов в видимом диапазоне волн на основании данных от оптико-электронного модуля. В основе системы лежит нейронная сеть, при помощи которой происходит обнаружение летательных аппаратов в зоне наблюдения. После обнаружения цель сопровождается и классифицируется.

Использование системы технического зрения в комплексах обнаружения беспилотных летательных аппаратов позволит повысить вероятность обнаружения, точность определения класса и типа летательного аппарата при этом снизив его стоимость.

Список использованных источников:

1. OpenCV [Электронный ресурс]. – Электронные данные - Режим доступа: <https://opencv.org>
2. Xilinx [Электронный ресурс]. – Электронные данные – Режим доступа: <https://www.xilinx.com>
3. Zedboard [Электронный ресурс]. – Электронные данные – Режим доступа: <http://www.zedboard.org>
4. Demuth H., Beale M. Neural Network Toolbox. For Use with MATLAB. The MathWorks Inc. 1992-2000.

IP – ЯДРО АЛГОРИТМА SHA – 1

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

Корбут А.А.

Станкевич А.В. – к.т.н., доцент

Необходимость обеспечения конфиденциальности и сохранения целостности данных в рамках систематизации повседневной жизни бесспорна и очевидна. В настоящее время – это одна из самых актуальных задач, решения которой совершенствуются с каждым днем. Secure Hash Algorithm 1 — алгоритм криптографического хеширования, являющийся классическим.

Множество алгоритмов разработаны и разрабатываются по сей день. Алгоритм MD4 послужил прототипом SHA – 1.

Организация SHA – 1 состоит из нескольких этапов[1]:

1. Инициализация переменных. Определение операций и констант.
2. Приведение исходного сообщения к необходимому для обработки виду.
3. Обработка главным циклом.

Инициализация.

Используются пять переменных размера 32 бит.

A = a = 0x67452301

B = b = 0xEFCDAB89

C = c = 0x98BADCFE

D = d = 0x10325476

E = e = 0xC3D2E1F0

Номер итерации t определяет действующую функцию и константу.

Табл. 1 – Функции и константы алгоритма SHA – 1

Подготовка сообщения.

Сообщение делится на блоки. Каждый блок не превышает 448 бит, кратен 512. К каждому блоку добавляется 1 и множество нулей для дополнения длины до 448 бит. Добавление осуществляется всегда, даже если сообщение уже имеет нужную длину. В последние 64 бита записывается длина сообщения в битах.

Главный цикл.

Основой алгоритма является модуль, состоящий из 80 циклических обработок, обозначенный как HSHA. Все 80 циклических обработок имеют одинаковую структуру.

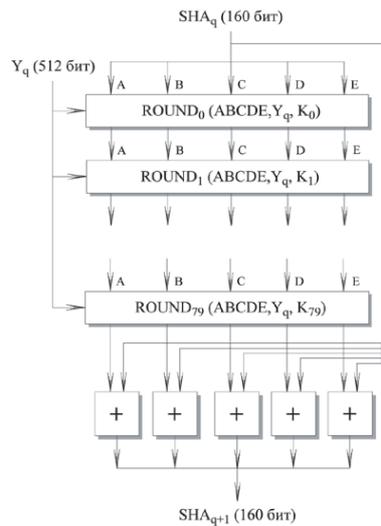


Рис.1 – Структура алгоритма

Состоит из 80 итераций от 0 до 79.

Логичу одной итерации цикла можно представить в виде[1]:

$A, B, C, D, E (CLS5(A) + ft(B, C, D) + E + W_t + K_t), A, CLS30(B), C, D,$

где A, B, C, D, E - пять слов из буфера;

t - номер цикла, 0 t 79;

ft - элементарная логическая функция;

CLSs - циклический левый сдвиг 32-битного аргумента на s битов;

W_t - 32-битное слово, полученное из текущего входного 512-битного блока;

K_t - дополнительная константа;

+ - сложение по модулю 2^{32} .

32-битные слова W_t получаются из очередного 512-битного блока сообщения следующим образом:

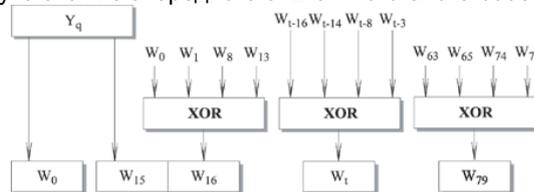


Рис.2 – Формирование слова

Первые 16 значений W_t берутся непосредственно из 16 слов текущего блока. Оставшиеся значения определяются следующим образом:

В первых шестнадцати циклах вход состоит из 32 – битного слова данного блока. Для оставшихся 64 циклов вход состоит из сложения по модулю 2 нескольких слов.

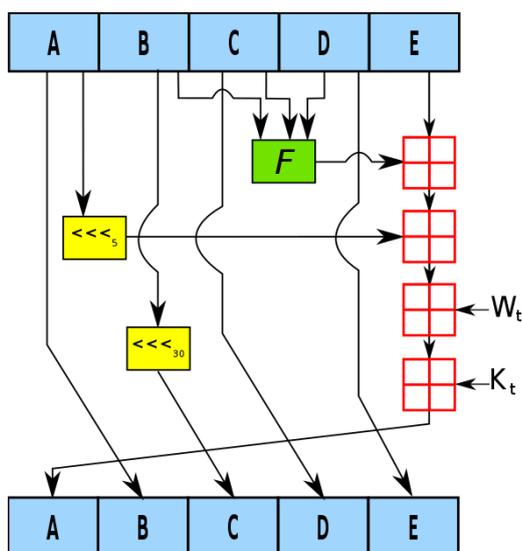


Рис. 3– Одна итерация цикла.

Длина дайджеста	160 бит
Размер блока обработки	512 бит
Число итераций	80
Число элементарных функций	3
Число дополнительных констант	4

Алгоритм прост в описании и реализации, однако достаточно устойчив к атакам грубой силы.

Список использованных источников:

1. D. Eastlake, P. Jones, RFC 3174 US Secure Hash Algorithm 1 (SHA1)// Cisco Systems – 2001.
2. Ярчук С. М. Конспект лекций по информационной безопасности/ С. Ярчук – Уральский Федеральный университет им. Б.Н. Ельцина «УПИ», 2011. - 75 с.

ОЦЕНКА РЕЧЕВОЙ МАСКИ ДЛЯ ИДЕНТИФИКАЦИИ ДИКТОРА

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

Кручок Д.Н.

Петровский А.А. – д.т.н., профессор

В данной работе рассмотрены методы оценивания маски для речевого сигнала. Речевая маска позволяет использовать реконструированный сигнал для извлечения характеристического вектора и дальнейшей идентификации диктора в условиях акустических шумов.

Речевой сигнал, даже если он зашумлен, обладает значительной степенью избыточности, и информация о речевых характеристиках присутствует и в случае сильного зашумления. Методы, которые используют это наблюдение (англ. missingdataapproaches) используют речевые маски для маркировки сигнала для каждой точки время-частота по наличию речи или шума в ней [1]. В дальнейшем, маску используют для реконструирования сигнала, получения из него характеристического вектора для идентификации диктора. В случае, когда шум известен заранее, маску можно построить по критерию отношения сигнал-шум (англ. signaltonoise, SNR) для каждого временно-частотного компонента. Если такую маску использовать для идентификации, то распознавание диктора будет иметь высокую шумоустойчивость даже в присутствии сильных шумов и искажений [2]. На практике, отсутствие априорного знания о шуме толкает на осуществление оценки речевой маски.

Подходы оценивания речевой маски делятся на [1]:

- 1) Методы, основанные на оценке SNR;
- 2) Методы, основанные на слуховом восприятии;
- 3) Методы, основанные на классификации параметров оценивания.

При оценке SNR решение надежности того или иного временно-частотного компонента принимается

прямым измерением спектральной плотности мощности зашумлённого сигнала. Для этого применяется метод спектрального вычитания [3], где оценка значения средней спектральной мощности шума получается из нескольких первых фреймов выражения (при условии тишины, когда в начале выражения отсутствует энергия речи). Для зашумленного сигнала, предполагаемый спектр чистой речи можно получить по формуле [3]:

$$\hat{S}(t, f) = \begin{cases} X(t, f) - \hat{N}(t, f), & \text{если } X(t, f) - \hat{N}(t, f) > \gamma X(t, f), \\ \gamma X(t, f), & \text{в противном случае,} \end{cases} \quad (1)$$

где,

$$\hat{N}(t, f) = \frac{1}{T_{ср\delta}} \sum_{t=1}^{T_{ср\delta}} X(t, f) - \text{спектр мощности для первых } T \text{ кадров, а } \gamma - \text{малый масштабирующий}$$

коэффициент ($\gamma \approx 0$), необходим, чтобы значения спектра мощности были неотрицательными. Критерий для принятия решений [1]:

$$m_{ff}^{ss}(\theta) = \begin{cases} 1, & \text{если } 10 \log_{10} \left(\frac{\hat{S}(t, f)}{\hat{N}(t, f)} \right) > \theta, \\ 0, & \text{в противном случае} \end{cases} \quad (2)$$

θ – порог в дБ. В работе [4] был предложен критерий отрицательной энергии, где временно-частотный компонент помечался как надёжный, если значение спектра зашумленного сигнала превышает значение спектра предполагаемого «шумового» сигнала. На рисунке 1 изображены спектрограмма зашумлённого речевого сигнала и полученная речевая маска с порогом надёжности $\theta = 0$ дБ. Речевой сигнал содержит фразу: «штурман просил продолжить разворот», частота дискретизации равна 16 кГц, тип используемого шума – *babble*, уровень зашумления SNR составляет минус 5 дБ.

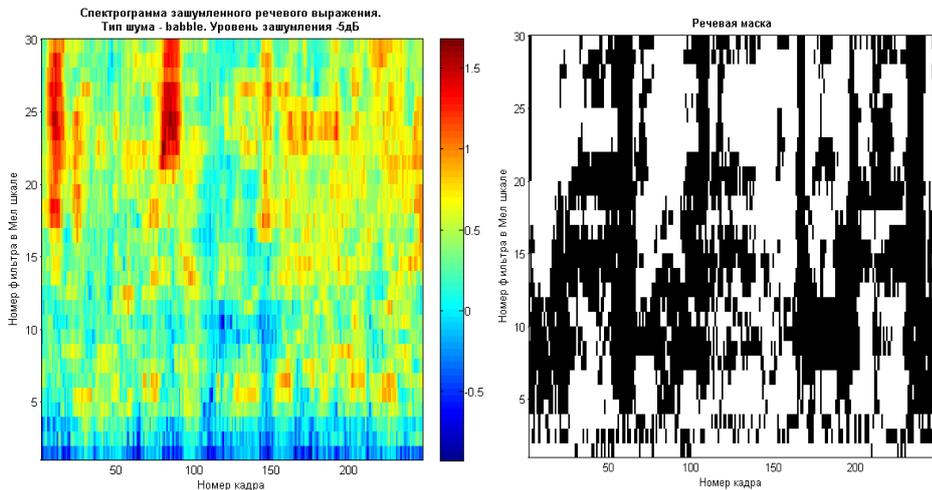


Рисунок 1. Спектрограммы речевого сигнала зашумленного шумом *babble* с уровнем SNR = -5дБ (слева) и полученная речевая маска с порогом надёжности $\theta = 0$ дБ (справа)

Недостатком методов, основанных на оценке SNR – это отсутствие шумоустойчивости к нестационарным и переходным шумам.

Методы оценки, основанные на слуховой системе человека, используют свойства разделения целевых сигналов от источников шумов. В работе [5] был предложен метод группировки гармоник, основанный на том, что энергия внутри голосового сигнала находится возле гармоник частоты основного тона. В работе [6] используется метрика – голосовая дистанция – евклидово расстояние между формантами спектра кратковременного сигнала и анализируемого окна. Это расстояние потом сравнивается с порогом, и сходно с формулой 2, выносится решение по каждому временно-частотному компоненту. Также методы могут использовать пространственное местонахождение источников сигнала для определения временно-частотных регионов в слуховой сцене с помощью бинауральной слуховой модели [7, 8].

Методы, основанные на классификации параметров оценивания, используют модель обучения для оценивания речевой маски. В байесовском подходе эта классификация достигается путем моделирования распределений набора признаков надёжности. Если признаки сконструированы таким образом, что они используют характеристики самого речевого сигнала (а не характеристики искажающего шума), то вероятностная оценка значений признаков будет показывать, надёжна ли связанная с ними временно-частотная точка.

Рассмотрим спектральный компонент x_{tf} и набор соответствующих признаков надёжности ss_{tf} . В соответствии с байесовским подходом решение о бинарной надёжности для x_{tf} можно получить [1]:

$$m_{ff}^{Bayes} = \begin{cases} 1, & \text{если } p(\lambda_{m=1}^{Bayes}(f)) p(ss_{tf} | \lambda_{m=1}^{Bayes}(f)) > p(\lambda_{m=0}^{Bayes}(f)) p(ss_{tf} | \lambda_{m=0}^{Bayes}(f)), \\ 0, & \text{в противном случае} \end{cases} \quad (3)$$

$\lambda_{m=1}^{Bayes}(f)$ и $\lambda_{m=0}^{Bayes}(f)$ являются обученными распределениями для надёжного и ненадёжного решения,

соответственно, в частотном канале f . А $p(ss_{if} | \lambda_{m=1}^{Bayes}(f))$ и $p(ss_{if} | \lambda_{m=0}^{Bayes}(f))$ соответствующие плотности вероятности надежности и ненадежности для канала f , основанные на признаках ss_{if} .

В качестве признаков используют различные параметры: в работе [9] – спектральная энергия полосы и её производные, в работе [8] были использованы: *коэффициент гребенчатого фильтра* для сравнения энергии в вокализованных областях с энергией в негармонических областях; *коэффициент автокорреляции*, для измерения периодичности сигнала; *отношение энергии поддиапазона к энергии полной полосы*, представляющее спектральную форму; *оценка энергии шума*; *коэффициент эксцесса*, используемый для измерения «остроты» пика сигнала; *коэффициент тональности*, для измерения SNR.

Используя предполагаемые маски, отдельные классификаторы обучаются для невокализованных и вокализованных типов внутри каждого канала. В оценках для шумоустойчивого распознавания оценки классификатора превосходят традиционные оценки спектрального вычитания во всех условиях шума, но особенно для нестационарных случаев [10].

Таким образом, методы, основанные на слуховой системе человека, так и основанные на классификации речевых параметров могут обеспечить лучшую эффективность распознавания в сравнении с методами на основе оценки только SNR. Преимуществом аудиторных подходов в оценке маски является их способность строить решения, основанные на свойствах спектра речевого сигнала. Это позволяет более точно идентифицировать доминирующие речевые спектральные области по сравнению с подходами на основе SNR, которые предполагают обобщение шумовых характеристик, наблюдаемых в небольшом числе свободных от речи кадров. Недостатком подхода, основанного на классификации речевых параметров, является его слабость к шуму, которые имеют сходные спектральные характеристики с характеристиками речевого сигнала. В этом случае речевые параметры не смогут отличить речевые и шумовые компоненты доминирующей частоты, что приводит к низкой точности маскировки.

Список использованных источников:

1. Togneri R., Pullella D. An Overview of Speaker Identification Accuracy and robustness Issues // IEEE Circuits and systems magazine. 2011. P. 23–58.
2. M. Cooke, P. Green, and M. Crawford, "Handling missing data in speech recognition," in Proc. IEEE Int. Conf. Acoustics Speech Signal Processing (ICASSP), 1994, pp. 1555–1558.
3. S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," IEEE Trans. Acoust. SpeechSignalProcess., vol. 27, no. 2, pp. 113–120, 1979.
4. M. El-Maliki and A. Drygajlo, "Missing features detection and handling for robust speaker verification," in Proc. European Conf. Speech Communication Technology (Eurospeech), Budapest, Hungary, 1999, pp. 975–978.
5. J. Barker, M. Cooke, and P. Green, "Robust asr based on clean speech models: An evaluation of missing data," in Proc. European Signal Process. Conf. (EUSIPCO), Aalborg, Denmark, 2001, pp. 213–216.
6. P. Jan'covi'c and M. Кцкьер, "Estimation of voicing-character of speech spectra based on spectral shape," IEEE Signal Process. Lett., vol. 14, no. 1, pp. 66–69, 2007.
7. K. J. Palomdki, G. J. Brown, and D. Wang, "A binaural processor for missing data speech recognition in the presence of noise and small-room reverberation," Speech Commun., vol. 43, no. 4, pp. 361–378, 2004.
8. Кручок, Д. Н. Эффект бинауральной маскировки для идентификации диктора в акустических шумах / Д.Н. Кручок // Современные технологии в науке и образовании – СТНО-2017 [текст]: сб. тр. междунар. науч.-техн. и науч.-метод. конф.: в 8 т. Т.3./ под общ. ред. О.В. Милонзорова. – Рязань: Рязан. гос. радиотехн. ун-т, 2017; Рязань. – 292 с. – С. 165–168.
9. B. Raj, "Reconstruction of incomplete spectrograms for robust speech recognition," Ph.D. dissertation, Pittsburgh, PA, Carnegie Mellon Univ., 2000.
10. M. L. Seltzer, B. Raj, and R. M. Stern, "A Bayesian classifier for spectrographic mask estimation for missing feature speech recognition," Speech Commun., vol. 43, no. 4, pp. 379–393, 2004.

СТИЛИЗАЦИЯ ГОЛОСА С ИСПОЛЬЗОВАНИЕМ ГЛУБОКОГО ОБУЧЕНИЯ

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

Крылов Н.Д.

Одной из самых сложных задач для вычислительной техники является обработка естественного языка. В частности, одной из таких задач является имитация человеческого голоса. Алгоритмы машинного обучения являются наиболее эффективным инструментом в этой области.

Для обучения нейронной сети необходим правильно составленный набор данных. Будет ли правильным обучающим примером для данной задачи сказанная одна, двумя разными людьми, фраза? Фраза может быть произнесена с различной скоростью, громкостью и темпом, входная и выходная последовательность будет иметь различную длину. Обучение нейронной сети на таких данных затруднительно, в таком случае хорошим решением является использование двух нейронных сетей. Первая нейронная сеть используется в качестве кодировщика и переводит данные в иное представление, вторая в качестве декодировщика, позволяет получить требуемый результат.

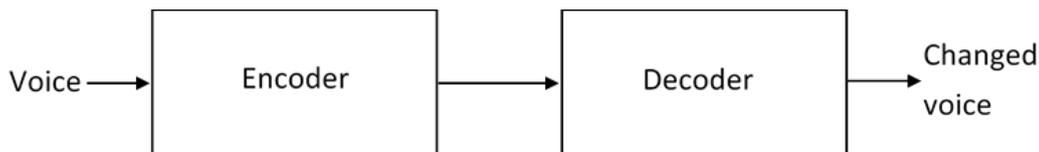


Рис. 1 – Общая структура модели

Такая модель успешно используется в задачах машинного перевода и подписи изображений. Для обучения такой модели необходим набор данных, содержащий пары звук-текст [1]. Обучение модели разделено на следующие шаги:

1. обучение системы распознавания речи (кодировщика), используя звук-текст обучающие примеры;
2. удаление Connection temporal classification и Softmax слоев нейронной сети;
3. блокировка всех слоев кодирующей нейронной сети чтобы избежать их переобучения;
4. подключение к модели синтеза голоса (декодера);
5. обучение конечной модели с использованием только звука в качестве входных и выходных данных.

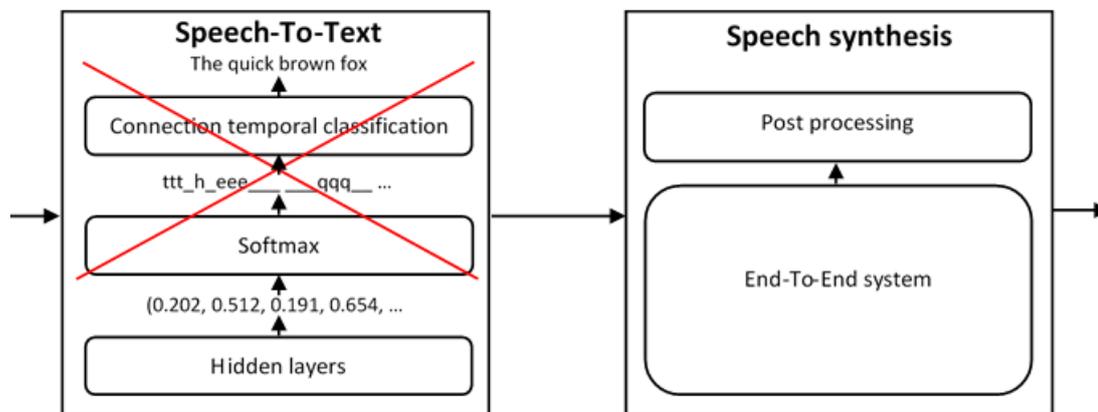


Рис. 2 – Подробная структура модели

В качестве кодировщика может быть использована предобученная нейронная сеть, такая как Speech-to-Text-WaveNet, DeepSpeech. Нейронная сеть для синтеза голоса должна быть обучена с нуля из-за нестандартного представления входных данных. Такое представление данных может содержать больше информации об интонации. Кодирующая нейронная сеть может быть использована для любых примеров, однако декодирующая должна быть дообучена в отдельности для синтеза конкретного голоса [2].

Список использованных источников:

1. AaronvandenOord, SanderDieleman, HeigaZen, KarenSimonyan, OriolVinyals, AlexGraves, NalKalchbrenner, AndrewSenior, KorayKavukcuoglu, WaveNet: AGenerativeModelforRawAudio [Электронный ресурс]. – 2016. – Режим доступа: <https://arxiv.org/abs/1609.03499>.
2. Wei Ping, Kainan Peng, Andrew Gibiansky, Sercan O. Arık, Ajay Kannan, Sharan Narang, Deep Voice 3: Scaling Text-to-Speech with Convolutional Sequence Learning [Электронныйресурс]. – 2018. – Режимдоступа: <https://arxiv.org/abs/1710.07654>.

ПОВЫШЕНИЕ КАЧЕСТВА ИЗОБРАЖЕНИЯ НА БАЗЕ АЛГОРИТМОВ НЕЙРОННЫХ СЕТЕЙ

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

Никитин Г.Ю.

Петровский А.А. – д.т.н., профессор

В настоящее время остро стоит проблема наличия различных шумов и артефактов на изображениях, что приводит к сильному снижению их качества. На практике наиболее распространённые шумы колеблются от аддитивного шума до мультипликативного шума. Такая деградация может оказать значительное влияние на точность методов компьютерной обработки в медицине, сделать анализ и распознавание изображений трудными и ненадёжными.

Существует большое множество алгоритмов, способных убрать с изображения шумы. В последнее время было проведено много исследований алгоритмов шумоподавления в вейвлет области, например, был предложен алгоритм на основе вейвлет преобразования с последующей отсечкой шумовых коэффициентов [1], или, к примеру, алгоритм оценки шумовых вейвлет коэффициентов, основывающийся на моделях

распределения Гаусса и Лапласа [2]. Также одним из алгоритмов, позволяющих убрать с изображения различные шумы, является алгоритм обработки изображения нейронной сетью в вейвлет области [3, 4].

Вейвлет преобразование представляет собой перевод изображения из временного представления в пространственно-временное представление. В данном случае рассматривается дискретное вейвлет преобразование (discretewavelettransform, DWT). Каждая часть такого вейвлет преобразования подаётся на свою нейронную сеть для обработки и сборки в обесшумленное изображение при помощи обратного вейвлет преобразования (inversediscretewavelettransform, IDWT). Структура алгоритма представлена на рисунке 1:

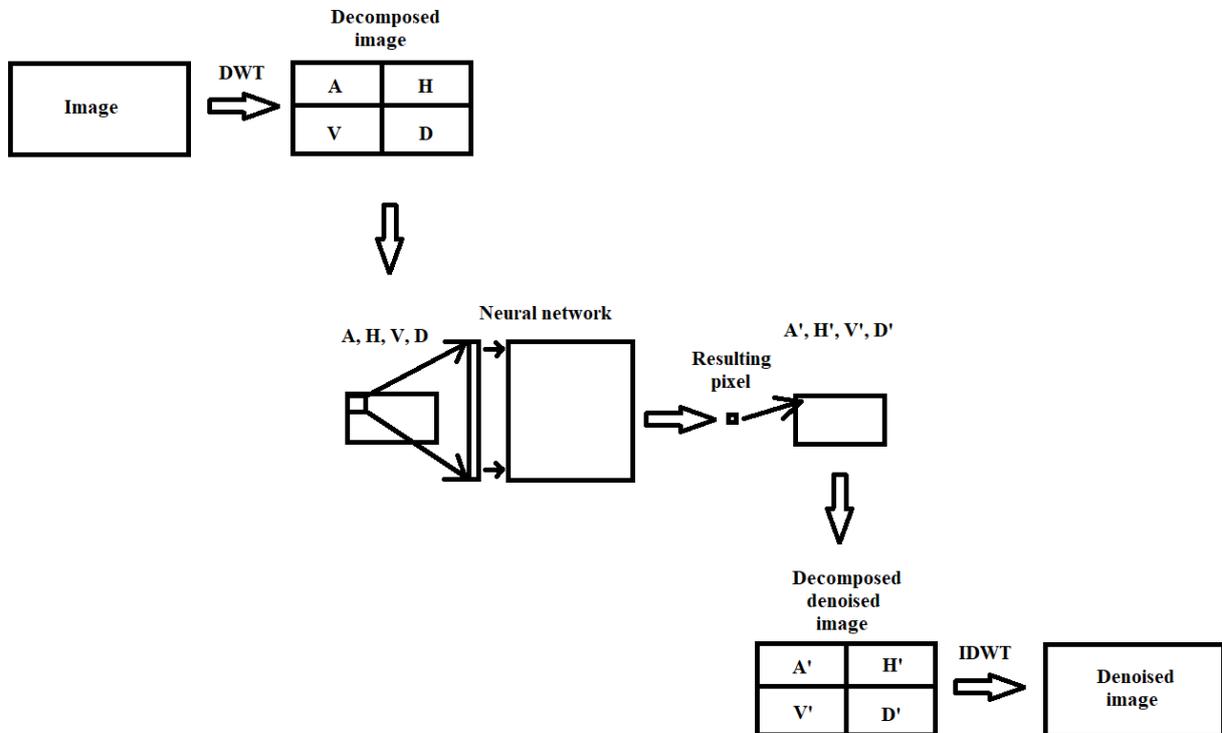


Рис. 1 – Структура алгоритма.

Как уже было сказано, в начале работы изображение проходит вейвлет преобразование (отмечено DWT). Затем каждая часть преобразованного изображения (отмечено A, H, V, D) подаётся на соответствующую нейронную сеть. При этом на обработку нейронным сетям (Neuralnetwork) подаются области некоторого размера (в данной работе был использован размер области 5x5 пикселей) с шагом в 1 пиксель. В результате обработки каждой области нейронная сеть возвращает уже обесшумленный пиксель (Resultingpixel). После завершения обработки всех областей полученные пиксели собираются вместе и выполняется обратное вейвлет преобразование (IDWT) и получение исходного изображения.

Нейронные сети представляют из себя двухслойные линейный нейронные сети, с 10 нейронами на 1 слое и 1 нейроном на 2 (выходном) слое. Обучение проводилось методом Левенберга-Марквардта.

Алгоритм был протестирован на нескольких тестовых изображениях с шумами (изображения не участвовали в выборке для обучения нейронных сетей). Результаты работы алгоритма представлены на рисунках 2а, 2б и 2в. Значения PSNRпредставлены в таблице 1.



Рис. 2а – Исходное изображение без шума.



Рис. 2б – Исходное изображение с шумом.



Рис. 2в – Изображение с шумом после обработки алгоритмом.

Табл. 1 – Значения PSNR для обрабатываемых изображений.

Изображение	PSNR (до обработки)	PSNR (после обработки)
Для обучения нейронных сетей	8.676	21.031
Тестовое	9.853	20.017
Тестовое	10.415	20.835

Таким образом была представлена система шумоподавления, использующая вейвлет преобразование и линейные нейронные сети. Нейронные сети ищут зависимость между исходными и зашумленными коэффициентами вейвлет преобразования, что позволяет эффективно подавлять различные шумы на изображениях.

Основные преимущества такого подхода:

- Обработка изображения в вейвлет области занимает значительно меньше времени, чем при обработке исходного изображения из-за возможности параллельной обработки нескольких областей вейвлет преобразования одновременно;
- Вышеописанное свойство также работает и для обучения нейронных сетей, что позволяет значительно сократить время работы.

Основной недостаток – данный алгоритм (как и многие, если не все) не даёт 100% шумоподавления и восстановления исходного изображения. Это объясняется тем, что часть информации теряется необратимо и лишь часть можно восстановить при использовании алгоритмов восстановления и шумоподавления.

Список использованных источников:

1. Donoho D., Adapting to unknown smoothness via wavelet shrinkage / D. Donoho, I. Johnstone // *Biometrika* – 1994.
2. Yu H., Image denoising using trivariate shrinkage filter in the wavelet domain and joint bilateral filter in the spatial domain / H. Yu, L. Zhao, H. Wang // *IEEE Trans. Image Proc.* – 2009.
3. Zhang S., Image denoising using a neural network based non-linear filter in wavelet domain / S. Zhang, E. Salari
4. Bhutada G., Image enhancement by wavelet-based thresholding neural network with adaptive learning rate / Bhutada G., Anand R., Saxena S. // *IET Image processing* – 2010.

СИСТЕМА МОДЕЛИРОВАНИЯ ЭФФЕКТА ПОТЕРИ СЛУХА ПО АУДИОГРАММЕ

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

Порхун М.И.

Вашкевич М. И. – кандидат технических наук, доцент

Предлагается система моделирования эффекта потери слуха по аудиограмме. Задача моделирования потери слуха сводится к обработке сигнала таким образом, чтобы человек с нормальным слухом услышал звук так, как его воспринимает тугоухий человек. Метод заключается в субполосной обработке сигнала с применением компрессоров динамического диапазона (КДД). Настройка компрессоров выполняется исходя из аудиограммы тугоухого человека.

Введение. Несмотря на то, что за последние годы были разработаны различные системы моделирования потери слуха, ни одна из них не получила широкого распространения. Методы моделирования потери слуха многими специалистами не воспринимается всерьёз, хотя их применение позволяет обеспечить реалистичную демонстрацию как коммуникативных, так и психосоциальных эффектов нарушения слуха. Основная цель моделирования потери слуха заключается в том, чтобы показать людям с

нормальным слухом как тугоухий человек воспринимает окружающую акустическую среду [1]. На основе модели потери слуха может быть оценена эффективность различных способов коррекции слуха.

Моделирование эффекта потери слуха. Для моделирования потери слуха предлагается использовать схему, приведённую на рисунке 1.

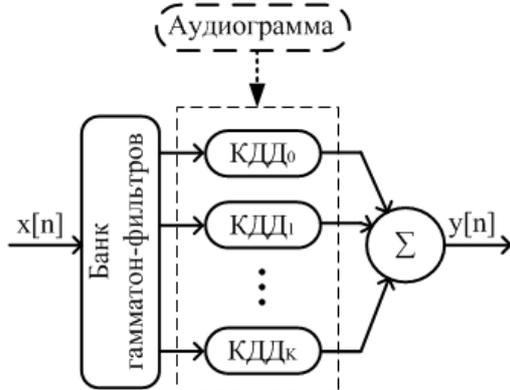


Рис. 1 – Общая структура модели потери слуха

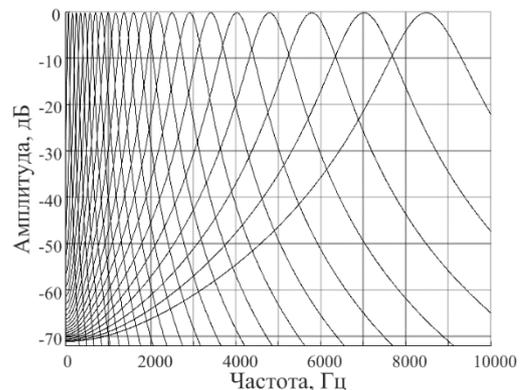


Рис. 2 – АЧХ банка фильтров

Входной сигнал $x[n]$ поступает на банк гамма-фильтров, где разделяется на субполосы. Полученные субполосные компоненты сигнала проходят через соответствующие компрессоры динамического диапазона (КДД). Задачей КДД является автоматический выбор коэффициентов ослабления субполосных компонент сигнала в зависимости от уровня мощности сигнала в данной полосе. Параметры КДД определяются из аудиограммы. Субполосные сигналы, обработанные в КДД, синтезируются путём суммирования субполосных компонент сигнала. Прослушав выходной сигнал $y[n]$ человек с нормальным слухом может получить представление о том, как слышит сигнал $x[n]$ тугоухий человек.

Банк фильтров. Для разделения сигнала на субполосные составляющие в работе используется банк гамма-фильтров, который имитирует механизм частотного разложения звука в улитке уха человека [2]. Импульсная характеристика гамма-фильтра с центральной частотой f_c имеет следующий вид:

$$g(t) = At^{N-1} e^{-2\pi b \text{ERB}(f_c)t} \cos(2\pi f_c t + \varphi) \quad (1)$$

где N – порядок фильтра, $\text{ERB}(f_c)$ – эквивалентная прямоугольная полоса пропускания фильтра, φ – начальная фаза, A – амплитуда, b – параметр, регулирующий ширину эквивалентной прямоугольной полосы пропускания фильтра.

При умеренных уровнях мощности сигнала $\text{ERB}(f)$ описывается следующим выражением [3]:

$$\text{ERB}(f) = 24.7 + 0.108 f \quad (2)$$

В данной работе порядок гамма-фильтра равен четырём ($N = 4$), а коэффициент $b = 1.019$ [4].

Гамма-фильтры могут быть реализованы в виде КИХ- или БИХ-фильтров, а также при помощи преобразований сигнала в частотной области [5]. В настоящей работе использовался 22-канальный банк фильтров (рисунок 2), реализованный в виде гребенки КИХ-фильтров (фильтрация выполнялась с применением БПФ).

Анализ аудиограммы. При моделировании потери слуха, источником информации о слухе является аудиограмма. Для настройки модели необходимо знать пороги слышимости на центральных частотах банка фильтров. Для их определения, на первом этапе, выполняется интерполяция аудиограммы на сетку центральных частот банка фильтров. На следующем этапе рассчитываются пороги слышимости (в норме) на центральных частотах банка фильтра в соответствии со стандартом ISO 226 [6].

На рисунке 3 приведён пример построения КДД для центральной частоты 838 Гц. Аудиограмма интерполируется на сетку центральных частот банка фильтров, что позволяет сформировать пороги слышимости тугоухого человека (рисунок 3, б). В соответствии со стандартом ISO 226, определяются пороги слышимости здорового человека (рисунок 3, а). Исходя из полученных диапазонов слышимости определяется порог компрессии КДД (рисунок 3, в).

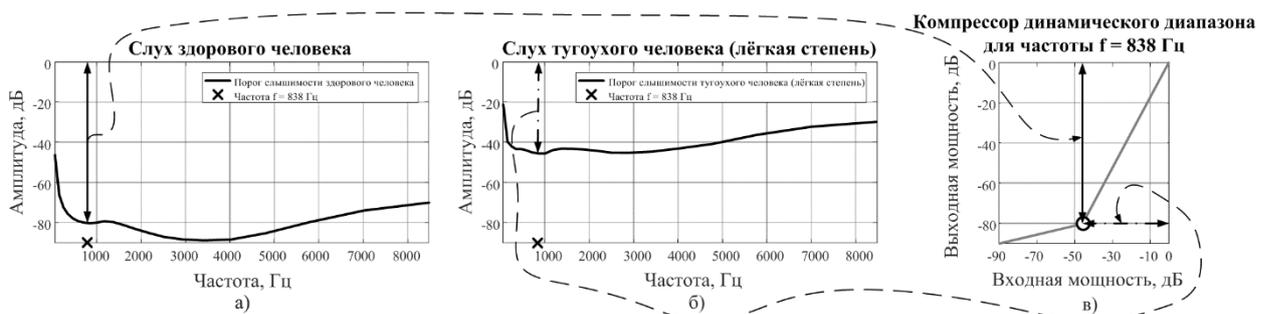


Рис. 3 – Построение компрессора динамического диапазона для частоты 838 Гц

Результаты экспериментов. Проверка предлагаемой системы моделирования эффекта потери слуха производилась на речевом сигнале (рисунок 4, в) и аудиограмме, соответствующей лёгкой степени потери слуха. Частотно-временное представление исходного сигнала приведено на рисунке 4, а. Частотно-временное представление выходного сигнала (рисунок 4, б) демонстрирует особенности восприятия исходного звукового сигнала тугоухим человеком, чья аудиограмма использовалась при моделировании.

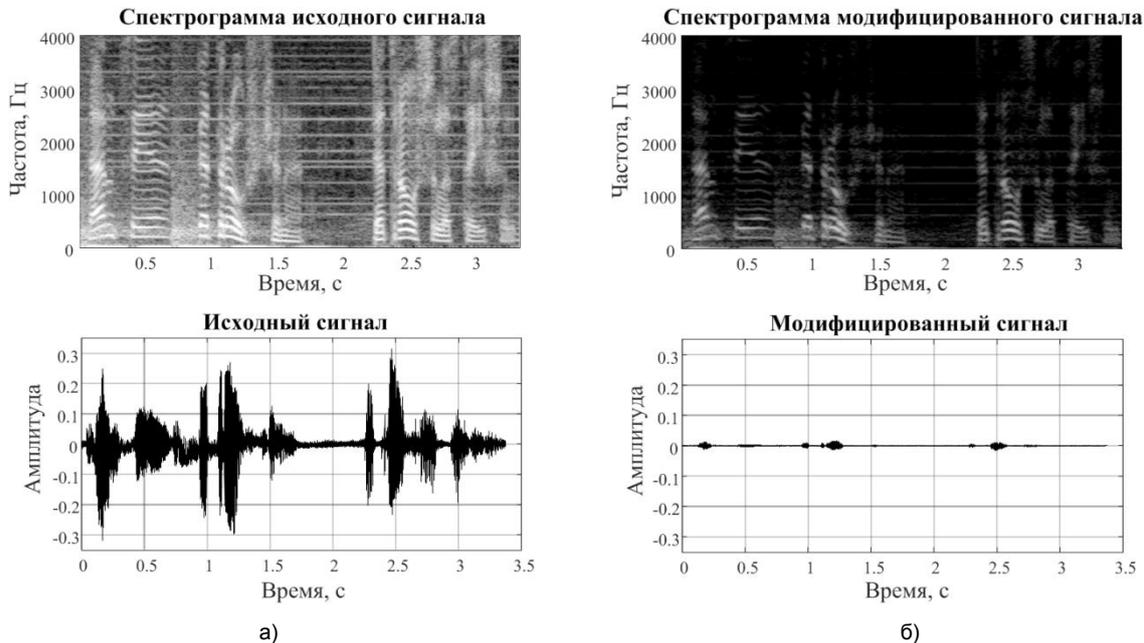


Рис. 4 – Результаты обработки сигнала предложенным методом: а) частотно-временное представление исходного сигнала; б) частотно-временное представление модифицированного сигнала

Вывод. В работе предложена система моделирования эффекта потери слуха. Особенностью системы является применение в ней банка гамматон-фильтров, использование которого позволяет выполнять обработку речевого сигнала, согласованную с работой слуховой системы человека. Другой особенностью предложенной системы является настройка КДД согласно аудиограмме тугоухого человека.

Список использованных источников:

1. Zurek P. M., Desloge J. G. Hearing loss and prosthesis simulation in audiology / The Hearing Journal, – 2007. – Vol. 60, Issue 7. – P. 32 – 33.
2. Cao L. t., Li R. w., Shi Y. q. and Wang S. Loudness compensation method based on human auditory for digital hearing aids / 7th International Conference on BioMedical Engineering and Informatics (BMEI 2014), Dalian, China, – 2014. – P. 335 – 340.
3. Lin L., Holines W.H., Anibikairajah E. Auditory filter bank inversion / IEEE International Symposium Circuits and Systems (ISCAS), Sydney, NSW, Australia, – 2001. – P. 537–540.
4. Jiang Y., Zu Y., Liu L., Wang Q., Ren P. and Zhou H. Gammatone filterbank based energy masking algorithm for active hearing protection system / 5th International Conference on BioMedical Engineering and Informatics (BMEI 2012), Chongqing, China, – 2012. – P. 537–540.
5. Ambikairajah E., Epps J. and L. Wideband Lin Speech and audio coding using gammatone filter banks / IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Salt Lake City, USA, – 2001. – Vol. 2. – P. 773–776.
6. Порхун М. И., Вашкевич М. И. Моделирование потери слуха / Цифровая обработка сигналов и ее применение: труды 20-й междунар. конф., Россия, Москва. – 2018. – Т. 1. – С. 228–233.
7. ISO 226:2003. Acoustics — Normal Equal-Loudness Level Contours / International Organization for Standardization, – Geneva, Switzerland, – 2003.

СИСТЕМА ОХРАННОЙ СИГНАЛИЗАЦИИ

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

Савко Н.Ю.

Станкевич А.В. – к.т.н., доцент

В настоящее время на рынке охранных систем присутствует большое разнообразие продуктов от разных производителей. Все эти системы предоставляют некоторый набор стандартных для такого рода устройств функций, а отдельные экземпляры могут предоставлять также некоторые дополнительные функции. Например, поддержка магнитных карт, или полное управление всеми датчиками из мобильного

приложения, или повышенная степень взломоустойчивости и тд. Несмотря на все это многообразие продукта на рынке, есть смысл в создании системы охранной сигнализации, которая могла бы составить конкуренцию существующим и занять свою нишу.

Сейчас огромное распространение получили системы беспроводной охранной сигнализации. Эта технология гораздо удобнее простого проводного соединения, т.к. не требуется соединять проводами устройства в различных помещениях. Особенно это заметно, если требуется наладить связь между большим числом устройств. Также такой способ беспроводной связи позволяет наладить управление системой с помощью дистанционных средств, например, с помощью пульта или брелока дистанционного управления, мобильного телефона и тд. Большинство охранных систем, являющихся актуальными на сей день и пользующихся спросом, имеют возможность подключения до 50, а иногда и более датчиков.

В данном проекте также будет поддерживаться возможность подключения такого количества датчиков с использованием радиоканала. Предполагается использовать в качестве основной частоты 433 МГц и приемник XY-MK-5V. В данной системе также предполагается поддержка до 4х каналов проводной связи. Это может быть полезно для небольших систем, где все датчики располагаются недалеко от главного модуля. Также это может пригодиться для снижения части затрат, так как проводное подключение дешевле беспроводного.

Предполагается использовать следующие типы датчиков: датчики движения, датчики открытия/закрытия и др. Важной деталью является то, что эти датчики должны работать на той же частоте, что и приемник. В данном случае – 433 МГц. Данная частота выбрана неслучайно. Большинство актуальных систем охранной сигнализации реализовано именно на частоте 433 МГц. Также существует большой ассортимент датчиков различных типов, работающих на данной частоте. Популярным является проектирование охранных систем, используя приемники на 2.4 ГГц. Один из таких приемников (NRF24L01) позволяет подключать к нему до 6 передатчиков одновременно. При этом каждый из них имеет до 126 собственных частот. Однако существует совсем небольшое количество датчиков, которые бы были совместимы с приемником такой частоты. Поэтому для реализации данного проекта был выбран способ реализации с применением приемника и передатчиков. Работающих на частоте 433 МГц.

При разработке охранной системы, которая будет поддерживать беспроводную связь между приемником и передатчиками, важен выбор типа кодировки. Самыми распространенными типами кодировки являются кодировки 1527 и 2262. Разница между ними заключается в следующем:

- количество уникальных адресов (6561 у кодировки-2262 и 1048576 и кодировки-1527);
- способ подключения новых датчиков и удаления старых (сложная, особенно для неподготовленного пользователя, процедура подключения новых и удаления из памяти старых датчиков у типа кодировки 2262, легкая процедура у 1527);
- стоимость чипа с кодировкой 2262 выше стоимости чипа с кодировкой 1527);

Алгоритм кодировки 1527:

- 1) посылается импульс преамбулы;
- 2) посылаются 24 импульса данных, из которых 4 последних бита – это адрес сработавшего первичного преобразователя;

Алгоритм кодировки 2262:

- 1) 24 импульса данных, 1 завершающий импульс
- 2) последние 8 бит (4 перемычки) в послылке - это код сработавшего первичного преобразователя

Преамбула - импульс 500 мкс и пауза 16 мс.

Единица - импульс 1500 мкс и пауза 500 мкс.

Ноль - импульс 500 мкс и пауза 1500 мкс.

Для данного проекта был выбран тип кодировки 1527.

Также важным является поддержка GSM технологии. С ее помощью возможно оповещение владельца охранной системы о том, что системой было отмечено какое-то событие. Будь то ложное срабатывание или взлом. Полезной функцией является сохранение истории срабатывания системы. Так владелец сможет узнать, когда, где и каким образом было вызвано срабатывание сигнализации. Предполагается использовать GSM модуль SIM900D, так как он способен задействовать большинство услуг сотовой связи в случае необходимости: совершать и принимать звонки, слать и получать SMS и MMS, использовать GPRS.

Структурная схема системы представлена на рисунке 1.

Предусмотрена работа устройства в нескольких режимах: режим полной охраны, режим частичной охраны и режим, при котором система отключена. Режим частичной охраны предназначен для того, чтобы информировать владельца, находящегося внутри охраняемого помещения о срабатывании некоторых датчиков. Например, открытия двери и тд. Смена режима работы возможна при помощи взаимодействия с лицевой панелью устройства, находящейся внутри охраняемого помещения, или с помощью пульта дистанционного управления. Например, после открытия двери у вошедшего есть какое-то определенное количество времени, чтобы отключить сигнализацию. Владелец может отключить систему или же перевести ее в режим частичной охраны. Человек, не знающий о нахождении в помещении охранной сигнализации или не знающий как ее отключить, инициирует срабатывание сигнализации, которая в свою очередь запустит звуковое оповещение, световую индикацию и оповестит владельца системы о проникновении посредством GSM модуля. Чтобы отключить систему охраны требуется ввести специальный код. Если код был введен верно, то система перейдет в состояние, когда охрана не осуществляется. Чтобы из этого состояния перевести систему в режим полной или частичной охраны требуется нажать на соответствующую кнопку на лицевой панели. Когда владелец собирается покинуть охраняемое помещение и включить систему, он должен нажать на соответствующую кнопку. После нажатия у него будет определенное количество времени

для того, чтобы покинуть помещение. На протяжении этого времени информация с датчиков передаваться не будет. Если система находится в режиме полной охраны, то владельцу сигнализации будет сообщаться с помощью SMS сообщений о срабатывании любого датчика. Если кто-то из тех, кто имеет доступ к помещению, зайдет внутрь и отключит систему или переведет ее в режим частичной охраны, то об этом также будет сообщено владельцу. Такая функция может пригодиться в случае того, когда кто-то посторонний смог взломать дверь и отключить сигнализацию. Если владелец знает, что в данный момент систему не мог отключить никто из доверенных лиц, то он поймет, что произошел несанкционированный доступ и сможет принять соответствующие действия. На лицевой панели также присутствует кнопка «SOS». Эта кнопка позволяет из любого состояния системы связаться со службой спасения или, например, милицией.

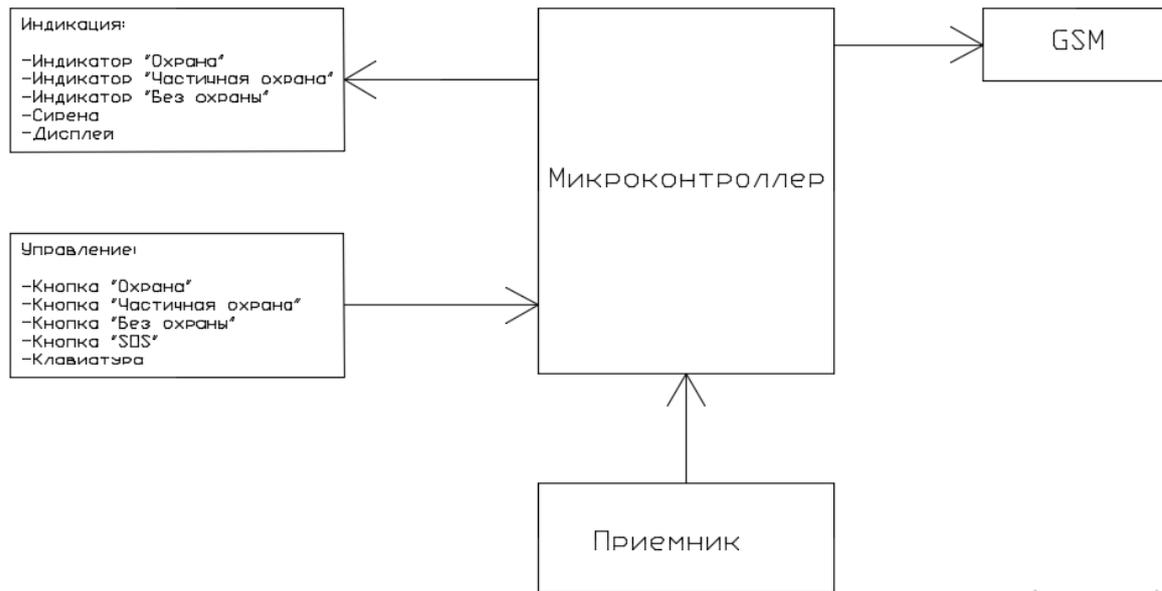


Рис. 1 - Структура системы

Брелок дистанционного управления будет иметь функционал, схожий с функционалом лицевой панели. На пульте будет располагаться 4 кнопки: включение полной охраны, включение частичной охраны, выключение охраны и кнопка SOS.

Список использованных источников:

1. Бессекерский В.А., Попов Е.П. Теория систем автоматического управления, Изд. 4-е, перераб. И доп. — Спб., Профессия, 2007. — 752с.
2. Шаратов В.М., Полищук Е.С., Кошевой Н.Д., Ишанин Г.Г., Минаев И.Г., Совлуков А.С. Датчики: Справочное пособие / . — Москва: Техносфера, 2012. — 624 с.
3. Рюмик С.М. 1000 и одна микроконтроллерная схема. Вып. 1. — М.: Додэка-XXI, 2010. — 356с.

СИСТЕМА КОМПЕНСАЦИИ ГРОМКОСТИ ДЛЯ СЛУХОВОГО АППАРАТА НА ОСНОВЕ КОМПРЕССИИ И ПЕРЕНОСА ВЫСОКОЧАСТОТНЫХ КОМПОНЕНТ РЕЧЕВОГО СИГНАЛА

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

Санько Н.С.

Вашкевич М.И. – к.т.н., доцент

Слух является одной из ключевых способностей человека, позволяющей ему общаться с другими людьми и окружающим миром. Нарушение работы органов слуха приводит к тому, что человек лишается значительной доли восприятий и ощущений, приносящих радость и наслаждение. Тем не менее, в современном мире тугоухость является часто встречающимся нарушением здоровья. Одним из перспективных методов его коррекции является применение слухового аппарата и новых методов обработки сигнала и шумоподавления. В работе рассматривается система компенсации громкости на основе компрессии и переноса высокочастотных компонент речевого сигнала.

Для построения системы компенсации громкости необходимо учитывать порог слуха – минимальную

величину звукового давления, при которой звук данной частоты может быть ещё воспринят, и болевой порог – величину звукового давления, при котором в слуховом органе возникают болевые ощущения. У пациентов с тугоухостью порог слуха выше, по сравнению с пациентами с обычным слухом, а болевой порог слышимости ниже. Таким образом они имеют меньший диапазон слышимости. Из этого следует, что с учетом оценки слуха пациента, полоса частот для нормального восприятия звука будет ужата, а высокочастотная составляющая звука будет болезненна для восприятия.

Цель алгоритма обработки заключается в переходе из более широкой полосы частот в более усреднённую для более комфортного восприятия звука с учетом пороговых значений слышимости. Затем сигнал проходит через квадратурно-зеркальный банк фильтров, который разделяет сжатый и смещенный в определённую полосу частот сигнал на четыре неравные полосы, затем производится компенсация громкости и усиление. Завершающей стадией является повторное пропускание сигнала через квадратурно-зеркальный банк фильтров для восстановления четырёх суб-полос широкополосный сигнал.

Система компенсации громкости для слухового аппарата на основе компрессии и переноса высокочастотных компонент речевого сигнала разделена на модули.

Первый модуль отвечает за высокочастотное сжатие сигнала и выполняет следующие действия:

- 1) выбор полосы частот, которую нужно сместить, а также выбор полосы, в которую она будет смещена (– максимальная частота исходной полосы; минимальная частота исходной полосы; максимальная частота выбранной полосы; минимальная частота выбранной полосы);
- 2) расчёт коэффициента сжатия:

(1)

- 3) преобразование Фурье;
- 4) сжатие сигнала с учетом частотной компрессии;

(1)

- 5) обратное преобразование Фурье.

После обработки сигнала первым модулем он проходит через квадратурно-зеркальный банк фильтров анализа. Квадратурно-зеркальный банк фильтров представляет собой набор фильтров с конечной импульсной характеристикой, частотная характеристика которого симметрична относительно половины частоты дискретизации. Далее следует модуль компенсации громкости и усиления сигнала. Последней стадией является восстановление обработанного сигнала с использованием квадратурно-зеркального банка фильтров синтеза.

Компенсация громкости и усиление является последним этапом в данной системе, так как при компрессии сигнала на выходе достаточно сложно разобрать обработанную речь, потому что становится достаточно тихой из-за алгоритма сжатия и сложно оценить результат исследований.

Схема реализации системы компенсации громкости представлена на рисунке 1.



Рис. 1 - Структура системы компенсации громкости

Система компенсации громкости для слухового аппарата на основе компрессии и переноса высокочастотных компонент речевого сигнала имеет ряд преимуществ:

- система может с помощью алгоритма компрессии и определённого рода фильтрации сигнала, компенсировать высокочастотную область речевого сигнала для комфортного восприятия пациентом;
- система основана на алгоритме, эффективность которого превосходит разработанные алгоритмы в данной тематике на 10-15%;

Результаты работы данной системы представлены на рисунке 2:

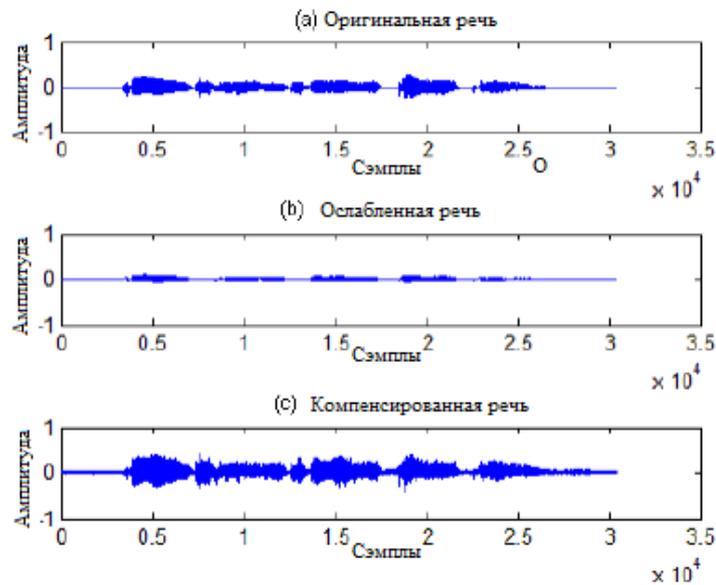


Рис. 2 - Результаты системы компенсации громкости

Таким образом данная система имеет явное превосходство над другими современными алгоритмами компенсации громкости, что делает её инновационной на данный момент времени.

Список использованных источников:

1. Deniz B., Robert V., Combined Effects of Frequency Compression-Expansion and Shift on Speech Recognition. Ear & Hearing, 2007, vol. 28, no. 3, pp. 277-289.
2. Miller-Hansen D.R., Nelson P.B., Widen J.E., Simon S.D. Evaluating the Benefit of Speech Recoding Hearing Aids in Children. American Journal of Audiology, 2003, vol. 12, pp. 106-113.
3. Wyrsh S., Kaelin A. Sub-band signal processing for hearing aids. Proceedings of the 1999 IEEE International Symposium on Circuits and Systems, 1999, vol. 3, pp. 29-32.

ПРИМЕНЕНИЕ ЛОГАРИФМИЧЕСКОЙ СИСТЕМЫ СЧИСЛЕНИЯ ДЛЯ АЛГЕБРЫ КВАТЕРНИОНОВ

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

Сапронова Ю.И.

Петровский Н.А. – к.т.н., доцент

Алгебра кватернионов находит применение в вычислительной механике, трехмерной графике, а также цифровой обработке сигналов. Умножение кватернионов является затратной операцией. Применение альтернативного представления кватернионов с использованием логарифмической системы счисления позволяет сократить количество операций, а также сократить затраты памяти на хранение коэффициентов.

Алгебра кватернионов (\mathbb{H}) представляет собой четырехмерную ассоциативную алгебру над полем \mathbb{R} , имеющую базис со следующими правилами умножения [1]:

(1)

Такая алгебра находит применение во многих приложениях. Особое место при этом занимает умножение кватернионов, которое имеет ряд особенностей. Во-первых, оно некоммутативно, а во-вторых, является довольно затратной операцией, потому как включает в себя 16 действительных умножений и 12

сложений. В связи с этим возникает задача сокращения количества операций.

Для решения данной задачи может применяться логарифмическая система счисления, позволяющая сократить количество операций.

Как показано в [2] применение логарифмической системы счисления для умножения кватернионов, представленных в виде $q = a + bi + cj + dk$, не приносит никакого выигрыша в количестве реализуемых операций ввиду некоммутативности произведения, т.е.

Другим способом представления кватернионов является конструкция Кэли-Диксона [3]:

$$Q = \begin{pmatrix} a + bi & 0 \\ 0 & a - bi \end{pmatrix} \begin{pmatrix} c + di & 0 \\ 0 & c - di \end{pmatrix} \quad (2)$$

где a, b, c, d и i, j, k представляют собой комплексные числа. Произведение кватернионов в данном случае можно найти как

$$Q_1 Q_2 = \begin{pmatrix} a_1 + b_1 i & 0 \\ 0 & a_1 - b_1 i \end{pmatrix} \begin{pmatrix} c_1 + d_1 i & 0 \\ 0 & c_1 - d_1 i \end{pmatrix} \quad (3)$$

где \bar{a} – комплексное сопряжение a и соответственно.

Таким образом, представление чисел в логарифмической системе счисления по произвольному основанию b [4] выглядит следующим образом:

$$N = \sum_{k=0}^{n-1} d_k b^k \quad (4)$$

где b – масштабирующий коэффициент, a и i – коэффициенты логарифма кватерниона q и i , представленных в виде (2).

Подставив значение коэффициентов (4) в выражение произведения (3) можно найти результирующее значение. Следует отметить, что в данном случае количество операций сокращается до 8 операций логарифмического умножения и двух операций комплексно-логарифмического сложения. Кроме того, применение логарифмической системы счисления в данном случае позволяет сократить затраты памяти на хранение коэффициентов, поскольку логарифм числа кодируется меньшим количеством битов, нежели само число.

Список использованных источников:

1. Parfieniuk, M., Petrovsky, A. Quaternion Multiplier Inspired by the Lifting Implementation of Plane Rotations // IEEE Transactions on Circuit and Systems, 2010. – pp. 2708-2717.
2. Arnold, Mark G. Towards a Quaternion Complex Logarithmic Number System // 20th IEEE Symposium on Computer Arithmetic, 2011. – pp. 33-42.
3. Sangwine, Stephen 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. Arnold, Mark G. A Dual-Purpose Real/Complex Logarithmic Number System ALU // 19th IEEE International Symposium on Computer Arithmetic, 2009. – pp.15-24

СИСТЕМА ОПРЕДЕЛЕНИЯ ГЕОМЕТРИЧЕСКИХ ПАРАМЕТРОВ И РАЗМЕРОВ КРИСТАЛЛОВ СЛОЖНОЙ ФОРМЫ

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

Сидорук И.И.

Шемаров А.И. – к.т.н., доцент

В наши дни потребности мирового алмазного рынка диктуют новые требования в области контроля и исследования необработанного алмазного сырья. Так, стараясь удовлетворить эти требования, все ключевые предприятия, специализирующейся в области сортировки и обработки алмазного сырья, вынуждены постоянно искать новые подходы к организации технологического процесса сортировки и аттестации кристаллов и заниматься техническим перевооружением производства. Однако существующие программно-аппаратные комплексы не в полной мере удовлетворяют предъявляемые запросы реального производства. Усовершенствование существующих и разработка новых систем сканирования алмазного сырья и ввода оцифрованных параметров геометрии кристалла в компьютер для решения задачи определения формы кристаллов, для их последующей разметки и разработки, – может позволить получить значительный экономический эффект.

Основным методом, используемым и в системах получения информации о геометрии кристалла, является метод математического моделирования. Параметры и свойства таких систем во многом определяются используемыми математическими моделями и алгоритмами [1].

Принцип получения оригинальной 3D-модели внешней поверхности кристалла в декартовой системе координат (рисунок 1) основывается на математическом описании объекта по определенному алгоритму с целью последующего построения поверхности кристалла [2].

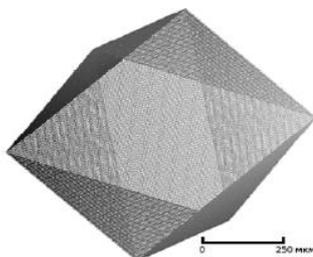


Рис. 1 – Пример 3D-модели кристалла алмаза в декартовой прямоугольной системе координат

Система (экспериментальная установка) содержит следующие составные части:

1. Держатель для необработанного объекта (кристалла сложной формы);
2. Систему сканирования, используемую для автоматического получения информации о геометрических параметрах и размерах объекта (кристалла сложной формы);
3. Систему освещения кристалла в виде двух (или более) лазерных пучков, направляемых по двум отдельным оптическим траекториям;
4. Систему формирования изображения кристалла сложной формы;
5. Вычислительное средство, реализующее возможность определения формы кристалла на основе принятого излучения от освещающей системы и указанной начальной информации о геометрической форме кристалла.

Устройство выполнено с возможностью приведения сканируемого объекта во вращение относительно осветительной системы [3] для предоставления пользователю более точной визуальной модели объекта.

Используя математическое описание, на основе знания пространственного расположения элементов системы освещения объекта, данные о показателе преломления исследуемых образцов кристаллов, можно построить трехмерные модели реальных кристаллов сложной геометрической формы. Полученный набор данных позволяет построить 3D виртуальную модель внешней поверхности кристалла и 3D виртуальные модели одного или нескольких видимых включений во внутреннем объеме кристалла. Эти данные могут быть использованы для последующего компьютерного анализа с целью предоставления пользователю информации, относящейся к визуальным характеристикам кристаллов сложной формы, которая может быть получена из исходного кристалла.

Разработка или усовершенствование существующих алгоритмов и систем определения геометрических параметров кристаллов сложной формы, являющихся целью представленной работы, существенно могут повысить эффективность алмазного производства, в том числе и ювелирного.

Список использованных источников:

1. Горбунова, Е.В. Возможности сортировки алмазного сырья опико-электронными методами / Е.В. Горбунова // Научно-технический вестник информационных технологий, механики и оптики. – СПб.: ИТМО, 2012. – №4 – С. 13 – 17.
2. Григорьев, Ю.М. Сравнительный анализ оптических свойств бриллиантов / Ю.М. Григорьев // Наука и образование. – М.: Наука, 2014. – №1 – С. 45 – 49.
3. Abraham Kerner., System for accurate 3d modeling of gemstones. PatentUS, no. 20160004926A1, 2012.

ДЕТЕКТИРОВАНИЕ ЛИЦ НА ИЗОБРАЖЕНИИ С ОДНОВРЕМЕННОЙ КЛАССИФИКАЦИЕЙ ЛИЦ С ПОМОЩЬЮ СВЁРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ

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

Суша А.В.

Вашкевич М.И. – к.т.н., доцент

Задача детектирования лиц является одной из самых важных в области компьютерного зрения, т.к. детектирование необходимо практически всем системам, которые оперируют изображениями человека. В работе рассматривается применение перспективного подхода детектирования лиц или объектов YOLO (You Only Look Once) на основе базовых моделей: VGG-16 и NasNet.

В настоящее время существует множество способов детектирования лиц. Большинство из них базируется на классических методах компьютерного зрения и машинного обучения (каскады Хаара,

HOG+SVM). Неоспоримым преимуществом этих подходов является высокая скорость работы. Однако данные методы имеют такие недостатки, как большое число ложных срабатываний и ограниченное число ракурсов лиц, на которых возможно проведение детектирования. Применение методов глубокого обучения позволяет решить эти недостатки. А более низкая скорость работы в настоящее время компенсируется наличием более производительных платформ.

Описываемый в данной работе метод детектирования лиц основывается на методе *YOLO*, описанный в [1] и усовершенствованный в [2]. Привлекательность метода *YOLO* заключается в том, что изображение анализируется свёрточной нейронной сетью только один раз, в отличие от подходов, базирующихся на применении скользящего окна. Также *YOLO* позволяет проводить некоторую дополнительную классификацию детектируемых объектов, например пол человека, выражение лица. В этих работах в качестве базовой модели использовалась архитектура сети *DarkNet*, разработанная авторами *YOLO*. Ключевым отличием настоящей работы является применение подхода *YOLO* к иным базовым моделям: *VGG-16* [3] и *NasNet* [4] и своя архитектура *FaceNet*. Также некоторые отличия имеются в применяемой для обучения функции.



На рисунке 1 изображён процесс детектирования объектов с помощью *YOLO*. Входное изображение разбивается на ячейки некоторого размера, которые формируют сетку размером $H \times W$. *YOLO* осуществляет предсказание как позиции рамки объекта, так и класса, к которому этот объект принадлежит. В общем виде процесс детектирования записывается следующим выражением:

где I – входное изображение;
 D – детектор;
 O – выход детектора;
 F – функция преобразования выхода сети в список объектов;
 L – список найденных объектов;

где p – вероятность обнаружения объекта, b_x и b_y – смещение по горизонтали и вертикали центра рамки объекта, в относительных координатах внутри ячейки, w и h – ширина и высота рамки в нормированных координатах относительно размера изображения, c – класс объекта, p_{obj} – вероятность обнаружения объекта, p_{cls} – вероятность принадлежности объекта к классу c .

В качестве функции ошибки для минимизации использовалась сумма функций ошибок для каждой величины, описывающей объект: для вероятности обнаружения объекта – средняя бинарная перекрёстная энтропия, для рамки лица – сумма средних квадратических ошибок для координат центра рамки и логарифмов ширины и высоты рамки и для класса объекта – средняя категориальная перекрёстная энтропия. Общая функция ошибки записывается следующим образом:

где N – число примеров;
 p_i – целевое значение вероятности обнаружения объекта i -го примера;
 \hat{p}_i – предсказанное значение вероятности обнаружения объекта i -го примера;
 $\{i\}$ – множество индексов примеров, для которых $p_i > 0$;
 $\{x_i, y_i, w_i, h_i\}$ – целевые значения смещения центра и размера рамки объекта i -го примера;
 $\{\hat{x}_i, \hat{y}_i, \hat{w}_i, \hat{h}_i\}$ – предсказанные смещения центра и размера рамки объекта i -го примера;
 K – число классов объектов;
 p_{ij} – целевое значение вероятности принадлежности объекта к j -го классу i -го примера;

–предсказанное значение вероятности принадлежности объекта к j -го классу i -го примера.

Обучение проводилось на базах данных *FaceSrub* [5] и *IMDB* [6]. Несмотря на наличие более 350 тысяч обучающих примеров, в указанных базах размечена рамка лица только одного человека даже если людей изображено несколько. Но как оказалось даже в таких условиях свёрточные сети способны обучаться. Для оценки качества обучения использовалась база *Fddb* [7] содержащая 2845 вручную размеченных изображений лиц. Результат тестирования приведен в таблице 1. В качестве метрик использовались точность, полнота, *f1-score* и также пересечение над объединением (англ. *IoU – intersection over union*). Пороговым значением вероятности детектирования – 0,5; Пороговое значение *IoU* для алгоритма подавления не максимумов – 0,4 и для фиксирования правильно сдетектированного лица – 0,5.

Таблица 1 – результаты тестирования

Базовая модель	Точность, %	Полнота, %	F1-Score, %	IoU, %
<i>VGG-16</i>	12,02	11,01	11,49	60,59
<i>NasNet</i>	99,07	85,84	91,98	78,07
<i>FaceNet</i>	88,70	80,72	84,52	70,28

Исходя из приведённых результатов можно заключить, что *VGG-16* обучилась довольно плохо. К тому же время, затраченное на обучение этой сети, в несколько раз больше времени, затраченного на обучение *NasNet*. С одной стороны *NasNet* имеет меньшее число параметров, но с другой – более сложную архитектуру. Это хорошо показывает, что архитектура нейронной сети оказывает существенное влияние на качество её работы. В настоящее время проводится работа по созданию и обучению собственной архитектуры базовой модели – *FaceNet*. Основной акцент при проектировании делается на возможности применения этой модели на мобильных вычислительных платформах с сохранением высокой точности детектирования.

Список использованных источников:

1. Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, You Only Look Once: Unified, Real-Time Object Detection // arXiv preprint, arXiv:1506.02640. – 2015. – 10 p.
2. Joseph Redmon, Ali Farhadi, YOLO9000: Better, Faster, Stronger // arXiv preprint, arXiv:1612.08242 – 2016. – 9 p.
3. Karen Simonyan, Andrew Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition // arXiv preprint, arXiv:1409.1556. – 2014. – 14 p.
4. Barret Zoph, Vijay Vasudevan, Jonathon Shlens, Quoc V. Le, Learning Transferable Architectures for Scalable Image Recognition // arXiv preprint, arXiv:1707.07012. – 2017. – 14 p.
5. Hong-Wei Ng, Stefan Winkler, A Data-driven approach to cleaning large face datasets // In Proc. IEEE International Conference on Image Processing – 2014. – pp. 27-30.
6. Rasmus Rothe, Radu Timofte, Luc Van Gool, Deep expectation of real and apparent age from a single image without facial landmarks // International Journal of Computer Vision. – 2016. – pp. 144-157.
7. Vidit Jain, Erik Learned-Miller, Fddb: A Benchmark for Face Detection in Unconstrained Settings // Technical Report UM-CS-2010-009, Dept. of Computer Science, University of Massachusetts. – 2010. – 11 p.

СИСТЕМЫ ФАЗОВОЙ АВТОПОДСТРОЙКИ ЧАСТОТЫ

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

Дорошкевич М.В., Зайцева А.И., Фоменок А.А.

Качинский М. В. – к.т.н., доцент

Вопросы синхронизации требуют особого внимания, когда речь идет о распространении цифровой информации. При использовании фазоманипулированных сигналов (*phase-shiftkeying* (*PSK*)) необходимо различать передаваемые символы по фазе. Однако опорные генераторы на передающей и приемной стороне не могут быть согласованы. Для решения этой проблемы используются следящие контуры (петли), основная задача которых производить подстройку генераторов опорных сигналов для когерентной демодуляции. Такие системы получили название ФАПЧ.

Система фазовой автоподстройки частоты — система автоматического регулирования, которая подстраивает фазу управляемого генератора так, чтобы достиглось равенство с фазой опорного сигнала, или отличие составляло бы заданную временную функцию. Системы ФАПЧ широко применяются для изменения частотных параметров сигнала, а также для выделения опорного колебания.

Основная схема ФАПЧ представлена на рисунке 1.



Рис. 2— Основная схема ФАПЧ

Как видно на схеме, система состоит из 4 основных компонентов: фазового детектора, фильтра нижних частот (ФНЧ), регулируемого генератора и делителя частоты.

Система ФАПЧ может быть реализована разными способами: с использованием полностью цифровых, полностью аналоговых или смешанных схем.

В настоящее время выделяют следующие разновидности ФАПЧ:

- Аналоговые или линейные (APLL)
- Цифровые (DPLL)
- Полностью цифровые (ADPLL)
- Программные (SPLL)
- Нейрональные (NPLL)

Принцип работы цифровой ФАПЧ похож на работу аналоговой, но вместо генератора, управляемого напряжением, используются системные часы и счётчик-делитель. ЦФАПЧ достаточно просты как в реализации, так и в разработке. В отличие от аналоговых ФАПЧ, цифровые менее чувствительны к шумам напряжения, однако обычно она допускает фазовый шум по причине наличия шума квантования при использовании цифрового генератора.

Еще одной важной проблемой систем ФАПЧ является проблема совмещения достаточной ширины полосы захвата с возможностью подавления дрожания фазы. Чем лучше характеристики подавления дрожания фазы, тем уже полоса захвата. Эту задачу можно решить с помощью метода имитационного моделирования с использованием пакета MatLab. В работе рассмотрена система, в которой используется петлевой фильтр нового типа. В этой системе отчасти решена обозначенная проблема.

Система ФАПЧ является одним из основных современных аппаратных узлов. Она позволяет эффективно заменить катушки индуктивности и дроссели при реализации частотной селекции, точной настройки фильтрации.

Список использованных источников:

1. Информационный портал HintFox [Электронный ресурс]. – Режим доступа: <http://www.hintfox.com/>. – Дата доступа: 05.04.2017.
2. Исследование цифровой фазовой автоподстройки частоты/Успенко В. Б.[и др.] – Пермь, Россия: ПНИПУ, 2015 – 20с
3. Цифровые системы фазовой синхронизации/ М.И. Жодзишский [и др.]: Советское радио, 1980. – 208с.: с ил.

ВЕРИФИКАЦИЯ VHDL-МОДЕЛЕЙ КОНЕЧНЫХ АВТОМАТОВ

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

Щепанский В.Е.

Бибилко П. Н. – д. т. н., профессор

Современная цифровая аппаратура на основе интегральных схем (ИС) представляет собой сложные устройства, проектирование и производство которых требует больших затрат ресурсов. Цена ошибки в аппаратуре может оказаться очень высокой: известный случай замены микропроцессоров IntelPentium с ошибкой деления обошелся компании приблизительно в 500 миллионов долларов. Так как исправление ошибок в уже готовых микросхемах невозможно, поиск и нахождение функциональных ошибок проводится на этапе проектирования HDL-описания устройства. Подобного рода деятельность, состоящая в проверке соответствия поведения аппаратуры, задаваемого HDL-описанием, его спецификации, называется верификацией [1]. Под верификацией будем понимать проверку правильности исходного VHDL-описания, т. е. проверку соответствия составленного синтезируемого VHDL-описания проектируемой цифровой системы спецификациям на проектирование [2]. Для полной верификации цифрового устройства с состояниями

требуется проверить наличие всех переходов в графе состояний проверяемого устройства. Это можно сделать двумя способами, подать на проверяемую модель большое количество псевдослучайных входных наборов, построить граф состояний и проверить наличие всех дуг, или подать “небольшой” тест, который при правильной работе модуля позволяет полностью пройти граф состояний, и проверить известные реакции модели. Ясно, что второй способ является более удобным и быстрым. Нахождение такого теста и есть одна из основных задач верификации VHDL-модели конечного автомата.

Под тестом понимается упорядоченная последовательность наборов значений входных сигналов, которые при моделировании подаются на входные порты VHDL-модели автомата. Тесты, которые позволяют проверить правильность функционирования исследуемой модели автомата, называются функциональными [1].

Для решения этой проблемы был составлен алгоритм нахождения теста, позволяющий пройти граф состояний по всем дугам. Суть алгоритма заключается в следующем:

- 1) генерируется тест, состоящий из псевдослучайных наборов входных сигналов;
- 2) на основе полученного теста проводится моделирование и в результате моделирования становятся известны реакции модели и пройденные состояния;
- 3) на основе пройденных состояний автомата формируется граф состояний;
- 4) находится путь обхода графа по всем дугам;
- 5) при помощи этого пути получаем тест для обхода графа и наборы ожидаемых выходных сигналов.

Полученного теста достаточно для проведения полной верификации автомата. Для примера рассмотрим граф переходов состояний конечного автомата (рис. 1).

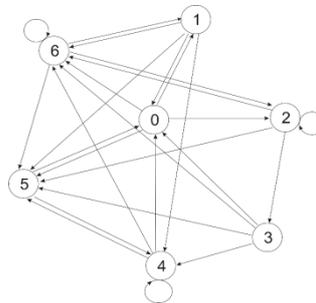


Рисунок 1 – Граф переходов состояний автомата

Для полного прохождения всех дуг графа были поданы тесты, состоящие из псевдослучайных наборов. Наименьший тест, который дал полное прохождение графа, состоял из 512 псевдослучайных наборов. После выполнения предложенного алгоритма был получен тест, включающий 54 набора. Переходы между состояниями этого автомата имеют вид:

0 → 6 → 6 → 1 → 4 → 4 → 5 → 4 → 6 → 2 → 2 → 6 → 5 → 0 → 6 → 1 → 6 → 5 → 4 → 6 → 1
 → 0 → 6 → 1 → 5 → 4 → 6 → 1 → 0 → 5 → 4 → 6 → 2 → 5 → 0 → 2 → 3 → 6 → 1 → 4 → 0 →
 2 → 3 → 4 → 5 → 0 → 2 → 3 → 5 → 0 → 2 → 3 → 0 → 1 → 0.

Число наборов полученного теста сократилось примерно в 10 раз.

Список использованных источников:

1. Чэнь М., Цинь К., Ку Х.-М., Мишра П. Валидация на системном уровне. Высокоуровневое моделирование и управление тестированием. М.: Техносфера, 2014. – 296 с.
2. Бибило П.Н., Романов В.И. Построение компактных тестов для функциональной верификации VHDL-описаний конечных автоматов // Управляющие системы и машины. - 2017. - № 1. – с. 35 – 45.

Научное издание

КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ

**МАТЕРИАЛЫ 54-Й НАУЧНОЙ КОНФЕРЕНЦИИ
АСПИРАНТОВ, МАГИСТРАНТОВ И СТУДЕНТОВ**

(Минск, 23-27 апреля 2018 года)

В авторской редакции

Ответственный за выпуск *Е.В. Калабухов*
Компьютерная верстка: *А.Ф. Шиханцова, З.К. Кот, А. Артамонов*