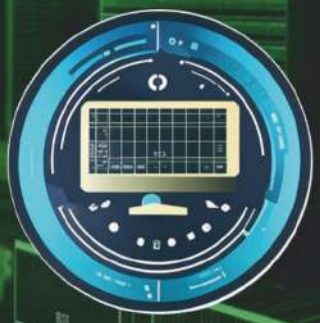
ISBN 978-5-907811-43-0

МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ 2024 ГОД

Межвузовский сборник научных трудов

МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ



2024

межвузовский сборник научных трудов

МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

РГРТУ, 2024

УДК 004.4
М 33

Математическое и программное обеспечение вычислительных систем: Межвуз. сб. науч. тр. / Под ред. Г. В. Овечкина – Рязань: ИП Коняхин А.В. (Book jet), 2024 – 206 с.

Рецензент:
к.т.н., доцент, зав. каф. «Информатики, информационных технологий и защиты информации» Липецкого государственного педагогического института Д. М. Скуднев

ISBN 978-5-907811-43-0

Представлены статьи, посвящённые различным аспектам разработки программных средств ЭВМ, вычислительных систем и сетей, вопросам математического моделирования, методам обработки информации.

Предназначен для научно–педагогических работников вузов, инженеров, аспирантов и студентов старших курсов.

Авторская позиция и стилистические особенности публикаций полностью сохранены.

ISBN 978-5-907811-43-0

ВВЕДЕНИЕ

Межвузовский сборник научных трудов содержит результаты научных исследований и разработок по следующим направлениям:

– программное обеспечение вычислительных систем, новые информационные технологии;

– прикладная математика, теория информации, искусственный интеллект;

– ЭВМ в системах обработки, управления и обучения;

– автоматизированное проектирование аппаратных средств вычислительных систем;

– информационные технологии в экономических и социальных системах.

Сборник сформирован на основе статей, в которых рассматриваются различные аспекты разработки программных средств ЭВМ, вычислительных систем и сетей, включая вопросы автоматизации проектирования, теории обработки информации и математического моделирования, и предназначен для студентов, аспирантов и преподавателей технических вузов и научных работников.

Материалы для сборника предоставлены сотрудниками и студентами:

– Федерального государственного бюджетного образовательного учреждения высшего образования «Московский государственный технический университет им. Н. Э. Баумана», г. Москва.

– Федерального государственного бюджетного образовательного учреждения высшего образования «Липецкий государственный педагогический университет имени П. П. Семенова-Тян-Шанского», г. Липецк.

– Федерального государственного бюджетного образовательного учреждения высшего образования «Рязанский государственный радиотехнический университет имени В. Ф. Уткина», г. Рязань.

– Федерального государственного бюджетного образовательного учреждения высшего образования «Рязанский государственный университет имени С.А. Есенина», г. Рязань.

УДК 004.932

С.Д. Антонушкина, Г.В. Овечкин, Т.А. Осипова

ИССЛЕДОВАНИЕ МЕТОДОВ ПРОСТРАНСТВЕННОЙ ФИЛЬТРАЦИИ ИЗОБРАЖЕНИЯ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

Рассмотрены различные модели шума и методы пространственной фильтрации изображения. Целью исследования является повышение эффективности фильтрации за счёт выбора наиболее эффективного метода для изображений с разными параметрами.
Ключевые слова: шум, методы устранения шума, оценка качества изображения.

На качество получаемых съемочной аппаратурой изображений влияет множество факторов. Зачастую в результате воздействия этих факторов на изображении могут возникать шумы, для устранения которых существует несколько методов фильтрации изображений. В силу распространенности и актуальности данной проблемы, необходимо установить наиболее эффективные методы для изображений для различных классов изображений.

В данной статье рассмотрены различные модели шума, а также представлены такие способы устранения шума, как размытие по Гауссу, среднеарифметический фильтр, среднегармонический фильтр, медианный фильтр, адаптивный медианный фильтр и фильтр срединной точки. Оценка качества обработанных изображений по оценке SSIM, а также сравнительный анализ полученных результатов, позволят выявить рекомендации по применению различных фильтров устранения шума с изображения.

Модели шума

Шум изображения – это случайное изменение яркости пикселей на изображении. Он может возникать как при передаче изображения из-за несовершенства канала, по которому оно передается, так и непосредственно при съемке.

Существует несколько моделей шума. В данной работе были рассмотрены такие модели как гауссов шум, равномерный шум, экспоненциальный шум, импульсный шум и гамма-шум.

Гауссов шум

Гауссов шум — это статистический шум, имеющий плотность вероятности, равную плотности вероятности нормального распределения (также известного как гауссово).

Плотность вероятности p в этом случае равна:

$$p(z) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{z-\bar{z}}{2\sigma^2}} \qquad (1)$$

где z – значение яркости, z – среднее значение случайной величины z, σ2 – ее среднеквадратическое отклонение. Дисперсией величины z является квадрат её среднеквадратического отклонения σ2.

Равномерный шум

Для равномерного шума функция плотности распределения вероятностей задается выражением:

$$p(z) = \begin{cases} \dfrac{1}{b-a}, & при\ a \le z \le b \\ 0 & в\ остальных случаях \end{cases} \quad (2)$$

Экспоненциальный шум

Функция плотности распределения вероятностей экспоненциального шума задается выражением:

$$p(z) = \begin{cases} ae^{-az}, & при\ z \ge 0 \\ 0, & при\ z < 0 \end{cases} \quad (3)$$

где a > 0.

Импульсный шум

Функция плотности распределения вероятностей (биполярного) импульсного шума задается выражением:

$$p(z) = \begin{cases} P_a\ , & при\ z = a \\ P_b\ , & при\ z = b \\ 1 - P_a - P_b & в\ остальных случаях \end{cases} \quad (4)$$

Импульсный шум называется униполярным, если одно из значений вероятности (P_a или P_b) равно нулю [2]. В этом случае шум выглядит как случайные белые или черный точки. Если же ни одна из вероятностей не равна нулю, то импульсный шум называют шумом типа «соль и перец», так как он выглядит как случайные белые и черные точки и визуально походит на крупицы перца и соли.

Гамма-шум

Функция плотности распределения вероятностей гамма-шума задается выражением:

$$p(z) = \begin{cases} \dfrac{a^b z^{b-1}}{(b-1)!} e^{-az}, & при\ z \ge 0 \\ 0, & при\ z < 0 \end{cases} \quad (5)$$

где a > 0, b — положительное целое число.

Устранение шума с изображения

Устранение шума с изображения посредством пространственной фильтрации происходит при помощи свертки изображения: в исходном изображении вокруг каждого пикселя выделяется окно фильтрации определенного размера (обычно берут нечетные значения, например, выбирают окно размером 3×3 или 5×5 пикселей), затем посредством вычислений над значениями яркостей пикселей данного окна получают новое значение яркости центрального пикселя. Благодаря этому можно исключить сильно выбивающиеся по яркости пиксели. В данной работе были рассмотрены такие способы устранения шума как фильтр Гаусса, среднеарифметический фильтр, среднегармонический фильтр, медианный фильтр, адаптивный медианный фильтр и фильтр срединной точки.

Фильтр Гаусса

Фильтр Гаусса – сглаживающий свёрточный фильтр, матрица свёртки которого состоит из элементов, вычисленных с помощью двумерной функции Гаусса:

$$G(x,y)=\frac{1}{\sqrt{\pi\sigma^2}}e^{-\frac{((x-\frac{k}{2})^2+(y-\frac{k}{2})^2)}{2\sigma^2}} \quad (6)$$

где x, y — координаты точки, k – размер окна фильтрации, а σ — среднеквадратическое отклонение нормального распределения. Чем больше σ, тем сильнее крайние пиксели окна влияют на результат свёртки (значения фильтра Гаусса стремятся к равномерному распределения), тем сильнее размывается изображение. Применение данного фильтра заключается в умножении соответствующих элементов матрицы свертки на соответствующие элементы окна фильтрации, а затем суммировании полученных результатов.

Среднеарифметический фильтр

Среднеарифметический фильтр – яркостное преобразование, в котором в качестве яркости пикселя результирующего изображения с координатами (x,y) используется среднее арифметическое яркостей пикселей исходного изображения, находящихся в окрестности m×n вокруг пикселя исходного изображения с координатами (x,y). То есть в этом фильтре вычисляется математическое ожидание яркостей пикселей, находящихся в окрестности пикселя с заданными координатами.

Пусть S_{xy} – прямоугольная окрестность (окно фильтрации) размерами m×n с центром в точке (x, y), а g(x, y) – значения искаженного изображения по этой окрестности, тогда значение восстановленного изображения $\hat{f}$ в произвольной точке (x,y) может быть представлено следующим образом:

$$\hat{f}(x,y) = \frac{1}{mn} \sum_{(s,t)\in S_{xy}} g(s,t) \quad (7)$$

Среднегармонический фильтр

Среднегармонический фильтр представляет собой следующее преобразование над каждым из пикселей изображения:

$$\hat{f}(x,y) = \frac{mn}{\sum_{(s,t)\in S_{xy}} \frac{1}{g(s,t)}} \quad (8)$$

Данный фильтр целесообразно применять в случае униполярного белого импульсного шума, но для униполярного черного шума среднегармонический фильтр не следует применять для устранения черного униполярного шума [2].

Медианный фильтр

Медианный фильтр – это яркостное преобразование, в котором в качестве яркости пикселя результирующего изображения выступает медиана, вычисленная по пикселям окрестности соответствующего пикселя:

$$\hat{f}(x,y) = \underset{(s,t)\in S_{xy}}{\text{med}} \{g(s,t)\} \quad (9)$$

Медианные фильтры приспособлены для подавления некоторых видов случайных шумов, при этом приводят к меньшему размыванию по сравнению с линейными сглаживающими фильтрами того же размера [3], что позволяет сохранить больше деталей.

Адаптивный медианный фильтр

Адаптивная медианная фильтрация помогает справиться с импульсным шумом. Еще одним преимуществом такого фильтра является то, что такой фильтр «старается сохранить детали» в областях, искаженных не импульсным шумом. Обычный медианный фильтр таким свойством не обладает.

Как и все рассмотренные ранее фильтры, адаптивный медианный фильтр осуществляет обработку в прямоугольной окрестности S_{xy}. Однако, в отличие от других фильтров, он изменяет (увеличивает) размеры окрестности S_{xy} во время работы в соответствии с условиями, описанными в [2].

Фильтр срединной точки

Фильтр срединной точки – это яркостное преобразование на основе максимального и минимального значений яркостей из окрестности пикселя:

$$\hat{f}(x,y)=\frac{1}{2}\left[\max_{(s,t)\in S_{xy}}\{g(s,t)\}+\min_{(s,t)\in S_{xy}}\{g(s,t)\}\right] \quad (10)$$

Оценки качества изображения

Чтобы определить уровень качества полученного изображения, используют различные методы оценки. Для оценки уровня качества изображения обычно берут исходное (не зашумленное) и обработанное изображения и сравнивают их по некоторым параметрам. Мы будем использовать такую оценку, как SSIM.

SSIM (от англ. Structure SIMilarity) – метрика оценки качества изображения по трем критериям: яркость, контрастность и структура. Принимает значения от 0 до 1, при этом чем выше значение, тем ниже искажения изображения и выше качество.

Отличительной особенностью метода, помимо упомянутых ранее (MSE и PSNR), является то, что метод учитывает «восприятие ошибки» благодаря учёту структурного изменения информации. Идея заключается в том, что пиксели имеют сильную взаимосвязь, особенно когда они близки пространственно. Данные зависимости несут важную информацию о структуре объектов и о сцене в целом.

SSIM метрика рассчитана на различные размеры окна. Разница между двумя окнами x и y, имеющими одинаковый размер N×N:

$$SSIM(x,y)=\frac{(2\mu_x\mu_y+c_1)(2\sigma_{xy}+c_2)}{(\mu_x^2+\mu_y^2+c_1)(\sigma_x^2+\sigma_y^2+c_2)} \quad (11)$$

Тут μ(x) – среднее значение для первой картинки, μ(y) – среднее значение для второй картинки, σ(x) – среднеквадратичное отклонение для первой картинки, σ(y) – среднеквадратичное отклонение для второй картинки, σ(x,y) – ковариация. Она находится, как σ(x,y) = μ(x,y) — μ(x) · μ(y). c_1 и c_2 – это поправочные коэффициенты, которые необходимы из-за малости знаменателя. Причем они равны квадрату числа, равному количеству цветов, соответствующему данной битности изображения, умноженной на и 0.01 и 0.03 соответственно.

Как было сказано выше, для проведения исследования были использованы такие типы шумов, как гауссов, равномерный, экспоненциальный, импульсный и гамма-шум. На исходное изображение накладывался определенный тип шума, после чего устранялся одним из методов, описанных ранее. После чего для исходного и полученного в результате обработки изображения подсчитывались значения оценки SSIM. Для более наглядного представления полученных результатов, данные были обработаны и представлены в графическом виде.

Исследования проводились для различных уровней зашумленности, полученное данные были усреднены. Ниже представлены

графики, на которых показаны значения оценки SSIM для различных типов шума (рисунки 1-5).

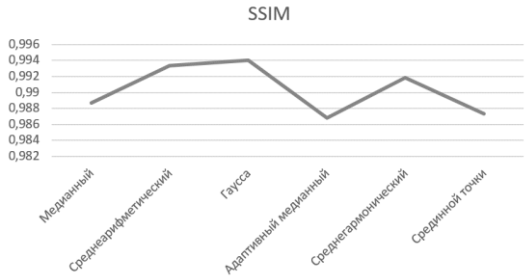


Рисунок 1 – Оценка SSIM для гауссова шума

Из графика видно, что наилучшим фильтром для гауссова шума является размытие по Гауссу, а наименее подходящим – адаптивный медианный фильтр.

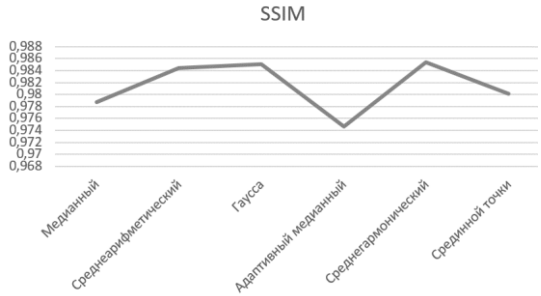


Рисунок 2 – Оценка SSIM для равномерного шума

Из графика видно, что наилучшим фильтром для равномерного шума является среднегармонический фильтр, а наименее подходящим – адаптивный медианный фильтр.

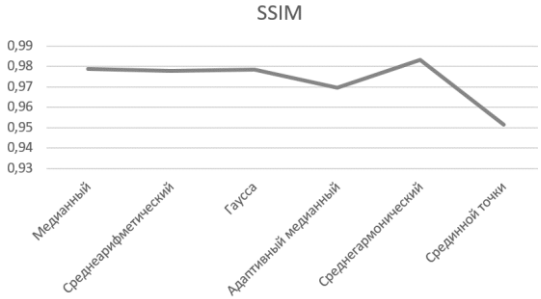


Рисунок 3 – Оценка SSIM для экспоненциального шума

Из графика видно, что наилучшим фильтром для экспоненциального шума является среднегармонический фильтр, а наименее подходящим – фильтр срединной точки.

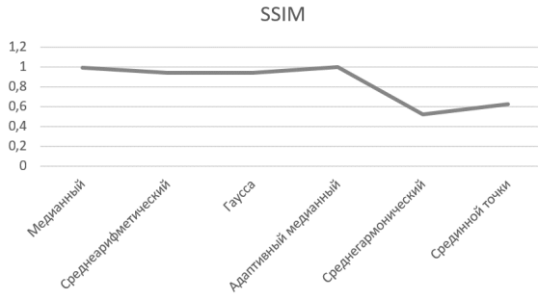


Рисунок 4 – Оценка SSIM для импульсного шума

Из графика видно, что наилучшим фильтром для импульсного шума является адаптивный медианный фильтр, а наименее подходящим – среднегармонический фильтр.

Приведем пример работы медианного и адаптивного медианного фильтров в случае импульсного шума. На рисунке 5 представлено зашумленное изображение, на рисунке 6 представлен результат фильтрации медианным фильтром, а рисунок 7 показывает результат фильтрации при использовании адаптивного медианного фильтра.

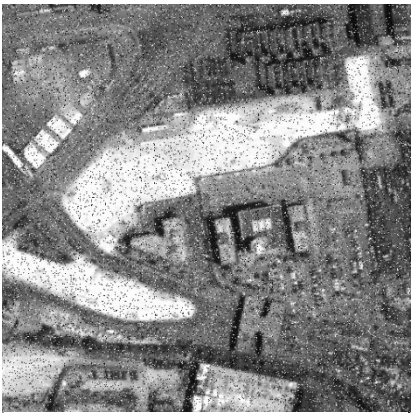


Рисунок 5 – Зашумленное изображение

Рисунок 6 – Медианная фильтрация

Рисунок 7 – Адаптивная медианная фильтрация

Данный пример наглядно демонстрирует большую степень сохранения краёв в случае использования адаптивного медианного фильтра по сравнению с обычным медианным фильтром.

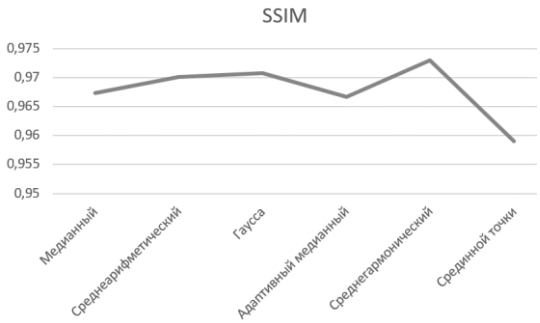


Рисунок 8 – Оценка SSIM для гамма-шума

Из графика видно, что наилучшим фильтром для гамма-шума является среднегармонический фильтр, а наименее подходящим – фильтр срединной точки.

Заключение

Проведенное исследование позволило оценить целесообразность применения тех или иных методов фильтрации изображения для заданного типа шума. Применение полученных результатов позволит более эффективно устранять шум с изображения. Дальнейшие исследования могут быть направлены на изучение других методов устранения шума с изображения.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Самойлин Е.А., Карпов С.А. Программная модель для исследования эффективности процедур выделения контуров зашумленных изображений // – 2018. 9с

2. Гонсалес Р., Вудс Р. Цифровая обработка изображений. М.: Техносфера // – 2012. 1104 с.

3. Грузман И.С., Киричук В.С., Косых В.П., Перетягин Г.И., Спектор А.А. Цифровая обработка изображений в информационных системах // – Учебное пособие.– Новосибирск: Изд-во НГТУ, 2000. – 168.

4. В.Г. Шубников, С.Ю. Беляев Подавление шума и оценка различий в изображениях // – Научно-технические ведомости СПбГПУ 3' (174) 2013 Информатика. Телекоммуникации. Управление

УДК 681.7.014.3

В. М. Архипкин

ПРИМЕНЕНИЕ ВЕЙВЛЕТ-ПРЕОБРАЗОВАНИЙ ХААРА ДЛЯ УЛУЧШЕНИЯ КАЧЕСТВА ИЗОБРАЖЕНИЙ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

В статье рассматривается улучшение качества изображений с использованием вейвлет-преобразования Хаара. Проблема решается применением дискретных вейвлет-преобразований для коррекции артефактов. Результаты показывают эффективность метода для фильтрации и восстановления изображений.

Ключевые слова: вейвлет-преобразование Хаара, улучшение качества изображений, дискретное вейвлет-преобразование, фильтрация изображений, обработка изображений.

В статье рассматривается задача улучшения качества изображений путем применения вейвлет-преобразования Хаара. Цель исследования — это изучение возможностей фильтровой реализации прямых и обратных дискретных вейвлет-преобразований.

Вейвлет-преобразования являются одними из наиболее эффективных и мощных инструментов обработки изображений. Эти методы предоставляют уникальные возможности для анализа, обработки и сжатия изображений, что делает их неотъемлемой частью современных технологий обработки видеоданных [1].

Дискретное двумерное вейвлет-преобразование - это метод математического анализа сигналов, который разбивает изображение на различные уровни масштаба и направления. Этот метод обеспечивает точное определение частотных характеристик сигнала и его

пространственное распределение, что позволяет эффективно анализировать разнообразные структуры и текстуры на изображении.

В работе рассматривается быстрое преобразование Хаара, основанное на использовании вейвлета Хаара. Этот вейвлет представляет собой кусочно-постоянные функции, определенные на конечных интервалах различных масштабов и принимающие значения {-1; +1}. Материнский вейвлет Хаара - функция, равная +1 на интервале [0; 1/2) и -1 на интервале [1/2; 1). Вейвлеты Хаара широко применяются в обработке дискретных сигналов, таких как аудиозаписи и цифровые изображения. Благодаря своей простой структуре и вычислительной эффективности, преобразование Хаара является отличным выбором для многих приложений обработки данных изображений [2].

Преобразование Хаара относится к одним из наиболее простых базисных вейвлет-преобразований. Рассмотрим одномерный дискретный сигнал.

Процедура Хаара разбивает каждый сигнал на две составляющие: среднюю и разностную [3].

Первое среднее значение подсигнала $a^1 = (a_1, a_2, ..., a_{N/2})$ на первом уровне для одного сигнала длиной N вычисляется по формуле:

$$a_n = \frac{f_{2n-1} + f_{2n}}{\sqrt{2}}, n = 1, 2, 3, ..., N/2 \quad (1)$$

А первый детализирующий подсигнал $d^1 = (d_1, d_2, ..., d_{N/2})$ на таком же уровне представляется формулой:

$$d_n = \frac{f_{2n-1} - f_{2n}}{\sqrt{2}}, n = 1, 2, 3, ..., N/2 \quad (2)$$

Эти значения формируют два новых сигнала:

$$a = \{a_n\}, n \in Z \quad (3)$$

$$d = \{d_n\}, n \in Z \quad (4)$$

Первый из которых является огрубленной версией исходного сигнала (каждой паре элементов f соответствует их среднее арифметическое), а другой содержит информацию (будем называть ее детализирующей), необходимую для восстановления исходного сигнала. Действительно:

$$f_{2n-1} = a_n + d_n, n \in Z \quad (5)$$

$$f_{2n} = a_n - d_n, n \in Z \quad (6)$$

Для декомпозиции изображения сначала применяется одномерное преобразование Хаара к каждой строке матрицы изображения, а затем к

каждому столбцу. Полученными значениями являются все детализирующие коэффициенты, за исключением единственного общего среднего коэффициента [3].

Другими словами, двумерное вейвлет-преобразование состоит в последовательном применении одномерных вейвлет-преобразований к строкам и столбцам матрицы. Сначала строки подвергаются одномерным вейвлет-преобразованиям, а затем вейвлет-преобразования применяются ко всем столбцам. В результате изображение разбивается на четыре равные части (см. рисунок 1) [3].

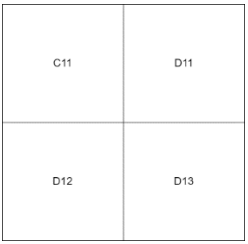


Рисунок 1 – Схема однократного применения
двумерного вейвлет-преобразования изображения

На рисунке 1 изображены четыре компонента преобразованного изображения: C11, D11, D12, D13. Компонент C11 отражает низкочастотные вейвлет-коэффициенты, а D* – высокочастотные. Например, D11 представляет вертикальные низкочастотные составляющие, D12 – горизонтальные, а D13 – диагональные [3].

Под N-кратным двумерным вейвлет-преобразованием понимается последовательное применение двумерного вейвлет-преобразования N раз. При этом каждое следующее двумерное вейвлет-преобразование применяется к низкочастотной части матрицы (компонент C11 на рисунке 1).

Тогда четырехкратное преобразование (N = 4) будет выглядеть, как показано на рисунке 2.

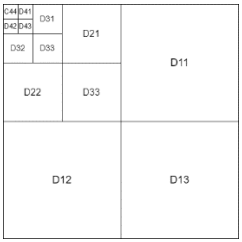


Рисунок 2 – Схема четырехкратного применения
двумерного вейвлет-преобразования изображения

Обратное двумерное вейвлет-преобразование рекурсивно восстанавливает младший компонент путем использования более высоких компонент. Например, для восстановления компонента C33 изображения на рисунке 2, необходимо использовать компоненты C44, D41, D42 и D43. После этого для восстановления компонента C22 используются компоненты C33, D31, D32, D33 и так далее. Таким образом, можно выполнять обратное вейвлет-преобразование N раз.

В рамках проведенного исследования было разработано приложение, которое выполняет преобразование Хаара до четвертого порядка. Демонстрация работы приложения представлена на рисунке 3.

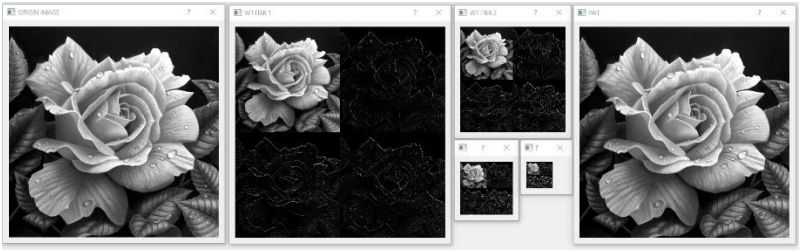


Рисунок 3 – Программная реализация преобразования Хаара

Рассмотренное преобразование может быть использовано для коррекции артефактов на изображениях, например «полосатости». Когда мы раскладываем изображение на составляющие, артефакты остаются только в некоторых компонентах. Такие компоненты можно «занулить» и после обратного преобразования получить исправленный снимок. Ниже представлен пример фильтрации «вертикальных полос» (рисунок 4).

Рисунок 4 – Пример фильтрации «вертикальных полос»

Согласно проведенным экспериментальным исследованиям, данный метод является довольно эффективным, однако для достижения оптимальных результатов необходима тщательная настройка, так как исключение отдельных компонентов может привести к размытию исходного изображения.

Подводя итог, двумерное дискретное вейвлет-преобразование (а в частности, преобразование Хаара) представляет собой мощный инструмент для анализа и обработки изображений. Использование данных технологий обеспечивает возможность эффективного анализа изображений на различных уровнях разрешения, локализации в пространственной и частотной областях, а также высокую вычислительную эффективность. Данные методы активно развиваются и находят новые применения в обработке изображений, сигналов, и системах высокоскоростной обработки видеоданных / мультимедийных системах.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Уэлстид С. «Фракталы и вейвлеты для сжатия изображений в действии» // Учебное пособие [Электронный ресурс] – Издательство «ТРИУМФ», 2003. - С.167 - 169. Режим доступа: https://clck.ru/3ARoAv.

2. Юдин М.Н., Фарков Ю.А. «Введение в вейвлет-анализ» // Учебное пособие [Электронный ресурс] – Издательство МГТА, 2014. – С. 30–31. Режим доступа: https://clck.ru/3ARoLo.

3. AnujBhardwaj, RashidAli «Image Compression Using Modified Fast Haar Wavelet Transform» // Статья [Электронный ресурс] – Издательство Department of Mathematics, Vishveshwarya Institute of Engineering and Technology, Dadri, G.B.Nagar - 203207, U.P. India, 2009. Режим доступа: https://goo.su/UE5d94Q.

УДК 004.8

В. М. Архипкин Д. Э. Бизяев

ИССЛЕДОВАНИЕ МЕТОДОВ И РАЗРАБОТКА ПО РАСПОЗНАВАНИЮ РУКОПИСНЫХ ЦИФР

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

В статье рассматривается разработка нейронной сети для распознавания рукописных цифр из набора данных MNIST. Проблема решается с помощью библиотеки PyTorch для создания модели и Flask для разработки веб-приложения, позволяющего вводить рукописные цифры и получать результаты в реальном времени. Результаты демонстрируют высокую точность модели, достигнувшей 96.25% на тестовых данных.

Ключевые слова: распознавание рукописных цифр, нейронная сеть, MNIST, PyTorch, Flask, автоматизация, машинное обучение.

В статье рассматривается задача распознавания рукописных символов, включая цифры, для автоматизации различных сфер, таких как банковское дело и медицинская отчетность. Цель исследования — разработка программного обеспечения для распознавания рукописных цифр с использованием нейронных сетей, что позволит автоматизировать ввод числовых значений и ускорить работу с данными.

Существует ряд аналогичных продуктов, таких как ReHand, IfImageEditor, Pen to Print online, 2OCR и Text-scan, предлагающих различные подходы к распознаванию рукописных текстов и цифр. Они предоставляют возможности автоматического распознавания и конвертации рукописных значений в машинный текст с использованием различных алгоритмов и технологий, таких как нейросети и оптическое распознавание символов (OCR). Эти инструменты позволяют ускорить процесс обработки данных и повысить эффективность работы в различных сферах, где требуется ввод и анализ числовых значений или текстовой информации.

Приложение состоит из нескольких ключевых компонентов: WSGI интерфейса, контроллера Flask, Flask сервера для обработки HTTP запросов, SSL для безопасного соединения, нейронной сети для распознавания данных, шаблонов для визуализации, и HTML, CSS и JavaScript для оформления и добавления интерактивности на веб-страницах. Эти компоненты обеспечивают связь между клиентом и сервером, обработку запросов, безопасное соединение, а также визуализацию и взаимодействие с пользователем (рисунок 1).

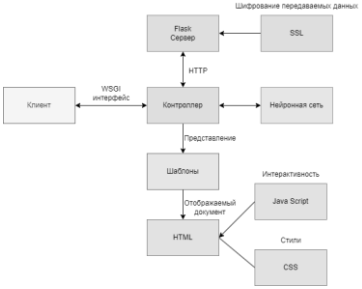


Рисунок 1 – Структура приложения

Для обработки данных мы использовали датасет MNIST, который содержит изображения рукописных цифр и соответствующие метки классов. Эти данные были подготовлены для обучения нейронной сети, которая состоит из входного слоя с 784 нейронами и выходного слоя с 10 нейронами, представляющими десять классов цифр от 0 до 9.

Создание нейронной сети было выполнено с использованием библиотеки PyTorch. Мы определили архитектуру сети, включающую входной, скрытый и выходной слои. Входной слой представляет вектор изображения, который преобразуется в тензор нужной размерности, а значения пикселей нормализуются к диапазону [0, 1].

Для обучения модели мы использовали кросс-энтропийную функцию потерь и стохастический градиентный спуск (SGD) с оптимизатором Adam. Обучение проводилось в течение 10 эпох. Мы также применили раннюю остановку для контроля процесса обучения и предотвращения переобучения.

В результате обучения нейронная сеть достигла точности предсказания на тестовых данных 96.25%. (рисунок 2).

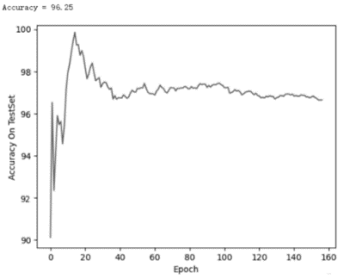


Рисунок 2 – Точность определения значения

После успешного обучения модели мы реализовали серверное приложение с использованием Flask для определения рукописных цифр, нарисованных пользователем. Веб-интерфейс был разработан с использованием HTML, CSS и JavaScript. Пользователь может рисовать рукописные цифры на холсте, отправлять их на сервер для идентификации и получать обратно определенные значения (рисунок 3).

Рисунок 3 – Внешний вид WEB-страницы

В ходе исследования была создана модель нейронной сети для распознавания рукописных цифр с использованием PyTorch. WEB-приложение на Flask было разработано для проверки модели в реальном времени. В будущем можно расширить модель для распознавания не только цифр, но и букв, чисел и слов. Также возможно интегрировать полученное решение в программное обеспечение для решения конкретных задач, например, автоматического решения примеров.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Рашид Т. «Создаем нейронную сеть» // Учебное пособие [Электронный ресурс] – Издательство «Альфа-книга», 2017. - С.167 - 169. Режим доступа: https://clck.ru/3AeNby.

2. Голованов В. «Нейросети и глубокое обучение, глава 1: использование нейросетей для распознавания рукописных цифр» // Статья [Электронный ресурс] – ХАБР, 2019. – С. 30–31. Режим доступа: https://clck.ru/3AeP37

3. Ashley « MNIST Digit Classification In Pytorch» // Статья [Электронный ресурс] – Medium.net, 2020. Режим доступа: https://clck.ru/3AeNge

УДК 004.42

А. П. Бабаян, Т. А. Дмитриева

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ОБУЧЕНИЯ КИТАЙСКОМУ ЯЗЫКУ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

В статье представлен интерфейс разработанного программного обеспечения китайскому языку. Рассмотрены дальнейшие направления разработки данного приложения.

Ключевые слова: обучение, китайский язык, веб-приложение, тестирование.

На текущий момент не существует удобного приложения для русскоговорящих пользователей, позволяющего самостоятельно заниматься изучением китайского языка [1]. Актуальность данной темы обусловлена растущей потребностью в изучении китайского языка, который становится все более востребованным в глобализированном мире. Образовательные технологии играют ключевую роль в упрощении доступа к знаниям и в повышении качества обучения, что делает разработку специализированного программного обеспечения особенно значимой.

Практическая значимость работы заключается в создании инструмента, который может быть использован широким кругом пользователей для самостоятельного изучения китайского языка. Данное

программное обеспечение может быть полезным не только для индивидуального обучения, но и в качестве вспомогательного ресурса в образовательных учреждениях.

В результате проведенной работы было разработано Java-приложение для обучения китайскому языку. Приложение включает клиентскую часть, реализованную с использованием Thymeleaf, серверную часть на базе Spring Framework и базу данных PostgreSQL, для которой использовано версионирование через Liquibase.

Рассмотрим интерфейс разработанного программного обеспечения.

Для регистрации в приложении необходимо открыть страницу регистрации и ввести в форму данные пользователя: адрес электронной почты, пароль, имя, фамилия. Повторная регистрация по адресу электронной почты невозможна. После нажатия на кнопку Зарегистрироваться, в случае успешной регистрации, пользователь будет перенаправлен на страницу авторизации. Страница регистрации приведена на рисунке 1.

Рисунок 1 – Страница регистрации

Для авторизации в приложение необходимо открыть главную страницу. В форму вводится информация о пользователе: логин (адрес почты) и пароль от аккаунта. Для входа в систему нужно нажать на кнопку Войти.

У пользователя есть возможность зайти в Личный кабинет. Для изменения информации о пользователе необходимо нажать на кнопку Редактировать. В открывшемся окне можно изменить свои личные данные и поменять пароль. Доступ в личный кабинет так же возможен по кнопке Личный кабинет в правой верхней части окна приложения. Для того чтобы выйти из аккаунта, необходимо нажать на соответствующую иконку в верхнем меню. Страница личного кабинета приведена на рисунке 2.

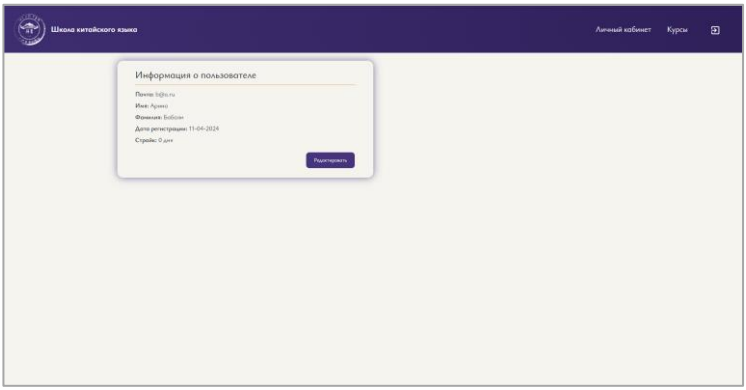


Рисунок 2 – Личный кабинет пользователя

Для записи на курс необходимо нажать на кнопку Курсы в верхнем меню. Откроется страница со списком курсов пользователя и доступными курсами для записи. Пользователь должен выбрать интересующий курс из списка. После чего нажать на кнопку Записаться, если он еще не записан на курс. Страница курса до записи приведена на рисунке 3.

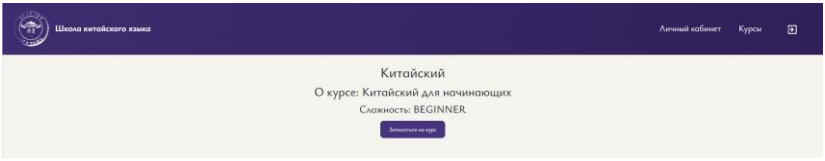


Рисунок 3 – Страница курса до записи

На странице курса можно получить доступ ко всем его урокам, статистике прохождения курса, справочным материалам. Курс состоит из уроков, их названия отображены слева. Под именем урока находится пиктограмма книги – по нажатию на нее пользователь получает доступ к справочному материалу. Для прохождения тестирования [2] необходимо выбрать любой доступный тест справа. Зеленый цвет означает, что урок пройден, фиолетовый – что он доступен, серый – не доступен. Страница прохождения курса приведена на рисунке 4.

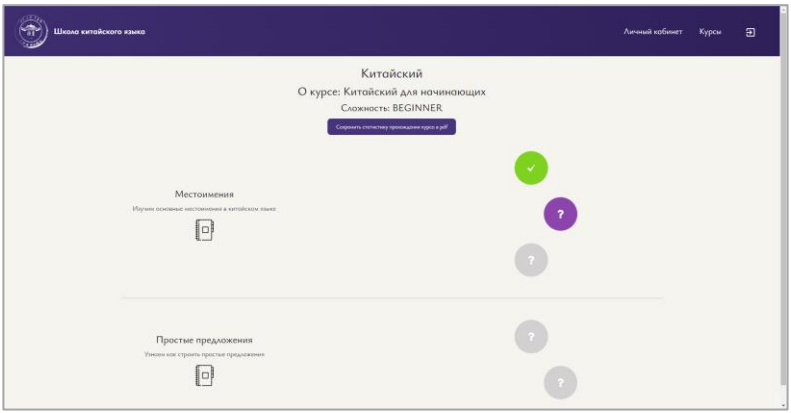


Рисунок 4 – Страница прохождения курса

Вопросы тестирования могут быть разных типов. Тестовые вопросы закрытого типа требуют выбора ответа из предложенных (рисунок 5). Вопросы открытого типа требуют полного ответа на задание. Например, предоставить перевод предложения (рисунок 6).

Рисунок 5 – Пример тестового вопроса

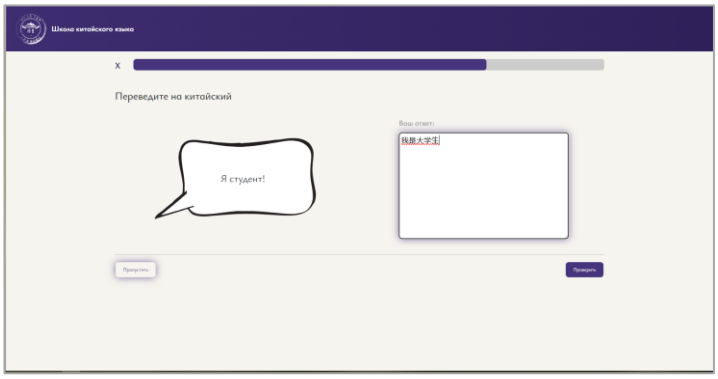


Рисунок 6 – Пример вопроса с открытым ответом

Для проверки ответа нужно нажать на соответствующую кнопку. В случае верного ответа, система отобразит соответствующее сообщение (рисунок 7). Если был дан неверный ответ, система покажет правильный, а сам вопрос перейдет в конец очереди (рисунок 8).

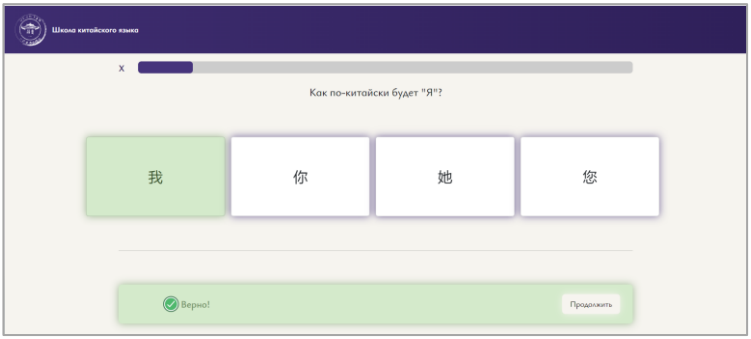


Рисунок 7 – Пример сообщения о верном ответе

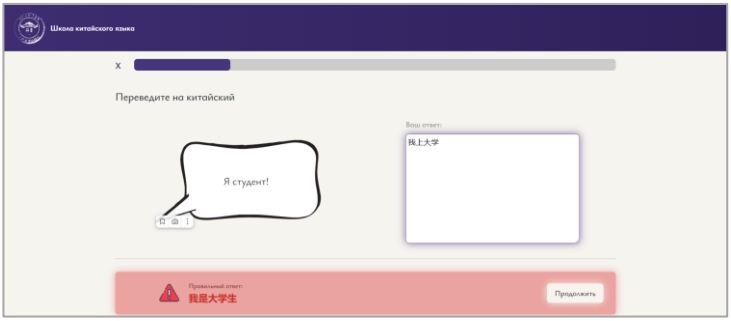


Рисунок 8 – Пример сообщения о неверном ответе

Для успешного завершения тестирования необходимо ответить на все вопросы. При желании вопрос можно пропустить – в этом случае он так же переносится в конец тестирования, но не помечается как неверный. По окончанию прохождения тестирования появится окно со статистикой ответов – количество вопросов, на которые были даны ответы и процент верных ответов (рисунок 9).

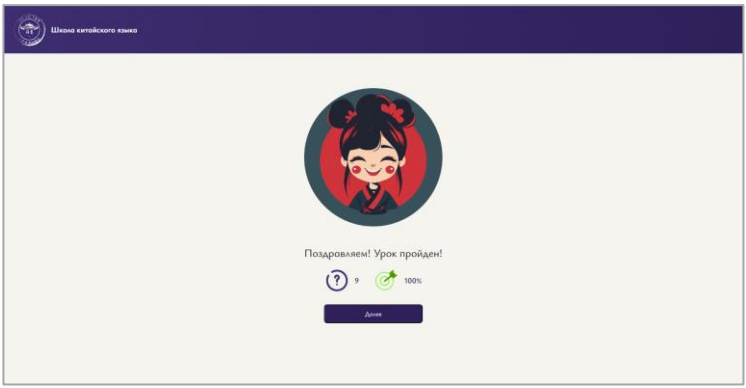


Рисунок 9 – Страница успешного прохождения теста

Если в процессе прохождения тестирования возникнет необходимость его прервать, то сделать это можно нажатием на крестик слева от школы прогресса. В этом случае система отобразить всплывающее окно с предупреждающей информацией (рисунок 10).

Для получения полноценного опыта работы с приложением рекомендуется внимательно изучать справочные материалы, предоставленные к каждому уроку. Для этого необходимо нажать на пиктограмму с книжкой под название урока. Если материал большой, то справа от открывшегося окна будет ползунок, который позволит прокрутить материал. Пример такой страницы приведен на рисунке 11.

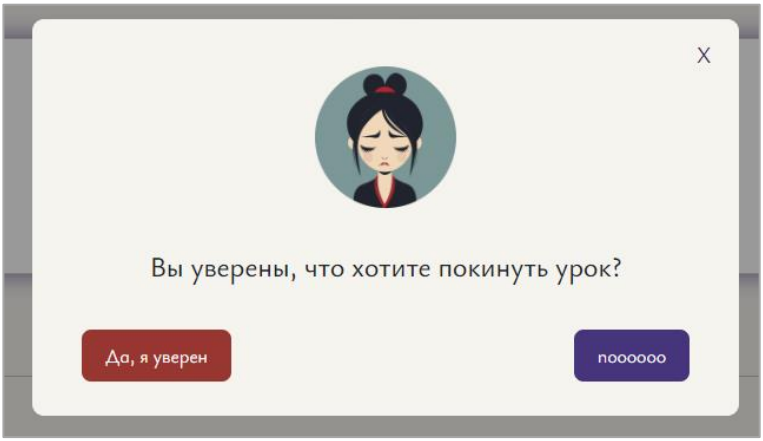


Рисунок 10 – Всплывающее окно о потере прогресса

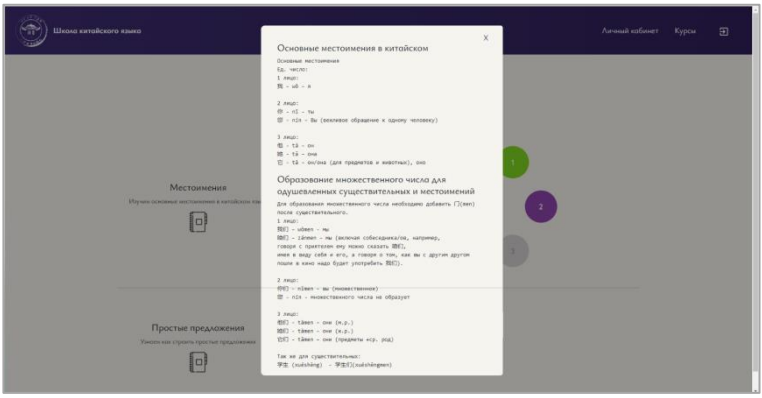


Рисунок 11 – Страница со справочной информацией

На странице курса есть кнопка «Сохранить статистику в pdf». При нажатии на нее будет сгенерирован pdf документ с текущими результатами пользователя: количество отвеченных вопросов за последние 7 дней, точность ответов, темы, в ответы на которые, за последние 14 дней было совершенно большое количество ошибок. Пример сгенерированного отчета приведен на рисунке 12.

В дальнейшем разработанное программное обеспечение можно улучшить в следующих направлениях: расширить функциональность (например, добавить новые типы упражнений и интерактивных заданий для разнообразия учебного процесса и повышения его эффективности); добавить мультиязычность, так как интерфейс и учебных материалов на других языках, позволят привлечь более широкую аудиторию пользователей из разных стран; реализовать мобильный вариант приложения; использовать технологии искусственного интеллекта.

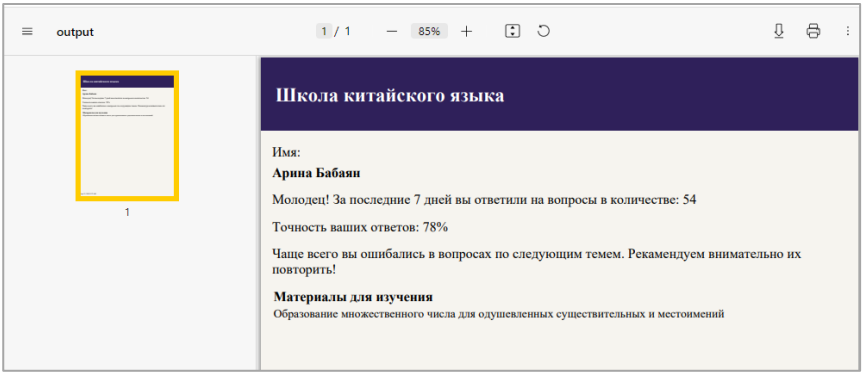


Рисунок 12 – Пример сгенерированного отчета

Таким образом, разработанное программное обеспечение обладает значительным потенциалом для дальнейшего развития и улучшения, что позволит сделать процесс изучения китайского языка еще более эффективным и доступным для широкого круга пользователей.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Бабаян А.П. Разработка программного обеспечения для обучения китайскому языку // 71-я студенческая научно-техническая конференция Рязанского государственного радиотехнического университета. – Рязань: 2024. – С. 50.

2. Бабаян А.П. Разработка формальной грамматики составления автоматизированных тестов контроля знаний китайского языка // Новые информационные технологии в научных исследованиях: материалы XXVIII Всероссийской научно-технической конференции студентов, молодых ученых специалистов. – Рязань: издательство ИП Коняхин А.В. (Book Jet), 2023. – С. 11–12.

УДК 004.8

Д. В. Белозерцев, С. О. Алтухова

ОБУЧЕНИЕ С НУЛЕВЫМ РАЗГЛАШЕНИЕМ

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Липецкий государственный педагогический университет имени П. П. Семенова-Тян-Шанского», Липецк

Применение технологии обучения с нулевым разглашением в различных сферах жизнедеятельности. Решение недостатков имеющихся моделей искусственного интеллекта.

Ключевые слова: искусственный интеллект, криптография, конфиденциальность, нулевое разглашение.

В последнее время искусственный интеллект набирает популярность. С каждым годом машинное обучение всё больше захватывает различные сферы деятельности. Благодаря этому появляются различные технологии, способные решить те проблемы, которые с трудом бы дались человеку.

Одной из таких технологий является обучение с нулевым разглашением (ZKML) – технология, использующая методы криптографии для создания гарантии, что процессы машинного обучения могут быть проверены и являются доверенными без разглашения конфиденциальных данных.

Обучение с нулевым раскрытием построена на основе созвучного криптографического метода – доказательство с нулевым разглашением, главная задача которого состоит в убеждении «пользователем» проверяющего в том, что ему известны конфиденциальные данные, без

раскрытия этих данных. Такие протоколы являются двухключевыми криптосхемами и включают три основных шага:

1. Генерация фиксатора в виде разового открытого ключа доказывающим.

2. Генерация случайного запроса проверяющим.

3. Вычисление доказывающим ответа на запрос.

Приведём в пример протокол Фиата-Шамира. Он основан на сложности извлечения квадратного корня по составному модулю, включающему не менее двух больших простых множителей, при условии, что разложение неизвестно. Для начала выбираются два больших простых числа p и q и вычисляется модуль $n = pq$ доказывающим. Затем выбирает случайное число s, такое, что $1 \leq s \leq n - 1$ в качестве личного секретного ключа, и вычисляется:

$$y = s^2 \, mod \, n \tag{1}$$

Доказательство с нулевым разглашением заключается в убеждении проверяющего в том, что доказывающий знает квадратный корень из y. Значение y используется в роли открытого ключа и объявляется всем участникам протокол для проверки знания s.

Протокол состоит из неоднократного повторения раунда, включающего три шага.

В первую очередь доказывающим выбирается такое число k, что $1 \leq k \leq n - 1$, вычисляется значение:

$$u = k^2 \, mod \, n \tag{2}$$

называемое фиксатором, и передаёт его проверяющему. В данном случае число k является разовым секретным ключём и обеспечивает защиту личного секретного ключа. Далее проверяющий отправляет доказывающему равновероятный случайный бит r. Затем доказывающий вычисляет значение:

$$w = ks^r \, mod \, n \tag{3}$$

Затем направляет его проверяющему. (Если $r = 1$, то $w = ks \, mod \, n$. Если $r = 0$, то $w = k$. По данным результатам видно, что секрет s вычисляется довольно просто, поэтому значения k должны удаляться после каждого раунда или после выполнения всего протокола). Проверяющий считает ответ верным, если выполняется соотношение:

$$w^2 = uy^r \, mod \, n \tag{4}$$

Если $r = 1$, то $w^2 = uy \, mod \, n$. Если $r = 0$, то получаем $w^2 \, mod \, n = u$. В ходе осуществления протокола выполняется z шагов. Вероятность того, что нарушитель (который не знает секрета s) при выполнении одного раунда может дать положительный ответ, равна 2^{-1}, следовательно, вероятность того, что нарушитель может быть принят за пользователя, знающего секрет s, составляет $2 - z$. Выбирая в протоколе

достаточно большое число раундов проверки, можно сделать сколь угодно низкой вероятность обмана [1].

Работа ZKML заключается в объединении машинного обучения и криптографических методов в децентрализованной сети. Обучаемые модели, которые содержат собственные части конфиденциальных данных на различных узлах распределённой сети создают «доказательства», что позволяет узлам искусственного интеллекта подтверждать определённые качества или характеристики своих данных, не раскрывая сами данные. В этом и заключается преимущество данного метода при использовании его в прозрачных системах, например в блокчейн-сетях. Некоторые крупные компании могут использовать систему хранения и отслеживания транзакций с виртуальной валютой, не раскрывая коммерческую тайну или личную информацию своих клиентов. Подобным образом могут функционировать искусственный интеллект, основанный о данных о пациентах. Больницы с разных концов мира могут использовать данную технологию, сохраняя врачебную тайну. Обучение с нулевым разглашением становится всё доступней для разработчиков приложений, что позволяет технологии использоваться не только в крупных организациях, но и в близких нам системах. Например, такая система проверки личности, как WorldID, благодаря данной технологии, развивается и становится точней, без раскрытия личных данных.

Одним из полезных свойств технологии обучения с нулевым разглашением является явное представление факта создания контента искусственным интеллектом путём применения определённой модели, без предоставления дополнительной информации или входных данных. Вторым свойством является прозрачная оценка производительности модели, путём получения лишь результата работы искусственного интеллекта [2].

Стандартные модели машинного обучения имеют определённые проблемы, и ZKML потенциально может решить некоторые из них. Её преимущество заключается в том, что разработчики могут участвовать в проверке результатов, полностью не раскрывая архитектуру модели, при этом сохраняя свою интеллектуальную собственность. Также рассматриваемая технология создаёт безопасную и сохраняющуюся конфиденциальную платформу, которая устраняет недостатки традиционного машинного обучения.

Однако криптографический метод доказательства с нулевым разглашением имеет свои недостатки, а значит, что и модель машинного обучения также подвержена риску. Известны два типа возможных атак. Атака на основе подобранного шифротекста состоит в предоставлении проверяющему большое количество данных, которое должно быть обработано и на основе ответных данных появляется возможность анализа результатов. Атака с помощью квантового компьютера актуальна

для многих, если не всех, криптографических методов. Так как при использовании грамотно построенного суперкомпьютера, который бы смог путём сложных вычислений производить большое количество итераций для поиска секретного ключа.

Обучение с нулевым разглашением новая технология, которая стоит на этапе зарождения, которой свойственны недостатки и разногласия с другими методами. Однако при постепенном развитии и попытках внедрения во всё большие и отличающиеся по своей структуре системы, есть вероятность, что недостатки получится убрать, а полезные особенности усилить.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Молдовян, А. А. Протоколы аутентификации с нулевым разглашением секрета / А. А. Молдовян, Д. Н. Молдовян, А. Б. Левина. – СПб: Университет ИТМО, 2016. – 55 с.

2. Горячев, А. А. Методические аспекты изложения протоколов с нулевыми знаниями: толкование термин «нулевое разглашение» в дисциплине «Криптографические протоколы» / А. А. Горячев, Р. В. Кишмар, Хо Нгок Зуй // Современное образование: содержание, технологии, качество. – 2011. – Т. 2. – С. 238-240.

УДК 000.000

А.А. Буланов, Д.Г. Котиков

ИЗУЧЕНИЕ МЕТОДОВ ГЕНЕРАЦИИ ПСЕВДОСЛУЧАЙНЫХ ЧИСЕЛ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

В данной статье рассматриваются пять популярных методов генерации псевдослучайных чисел: метод середины квадрата, линейный конгруэнтный метод, генератор Фибоначчи с запаздыванием, Xorshift и Xoroshiro128+.

Ключевые слова: детерминированные генераторы псевдослучайных чисел, критерий Колмогорова, генерация псевдослучайных чисел.

Введение. Генераторы псевдослучайных чисел (ГПСЧ) играют ключевую роль в различных областях науки и техники, таких как криптография, статистический анализ, моделирование и компьютерные игры. Эти алгоритмы создают последовательности чисел, которые обладают свойствами случайности, но являются детерминированными при известных начальных условиях. Целью работы является проведение сравнительного анализа различных методов генерации псевдослучайных чисел и формирование рекомендаций по использованию определенных методов при решении определенных задач.

Методы исследований. Для проведения исследования были выбраны следующие генераторы псевдослучайных чисел:

- метод середины квадрата;
- линейный конгруэнтный метод;
- генератор Фибоначчи с запаздыванием;
- xorshift;
- xoroshiro128+.

Каждый из этих методов был проанализирован по следующим критериям:

- период генерации;
- статистические свойства (равномерность распределения, корреляция);
- скорость генерации;
- простота реализации;
- типичные области применения.

Метод середины квадрата был предложен Джоном фон Нейманом в 1949 году. Он основан на следующем принципе: квадрат числа берётся, а затем из середины результата извлекается нужное количество цифр для формирования следующего числа в последовательности. На рисунке 1 изображена гистограмма распределения для данного генератора при генерации 100000 чисел.

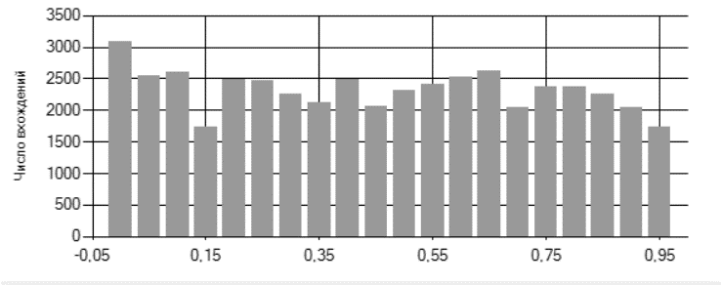


Рисунок 1 - Гистограммы распределения для метода середины квадратов

Линейный конгруэнтный метод является одним из наиболее популярных и простых ГПСЧ. Он основан на рекуррентном соотношении (1):

$$X_{\{n+1\}} = \left(aX_n + c\right) \bmod m, 1 \qquad (1)$$

где (a), (c) и (m) – параметры метода. На рисунке 2 изображена гистограмма распределения для данного генератора при генерации 100000 чисел.

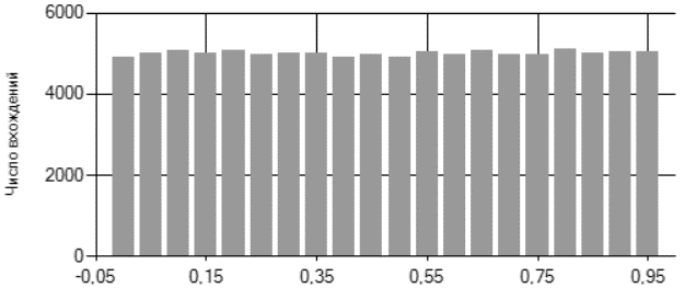


Рисунок 2– Гистограммы распределения для линейно конгруэнтного генератора

Генератор Фибоначчи с запаздыванием основан на модификации рекуррентного соотношения Фибоначчи (2):

$$X_n = (X_{\{n-j\}} + X_{\{n-k\}}) \bmod m, \tag{2}$$

где (j) и (k) – задержки. На рисунке 3 изображена гистограмма распределения для данного генератора при генерации 100000 чисел.

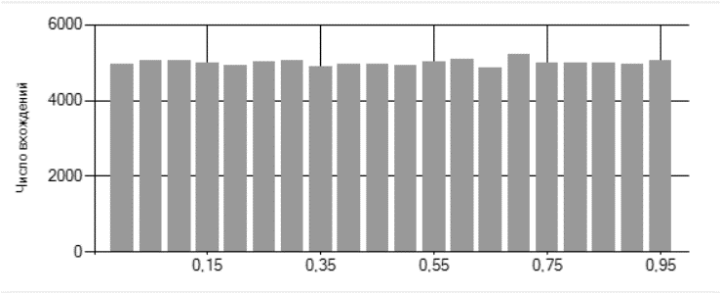


Рисунок 3 - Гистограмма распределения для генератора Фибоначчи с запаздыванием

Xorshift использует битовые операции для создания псевдослучайных чисел. Он реализуется путем выполнения следующих действий (3):

$$\begin{aligned} &1. \quad y \oplus = \left(y \ll a\right); \\ &2. \quad y \oplus = \left(y \gg b\right); \\ &3. \quad y \oplus = \left(y \ll c\right); \end{aligned} \tag{3}$$

где $(\ll)$ и $(\gg)$ – битовые сдвиги влево и вправо, а $(\oplus)$ – операция XOR. На рисунке 4 изображена гистограмма распределения для данного генератора при генерации 100000 чисел.

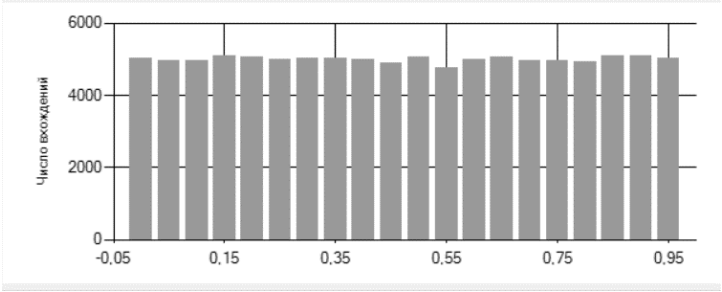


Рисунок 4 - Гистограмма распределения для XORShift

Xoroshiro128+ является улучшенной версией Xorshift, предложенной Себастьяном Виртмом и Дэвидом Блэкманом. Он сочетает битовые операции со сложением. На рисунке 5 изображена гистограмма распределения для данного генератора при генерации 100000 чисел.

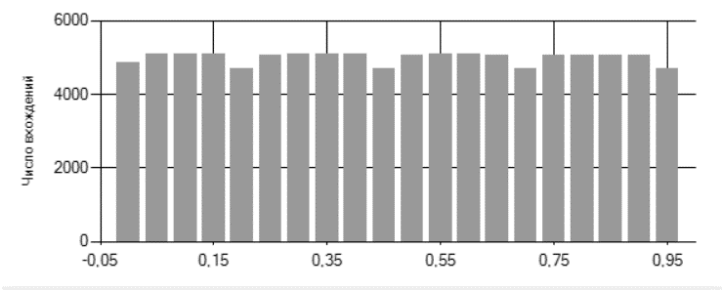


Рисунок 5 - Гистограмма распределения для Xoroshiro128+

Результаты исследования показывают, что каждый из рассмотренных методов имеет свои преимущества и недостатки, которые определяют их пригодность для конкретных задач. В таблице 1 указаны значения критериев сравнения для каждого генератора в конкретной реализации.

Таблица 1 – Значения характеристик для каждого генератора

	Критерии сравнения				
Генераторы	**Среднее время генерации 100000 элементов (мс)**	**Критерий Колмогорова (уровень значимости)**	**Математическое ожидание**	**Дисперсия**	**Период генерации**
Метод середины квадратов	33.94	7.22 (1)	0.48	0.085	10^6
Линейно конгруэнтного метода	3.12	0.78 (0.5)	0.502	0.083	2^{32}
Генератор Фибоначчи с запаздыванием	9.11	0.75 (0.4)	0.5	0.083	2^{32}
Xorshift	1.65	0.68 (0.3)	0.5	0.084	2^{32}
Xoroshiro128+	1.77	1.17 (0.9)	0.499	0.083	2^{32}

Стоит отметить, что по критерию Колмогорова наилучшими генераторами с точки зрения приближения к равномерному закону распределения случайной величины являются генератор Фибоначчи с запаздыванием и Xorshift. Xoroshiro128+ имеет уязвимость в младших битах, поэтому в данном исследовании показал плохой результат. Кроме того, метод середины квадратов не применим для генерации псевдослучайных последовательностей из-за неравномерного распределения случайной величины.

Метод середины квадрата имеет наименьший период генерации по сравнению с остальными генераторами, что ограничивает его использование в задачах, требующих большого объема экспериментов. Лучшие статистические свойства продемонстрировали Xorshift и генератор Фибоначчи с запаздыванием. Линейный конгруэнтный метод и Xoroshiro128+ также могут обеспечивать хорошие свойства, но требуют тщательной настройки параметров. Метод середины квадрата имеет худшие статистические свойства, часто приводя к корреляциям и циклам. Наибольшую скорость генерации показывают Xorshift и Xoroshiro128+ благодаря использованию эффективных битовых операций. Линейный конгруэнтный метод также достаточно быстр, тогда как метод середины квадрата и генератор Фибоначчи с запаздыванием требуют больше вычислительных ресурсов. Самыми простыми в реализации являются метод середины квадрата и линейный конгруэнтный метод. Xorshift также относительно прост, тогда как генератор Фибоначчи с запаздыванием и Xoroshiro128+ требуют более сложного программирования.

Выводы. Исходя из проведённого исследования и анализа, можно утверждать, что каждый из рассмотренных генераторов обладает своими

преимуществами и недостатками, определяющими их уместность для конкретных задач. Для приложений, требующих высокой скорости и отличных статистических свойств, наилучшим выбором являются Xorshift и Xoroshiro128+. Линейный конгруэнтный метод остаётся популярным благодаря своей простоте и гибкости. Метод середины квадрата и генератор Фибоначчи с запаздыванием могут быть полезны в специфических ситуациях, но требуют более тщательного анализа и настройки.

Таким образом, анализ и сравнение различных генераторов псевдослучайных чисел позволяют выбрать оптимальный метод для конкретных применений, обеспечивая баланс между простотой реализации, скоростью и статистическими свойствами.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Агибалов Г. П., Теория псевдослучайных генераторов. // Учебное пособие - Томск, 2019. - С. 18-54.

2. Градов В.М., Овечкин Г.В., Овечкин П.В., Рудаков И.В., Компьютерное моделирование. // Учебник – Москва, 2017. – С.26-63.

3. Слеповичев И.И., Генераторы псевдослучайных чисел. // - 2017. - С.118.

4. Кнут Д., Искусство программирования. Том 2. Получисленные алгоритмы. // Учебное пособие – 2002. - С. 12-120.

5. Marsaglia G., Xorshift RNGs. // - 2003. - С.6.

УДК 004.032.26

А. В. Крошилин, Е. Г. Вапилина

РАЗРАБОТКА СВЕРТОЧНОЙ НЕЙРОННОЙ СЕТИ ДЛЯ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА СОДЕРЖАНИЯ СООБЩЕНИЙ ЧЕЛОВЕКА С ЦЕЛЬЮ СОСТАВЛЕНИЯ ЕГО ПСИХОЛОГИЧЕСКОГО ПОРТРЕТА

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

В статье рассматривается проблема анализа психологического портрета человека на основе данных из социальных сетей. Для решения данной проблемы предлагается использование сверточных нейронных сетей, которые позволяют автоматически извлекать и классифицировать текстовые признаки, ускоряя обработку данных и повышая эффективность анализа.

Ключевые слова: сверточные нейронные сети, психологический портрет, текстовый анализ, социальные сети, обработка естественного языка, классификация текста.

Концепция использования нейронных сетей для построения психологического портрета человека на основе его социальных сетей строится на анализе текстовых данных сообщений с целью выявления некоторых характеристик личности [1].

Сверточные нейронные сети (СНС) широко используются для обработки естественного языка. Они имеют способность к автоматическому извлечению признаков из текста, что делает их эффективными в задачах классификации и анализе тональности текстов.

Преимущества СНС:

- позволяет эффективно находить полезные комбинации слов или фраз, независимо от их положения в тексте;
- обрабатывает текстовые данные параллельно, что значительно ускоряет обучение и предсказание;
- работает с текстами различной длины без необходимости явного нормирования или обрезки;
- строит иерархические представления текста: от простых сочетаний символов и слов к более сложным фразам и семантическим единицам;
- компенсируют небольшие смещения и искажения в данных;
- интегрируются с предобученными словарными эмбеддингами;
- требуют мало вычислительных ресурсов и легко настраиваются.

Ограничения СНС:

- плохо захватывают долгосрочные зависимости в тексте;
- имеют ограниченную способность захватывать паттерны различной длины из-за фиксированного размера фильтров;
- недостаточно эффективно захватывают контекстную информацию;
- чувствительны к порядку слов;
- не подходят для задач, требующих последовательного или пошагового вывода.

Хотя у СНС есть свои ограничения, их преимущества делают их одним из наиболее востребованных инструментов для обработки естественного языка.

Схематическое представление архитектуры сверточной нейронной сети изображено на рисунке 1 [2].

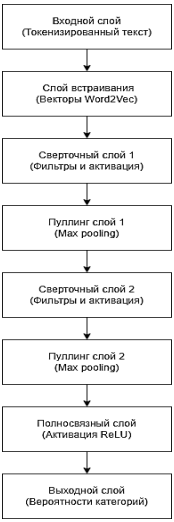


Рисунок 1 — Архитектура сверточной нейронной сети

Для обучения сети сначала текст преобразуется в числовой формат с помощью векторизации. Затем он представляется в виде матрицы, где строки — это векторы слов. Сверточные слои применяют фильтры для выявления локальных текстовых шаблонов, а пуллинг уменьшает размерность данных, выделяя ключевые признаки. Извлеченные признаки обрабатываются полносвязными слоями, а финальный слой выполняет классификацию текста. Модель обучается с использованием функции потерь и оптимизации, а техники регуляризации, такие как Dropout, помогают избежать переобучения.

Алгоритм создания СНС для выявления личностных качеств человека представлен на рисунке 2.

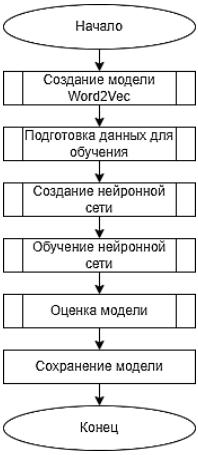


Рисунок 2 — Алгоритм создания сверточной нейронной сети

Алгоритм функции обучения нейронной сети представлен на рисунке 3 [3].

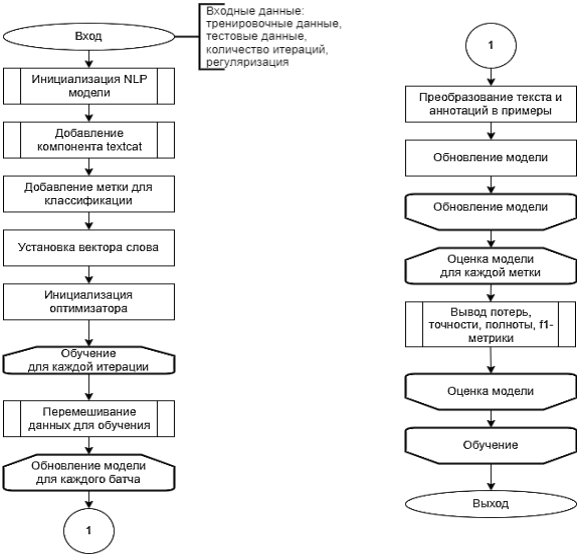


Рисунок 3 — Обучение нейронной сети

Для оценки модели выполняется расчет функции потерь, точности, полноты и F1-метрики. Обученную модель сохраняют для последующего использования.

В заключении можно отметить, что анализ психологического портрета человека на основе данных из его блога с применением

нейронных сетей представляет собой перспективное направление исследований, которое может принести новые знания и упростить процесс изучения личности.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Исследование алгоритмов анализа информации в социальных сетях // Современные наукоемкие технологии URL: https://top-technologies.ru/ru/article/view?id=37998&ysclid=lx54bk12sr224715399 (дата обращения: 06.06.2024).

2. Вакуленко С.А., Жихарева А.А. Практический курс по нейронным сетям – СПб: Университет ИТМО, 2018. – 71 с.

3. Классификация текста сверточной нейронной сетью на основе TensorFlow // UnderSkyAi URL: https://under-sky-ai.ru/post/klassifikatsiya_teksta_svertochnoy_neyronnoy_setyu_na_osnove_tensorflow (дата обращения: 06.06.2024).

УДК 004.032.26

В.А. Володин, С.В. Крошилина

ОСОБЕННОСТИ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ СТРОИТЕЛЬНОЙ КОМПАНИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

В статье рассмотрено и описано экспертное оценивание, помимо этого улучшен метод для решения задач строительной компании.

Ключевые слова: экспертные оценки, определение относительных весов объектов, степень обобщённости мнений специалистов.

Экспертные оценки (ЭО) представляют из себя процесс получения результата проблемы в зависимости от мнений экспертов. [1].

МЭО выглядят, как совокупность решений для работы со специалистами и обработки их ответов. Суть приёмов заключается в прогнозе, закладываемому на суждение одного или группы специалистов, базирующееся на их навыках, научных достижениях.

Рассмотрим подходы, применяемые во время разбора оценок [2].

В процессе анализа данных, полученных при опросе, наиболее часто применяется математическая статистика [3].

Исходя их целей экспертного анализа при работе с оценками необходимо выполнить ряд задач:

- формирование обобщенной оценки;
- определение относительных весов объектов;
- установление степени согласованности мнений специалистов и др.

Ниже будут проанализированы и рассмотрены различные подходы и методы, направленные на решение каждой из упомянутых проблем [4].

1) Получение общей оценки.

Пусть группа специалистов оценила некоторый объект, тогда xj – оценка j-го специалиста, j m = 1, где m – число специалистов.

Для формирования общей оценки группы специалистов обычно используются средние значения. Например, медиана (ME), которая представляет собой оценку, относительно которой количество оценок, превышающих её значение, равно количеству оценок или меньших её значения.

Также может быть применена точечная оценка группы специалистов, рассчитываемая как математическое ожидание:

$$\overline{x_э} = \frac{\sum_{j=1}^{m} x_j}{m} \qquad (1)$$

2) Определение относительной важности объектов. В некоторых ситуациях требуется оценить степень важности или существенности определённого фактора (объекта) в соответствии с заданным критерием. Этот процесс включает в себя определение весовых коэффициентов для каждого фактора. Один из методов определения весовых коэффициентов заключается в следующем. Пусть n – число сравниваемых сущностей, полученная j-ым специалистом, xij – оценка события i, i n = 1, , j m = 1, , m – число специалистов. Тогда вес i-го сущностей, подсчитанный по оценкам всех специалистов (wi), равен:

$$w_i = \frac{\sum_{j=1}^{m} w_{ij}}{m}, \qquad (2)$$

где wij – вес i-го объекта, рассчитанный по оценкам j-го специалиста, равен:

$$w_{ij} = \frac{x_{ij}}{\sum_{i=1}^{n} x_{ij}}, \ j = \overline{1,m}, \ i = \overline{1,n} \qquad (3)$$

3) Степень обощённости мнений специалистов.

При опросе с несколькими специалистами, расхождения в их оценках неизбежны. Однако величина этого расхождения имеет существенное значение. Групповая оценка может считаться достоверной только при условии хорошей согласованности ответов отдельных специалистов.

Формула расчёта вариационного размаха

$$(R) = x_{\max} - x_{\min},$$

Когда xmin – наименьшая оценка сущности; xmax – наивысшая оценка сущности.

Среднее квадратическое отклонение, вычисляемое по известной формуле:

$$\sigma = \sqrt{\frac{\sum_{j=1}^{m}\left(x_{j_3} - \overline{x}\right)^2}{m-1}}, \qquad (4)$$

Когда xj – оценка, данная j-ым специалистом; m – количество специалистов.

Коэффициент вариации (V), можно выразить следующим образом:

$$V = \frac{\sigma}{\overline{x}_3} \cdot 100\% \qquad (5)$$

Анализ степени Обобщённости экспертных мнений

В случае привлечения к опросу нескольких специалистов, различия в их оценках неизбежны. Однако степень этих различий имеет существенное значение. Групповая оценка может считаться надёжной только при условии хорошей согласованности ответов отдельных специалистов. (для специалиста j): x1j, x2j, …, xnj.

Для определения степени согласованности между ранжировками, предложенными двумя специалистов, можно использовать коэффициент ранговой корреляции Спирмена.:

$$\rho = 1 - \frac{6\sum_{i=1}^{n}\left(x_{ij} - x_{ik}\right)^2}{n\left(n^2-1\right)} = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n\left(n^2-1\right)}, \qquad (6)$$

где ik – ранг, присвоенный i-й сущности k-у специалисту; xij – ранг, присвоенный i-й сущности j-у специалисту; i – разница между рангами, присвоенными i-й сущности.

Значение коэффициента ранговой корреляции Спирмена может находиться в диапазоне от –1 до + 1. Если оценки специалистов полностью совпадают, коэффициент принимает значение, равное единице. И наоборот, при наибольшем различии в оценках специалистов коэффициент будет равен минус единице. Используется средний коэффициент ранговой корреляции для группы, включающей в себя m специалистов:

$$W = \frac{12 \cdot S}{m^2\left(n^3 - n\right)}, \qquad (7)$$

где

$$S = \sum_{i=1}^{n}\left(\sum_{j=1}^{m} x_{ij} - \frac{1}{2}m\left(n+1\right)\right)^2 \qquad (8)$$

Очевидно, что вычитаемое в скобках представляет собой среднюю сумму рангов, полученных i объектами от экспертов. Коэффициент согласованности W изменяется в диапазоне от 0 до 1, где значение 1 означает полное согласие между экспертами в присвоении рангов объектам, а значение близкое к нулю указывает на несогласованность в оценках экспертов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Крошилина С.В. Предметно-ориентированные информационные системы: учебное пособие / А.В. Крошилин, С.В. Крошилина, Г.В. Овечкин. — Москва: Курс, 2023. — 176 с. — (Естественные науки).

2. Крошилин А. В., Крошилина С. В. Интеллектуальные поисковые системы на основе нечеткой логики. Учебное пособие для вузов. - М.: Горячая линия – Телеком, 2023. – 140 с.: ил.

3. Володин В.А. Разработка и исследования интеллектуальных алгоритмов для решения задач в строительной сфере // Материалы IX научно-технической конференции магистрантов Рязанского государственного радиотехнического университета имени В.Ф. Уткина. – Рязань: РГРТУ, 2024 - 419 с. (199-200).

4. Гвоздинский А.Н. Методы аналитиче- ской обработки информации [Текст] / А.Н. Гвоздин- ский, Е.Г. Климко // Радиоэлектроника и информати- ка. 2000. №4. С.111-112.

УДК 004.72

Д. А. Перепелкин, А. И. Ковердяев

ОСОБЕННОСТИ ПРИМЕНЕНИЯ ГЕНЕТИЧЕСКОГО АЛГОРИТМА В ПРОГРАММНО-КОНФИГУРИРУЕМЫХ СЕТЯХ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

В статье исследуются особенности использования генетического алгоритма применительно к программно-конфигурируемым сетям. Рассматривается использование алгоритма для оптимизации кратчайших маршрутов и балансировки нагрузки. Приводятся преимущества и недостатки использования генетических алгоритмов в программно-конфигурируемых сетях.
Ключевые слова: программно-конфигурируемые сети, генетический алгоритм, многопутевая маршрутизация, качество обслуживания.

Программно-конфигурируемые сети (Software Defined Network, SDN, ПКС) представляют собой современное направление в построении сетей связи нового поколения. ПКС обеспечивают гибкое управление потоками данных благодаря разделению плоскости управления и плоскости передачи данных. Основная концепция ПКС заключается в упрощении сетевых элементов передачи данных через логическую централизацию управления, контроля и поддержки сетевых потоков с использованием специализированного программного обеспечения. Это позволяет перейти от управления отдельными сетевыми элементами к управлению сетевой платформой в целом.

Генетический алгоритм (ГА) — это метод оптимизации, основанный на принципах естественного отбора и генетики. Он используется для нахождения приближённых решений сложных задач оптимизации и поиска. Исходя из определения можно выделить следующие особенности использования ГА:

1) Эффективность при высоком уровне сложности (в пространстве поиска может быть множество локальных экстремумов);

2) Алгоритм может использоваться для широкого спектра задач;

3) При задании оптимальных параметров решение стремится к глобальному экстремуму;

4) ГА может предоставить не одно, а множество решений за счет использования понятия «популяция» в работе алгоритма.

Эти особенности генетических алгоритмов позволяют решать множество различных задач в программно-конфигурируемых сетях. Основные области, где могут применяться ГА в ПКС:

1) Оптимизация маршрутизации и управления трафиком. ГА могут использоваться для оптимизации маршрутов в сети с целью

уменьшения задержек, повышения пропускной способности и надежности. Они могут находить оптимальные маршруты для трафика, адаптируясь к изменениям в сети.

2) Распределение ресурсов и балансировка нагрузки. ГА помогают эффективно распределять ресурсы сети и балансировать нагрузку между различными сетевыми устройствами и маршрутами, что позволяет избежать перегрузок и улучшить общее качество обслуживания (QoS).

3) Управление пропускной способностью и QoS. ГА могут использоваться для динамического управления пропускной способностью и приоритетами трафика в зависимости от текущих потребностей пользователей и приложений.

4) Масштабируемость. ГА хорошо масштабируются, что позволяет их применять в больших сетях с большим количеством устройств и сложной топологией.

В работе [2] проводится исследование применения генетического алгоритма для исследования процесса многопутевой маршрутизации в ПКС. Можно сделать вывод, что использование генетического алгоритма для получения набора кратчайших маршрутов является в общем случае оптимальным при правильном подборе параметров. Оценка качества работы алгоритма оценивается относительно известного алгоритма Йена. Одновременно с нахождением набора кратчайших маршрутов генетический алгоритм может получить оптимальное значение балансировки нагрузки.

Экспериментальные результаты показали, что модифицированный генетический алгоритм превосходит классический алгоритм Йена в нахождении оптимальных маршрутов. При различных параметрах алгоритма (число итераций, вероятность мутации, число маршрутов) генетический алгоритм продемонстрировал высокую стабильность и эффективность.

Как отмечают авторы [3], генетический алгоритм может быть адаптирован для решения задачи построения оптимальной топологии сети с заданными характеристиками. Использование ГА позволяет уйти от полного перебора вариантов топологий сети. Авторы подводят итог, что для сети с достаточно большим количеством узлов, генетический алгоритм позволяет значительно ускорить задачу проектирования.

Одной из главных проблем использования генетических алгоритмов является вычисление параметров алгоритма. Неверный подбор параметров может значительно отдалить полученное решение от оптимального. Основными параметрами являются вероятность мутации, количество поколений, количество маршрутов.

Кроме того, хоть генетические алгоритмы и показывают хорошие показатели при определенных параметрах, не стоит исключать тот факт,

что этот алгоритм является эвристическим, т. е. оптимальное решение может быть и не найдено в конкретном случае.

Основным достоинством генетических алгоритмов применительно к ПКС является возможность его настройки для решения конкретных задач. Используя специализированные программные средства, можно динамически настраивать параметры алгоритма для получения наилучшего результата. К преимуществам использования генетических алгоритмов также можно отнести инвариантность к топологии сети.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Корячко, В. П., Перепелкин, Д. А. Программно-конфигурируемые сети. Учебник для вузов. М.: Горячая линия – Телеком, 2020. 288 с.

2. Перепелкин, Д. А., Нгуен, В. Т. Исследование и анализ процессов многопутевой маршрутизации и балансировки потоков данных в программно-конфигурируемых сетях на основе генетического алгоритма.

3. Балашова, Т. И. Обеспечение отказоустойчивости сети повышением надежности её топологии // ФГБОУ ВПО «Нижегородский государственный технический университет им. Р.Е. Алексеева». URL: https://science-education.ru/ru/article/view?id=16846 (дата обращения: 20.05.2024).

УДК 004.942

Д. А. Перепелкин, А. И. Ковердяев

ИССЛЕДОВАНИЕ МЕТОДОВ ПОСТРОЕНИЯ МАРШРУТОВ ПО ТОЧКАМ ДЛЯ БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

В работе рассматриваются различные алгоритмы построения маршрута для тандема БПЛА по точкам. Подчеркнуты особенности приведенных методов. На основе анализа может быть выбран один из алгоритмов.

Ключевые слова: построение маршрута, интерполяция, кубический сплайн, кривые Безье, БПЛА.

Одним из ключевых аспектов эффективного использования беспилотных летательных аппаратов (БПЛА) является разработка и оптимизация маршрутов их полета. Алгоритмы построения маршрутов играют важную роль в обеспечении точности, безопасности и экономичности выполнения заданий, особенно когда речь идет о полетах по заранее определенным точкам.

На практике часто бывают заданы координаты точек, по которым должен пролететь БПЛА. Однако, в случае, когда маршрут строится на

основе простых отрезков между точками, могут возникнуть изломы в траектории, где угол между отрезками очень маленький. Такие места могут представлять опасность для БПЛА. В данной статье рассматриваются основные подходы к построению гладких маршрутов для БПЛА, анализируются их преимущества и недостатки.

Самый простой способ – ручное исправление маршрута оператором. Исправление может достигаться перемещением точек, или добавлением новых. Данные действия проводятся для изменения угла излома в критических местах. Этот способ не является удобным, но с другой стороны он практически не требует вычислительных затрат.

Кривые Безье. Кривые Безье играют ключевую роль в задачах сглаживания данных по точкам, обеспечивая плавный вид линий и контуров в графических и инженерных приложениях. Их уникальные математические свойства позволяют эффективно сглаживать переходы между точками, минимизируя резкие изменения и создавая гармоничные формы. Кривые Безье описываются в параметрической форме:

$$x = P_x(t) \tag{1}$$

$$y = P_y(t) \tag{2}$$

Значение t выступает как параметр, которому отвечают координаты отдельной точки линии (t ∈ [0,1]). Многочлены Безье для P_X и P_y имеют следующий вид:

$$P_x(t) = \sum_{i=0}^{m} C_m^i t^i (1-t)^{m-i} x_i \tag{3}$$

$$P_y(t) = \sum_{i=0}^{m} C_m^i t^i (1-t)^{m-i} y_i \tag{4}$$

где C_m^i – сочетание n по i, x_i, y_i – координаты заданных точек. Степень полинома m будет на единицу меньше, чем количество заданных точек n.

Отсюда вытекает очевидная проблема использования кривых Безье: сложность вычислений растет экспоненциально с увеличением количества точек. Кроме того, в общем случае кривая Безье будет проходить только через первую и последнюю точки траектории (рисунок 1), что может быть непозволительно для некоторых задач.

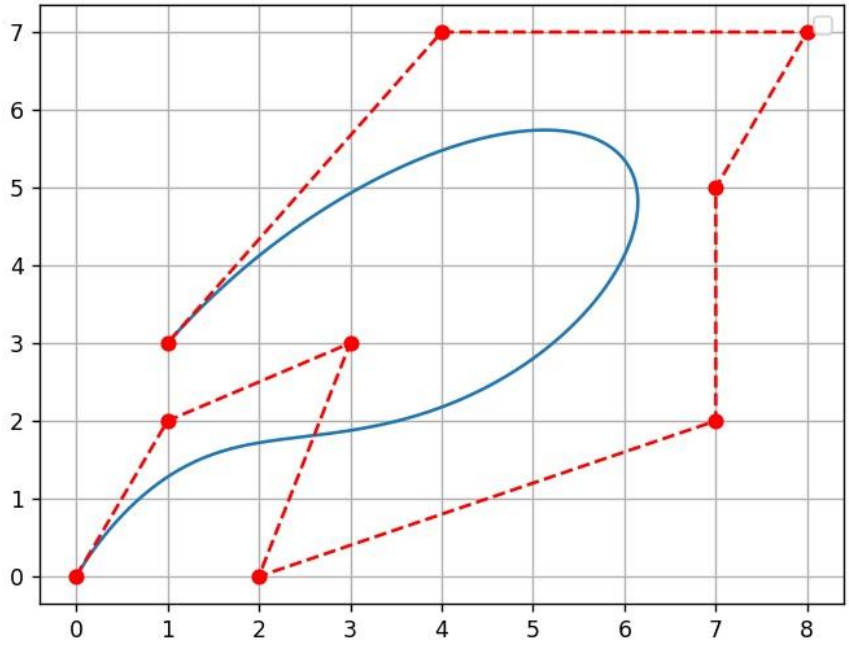


Рисунок 1 – Кривая Безье (n-1)-го порядка

Для того, чтобы уменьшить вычислительную сложность, можно использовать кривые Безье второй степени для каждой тройки точек (рисунок 2). Также такой подход позволяет увеличить количество точно охваченных точек до [n/2] + 1. Из рисунка явно виден недостаток такого подхода: общая кривая хоть и непрерывна, но в местах стыка двух кривых Безье второго порядка могут образовываться изломы. Эта ситуация может быть решена добавлением в траекторию новых точек, которые будут сглаживать места стыков (рисунок 3).

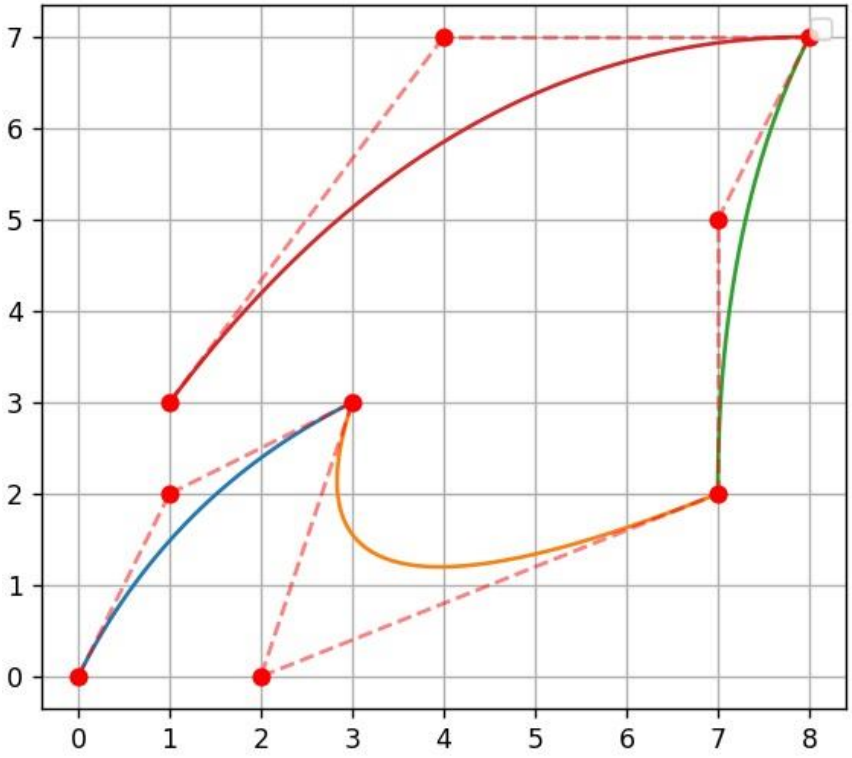


Рисунок 2 – Кривые Безье второго порядка

Таким образом, можно выделить следующие преимущества использования кривых Безье для построения маршрута по точкам:

1) Простота вычислений для кривых второго порядка;

2) Кривая Безье аффинно инвариантна, т. е. ее можно масштабировать и изменять не нарушая ее стабильность;

3) Возможность изменять маршрут и приближать его к желаемому в интерактивном режиме путем добавления новых точек;

4) Кривые Безье не зависят от расположения точек в пространстве.

Недостатками такого подхода являются:

1) Неполный охват точек изначальной траектории;

2) Необходимость совершения дополнительных действий для решения нештатных ситуаций (добавление новых точек или изменение координат для избежания образования изломов кривой).

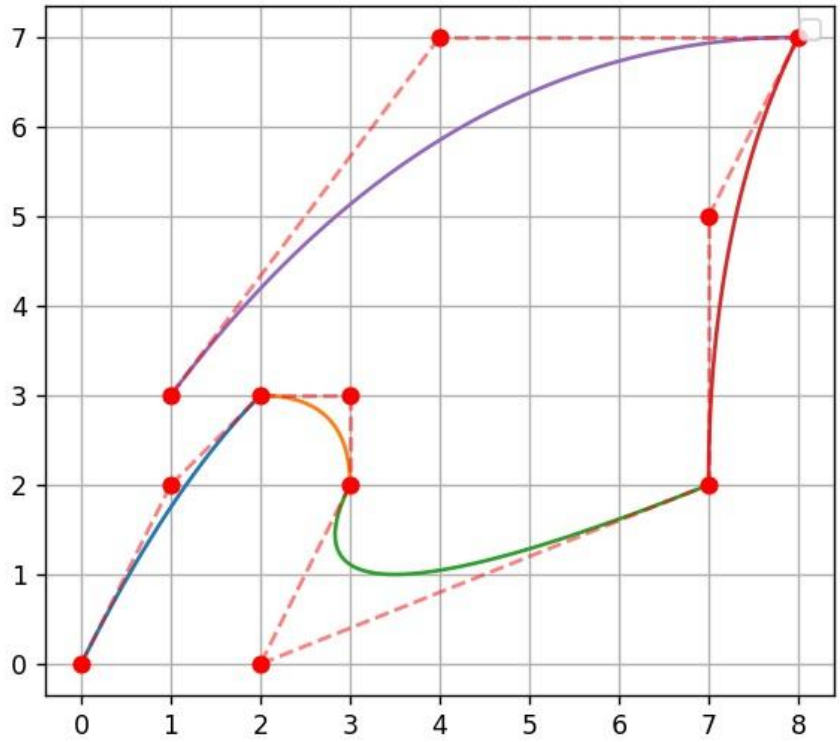


Рисунок 3 – Кривые Безье второго порядка с корректирующими точками

Многочлен Эрмита. Идейным продолжением использования совокупности кривых Безье второго порядка является интерполяционный многочлен Эрмита. Если помимо значений координат точек даны значения из производных до некоторого $k_i - 1$ порядка, то многочлен $P_m(x)$ степени $m = k_0 + k_1 + \ldots + k_n - 1$ удовлетворяющий условиям:

$$P_m(x_i) = y_i; P'_m(x_i) = y'_i; \ldots; P_m^{(k_i-1)}(x_i) = y_i^{(k_i-1)}; \quad (5)$$

где $i = (0,1,\ldots,n)$, называется интерполяционным многочленом Эрмита и является единственным. Нетрудно убедиться в том, что для двух соседних точек с заданными первыми производными многочлен Эрмита будет кубическим и имеет вид:

$$P_3(x) = y_{i-1}\frac{(x-x_i)^2(2(x-x_{i-1})+h_i)}{h_i^3} + y'_{i-1}\frac{(x-x_i)^2(x-x_{i-1})}{h_i^2} + \\ + y_i\frac{(x-x_{i-1})^2(2(x_i-x)+h_i)}{h_i^3} + y'_i\frac{(x-x_{i-1})^2(x-x_i)}{h_i^2} \quad (6)$$

где $h_i = x_i - x_{i-1}$.

В данной задаче, как и в большинстве реальных, значения производных неизвестны. Как показано в [1], значения производных можно найти в качестве наклонов сплайна $s_i = y'_i (i = 0,1,\dots,n)$, решив СЛАУ размера $n \times n$.

Тогда маршрут по точкам представляет собой совокупность многочленов Эрмита через каждую пару точек траектории (рисунок 4).

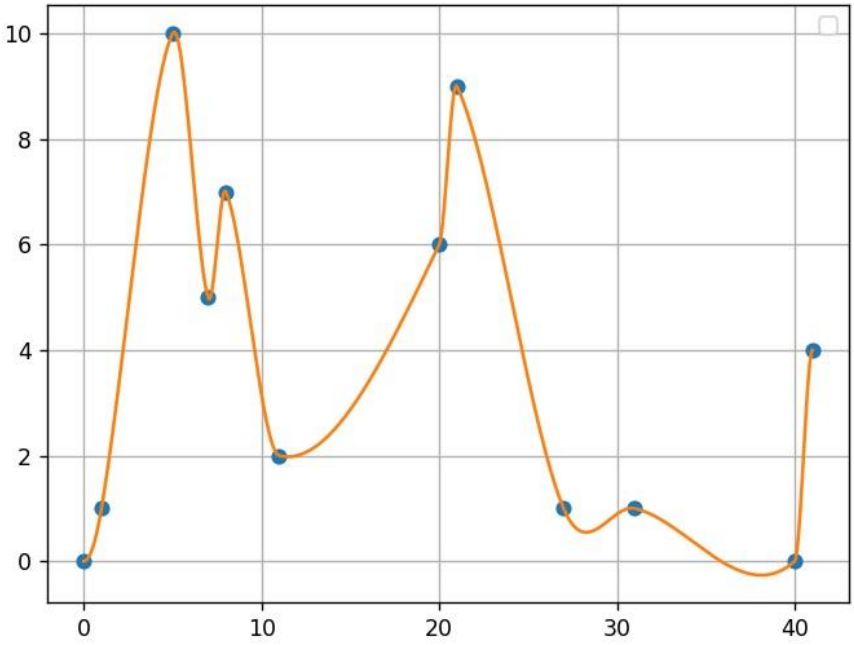


Рисунок 4 – Интерполяция с помощью многочлена Эрмита

Такой способ обеспечивает гладкость кривой на всем ее протяжении за счет равенства первых и вторых производных в заданных точках. Интерполяция с помощью многочлена Эрмита позволяет не проводить дополнительную коррекцию маршрута, после выполнения алгоритма. При добавлении новой точки в траекторию потребуется пересчитать лишь 2 кубических многочлена.

Главным недостатком данного алгоритма является то, что точки должны быть отсортированы по возрастанию координаты x. Это не позволяет описывать маршрут произвольного вида, что зачастую критично для БПЛА. Для устранения данного недостатка можно разбить несортированную последовательность точек на несколько функций, каждая по отдельности из которых будет состоять из сортированного по координате x набора точек.

Кубические сплайны с центростремительной параметризацией. В основе данного алгоритма лежит интерполяция кубическими сплайнами. Метод центростремительной параметризации зарекомендовал себя как одновременно простой и надежный способ выбора параметрической шкалы в задачах интерполяции кривых. Известно [3], что этот тип параметризации дает в целом лучшие результаты, чем хордовый. Вычисление параметров сплайнов происходит по формуле:

$$t_0 = 0$$
$$t_i = t_{i+1} + \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \quad (7)$$
$$t_n = 1$$

После нахождения всех параметров можно построить кубические сплайны от параметра t $S_i(t), (i = 0,1,\dots,n-1)$ для всех пар заданных точек. Полученная интерполяция (рисунок 5) является самой лучшей для решения данной задачи.

Преимуществами данного способа построения маршрута по точкам являются:

1) Кривая охватывает все заданные точки;

2) Кривая является гладкой, поскольку она непрерывно дифференцируема, что доказано в [2];

3) Интерполирующая кривая не имеет больших отклонений линейной интерполяции;

4) Инвариантность к расположению точек в пространстве.

Заключение. В данной статье мы рассмотрели несколько методов построения маршрутов для БПЛА по заданным точкам. Были изучены кривые Безье, интерполяционные многочлены Эрмита и интерполяция сплайнами с использованием центростремительной параметризации. Каждый из этих методов имеет свои уникальные преимущества и применяется в зависимости от конкретных требований и условий задач.

Кривые Безье предоставляют простую и интуитивно понятную методику построения гладких кривых. Их основное преимущество заключается в легкости управления формой кривой с помощью контрольных точек. Однако, для сложных траекторий с большим числом точек требуется использование кусочных кривых Безье, что может усложнить реализацию.

Интерполяционные многочлены Эрмита позволяют точно учитывать как значения функции в узловых точках, так и их производные, что обеспечивает высокую точность интерполяции.

Интерполяция сплайнами с центростремительной параметризацией представляет собой гибкий и мощный метод, позволяющий строить плавные и естественные кривые даже для сложных наборов точек. Центростремительная параметризация особенно

эффективна для неравномерно распределенных точек, обеспечивая устойчивость и точное следование форме маршрута.

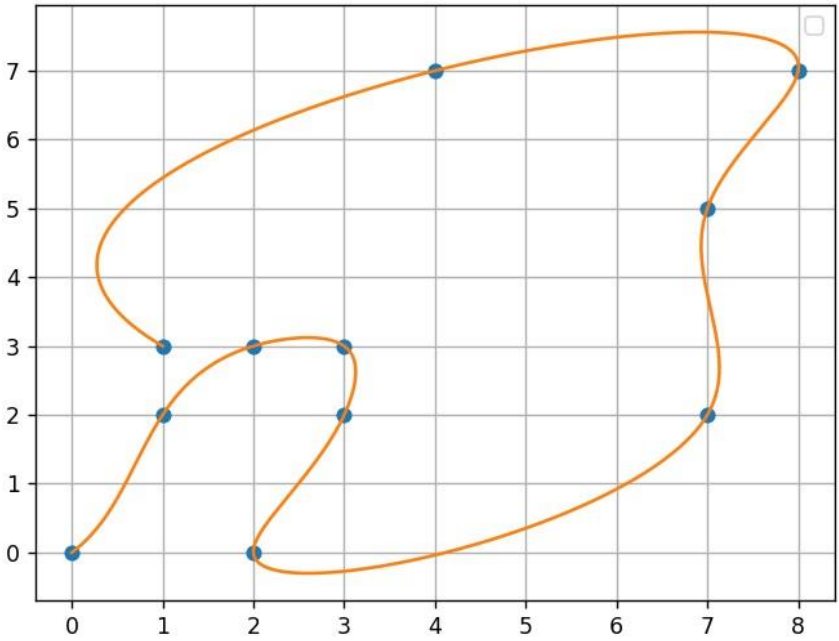


Рисунок 5 – Интерполяция кубическими сплайнами с использованием центростремительной параметризации

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Михеева, Л.Б., Скворцов, С.В. Методы вычислительной математики: Учеб. пособие / Рязан. гос. радиотехн. акад. — Рязань: РГРТА, 2005. — 80 с. — ISBN 5-7722-0258-8.

2. Кузнецов, Е.Б., Якимович, А.Ю. Наилучшая параметризация в задачах приближения кривых и поверхностей. — Москва: МАИ, 2005.

3. Простейшая интерполяция кривой без ограничений [Электронный ресурс] // quaoar.su. — URL: https://quaoar.su/blog/page/prostejshaja-interpoljacija-krivoj-bez-ogranichenij (дата обращения: 12.05.2024).

4. Кривые Безье [Электронный ресурс] // studfile.net. — URL: https://studfile.net/preview/954969/page:11/ (дата обращения: 08.05.2024).

УДК 004.65

Е. С. Гук, С. В. Крошилина

ОБРАБОТКА ЕСТЕСТВЕННОГО ЯЗЫКА В СИСТЕМЕ ВОПРОС-ОТВЕТ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

Статья посвящена обработке естественного языка в системах вопрос-ответ. Рассматриваются ключевые этапы, такие как морфологический, синтаксический, семантический и прагматический анализы, используемые для понимания вопросов и формирования ответов. Особое внимание уделяется применению машинного обучения и глубокого обучения для улучшения точности и релевантности ответов, а также вопросам многоязычности, контекстуальности и этики.

Ключевые слова: обработка естественного языка, система вопрос-ответ, машинное обучение, глубокое обучение, семантический анализ, многоязычность, контекстуальность.

Обработка естественного языка (ОНЯ) представляет собой одно из самых значимых и активно развивающихся направлений в области искусственного интеллекта. Она нацелена на создание систем, способных взаимодействовать с человеческим языком так, чтобы они (системы) понимали, интерпретировали и генерировали текстовые данные. Одним из важных приложений обработки естественного языка является система вопрос-ответ (QA), которая используется для поиска и предоставления ответов на вопросы, заданные на естественном языке. Эти системы находят широкое применение в различных областях, таких как виртуальные ассистенты, поисковые системы, образовательные платформы и службы поддержки клиентов [1-3]. В статье будет рассмотрены теоретические аспекты и принципы работы систем вопрос-ответ, основанных на обработке естественного языка.

Первым этапом в системе вопрос-ответ является понимание вопроса. Для этого используется ряд методов и техник, направленных на анализ текста. Эти методы включают морфологический анализ, синтаксический анализ, семантический анализ и прагматический анализ. Морфологический анализ включает разбор слов на морфемы и определение их грамматических характеристик. Морфемы - это наименьшие значимые единицы языка, такие как корни, приставки и суффиксы. Разбиение слова на морфемы помогает определить его базовое значение и грамматическую форму. Например, слово "переходящий" можно разложить на морфемы "пере-", "ход" и "-ящий". Синтаксический анализ направлен на определение грамматической структуры предложения. Он включает в себя построение синтаксических деревьев, которые отображают отношения между словами в предложении.

Синтаксический анализ позволяет понять, какое слово является подлежащим, какое - сказуемым, а какие - дополнениями или обстоятельствами. Это помогает выявить структуру вопроса и его основные компоненты. Семантический анализ связан с пониманием значения слов и фраз в контексте. Он включает задачи, такие как определение смысла слов, распознавание именованных сущностей и разрешение многозначности слов. Семантический анализ позволяет понять, о чем идет речь в вопросе, и интерпретировать его в зависимости от контекста. Например, слово "банк" может означать финансовое учреждение или берег реки, и семантический анализ помогает определить правильное значение. Прагматический анализ учитывает контекст и намерения говорящего. Он помогает понять, что именно хочет узнать пользователь и каким образом следует интерпретировать его вопрос. Прагматический анализ важен для понимания подтекста, сарказма, иронии и других аспектов, которые не всегда явно выражены в тексте. Например, если пользователь спрашивает: "Какая сегодня погода?", система должна интерпретировать это как запрос на предоставление текущих метеорологических данных.

После анализа и понимания вопроса система переходит к этапу поиска информации [2]. Информация для ответа на вопрос может храниться в различных источниках, таких как базы данных, текстовые документы, веб-страницы или специализированные знания. В современном мире объем данных настолько велик, что для эффективного поиска необходимо использовать продвинутые методы индексирования и поиска. Системы вопрос-ответ часто используют технологии машинного обучения и глубокого обучения для улучшения поиска информации. Алгоритмы машинного обучения позволяют моделям обучаться на больших объемах данных и распознавать паттерны, что улучшает точность и релевантность найденных ответов [1]. Глубокое обучение, основанное на нейронных сетях, особенно эффективно для задач, связанных с обработкой естественного языка. Эти модели обучаются на огромных объемах текстовых данных и способны учитывать контекст, что позволяет им находить более точные и релевантные ответы [2].

Когда релевантная информация найдена, система должна сформировать ответ. Этот этап также включает использование семантического и прагматического анализа для обеспечения точности и полезности ответа. Семантический анализ помогает понять смысл найденной информации, а прагматический анализ учитывает контекст и намерения пользователя [3]. Например, если пользователь спрашивает о погоде, система должна не только найти данные о погоде, но и представить их в удобном для восприятия формате. Это может включать использование графиков, таблиц или простого текста, в зависимости от предпочтений пользователя и контекста вопроса.

Несмотря на значительные успехи, системы вопрос-ответ сталкиваются с рядом вызовов, которые необходимо учитывать при их разработке и применении.

Многоязычность является одним из таких вызовов. Обработка вопросов на различных языках требует учета специфических лингвистических особенностей каждого языка. Это включает различия в грамматике, лексике и синтаксисе. Разработка многоязычных моделей и ресурсов остается сложной задачей, так как необходимо учитывать синтаксические, морфологические и семантические различия между языками. Это требует больших объемов данных и значительных вычислительных ресурсов для обучения моделей.

Контекстуальность вопросов также представляет собой вызов. Значение вопроса может сильно зависеть от контекста, в котором он был задан. Например, вопрос "Кто является президентом?" требует различных ответов в зависимости от страны, о которой идет речь. Для решения таких задач системы вопрос-ответ должны использовать контекстную информацию, которая может быть получена из предыдущих взаимодействий с пользователем или внешних источников данных.

Этика и конфиденциальность играют важную роль в разработке и применении систем вопрос-ответ. При анализе и хранении текстовых данных важно учитывать вопросы конфиденциальности и защиты персональных данных. Использование данных должно быть прозрачным и соответствовать правовым нормам и стандартам. Разработчики QA-систем должны учитывать этические аспекты, чтобы избежать предвзятости и дискриминации, которые могут возникать в процессе обработки данных.

Обработка естественного языка в системах вопрос-ответ является ключевым направлением в области искусственного интеллекта. Эти системы играют важную роль в улучшении взаимодействия человека и компьютера, делая его более естественным и интуитивным. Они находят широкое применение в различных сферах, таких как виртуальные ассистенты, которые помогают пользователям в повседневных задачах, поисковые системы, обеспечивающие быстрый доступ к информации, образовательные платформы, которые поддерживают обучение и развитие, и службы поддержки клиентов, которые помогают решать проблемы и отвечать на вопросы пользователей.

Будущее развитие этой области будет направлено на создание интеллектуальных и адаптивных систем, способных понимать сложные вопросы и предоставлять точные ответы в реальном времени. Это включает улучшение моделей глубокого обучения, разработку новых алгоритмов для обработки естественного языка и создание более богатых лингвистических ресурсов. Системы вопрос-ответ станут еще более

эффективными и полезными, удовлетворяя растущие потребности пользователей в получении точной и релевантной информации.

Таким образом, обработка естественного языка представляет собой мощный инструмент для создания систем вопрос-ответ, который позволяет извлекать полезные знания из текстовой информации и применять их в различных областях. Развитие этой области открывает новые горизонты для создания интеллектуальных систем, которые смогут удовлетворить растущие потребности пользователей в получении точной и релевантной информации.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Крошилина С.В. Предметно-ориентированные информационные системы: учебное пособие / А.В. Крошилин, С.В. Крошилина, Г.В. Овечкин. — Москва: Курс, 2023. — 176 с. — (Естественные науки).

2. Крошилин А. В., Крошилина С. В. Интеллектуальные поисковые системы на основе нечеткой логики. Учебное пособие для вузов. - М.: Горячая линия – Телеком, 2023. – 140 с.: ил.

3. Каширин И.Ю., Крошилин А.В., Крошилина С.В. Автоматизированный анализ деятельности предприятия с использованием семантических сетей. - М.: Горячая линия - Телеком, 2011. - 140 с.: ил.

УДК 004.932.2

А.В. Елисеева, Л.А. Шестопалов, Г.В. Овечкин

ИССЛЕДОВАНИЕ АЛГОРИТМОВ СЖАТИЯ ДАННЫХ БЕЗ ПОТЕРЬ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

Рассмотрены методы сжатия без потерь на разных входных данных.

Ключевые слова: сжатие без потерь, алгоритм Хаффмана, алгоритм Лемпеля-Зива-Велча.

Алгоритмы сжатия данных играют важную роль в современном мире информационных технологий, обеспечивая эффективное уменьшение объема хранимых и передаваемых данных без потери информации. Эти алгоритмы являются основой для множества приложений, начиная от сжатия изображений и видео до передачи файлов в сети Интернет.

Алгоритмы сжатия данных могут быть разделены на две основные категории: сжатие данных без потерь и сжатие данных с потерями. Вот их основные отличия:

Алгоритмы сжатия данных без потерь уменьшают размер данных, не теряя никакой информации. Это означает, что после сжатия и

последующего восстановления данных, они будут полностью идентичны оригинальным данным. К таким алгоритмам относятся алгоритм Хаффмана, алгоритм Лемпела-Зива-Велча (LZW), алгоритм дельта-сжатия.

Алгоритмы сжатия данных с потерями позволяют достичь более высокой степени сжатия за счет удаления некоторой информации, которая может быть восстановлена только приблизительно. Это приводит к некоторым изменениям в исходных данных после восстановления. К таким алгоритмам относятся алгоритм JPEG (для сжатия изображений), алгоритм MP3 (для сжатия звуковых файлов), алгоритм MPEG (для сжатия видеофайлов).

В данной статье будут исследованы алгоритмы сжатия данных без потерь.

Исследование алгоритмов сжатия данных без потерь остается актуальным по нескольким причинам:

1. Экономия места: Сжатие данных без потерь позволяет уменьшить объем хранимой информации, что особенно важно в условиях ограниченного пространства на устройствах хранения данных, в облаке и при передаче данных по сети. Это может быть критически важным для экономии места на жестких дисках, флэш-накопителях, серверах и других устройствах.

2. Сохранение качества: При сжатии данных без потерь сохраняется каждый бит исходной информации, что важно для приложений, где даже незначительные потери данных могут быть недопустимы. Например, в медицинских системах, финансовых данных или при передаче текстовой информации.

3. Безопасность и целостность данных: Сжатие данных без потерь обеспечивает сохранность исходной информации, что критически важно для областей, где целостность данных имеет первостепенное значение. Например, в архивировании файлов, передаче документов или хранении важных данных.

4. Увеличение скорости передачи данных: Сжатие данных без потерь может также способствовать увеличению скорости передачи данных по сети за счет уменьшения объема передаваемой информации. Это особенно важно для оптимизации производительности сетей и снижения времени передачи данных.

5. Развитие новых технологий: Исследования в области сжатия данных без потерь способствуют развитию новых технологий и методов, которые могут быть применены в различных областях, таких как медицина, финансы, наука и технологии.

Таким образом, исследования алгоритмов сжатия данных без потерь остаются актуальными из-за их значительного влияния на эффективное использование ресурсов хранения, безопасность и

целостность данных, а также оптимизацию скорости передачи информации.

Новизна данной работы заключается в исследовании алгоритмов сжатия на разных входных данных и анализ эффективности этих алгоритмов.

Данное исследование также предполагает обзор известных алгоритмов сжатия данных без потерь и анализ их недостатков.

Цель исследования: проверить алгоритмы сжатия данных без потерь на разных входных данных и определить наиболее эффективный из них.

Задачи:

- изучить алгоритмы сжатия данных;
- оценить эффективность сжатия;
- сравнить алгоритмы сжатия

Нами были изучены и исследованы алгоритмы Хаффмана и Лемпела-Зива-Велча (LZW).

Алгоритм Хаффмана – это алгоритм сжатия данных без потерь, разработанный Дэвидом Хаффманом в 1952 году. Суть алгоритма заключается в том, что он строит оптимальный префиксный код (код, в котором ни один кодовый символ не является префиксом другого) для каждого символа входного текста на основе их частоты встречаемости.

Суть работы алгоритма:

1. Подсчитывается частота каждого символа во входном тексте.

2. Создается дерево Хаффмана, в котором каждый символ представлен листом, а внутренние узлы образуются путем объединения узлов с наименьшей частотой.

3. Коды символов строятся следующим образом: движемся от корня дерева к каждому листу, присваивая "0" для левой ветви и "1" для правой ветви.

Нами была разработана программа, содержащая реализацию данного алгоритма. Программа была протестирована на нескольких типах входных данных: текст и изображение.

Результаты тестирования представлены в таблице 1.

Таблица 1 – Результаты тестирования алгоритма Хаффмана

Тип файла	**Расширение файла**	**Исходный размер файла**	**Размер файла после сжатия**	**Время сжатия**	**Степень сжатия**
Английский текст	.txt	7 КБ	4 КБ	0 с	1.75
Русские и английские спец символы (пример 1)	.txt	1015 КБ	772 КБ	3 с	1.32
Русские и английские спец символы (пример 2)	.txt	102 КБ	78 КБ	1 с	1.31
Русские и английские спец символы (пример 3)	.txt	51 КБ	39 КБ	0 с	1.31
Русские и английские спец символы (пример 4)	.txt	11 КБ	8 КБ	0 с	1.375
Русские и английские спец символы (пример 5)	.txt	2 КБ	1 КБ	0 с	2
Русский текст	.txt	10 КБ	6 КБ	0 с	1.67
Изображение (пример 1)	.png	7 КБ	3 КБ	0 с	2.33
Изображение (пример 2)	.png	20 КБ	19 КБ	1 с	1.05
Изображение (пример 3)	.png	2528 КБ	2510 КБ	1 мин 42 с	1.01
Изображение (пример 4)	.png	5250 КБ	5204 КБ	47 с	1.01
Изображение (пример 5)	.png	11759 КБ	11627 КБ	2 мин 1 с	1.01

Исходя из результатов тестирования, мы выявили плюсы и минусы данного алгоритма.

Плюсы:

1. Эффективность: Алгоритм Хаффмана позволяет достичь хорошего уровня сжатия данных, особенно если символы имеют различные частоты появления.

2. Быстродействие: Он отличается высокой скоростью работы при кодировании и декодировании данных.

3. Простота реализации: Алгоритм Хаффмана относительно прост в реализации и может быть легко понят и использован.

Минусы:

1. Неэффективность для некоторых типов данных: в случае, если все символы встречаются с примерно одинаковой частотой, алгоритм может не обеспечить значительного сжатия данных.

2. Необходимость передачи словаря: при передаче данных необходимо передавать словарь, содержащий информацию о построенных кодах символов.

3. Не всегда оптимальность: в некоторых случаях алгоритм может не построить оптимальный код из-за специфики данных или выбора начальных условий.

Алгоритм Лемпела-Зива-Велча (LZW) является одним из алгоритмов сжатия данных без потерь, который использует словарное кодирование для поиска и замены повторяющихся фрагментов данных.

Суть алгоритма:

1. Словарное кодирование: Алгоритм LZW строит словарь из фрагментов текста, которые встречаются повторно во входных данных.

2. Замена повторяющихся фрагментов: Вместо повторяющихся фрагментов данных алгоритм использует ссылки на предыдущие вхождения этих фрагментов в словаре.

Функционал:

1. Сжатие данных: LZW позволяет сжимать данные путем замены повторяющихся фрагментов на ссылки на эти фрагменты.

2. Построение словаря: Алгоритм строит словарь по мере прохождения по входным данным и обновляет его при необходимости.

В разработанной нами программе также содержится реализация данного алгоритма. Программа была протестирована на нескольких типах входных данных: текст и изображение.

Результаты тестирования представлены в таблице 2.

Таблица 2 – Результаты тестирования алгоритма Лемпеля-Зива-Велча

Тип файла	**Расширение файла**	**Исходный размер файла**	**Размер файла после сжатия**	**Время сжатия**	**Степень сжатия**
Английский текст	.txt	7 КБ	4 КБ	0 с	1.75
Русские и английские спец символы (пример 1)	.txt	1015 КБ	661 КБ	8 с	1.54
Русские и английские спец символы (пример 2)	.txt	102 КБ	67 КБ	1 с	1.52
Русские и английские спец символы (пример 3)	.txt	51 КБ	34 КБ	0 с	1.5
Русские и английские спец символы (пример 4)	.txt	11 КБ	7 КБ	0 с	1.57
Русские и английские спец символы (пример 5)	.txt	2 КБ	1 КБ	0 с	2
Русский текст	.txt	10 КБ	6 КБ	0 с	1.67
Картинка (пример 1)	.png	7 КБ	4 КБ	0 с	1.375
Картинка (пример 2)	.png	20 КБ	18 КБ	1 с	1.11
Картинка (пример 3)	.png	2528 КБ	2204 КБ	1 мин 49 с	1.15
Картинка (пример 4)	.png	5250 КБ	4569 КБ	2 мин 10 с	1.15
Картинка (пример 5)	.png	11759 КБ	10199 КБ	5 мин 4 с	1.15

Исходя из результатов тестирования, мы выявили плюсы и минусы данного алгоритма.

Плюсы:

1. Хорошее сжатие текстовых данных: LZV хорошо работает на текстовых данных с повторяющимися фрагментами.

2. Простота реализации: Алгоритм относительно прост в реализации и требует минимальных ресурсов для работы.

Минусы:

1. Неэффективность на некоторых типах данных: На некоторых типах данных, где повторяющиеся фрагменты отсутствуют или их мало, LZV может показать менее эффективное сжатие.

2. Размер словаря: Увеличение размера словаря может потребовать больше памяти, что может быть проблемой при работе с большими объемами данных.

Исходя из проведенных исследований, мы можем сделать следующие выводы:

При применении алгоритма Хаффмана для текстовых файлов степень сжатия варьируется от 1.31 до 2. Это связано с тем, что алгоритм Хаффмана хорошо работает с текстовой информацией, особенно когда в файле присутствует повторяющиеся символы или шаблоны.

При применении алгоритма Хаффмана для изображений степень сжатия составляет около 1.01, что говорит о незначительном сжатии.

При применении алгоритма Лемпеля-Зива-Велча для текстовых файлов степень сжатия варьируется от 1.5 до 1.75.

При применении алгоритма Лемпеля-Зива-Велча для изображений степень сжатия составляет от 1.11 до 1.375, что также указывает на то, что алгоритм Лемпеля-Зива-Велча может быть эффективным при сжатии изображений.

Таким образом, с помощью исследований мы выяснили, что алгоритм Хаффмана лучше подходит для сжатия текстовых файлов, а алгоритм Лемпеля-Зива-Велча – для сжатия изображений.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Основы теории информации: учеб. пособие/ Бубнов С.А., Овечкин Г.В., Филатов И.Ю.; Рязан. гос. радиотехн. ун-т. Рязань, 2023.

2. T. A. Welch, “A technique for high-performance data compression”, IEEE Computer, 17:6
(1984), 8–19.

3. J. Ziv, A. Lempel, “A universal algorithm for sequential data compression”, IEEE
Transactions on Information Theory, 23:3 (1977), 337–343.

4. J. Ziv, A. Lempel, “Compression of individual sequences via variable-rate coding”, IEEE Transactions on Information Theory, 24:5 (1978), 530–536.

УДК 519.687.7

Д. А. Елумеев, С.О. Алтухова

ОПТИМИЗАЦИЯ АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ ДЛЯ УСКОРЕНИЯ РАБОТЫ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Липецкий государственный педагогический университет имени П.П. Семенова-Тян-Шанского», Липецк

В статье рассматривается важность оптимизации в машинном обучении для повышения точности и эффективности моделей. Оптимизация включает в себя настройку гиперпараметров, выбор подходящих алгоритмов и улучшение общей эффективности модели. Также дается описание различных видов оптимизационных задач, таких как оптимизация функции потерь, размера и скорости модели, а также адаптация к нестационарным данным.

Ключевые слова: машинное обучение, оптимизация, минимизация функции потерь, градиентный спуск, сложные модели, адаптивные алгоритмы, квантовые алгоритмы.

В сфере машинного обучения оптимизация служит основой, которая позволяет алгоритмам обнаруживать закономерности и принимать решения с возрастающей точностью. По своей сути оптимизация заключается в поиске наилучших параметров или набора параметров для данной модели, которые минимизируют функцию затрат, которая является мерой того, насколько далеки прогнозы модели от фактических результатов. Этот процесс – это не просто математическое исследование; это стратегическое начинание, которое может существенно повлиять на бизнес-результаты за счет совершенствования прогнозных моделей, чтобы они были более точными и надежными. Выделяются следующие основные методы оптимизации в машинном обучении:

1. Градиентный спуск: это краеугольный камень оптимизации в машинном обучении. Это итеративный подход, при котором параметры модели обновляются в направлении, снижающем функцию затрат. Например, в модели логистической регрессии, прогнозирующей отток клиентов, градиентный спуск скорректирует веса, присвоенные характеристикам клиентов, чтобы минимизировать разницу между прогнозируемыми и фактическими показателями оттока.

2. Регуляризация: для предотвращения переобучения, когда модель хорошо работает с обучающими данными, но плохо с невидимыми данными, используются такие методы регуляризации, как L1 (лассо) и L2 (гребень). Эти методы добавляют штрафной член к функции затрат, поощряя более простые модели, которые лучше обобщают. Рассмотрим систему рекомендаций для платформы электронной коммерции; регуляризация гарантирует, что модель не будет слишком сильно зависеть от редких взаимодействий пользователя с товаром, которые могут не предсказывать будущее поведение.

3. Настройка гиперпараметров: помимо параметров модели, существуют гиперпараметров, которые определяют общее поведение алгоритма обучения. Для нахождения оптимального набора гиперпараметров используются такие методы, как поиск по сетке, случайный поиск и байесовская оптимизация. Например, в машине опорных векторов (SVM), используемой для классификации изображений, настройка гиперпараметров может включать поиск правильного типа ядра и штрафного параметра для максимальной точности классификации [1].

4. Стохастические методы: при работе с большими наборами данных стохастические методы, такие как стохастический градиентный спуск (SGD) и мини пакетный градиентный спуск, предлагают более приемлемый с вычислительной точки зрения подход. Эти методы обновляют параметры, используя подмножество данных, что ускоряет процесс оптимизации. Примером может служить обучение глубокой нейронной сети языковому переводу в реальном времени, где SGD может использоваться для эффективной обработки миллионов пар предложений.

5. Эволюционные алгоритмы: эти алгоритмы, основанные на естественном отборе, используют такие механизмы, как мутация, скрещивание и отбор, для разработки набора решений, повышающих производительность. Они особенно полезны, когда функция затрат не дифференцируема или пространство параметров дискретно. Практическим применением может быть оптимизация расположения компонентов на печатной плате, где эволюционные алгоритмы могут исследовать различные конфигурации для минимизации помех от сигнала.

Благодаря этим методам оптимизации, модели машинного обучения становятся более приспособленными к таким задачам, как прогнозирование тенденций рынка, персонализация взаимодействия с клиентами и автоматизация процессов принятия сложных решений. Конечная цель – использовать прогностические возможности этих моделей для стимулирования роста бизнеса, оптимизации операций и улучшения взаимодействия с клиентами [2]. Постоянно совершенствуя эти модели, предприятия могут сохранять конкурентное преимущество на постоянно развивающемся рынке.

В динамичной среде бизнеса способность быстро усваивать новые данные и соответствующим образом корректировать стратегии имеет первостепенное значение. Эта гибкость обеспечивается моделями машинного обучения, которые предназначены не только для прогнозирования, но и для непрерывного обучения на основе данных в реальном времени. Такие модели играют ключевую роль в выявлении едва заметных изменений в рыночных тенденциях, поведении потребителей и операционной эффективности, тем самым позволяя

предприятиям оставаться на опережение. Ключевые аспекты непрерывного обучения и адаптивных алгоритмов в машинном обучении:

1. Непрерывное обучение модели: традиционные модели машинного обучения обучаются на исторических данных, которые могут быстро устареть. Напротив, модели, оснащенные обучением в режиме реального времени, усваивают новые данные по мере их поступления, обновляя свои параметры "на лету". Это гарантирует, что прогнозы и решения всегда основываются на самой актуальной информации.

2. Адаптивные алгоритмы: алгоритмы, поддерживающие обучение в режиме реального времени, часто используют такие методы, как онлайн-обучение или инкрементное обучение. Эти методы позволяют модели адаптироваться к новым шаблонам без необходимости переобучения с нуля, экономя ценное время и вычислительные ресурсы.

3. Циклы обратной связи: внедрение циклов обратной связи имеет решающее значение для адаптации. Систематически учитывая результаты предыдущих решений, модель уточняет свои прогнозы на будущее, что приводит к успешному циклу улучшений.

4. Передовые вычисления: развитие передовых вычислений позволяет осуществлять обучение в режиме реального времени непосредственно там, где генерируются данные. Это сокращает задержки и обеспечивает более быстрое время отклика, что критически важно для таких приложений, как автономные транспортные средства или обнаружение мошенничества в режиме реального времени.

Пример: рассмотрим систему рекомендаций для платформы электронной коммерции. Модель, которая адаптируется в режиме реального времени, может сразу включать последние взаимодействия с пользователем, покупки и даже возвраты. Это означает, что рекомендации всегда соответствуют текущим предпочтениям пользователей, что потенциально повышает удовлетворенность клиентов и продажи.

Используя эти принципы, предприятия могут превратить свои модели машинного обучения в проактивные двигатели роста, способные ориентироваться в сложностях современных постоянно развивающихся рынков.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Анафиев А. С., Карюк А. С. ОБЗОР ПОДХОДОВ К РЕШЕНИЮ ЗАДАЧИ ОПТИМИЗАЦИИ ГИПЕРПАРАМЕТРОВ ДЛЯ АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ // ТВИМ. 2022. №2 (55). – URL: https://goo.su/tT5mP (дата обращения: 15.05.2024).

2. Корсун А. С. Алгоритмы оптимизации параметров с вычислением производной первого порядка в линейных методах машинного обучения // StudNet. 2020. №6. – URL https://goo.su/yG5tt (дата обращения: 16.05.2024).

УДК 004.9

С.Ю. Жулева

ПРИМЕНЕНИЕ ТЕНОЛОГИЙ БОЛЬШИХ ДАННЫХ BIG DATE В ЗДРАВООХРАНЕНИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

Рассмотрены особенности применения технологий Big Data в здравоохранении.

Ключевые слова: большие данные, Big Data, здравоохранение, информационные технологии.

Здравоохранение – это система взаимодействия между пациентами, врачами, больницами, фармацевтическими компаниями и управленческим аппаратом, поток информации между которыми подвергается анализу на основе методов и инструментов, предназначенных для обработки больших данных *Big Data*. *Big Data* обрабатывают постоянно генерируемые наборы цифровой информации с использованием информационных технологий. При этом набор данных, имеющий большой объем, разнообразие, несогласованность, требует постоянной проверки на достоверность, ценность и интерпретируемость, а также постоянно обновляется в режиме реального времени.

Применение информационных технологий для организации управления медицинского учреждения должно охватывать все подразделения и осуществлять формирование и отслеживание электронного документопотока (рис.1). Вся обрабатываемая информация представляет собой набор неструктурированных разнородных данных, которые невозможно обработать традиционными инструментами программного и аппаратного обеспечения.

Рисунок 1 – Информационная система медицинской организации

Технологии *Big Data* характеризуются: объемом (*Volume*), разнообразием (*Variety*), изменчивостью (*Variability*), скоростью (*Velocity*), достоверностью (*Veracity*) и ценностью (*Value*) [1, 2, 3].

1. Клинические исследования, банк данных пациентов, рентгеновского и магниторезонансного воздействия, статистические данные о пациентах, заболеваниях и банк знаний врачей – это данные, требующие непрерывного генерирования, накопления и объемного хранилища.

2. Медицинская информация отличается разнообразием и слабой структурированностью: работа с пациентами на уровне приема

специалистов (электронная регистратура, анамнез, постановка диагноза, используемая терапия, дополнительные исследования при помощи высокотехнологичных аппаратов), работа с фармакологическими компаниями, административно-управленческая деятельность, включая формирование документации внутри учреждения. При этом технология *Big data* на основе ретроспективного анализа позволяет закономерно формировать большие объемы данных.

3. За счет сезонной периодичности заболеваний трудно предсказать и спрогнозировать медицинские процессы и процесс принятия медицинских решений.

4. Непрерывно поступающий поток разнообразной информации требует реагирования и анализа в режиме реального времени.

5. Результаты анализа *Big Data* должны быть достоверными и максимально защищены от ошибок.

6. Ценность полученных сложных медицинских данных является базой для принятия медицинских решений в условиях постоянно меняющихся оценочных факторов.

В качестве источников медицинских данных при диагностировании могут выступать электронные медицинские карты, данные с датчиков устройств, данные о лекарственных средствах, данные медицинской науки; при организации административной работы – административно-паспортные данные, нормативно-законодательные документы.

Используемые системы в здравоохранении должны обеспечивать интерпретируемость сложных данных (рис. 2). Задача *Big Data* обработать большой объем данных для последующего принятия решений и тем самым повысить эффективность сбора, хранения, анализа и визуализации информации в сфере здравоохранения.

Работа с большими данными требует решения проблем фрагментации, стандартизации под используемые форматы, обеспечение безопасности и конфиденциальности медицинских данных. Негативное влияние оказывает нестабильность и неструктурированность исходных данных, постоянная проверка в режиме реального времени их достоверности, необходимость исключения лишней информации, удовлетворение требованиям медицинского учреждения, правовые и этические аспекты использования *Big Data* в соответствии с критериями доказательной медицины [2, 3].

Рисунок 2 – Технологии обработки больших данных

Таким образом, использование *Big Data* нацелено улучшить процессы исследования, анализа больших объемов медицинской информации: электронного документооборота медицинских учреждений, фармацевтического направления, страховых компаний, а также больших объемов медицинских данных, которые помогут улучшить работу с пациентами на этапе раннего диагностирования и последующего предотвращения развития заболевания, а также экстренного принятия мер во время возникновения эпидемий.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Цветкова Л.А., Черченко О.В. Внедрение технологий big data в здравоохранение: оценка технологических и коммерческих перспектив // Экономика науки. 2016. Том 2, № 2. С. 138-150.

2. Мамедова М.Г. Big Data в электронной медицине: возможности, вызовы и перспективы. İnformasiya texnologiyaları problemləri. 2016. №2. С. 9–29.

3. Карнаухов Н.С., Ильюхин Р.Г. Возможности технологий «Big Data» в медицине. Врач и информационные технологии. 2019, № 1. С. 59-63.

УДК 004.42

П. В. Журавлев, Т. А. Дмитриева

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ АНАЛИЗА БИРЖЕВОГО РЫНКА

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

В статье представлен интерфейс разработанного программного обеспечения для анализа биржевого рынка.

Ключевые слова: биржевой ранок, программное обеспечение, мобильное приложение.

На текущий момент не существует удобного мобильного приложения [1], оптимизирующего и автоматизирующего процессы

анализа финансовых данных. Требуется разработка высокопроизводительной системы [2], которая сможет оперативно обрабатывать большие массивы данных и предоставлять пользователю глубокий и структурированный анализ, учитывая расчет ликвидности, а также проверку закрытости ввода и вывода на биржах [3]. Поэтому вопрос разработки подобного программного обеспечения несомненно является актуальным и практически значимым.

Программный продукт «T-trade» устанавливается на систему Android или iOS. Для запуска программного продукта необходимо нажать на иконку приложения «T-trade» на смартфоне (рисунок 1). Далее необходимо авторизоваться под своей учётной записью в приложении или зарегистрировать новый аккаунт (рисунок 2).

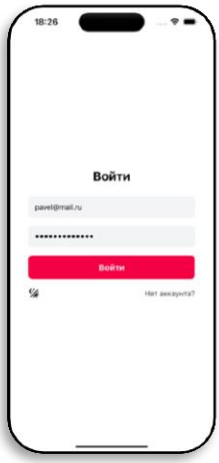


Рисунок 1 – Начальный экран приложения «T-trade»

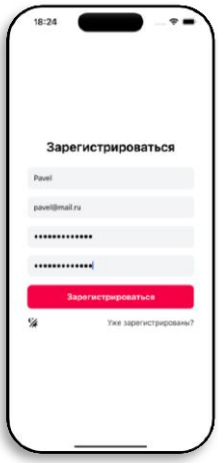


Рисунок 2 – Страница регистрации нового аккаунта

После авторизации открывается главная страница с графиком истории доходности, в верхней части размещены доступные биржи, а также выбранная основная валюта пользователя и его возможный доход (рисунок 3). В нижнем меню разработанного продукта можно переключаться между разработанными страницами приложения, на которых можно использовать все доступные функции. На рисунке 3 видно, что в приложении есть четыре основные страницы: главная, цепочки, доходы, профиль. Далее будет рассмотрена работа каждой из этих страниц.

После нажатия в меню на страницу «Chains» открывается страница с цепочками (рисунок 4).

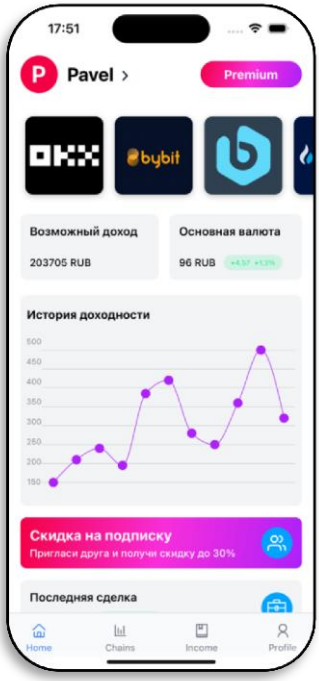


Рисунок 3 – Главная страница приложения

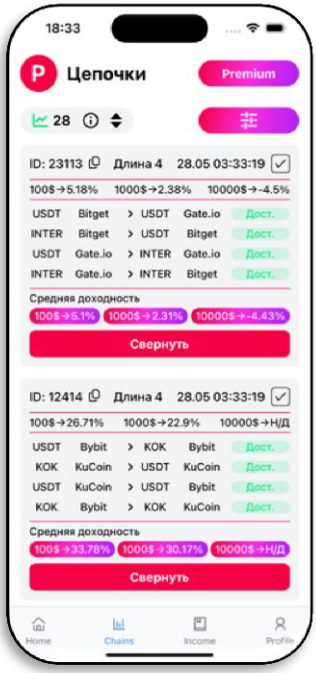


Рисунок 4 – Страница с цепочками приложения

Для фильтрации данных найденных цепочек необходимо нажать на иконку фильтров в правом верхнем углу экрана. В открывающемся окне можно задать различные параметры для фильтрации найденных цепочек, выбрав или введя соответствующие параметры (рисунок 5).

При отсутствии цепочек по заданным параметрам фильтров может быть выведено следующее сообщение «На данный момент прибыльных ликвидных цепочек не найдено». В случае если при применении фильтров нашлись цепочки, которые удовлетворяют условию, осуществляется вывод отфильтрованных цепочек (рисунок 6).

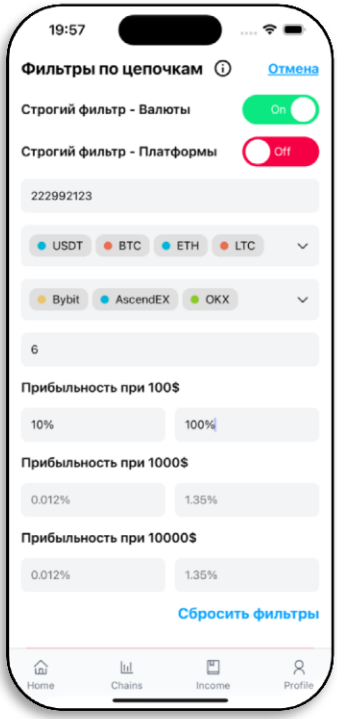


Рисунок 5 – Страница с фильтрами цепочек

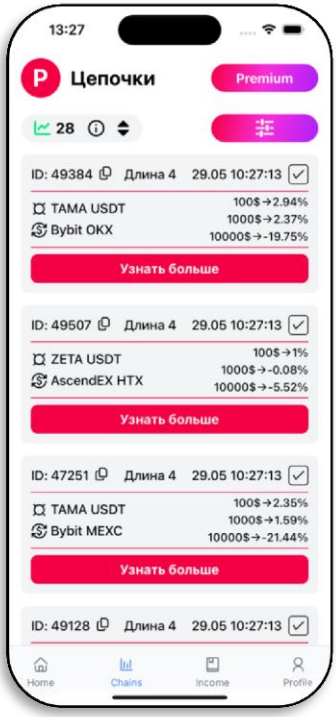


Рисунок 6 – Страница с примененным фильтром по цепочкам

Для ведения отчета о своих доходах, необходимо указать id цепочки, сумму входа и выхода, а также нажать на кнопку «Добавить», после ваши данные запишутся в историю всех сделок и отобразятся на графике выше (рисунок 7). На рисунке показано, как была сформирована личная история доходов на основе совершенных сделок.

После нажатия в меню на страницу «Profile» открывается профиль пользователя (рисунок 8). В нем можно изменить свои данные, сменить пароль или внести персональные настройки, а также контролировать количество привязанных устройств.

Таким образом, доступны возможности просмотра необходимой информации о найденных арбитражных цепочках, фильтрация их по различным параметрам, ведение собственной истории доходов, настройки личного профиля и контроль количество привязанных устройств.

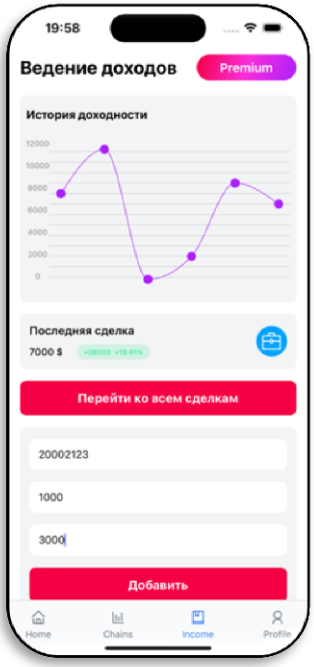


Рисунок 7 – Страница ведения доходов

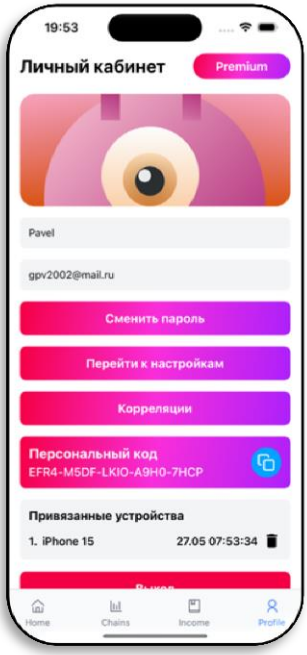


Рисунок 8 – Страница профиля пользователя

Таким образом, разработанный программный продукт «T-trade» обладает всеми функциональными возможностями, необходимыми для получения актуальной информации об арбитражных цепочках и их анализа с учетом ликвидности биржевого стакана. Тестирование программного продукта показало его работоспособность и эффективность в реальных условиях биржевой торговли. Пользователи системы могут оперативно получать информацию о возможностях для арбитража, что способствует более обоснованному и быстрому принятию решений в условиях динамично изменяющегося рынка.

Результаты работы подтвердили актуальность и практическую значимость темы исследования. Разработанный программный продукт вносит вклад в оптимизацию и автоматизацию процессов управления портфелями инвестиций и трейдинга, что способствует повышению доходности и снижению рисков для пользователей системы.

Выполненные исследования и разработки могут служить основой для дальнейших улучшений в области финансовых технологий и арбитража.

Для дальнейшего развития программного продукта «T-trade» возможны следующие улучшения.

1. Интеграция с дополнительными биржами: расширение списка поддерживаемых бирж увеличит возможности для арбитража и позволит пользователям системы получать более комплексный анализ рынка.

2. Улучшение алгоритмов машинного обучения: применение продвинутых методов машинного обучения для анализа рыночных данных может значительно повысить точность прогнозирования и эффективность арбитражных операций.

3. Улучшение системы безопасности: поскольку система обрабатывает конфиденциальную финансовую информацию, повышение уровня безопасности будет способствовать защите данных пользователей от несанкционированного доступа.

4. Разработка инструментов для визуализации данных: улучшенные графические интерфейсы и инструменты визуализации могут помочь пользователям лучше анализировать рыночные условия и принимать обоснованные решения.

5. Расширение функциональности анализа рисков: внедрение дополнительных инструментов для анализа и управления рисками повысит надежность и стабильность работы системы, что особенно важно в условиях высокой волатильности биржевых рынков.

Эти направления позволят не только улучшить существующую систему, но и обеспечат её долгосрочное развитие, а также повысят уровень удовлетворенности и лояльности пользователей.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Виггинс Дж. Разработка алгоритмов для анализа финансовых данных / Дж. Виггинс. – М.: ДМК Пресс, 2021. – 352 с.

2. Жуков С.В. Алгоритмическая торговля на финансовых рынках / С.В. Жуков. – М.: ДМК Пресс, 2023. – 448 с.

3. Кондратьев С.А. Анализ данных для финансовых рынков / С.А. Кондратьев. – М.: Финансы и статистика, 2022. – 396 с.

УДК 004.93'14

А.Н. Кабочкин, Г.В. Овечкин

ГЛУБОКИЕ СВЁРТОЧНЫЕ НЕЙРОННЫЕ СЕТИ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

Свёрточные нейронные сети (CNN) представляют собой один из ключевых подходов в области машинного обучения, способствующий значительным успехам в распознавании образов, анализе изображений и других задачах компьютерного зрения. В данной статье рассматриваются основные

принципы работы CNN, их архитектуры, обучения, а также примеры применения в различных областях.
Ключевые слова: Свёрточные нейронные сети (CNN), принципы работы CNN, распознавание образов, компьютерное зрение.

Свёрточные нейронные сети (CNN) – это класс глубоких нейронных сетей, разработанных для обработки данных с сетевой структурой, таких как изображения. CNN значительно улучшили точность и эффективность в задачах распознавания образов и классификации, став основой многих современных приложений в области компьютерного зрения. Простыми словами берётся изображение, которое пропускается через несколько свёрточных, нелинейных слоёв, слоёв объединения и полносвязных слоёв, и после чего генерируется вывод. Выводом может быть класс или вероятность классов, которые лучше всего описывают изображение.

Задача классификации изображений заключается в том, чтобы на основе входного изображения определить его класс (например, кошка, собака и т.д.) или группу наиболее вероятных классов, которые наилучшим образом описывают изображение (см. Рисунок 1). Этот процесс является одним из первых навыков, которые человек начинает развивать с самого рождения.

Рисунок 1 - Пример преобразования изображения

Основные принципы работы CNN

CNN отличаются от традиционных нейронных сетей использованием свёрточных и пулинг слоёв. Эти слои позволяют CNN извлекать и иерархически обрабатывать пространственные особенности входных данных (рисунок 2).

Свёрточный слой использует фильтры (ядра свёртки), которые сканируют входное изображение, создавая карты признаков. Это позволяет выделять такие характеристики, как края, углы и текстуры на различных уровнях абстракции.

Пулинг слои (максимальный и средний пулинг) используются для уменьшения размерности карт признаков, что снижает вычислительную сложность и повышает устойчивость к небольшим трансформациям и смещениям входных данных.

Полносвязные (fully connected) слои располагаются в конце сети и служат для объединения выделенных признаков и принятия окончательного решения, например, классификации изображения.

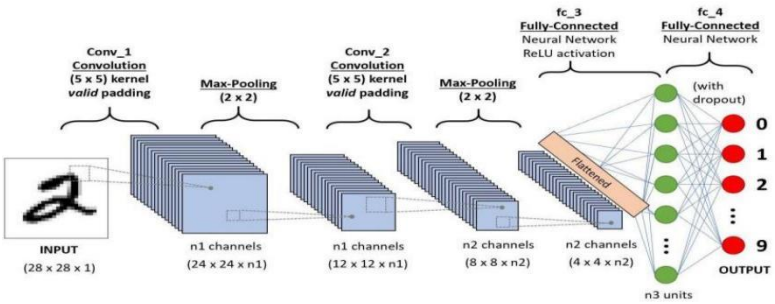


Рисунок 2 - Процесс работы сверточной нейронной сети

Когда компьютер обрабатывает изображение, оно представляется ему в виде массива пикселей. Размер этого массива зависит от разрешения и формата изображения. Например, цветное изображение в формате JPG размером 480x480 пикселей будет представлено массивом размером 480x480x3, где 3 соответствуют каналам RGB. Каждый пиксель в этом массиве характеризуется числовым значением от 0 до 255, отражающим его цветовую интенсивность.

Эти числа сами по себе не имеют смысла для человека, когда мы пытаемся понять содержание изображения. Однако компьютер использует эту матрицу для вычисления вероятностей принадлежности изображения к определённым классам (.80 для кошки, .15 для собаки, .05 для птицы и так далее).

Оптимизация вычислительной эффективности и устойчивости CNN

Несмотря на значительный прогресс в развитии CNN, существует ряд проблем, которые требуют решений для дальнейшего повышения эффективности и точности моделей.

Одна из основных проблем, связанных с применением CNN, заключается в высокой вычислительной сложности и необходимости значительных ресурсов для обучения и эксплуатации моделей. Это особенно актуально для систем с ограниченными вычислительными ресурсами, таких как мобильные устройства и встроенные системы. Кроме того, устойчивость моделей к небольшим трансформациям и смещениям входных данных также остаётся важной задачей.

Для решения данной проблемы предлагается использование максимального и среднего пулинг слоёв, которые позволяют значительно уменьшить размерность карт признаков, что снижает вычислительную сложность и повышает устойчивость моделей к небольшим трансформациям и смещениям входных данных.

Применение пулинг слоёв для повышения эффективности

Пулинг слои (или подвыборка) являются важным компонентом архитектуры свёрточных нейронных сетей, существенно влияющим на их производительность и устойчивость. Существуют два основных типа пулинга: максимальный и средний (рисунок 3).

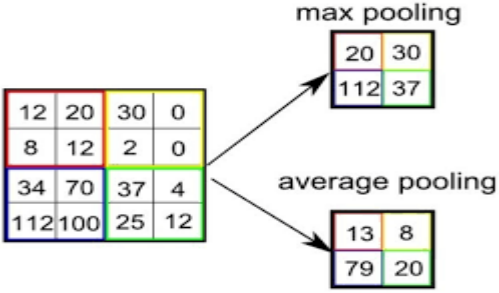


Рисунок 3 - Пример распределения пулинг слоёв

Максимальный пулинг выбирает максимальное значение из заданного окна (например, 2x2 пикселя), проходящего по входному изображению с определённым шагом. Этот процесс позволяет извлечь наиболее выраженные признаки, такие как края и углы, и уменьшить размерность выходных данных, сохраняя важные пространственные характеристики.

Средний пулинг (или усредняющий пулинг) вычисляет среднее значение в пределах окна. Этот метод также уменьшает размерность данных, но сохраняет больше информации о контексте, хотя и менее выраженно, чем максимальный пулинг.

Снижение вычислительной сложности

Пулинг слои уменьшают размерность выходных карт признаков, что ведёт к уменьшению количества вычислений в последующих слоях сети. Например, если входная карта признаков имеет размер 28x28 пикселей, применение максимального пулинга с окном 2x2 и шагом 2 приведёт к выходной карте размером 14x14 пикселей. Это сокращение в 4 раза уменьшает объём данных, которые необходимо обрабатывать, что существенно снижает вычислительную сложность.

Кроме того, уменьшение размерности данных позволяет сократить количество параметров в последующих полносвязных слоях, что дополнительно уменьшает объём вычислений и память, необходимую для хранения параметров модели.

Повышение устойчивости

Пулинг слои также способствуют повышению устойчивости моделей к небольшим трансформациям и смещениям входных данных. Максимальный пулинг, например, сохраняет наиболее выраженные

признаки независимо от их точного положения в окне. Это означает, что небольшие сдвиги или изменения в масштабе входного изображения не сильно повлияют на выходные карты признаков, что делает модель более устойчивой к вариациям входных данных.

Средний пулинг, усредняя значения в окне, снижает чувствительность модели к незначительным шумам и колебаниям в данных. Таким образом, модель становится более стабильной и менее подверженной ошибкам из-за несущественных изменений входных данных.

Архитектуры CNN

Существуют различные архитектуры CNN, каждая из которых разработана для улучшения точности и эффективности. Рассмотрим несколько популярных архитектур:

Одной из первых успешных архитектур CNN была LeNet, разработанная Яном Лекуном для распознавания рукописных цифр. Она состоит из нескольких свёрточных и пулинг слоев, за которыми следуют полносвязные слои.

AlexNet значительно улучшила результаты на ImageNet, увеличив глубину сети и применяя методы регуляризации, такие как Dropout. Введение нелинейной функции активации ReLU также способствовало успеху этой архитектуры.

VGGNet использует небольшие свёрточные ядра (3x3), что позволило создавать более глубокие сети с более управляемым числом параметров. VGGNet продемонстрировала важность глубины сети для повышения точности.

ResNet (Residual Networks) предложила концепцию остаточных соединений (skip connections), что позволило создавать чрезвычайно глубокие сети, избегая проблемы затухания градиентов.

Обучение

Важно понять, как нейронная сеть автоматически настраивает свои веса (или фильтры) в процессе обучения. Процесс, известный как обратное распространение ошибки, является основой для обучения нейронных сетей и позволяет им эффективно извлекать признаки из данных.

Обратное распространение ошибки можно разбить на четыре основных этапа: прямое распространение, функцию потерь, обратное распространение и обновление весов. Прямое распространение начинается с входного изображения размером 32x32x3 и проходит через все слои сети. В начале обучения, когда все веса или значения фильтров случайно инициализированы, выходной результат может быть равным, например, [0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1], что не дает предпочтения определенному классу. Сеть с такими весами не может находить базовые свойства изображения и корректно определять его класс. Это приводит к

функции потерь, которая измеряет расхождение между фактическим и предсказанным классами изображения.

Функция потерь часто выражается через среднеквадратичную ошибку, формула (1), где разница между реальным и предсказанным значением возводится в квадрат и делится на 2. Например, если первое изображение в тренировочном наборе представляет собой цифру 3, его метка будет [0 0 0 1 0 0 0 0 0 0]. Тогда функция потерь покажет, насколько сильно предсказание отличается от этой метки.

$$E_{total} = \sum \frac{1}{2}(target - output)^2 \quad (1)$$

Чтобы сеть могла учиться правильно предсказывать классы, необходимо минимизировать функцию потерь. Это эквивалентно математической задаче оптимизации, где требуется найти наименьшие значения функции потерь путем корректировки весов (или фильтров) сети. Для визуализации этого процесса можно представить трехмерный график (Рисунок 4), где веса нейронной сети являются независимыми переменными, а функция потерь — зависимой переменной. Оптимальная точка на графике соответствует минимальной потере, и цель состоит в том, чтобы настроить веса сети в направлении наискорейшего убывания потерь.

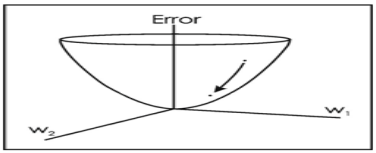


Рисунок 4 - График для функции потерь

Математически это выражается как производная функции потерь по весам (dL/dW), где W — веса определенного слоя. Обратное распространение ошибки вычисляет, какие веса вносят наибольший вклад в общую потерю и как их следует корректировать для уменьшения потерь. После вычисления производной происходит последний этап — обновление весов, где каждый вес фильтра обновляется в направлении, обратном градиенту функции потерь.

Одно из наиболее известных применений CNN – это распознавание объектов на изображениях. Современные системы способны распознавать сотни категорий объектов с высокой точностью.

В медицине CNN используются для анализа медицинских изображений, таких как рентгеновские снимки и МРТ, помогая в диагностике заболеваний, включая рак.

В системах автономного вождения CNN применяются для распознавания дорожных знаков, пешеходов и других транспортных средств, обеспечивая безопасность на дорогах.

CNN также находят применение в задачах обработки естественного языка, таких как анализ тональности текстов и машинный перевод, особенно когда данные можно представить в виде матриц.

Заключение

Свёрточные нейронные сети продолжают оставаться одним из самых эффективных инструментов для анализа визуальных данных. Постоянное развитие архитектур и алгоритмов обучения делает CNN всё более мощными и универсальными, открывая новые возможности в различных областях. В будущем можно ожидать дальнейшего совершенствования этих технологий и расширения их применения в новых сферах, таких как робототехника, виртуальная реальность и многое другое.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Рафаэл С. Гонсалес, Ричард Е. Вудс Цифровая обработка изображений/Пер. с англ. - М.: Техносфера, 2012. – 1105с.

2. Митрофанова, Е. Ю. Нейросетевые технологии обработки информатики. Методы и технологии глубокого обучения : учебное пособие / Е. Ю. Митрофанова, А. А. Сирота, М. А. Дрюченко .— Воронеж: Издательский дом ВГУ, 2019 .— 197 с.

3. Бутенко, В. В.Поиск объектов на изображении с использованием алгоритма адаптивного усиления / В. В. Бутенко // Молодой ученый. – 2015. – No4. – С. 52–56.

4. R-CNN, Fast R-CNN, Faster R-CNN, YOLO – Object Detection Algorithms –https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e.

УДК 007:681.512.2

И.Ю. Каширин

ПРОБЛЕМА НОРМАЛИЗАЦИИ ИЕРАРХИЧЕСКИХ ЧИСЕЛ ДЛЯ ИХ ИСПОЛЬЗОВАНИЯ В ОБУЧЕНИИ НЕЙРОННЫХ СЕТЕЙ ВОПРОСНО-ОТВЕТНЫХ СИСТЕМ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

Рассмотрены алгоритмы нормализации бинарных иерархических чисел с целью их перевода в десятичную форму, позволяющую дальнейшее применение таких чисел в нейронных сетях. Речь идет о языковых BERT-моделях глубокого обучения и технологии их использования в проектировании естественно-языковых вопросно-ответных систем.

Ключевые слова: бинарные иерархические числа, BERT-модели, анализ текстов, вопросно-ответные системы, нейронные сети.

Теория иерархических чисел [1] является разделом общей теории чисел и может использоваться при решении следующих задачах:

- проектирование баз данных;
- администрирование Интернет-сетей;
- библиотечная классификация;
- модели представления знаний искусственного интеллекта.

Десятичные иерархические числа – это числа вида

$$[\ z\ ]\ a_0\ .\ a_1\ .\ a_2\ .\ \ \ldots\ .\ a_i\ .\ \ldots\ \ .a_n\ ,$$

где a_i – целые числа из множества целых чисел

$$N = \{\ldots, -2, -1, 0, 1, 2, 3\ldots\}.$$

z – символ знака «+» или «-», причем положительный знак можно опустить.

Примерами десятичных иерархических чисел являются:

0.0.2.-47.0 или -20.75.0.1.8. Если не возникает коллизий, множество иерархических чисел можно обозначать символом H. Чаще всего в технологиях, связанных с компьютерным программированием, используются исключительно положительные иерархические числа.

Особый случай представляют собой бинарные иерархические числа, где вместо множества N используется множество B = { 0, 1 } или расширенное множество B^+ = { -1, 0, 1 }. Примерами таких чисел являются: 1.0.0.1.0 или 0.0.1.-1.0.1 .

Последний из рассмотренных случаев интересен тем, что он может с эффективностью использоваться в моделях представления знаний или нейронных сетях искусственного интеллекта [2]. Этими числами индексируются понятия (концепты) баз знаний в родовидовых, причинно-следственных и ассоциативных таксономиях [3].

Пример простой дихотомической таксономии с базовым родовидовым отношением дан на рисунке 1.

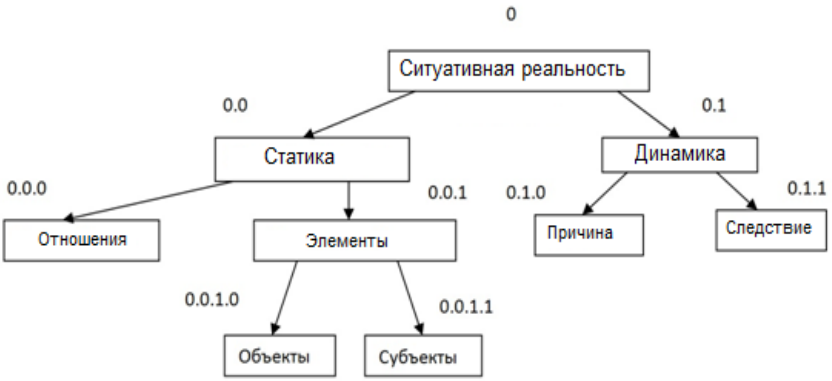


Рисунок 1 – Простая родовидовая дихотомическая таксономия

При использовании технологии языковых нейросетевых моделей со знаниями [4] для анализа естественно-языковых конструкций каждая вершина таксономии на развитых уровнях иерархии может соответствовать какому-либо токену. Когда речь идет об онтологиях общего уровня, описывающих наиболее абстрактные понятия естественного языка, схема бинарной классификации полностью аналогична родовидовой дихотомической таксономии рисунка 1. Если рассматриваются более конкретные концепты модели знаний, используется прикладная таксономия с множественным наследованием, схема которой использует обратное направление ребер таксономии. Таким примером является рисунок 2.

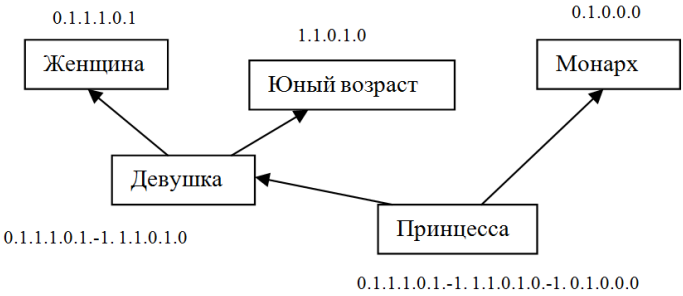


Рисунок 2 – Фрагмент прикладной онтологии с множественным наследованием

Здесь бинарное иерархическое число, соответствующее композиции двух концептов онтологии «Девушка» и «Монарх», представляет собой число, составленное из чисел двух базовых понятий композиции и разделенных символом -1. Такой синтез нового иерархического числа позволяет просто реализовать идею токенизации предложений естественного языка, в которой числовое представление токена «Принцесса» выглядит так:

Принцесса = Женщина + Юный возраст + Монарх.

Однако для использования в языковой нейронной сети иерархическое число должно быть представлено десятичным числом из отрезка [0,1] или [-1,1]. Перевод в такую форму числа называют нормализацией.

Одним из возможных методов нормализации является представление иерархического числа в тернарной системе счисления с добавлением первой значащей единицы затем перевод в десятичную систему счисления:

$$0.1.1.1.0.1.\text{-}1.1.1.0.1.0 \rightarrow 1011101211010 \rightarrow 618060.$$

Для получения числа из положительного отрезка [0,1] полученное дробное число полностью переводится в десятичную дробь, добавлением в целую часть нуля:

$$618060 \rightarrow 0.618060 .$$

Положительным свойством приведенного алгоритма кодирования является возможность преобразовать результирующее число обратно в иерархическое.

В то же время проблема нормализации иерархических чисел является открытой для получения более эффективных алгоритмов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Каширин И.Ю. Бинарные иерархические числа для вычисления семантической близости предложений естественного языка. Рязань. Вестник РГРТУ. 2023. № 85 С.110-121.

2. Каширин И.Ю. Применение теории иерархических чисел в проектировании ICF-таксономии для оптимизации нейронных сетей. Вестник РГРТУ. 2022. № 80 с.118-126.

3. Каширин И.Ю. Bert-модели с концентрацией внимания для анализа текстовой информации web-ресурсов на наличие токсичности. Межвуз.сб. Математическое и программное обеспечение вычислительных систем, РГРТУ, 2024, с.41-43.

4. Каширин И.Ю. Идентификация достоверности новостей с помощью моделей машинного обучения. Вестник РГРТУ. 2023. № 83. с.36-47.

УДК 007:681.512.2

И.Ю. Каширин

МНОГОМЕРНЫЙ АНАЛИЗ ЭЛЕКТРОННЫХ МАТЕРИАЛОВ СРЕДСТВ МАССОВОЙ ИНФОРМАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

Рассмотрен новый проект анализа электронных материалов средств массовой информации, основанный на разносторонней идеологической оценке. Анализ включает как учет прозападных взглядов на существующий миропорядок, так и духовно-нравственные основы интернационализма. Теоретическая новизна проекта заключается в применении моделей знаний и теории иерархических чисел, позволяющих производить семантическую векторизацию естественно-языковых текстов с последующим применением языковых BERT-моделей глубокого обучения и технологии их использования в проектировании вопросно-ответных систем.

Ключевые слова: иерархические числа, средства массовой информации, BERT-модели, анализ текстов, идеологическая окраска текста.

Языковые нейронные сети вызвали к жизни широкий спектр web-сервисов, предназначенных для анализа текстовых материалов, например, новостных статей средств массовой информации (СМИ). С разной степенью достоверности такие сервисы предлагают проанализировать статьи на фейк-новости и токсичность.

Задачи и методы анализа СМИ до настоящего времени базировались на следующих фактах.

1. Существующие аналитические средства выросли из поисковых задач для разработки высоких технологий или составления тематических подборок материалов в гуманитарной области.

2. Большее число нейронных сетей дает возможность анализировать материалы СМИ для синтеза аннотаций, составления обзоров или новых производных публикаций.

3. Языковые нейронные сети для идентификации фейк-новостей, выявления материалов гневного характера, определения токсичности публикации и эмоционального настроения текста чаще всего разрабатываются как локальные, для решения частных задачах оценки рейтинга на выборах и ранга популярности СМИ.

4. Современные языковые модели для исследования публикаций СМИ построены на основе концепции концентрации внимания с применением Bert-сетей. Эти нейронные сети уже обучены на большом количестве статьей западных СМИ с соответствующей идеологией полного подчинения американским инвесторам. В частности, это предполагает оценку любых российских публикаций как заведомо ложных ложных.

Перечисленные обстоятельства направляют живой интерес молодежи к новым сервисам оценки СМИ в строну безальтернативного выбора однополярного мира с единственным американским гегемоном.

Действиетльно, автоматизированные средства разоблачения фейков [1-3] становится весьма популярными в молодежной среде и вскоре станут настолько же востребованными наиболее продвинутой аудиторией, как и специализированное программное обеспечение информационной безопасности, включая антивирусные программы и программы антиплагиата. Современные технологии искусственного интеллекта в области исследования больших данных (Big Data) [4] предоставляют возможность автоматизированного или даже автоматического анализа электронных новостных лент, социальных сетей и других информационных web-сервисов на предмет достоверности содержащегося в этих сервисах контента.

Аналогично антивирусным программам средства автоматизированного разоблачения фальсифицирующей пропаганды должны содержать базы данных, а чаще, базы знаний, оперирующие знаниями о конкретных авторах, информационных источниках и

электронных изданиях. Этим самым может быть решена еще одна важнейшая проблема – проблема вычисления фейк ранга (Fake Rank) авторов и средств массовой информации. Эта проблема тем более актуальна, что аналогичные разработки так же могут создаваться недобросовестными международными организациями для фальсификации самих фейк рангов.

Предварительное обучение языковой модели, такое как Bert, значительно улучшило производительность многих задач обработки естественного языка. Однако предварительно обученные языковые модели обычно требуют больших вычислительных затрат, поэтому их сложно эффективно выполнять на устройствах с ограниченными ресурсами. Чтобы ускорить вывод и уменьшить размер модели при сохранении точности, в [5] сначала предлается новый метод дистилляции Transformer, специально разработанный для дистилляции знаний (KD) моделей на основе Transformer. Используя этот новый метод KD, множество знаний, закодированных в большом BERT учителя, можно эффективно передать малой модели Tiny-BERT. Затем выбирается новая структура двухэтапного обучения для TinyBERT, которая выполняет дистилляцию Transformer как на этапе предварительного обучения, так и на этапе обучения для конкретных задач. Эта структура гарантирует, что TinyBERT может собирать как общие, так и конкретные знания Bert. TinyBERT с 4 слоями эмпирически эффективен и достигает более чем 96,8% производительности учителя BERTBASE в тесте GLUE, при этом он в 7,5 раз меньше и в 9,4 раза быстрее при выводе. TinyBERT с 4 слоями также значительно лучше, чем современные 4-слойные базовые линии при дистилляции Bert, имея только около 28% параметров и около 31% времени вывода из них. Более того, TinyBERT с 6 слоями работает на одном уровне со своим учителем BERTBASE.

На кафедре вычислительной прикладной математики Рязанского государственного радиотехнического университета им. В.Ф.Уткина в настоящее время реализуется новый проект «Получение новых методов достоверной идентификации информационного влияния электронных средств массовой информации применением языковых нейронных сетей со знаниями на основе теории иерархических чисел». В проекте планируется построение нового метода проектирования ML-моделей с помощью привлечения теории представления знаний [6] в прикладные разработки программ интеллектуального анализа данных (Data Mining) с целью получения эффективного инструментария для выявления недостоверных новостных материалов.

Актуальность нового проекта, в первую очередь, основана на разносторонней оценке новостных материалов как с точки зрения однополярных предпочтений, так и с точки зрения интернациональных духовно-нравственных основ. Кроме того, актуальность проекта

обусловлена ***многомерностью*** анализа материалов СМИ, в том числе: анализ настроений, разжигание ненависти, подготовка цветных революций, токсичное давление на аудиторию.

Научная значимость проекта основывается на применении нового теоретического аппарата иерархических чисел [7,8] для создания новой технологии проектирования языковых нейронных моделей, превосходящих по возможностям существующие модели.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Каширин И.Ю. Идентификация достоверности новостей с помощью моделей машинного обучения. Вестник рязанского государственного радиотехнического университета. 2023. № 83. с.36-47.

2. I.Augenstein,Ch.Lioma, Multifc: A real-world multi-domain dataset for evidencebased fact checking of claims. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP).

3. Демидова Л.А., Морошкин Н.А. Аспекты разработки архитектуры вопросно-ответной системы для обработки больших данных на основе нейросетевого моделирования. // Вестник рязанского государственного радиотехнического университета. Рязань 2023. № 86. С.145-155.

4. Shaden Shaar, Nikolay Babulkov, Giovanni Da San Martino, and Preslav Nakov. That is a known lie: Detecting previously fact-checked claims. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020.

5. Xiaoqi Jiao, Yichun Yin, Lifeng Shang, TinyBERT: Distilling BERT for Natural Language Understanding.// ACL Anthology. 2020.

6. Clark E. et al., All that's 'human' is not gold: Evaluating human evaluation of generated text. InProceedings of the 59th Annual Meeting of the Association for Computational Linguisticsandthe11thInternational Joint Conference on Natural Language Processing (Volume1: LongPapers), 2021.

7. Definition of Hierarchial Numbers [Electronic resource]. Update date: 02.04.2024 URL: https://kashirin.net/definition-of-hierarchical-numbers. (date of application: .09.04.2024).

8. The Idea of ICF Relationships and ICF Ontologies [Electronic resource]. Update date: 18.02.2024 URL: https://kashirin.net/the-idea-of-icf-ontologies. (date of application: 05.04.2024).

УДК 004.891.3

О.А. Куликов

ИССЛЕДОВАНИЕ ВЛИЯНИЯ МЕТОДОВ ОПТИМИЗАЦИИ И ФУНКЦИИ ПОТЕРЬ ПРИ ОБУЧЕНИИ СВЕРТОЧНОЙ НЕЙРОННОЙ СЕТИ ДЛЯ ЗАДАЧИ КЛАССИФИКАЦИИ РАКОВЫХ ОПУХОЛЕЙ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

Статья посвящена исследованию влияния метода оптимизации и функции потери на обучение сверточной нейронной сети, классифицирующей изображения ультразвукового исследования молочных желез пациента. В ходе исследования использовался набор данных, включающий 750 изображений, разделенных на три класса. Результаты показали, что наилучшую эффективность продемонстрировала пара оптимизатора Lion с функцией потерь Categorical Focal Crossentropy, достигнув точности 76%.

Ключевые слова: раковые заболевания, машинное обучение, сверточная нейронная сеть, оптимизатор, функция потерь.

Раковые заболевания являются одной из основных причин смертности по всему миру. Существует более 100 различных разновидностей рака. Диагностика рака на ранних стадиях существенно повышает шансы на успешное лечение.

В последние годы искусственные нейронные сети стали активно применяться для улучшения и автоматизации процесса диагностики раковых заболеваний у человека. Искусственные нейронные сети представляют собой математическую модель, имитирующую работу нервной системы, и могут обрабатывать большие объемы данных с высокой точностью и скоростью [1, 2].

Основная цель распознавания вида опухоли молочных желез на изображении ультразвукового исследования пациента заключается сокращении времени на постановку диагноза врачом первичного звена. Также, сюда косвенно относится и снижение количества ошибочных диагнозов. Они могут возникать вследствие невнимательности, усталости или низкого уровня квалификации врача первичного звена лечебно-профилактического учреждения.

Задача классификации в машинном обучении заключается в присвоении объектам определенной категории или класса на основе их признаков. В процессе обучения модели классификации используются различные алгоритмы и методы, чтобы модель могла правильно классифицировать новые, ранее неизвестные данные.

В настоящей статье проанализировано влияние метода оптимизации и функции потери при обучении сверточной нейронной

сети. Задачей рассматриваемой сети является классификация изображений ультразвуковых исследований молочных желез пациентов.

В качестве обучающего набора данных для исследуемой сверточной нейронной сети был использован архив изображений ультразвуковых исследований. Набор данных включает в себя 750 изображений, распределенных три класса: изображения, не содержащие опухоли (normal); изображения с доброкачественными опухолями (benign) и изображения со злокачественными опухолями (malignant). Набор был разделен равномерно на обучающую (540 изображений), валидационную (135 изображений) и тестовую (75 изображений) выборки. Примеры изображений каждого класса приведены на рисунке 1.

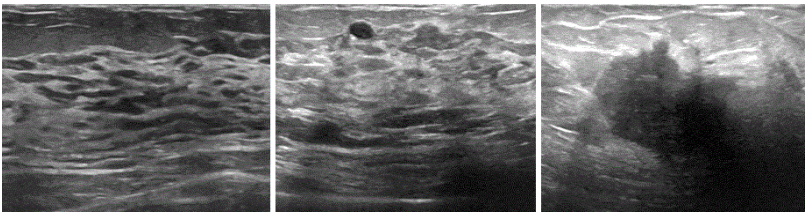


Рисунок 1 – Примеры изображений ультразвуковых исследований: слева – без опухоли, в центре – доброкачественная опухоль, справа – злокачественная опухоль

Для решения задачи классификации использовалась последовательная модель сверточной нейронной сети. Архитектура сети состоит из 3 блоков: блок свертки, связующий блок, классификатор. Структура сети показана на рисунке 2.

Блок свертки состоит из 3 последовательно расположенных пар слоев классов Conv2D с функцией активации Redu и MaxPooling2D. Задачей блока является выделение признаков на изображении.

Связующий блок представлен слоем класса Flatten. Данный блок обеспечивает плавную интеграцию блока свертки с классификатором.

Классификатор состоит из 3 полносвязных слоев класса Dense с функцией активации Relu и выходного слоя класса Dense с функцией активации Softmax. Для предотвращения сложных коадаптаций отдельных нейронов между полносвязными слоями применены прореживающие слои класса Dropout [3].

Метод оптимизации в машинном обучении (оптимизатор) – это специальный алгоритм, который используется для настройки параметров модели таким образом, чтобы минимизировать ошибку предсказания. Оптимизаторы играют ключевую роль в процессе обучения моделей машинного обучения, так как они определяют, каким образом будут обновляться веса и смещения модели в процессе обучения.

Существует множество различных оптимизаторов, каждый из которых имеет свои собственные преимущества и недостатки в

зависимости от задачи обучения. Для исследуемой нейронной сети использовались оптимизаторы: Adam, SGD, RMSprop, Lion [4].

Функции потерь в машинном обучении используются для оценки того, насколько модель хорошо работает на задаче обучения путем сравнения предсказанных значений с истинными метками данных. Цель функции потерь – минимизировать ошибку модели и улучшить ее предсказательную способность.

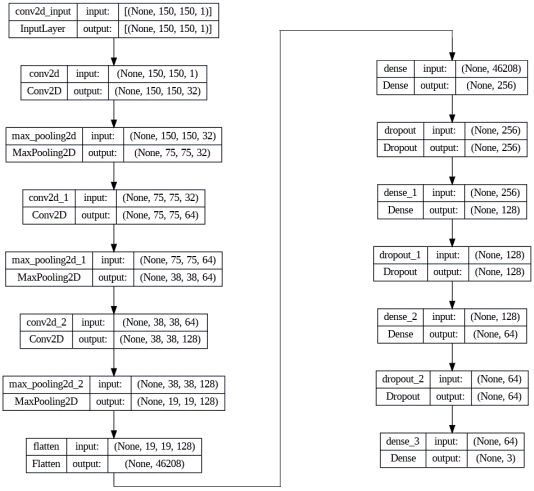


Рисунок 2 – Структура модели сверточной нейронной сети

Существует множество функций потерь, которые выбираются в зависимости от типа задачи и данных. Для задач классификации характерно использование перекрестной энтропия, функции потерь Пуассона. Для исследуемой нейронной сети использовались следующие функции потерь: Categorical Crossentropy, Categorical Focal Crossentropy, Poisson function, Sigmoid Focal Crossentropy [5, 6].

Для сравнения результатов обучения нейронной сети в качестве основного критерия использовалась точность модели на тестовой выборке. В качестве дополнительного критерия анализировалась точность модели для каждого класса. Также для каждого случая были построены матрицы ошибок многоклассовой классификации (Confusion matrix).

Модель обучалась с указанными выше оптимизаторами и функциями потерь на протяжении 30 эпох. Дополнительно был использован параметр количества обучающих примеров, используемых в одной итерации равный 20. Также осуществлялось сохранение значений весов для наилучшего показателя точности на валидационной выборке. В

таблице 1 представлены результаты обучения нейронной сети с использованием разных оптимизаторов и функций потерь.

Самый лучший результат по точности нейронная сеть показала при использовании оптимизатора Lion с функцией потерь Categorical Focal Crossentropy. При этом стоит отметить высокие и равномерные показатели точности среди выявленных классов. На рисунке 3 представлены графики точности и потерь на обучающей и валидационной выборках при обучении сети с указанными параметрами.

Наихудший результат по точности был выявлен при использовании оптимизатора SGD с функцией потерь Poisson function.

На рисунке 4 представлена матрица ошибок многоклассовой классификации (Confusion matrix) при использовании оптимизатора Lion с функцией потерь Categorical Focal Crossentropy.

Таблица 1 – Результаты обучения нейронной сети

Оптимизатор	**Функция потерь**	**Точность модели, (%)**	**Точность модели по каждому классу (normal / benign / malignant), (%)**
Adam	Categorical Crossentropy	64,0	50,0 / 66,0 / 62,0
Adam	Categorical Focal Crossentropy	70,7	60,0 / 70,0 / 72,0
Adam	Poisson function	74,7	70,0 / 77,0 / 73,0
Adam	Sigmoid Focal Crossentropy	68,0	46,0 / 67,0 / 80,0
SGD	Categorical Crossentropy	62,7	25,0 / 61,0 / 75,0
SGD	Categorical Focal Crossentropy	62,7	13,0 / 59,0 / 0,0
SGD	Poisson function	61,3	13,0 / 61,0 / 100,0
SGD	Sigmoid Focal Crossentropy	62,7	25,0 / 70,0 / 62,0
RMSprop	Categorical Crossentropy	70,7	57,0 / 68,0 / 77,0
RMSprop	Categorical Focal Crossentropy	64,0	67,0 / 61,0 / 67,0
RMSprop	Poisson function	72,0	75,0 / 70,0 / 76,0
RMSprop	Sigmoid Focal Crossentropy	66,7	40,0 / 69,0 / 86,0
Lion	Categorical Crossentropy	65,3	71,0 / 66,0 / 61,0
Lion	Categorical Focal Crossentropy	76,0	86,0 / 73,0 / 82,0
Lion	Poisson function	70,7	83,0 / 70,0 / 68,0
Lion	Sigmoid Focal Crossentropy	68,0	42,0 / 69,0 / 80,0

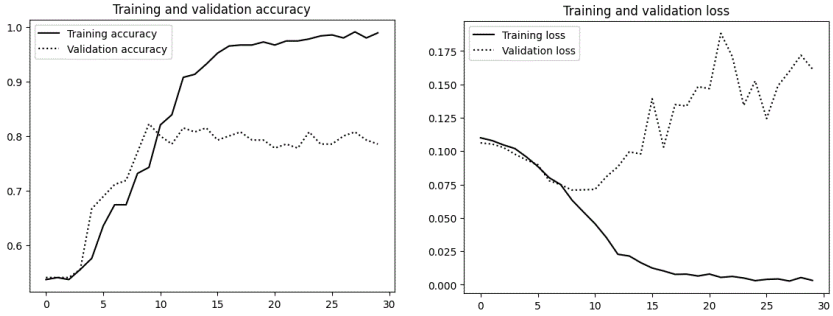


Рисунок 3 – Графики точности (слева) и потерь (справа) на обучающей (непрерывная линия) и валидационной (пунктирная линия) выборках

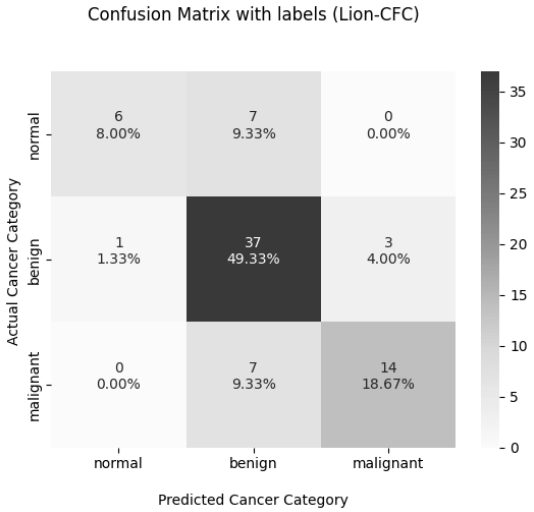


Рисунок 4 – Матрица ошибок для пары оптимизатора Lion с функцией потерь Categorical Focal Crossentropy

В результате проведенных исследований были получены следующие результаты. Среди оптимизаторов проявили себя Lion (средняя точность – 70%) и Adam (средняя точность – 69,4%). Среди функций потерь Poisson function (средняя точность – 69,7%) и Categorical Focal Crossentropy (средняя точность – 68,4%).

При обучении нейронной сети было осуществлено сравнение эффективности различных оптимизаторов и функций потерь по критерию точности на тестовой выборке.

Результаты обучения нейронной сети для классификации изображений ультразвуковых исследований молочной железы пациентов

выявили наиболее эффективную пару оптимизатора и функции потерь. Пара оптимизатора Lion и функции потерь Categorical Focal Crossentropy продемонстрировала точность 76%, т.е. нейронная сеть правильно классифицировала 57 изображений из 75. Такой результат можно считать неплохим, по причине ограниченного количества и неравномерного распределения по классам исследуемых изображений.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Каширин И.Ю., Чистяков П.А. Интеллектуальная диагностика с помощью алгебры иерархических чисел // ИИАСУ’23 – Искусственный интеллект в автоматизированных системах управления и обработки данных : Сборник статей II Всероссийской научной конференции Москва, 27–28 апреля 2023 г.) : в 5 т. – М.: «КДУ», «Добросвет», 2023. – Т. 2. – 406 с. – Электронное издание сетевого распространения. – URL: https://bookonlime.ru/node/72807 – doi: 10.31453/kdu.ru.978-5-7913-1352-2-2023-406 P-71-75.

2. Куликов О.А. Применение искусственных нейронных сетей в диагностике раковых заболеваний // Материалы XXVIII всероссийская научно-техническая конференция студентов, молодых ученых и специалистов: Рязанский государственный радиотехнический университет имени В.Ф. Уткина. 2023 — Рязань. С. 189-191.

3. Keras layers API [Электронный ресурс]. — Режим доступа: https://keras.io/api/layers/ — (дата обращения: 20.04.2024).

4. Keras Optimizers [Электронный ресурс]. — Режим доступа: https://keras.io/api/optimizers/ — (дата обращения: 20.04.2024).

5. Keras Losses [Электронный ресурс]. — Режим доступа: https://keras.io/api/losses/ — (дата обращения: 20.04.2024).

6. TensorFlow Addons [Электронный ресурс]. — Режим доступа: https://www.tensorflow.org/addons/api_docs/python/tfa/losses/SigmoidFocalCrossEntropy — (дата обращения: 20.04.2024).

УДК 004.056.5(075.8)

С. В. Крошилина, О. В. Курочкина

МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ В ПРОЦЕССАХ СКЛАДСКОГО И ЛОГИСТИЧЕСКОГО УЧЕТА

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

В статье рассматривается важность эффективного менеджмента складских и логистических процессов в современном бизнесе и преимущества использования, специализированного математического и программного обеспечения для вычислительных систем в области обслуживания бассейнов.

Ключевые слова: автоматизация, логистика, программное обеспечение, математическое обеспечение, цепь поставок.

Сегодняшний бизнес требует эффективного менеджмента складских и логистических процессов для обеспечения бесперебойного обслуживания клиентов. В этой связи, использование специализированного математического и программного обеспечения для вычислительных систем [1] становится ключевым фактором успеха в отрасли. Приведем более подробное описание, какие именно возможности и выгоды принесет автоматизация в обслуживании бассейнов.

Математическое обеспечение: оптимизация и прогнозирование. Одной из главных задач математического обеспечения является оптимизация процессов складского хранения и логистики. Современные алгоритмы оптимизации помогают минимизировать издержки на складское хранение, оптимизировать использование пространства и рационально распределять товары в рамках бассейнов. Кроме того, математические модели прогнозирования спроса позволяют более эффективно управлять запасами и предсказывать изменения в спросе на товары.

Программное обеспечение: оперативность и контроль. Программные решения для автоматизации складского и логистического учета обеспечивают оперативное выполнение заказов, контроль над движением товаров, планирование маршрутов доставки и многое другое. Интеграция таких систем с технологией сбора данных позволяет в реальном времени отслеживать положение товаров на складе, обеспечивая точную и актуальную информацию о наличии и движении товаров [2].

Преимущества использования автоматизации в обслуживании бассейнов. Эффективное использование математического и программного обеспечения в системах складского учета и логистики в обслуживании бассейнов приносит ощутимые преимущества. Среди них можно выделить повышение производительности труда, снижение издержек, улучшение качества обслуживания клиентов, сокращение времени обработки заказов и увеличение прозрачности процессов управления [3].

Инновации в области роботизации складских операций. Помимо математического и программного обеспечения, современные предприятия также активно внедряют инновационные технологии в области роботизации складских операций. Автоматизированные складские роботы и автономные транспортные средства становятся неотъемлемой частью логистических систем, обеспечивая быструю и эффективную обработку и перемещение товаров на складе. Исследования показывают, что роботизация складских процессов позволяет сократить временные затраты, уменьшить вероятность ошибок и повысить общую производительность складского хозяйства.

Экологическая устойчивость в логистике. Одним из важных аспектов современной логистики является экологическая устойчивость логистических операций. Внедрение зеленых технологий, таких как электрические грузовики, солнечные панели на складских помещениях, системы энергоэффективного освещения и управления отходами, помогает снизить негативное воздействие логистики на окружающую среду. При этом компании также получают дополнительные преимущества в виде снижения затрат на энергию, улучшения имиджа и привлечения экологически осознанных потребителей [4].

Анализ ключевых тенденций в цепи поставок и логистике. Проведение анализа ключевых тенденций в цепи поставок и логистике позволяет компаниям оперативно реагировать на изменения рыночной среды и принимать обоснованные стратегические решения. Важными аспектами анализа являются обзор рыночных тенденций, оценка конкурентной среды, мониторинг изменений в потребительском спросе, оценка эффективности логистических операций и поиск новых возможностей для оптимизации цепи поставок.

Следовательно, использование специализированного математического и программного обеспечения в процессах складского и логистического учета в обслуживании бассейнов является необходимым шагом для современных предприятий. Благодаря этому, компании могут улучшить эффективность своей деятельности, повысить уровень обслуживания и оставаться конкурентоспособными в условиях динамично меняющегося бизнес-окружения.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Крошилина С.В. Предметно-ориентированные информационные системы: учебное пособие / А.В. Крошилин, С.В. Крошилина, Г.В. Овечкин. — Москва: КУРС, 2023. — 176 с. — (Естественные науки).

2. Курочкина О.В., Крошилин А.В. Особенности разработки программного обеспечения складского и логистического учета сервисного обслуживания бассейнов // Математическое и программное обеспечение вычислительных систем: Межвуз. сб. науч. тр. / Под ред. Г.В. Овечкина - Рязань: РГРТУ им. В.Ф. Уткина, январь 2024 - 202 с. (44-48)

3. Курочкина О.В., Крошилин А.В. Автоматизация процессов складского и логистического учета в обслуживании бассейнов // Новые информационные технологии в научных исследованиях: материалы XXVIII Всероссийской научно-технической конференции студентов, молодых ученых и специалистов. РГРТУ им. В.Ф. Уткина. т.1, Рязань: 2023. 197с.(29-31)

4. Зарипова Р.С. Автоматизация складских процессов на предприятиях / Р.С. Зарипова, О.А. Рочева, Ф.Р. Хамидуллина // Наука Красноярья. – 2021. – Т. 10, № 3-3. – С. 65-70.

УДК 004.056.53

Д. Н. Кустов, С. В. Мицук

РЕАЛИЗАЦИЯ АТАКИ НА VPS СЕРВЕР ЧЕРЕЗ TELEGRAM-СЕССИЮ

Федеральное государственное бюджетное образовательное учреждение высшего образования “Липецкий государственный педагогический университет имени П.П. Семенова-Тян-Шанского”, Липецк

В статье рассматривается проблема атак на VPS сервера. На практике рассмотрен процесс атаки через Telegram-сессию, произведён анализ каждого этапа работы, исследованы побочные вектора угроз, их модели, приведены графические иллюстрации содержимого и исследованы возможные последствия. Ключевые слова: VPS, информационная безопасность, сервер, база данных

Проблема информационной безопасности сетевых ресурсов становится всё более актуальной в процессе развитий IT технологий. Как правило, функционирование таких систем осуществляется с использованием серверов, которые при определённых условиях могут стать уязвимыми для внешних и внутренних атак. Для описания угроз может применяться модель STRIDE, включающая в себя:

- Подделку личности.
- Изменение данных.
- Отказ от действий.
- Разглашение информации.
- Отказ в обслуживании.
- Повышение привилегий.

Смоделированная атака будет включать в себя следующие этапы:

1. Сбор предварительных данных
2. Рассылка фишинг писем
3. Компрометация данных админа
4. Повышение уровня доступа
5. Поиск наиболее уязвимых к аутентификации сервисов
6. Запуск стороннего скрипта через уязвимость.
7. Чтение базы данных через сторонний скрипт

Первым этапом осуществляется сбор предварительных данных. Интерес представляют номера телефонов сотрудников, их почты, другие контактные данные. Как правило, эта информация может быть доступна на сайте организации, социальных сетях.

Вторым этапом является отправка фишинг-письма. При должном уровне маскировке жертва имеет шанс перейти по письму, раскрыв какие-либо свои данные[1].

На третьем этапе происходит компрометация данных. Ценность представляют активные сессии, данные аутентификации, другая информация, которая может пригодиться для дальнейшей атаки. В рассматриваемом примере это будет активная сессия Desktop версии официального приложения Telegram для компьютера. [3] Такая версия использует файл сессии для доступа к аккаунту. Он располагается в каталоге «tdata» (рисунок 1). Заполучив такой файл, злоумышленник получает полный доступ к аккаунту жертвы.

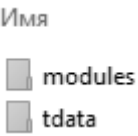


Рисунок 1 – Искомый каталог

Использование облачного пароля, двухфакторной аутентификации и других методов защиты не имеют определяющего значения. Они требуются для старта новой сессии, при этом полученный файл уже является сессией.

Программно реализуема автоматизация поиска нужной информации через Telethon библиотеку для Python. Она представляет собой client для автоматизации действий, осуществляемых от имени аккаунта. Авторизация представляет собой отправку номера телефона, получение кода авторизации, ввод облачного пароля. Номер телефона уже известен, так как есть доступ к аккаунту, код авторизации присылается внутри приложения и, как итог, тоже становится известен. Однако такая авторизация видна жертве, даже несмотря на возможность замены данных авторизации используемого client (рисунок 2).

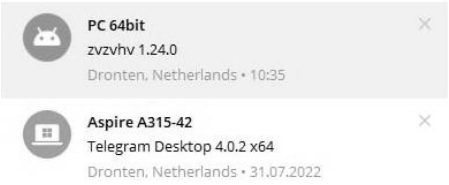


Рисунок 2 – Список сессий

В данном примере изображены активные сессии устройства. Нижняя – обычная, верхняя – создаваемая скриптом. Предварительно известно, что используемое устройство обеих сессий не отличается. Сравним предоставляемые данные, используя таблицу 1.

Таблица 1 – Параметры сессий

Параметр	**Стандартная сессия**	**Сгенерированная сессия**
ОС	Windows	Android
Имя устройства	Aspire A315-42	PC 64bit
Используемое приложение	Telegram Desktop 4.0.2 x64	Zvzvhv 1.24.0
Страна авторизации	Dronten, Netherlands	Dronten, Netherlands

Как можно заметить, сгенерированная сессия позволяет программно заменить все свои параметры, кроме страны авторизации. Но это без проблем решается использованием VPN. Использование дополнительной сессии имеет свои плюсы и минусы. К плюсам можно отнести возможность ускорения перебора данных, их автоматическая выгрузка и анализ. Из минусов – появление дополнительной сессии видно жертве, а также невозможность запуска при наличии облачного пароля, который неизвестен.

На четвёртом этапе происходит повышение уровня доступа. В внутри скомпрометированного аккаунта могут содержаться пароли авторизации, данные аутентификации для других сервисов. Например, это может быть логин/пароль к какому-либо серверу. Таким образом, злоумышленник который не имеет прав изначально, приобретает права путём получения доступа от чужого имени. Этот процесс также рассматривается в модели угроз Stride как «отказ от действий», когда злоумышленник может совершать действия от чужого лица и не признавать это. Более того, он может маскировать такую активность.

На пятом этапе потенциальный злоумышленник производит поиск наиболее уязвимых к аутентификации сервисов [2]. В рассматриваемом варианте такими сервисами выступают Putty, WinSCP. Применяя найденный в Telegram аккаунте пароль, получаем доступ к процессам сервера (рисунок 3) и его файлам.

```
1151640 postgres   20   0  213M 15360 14600 S  0.0  0.8  1:02.34 /usr/lib/postgresql/14/bin/p
1151678 postgres   20   0  213M 15732 14944 S  0.0  0.8  0:02.46 postgres: 14/main: checkpoin
1151679 postgres   20   0  213M  8984  8300 S  0.0  0.4  0:04.15 postgres: 14/main: backgroun
1151680 postgres   20   0  213M  8256  7592 S  0.0  0.4  0:17.01 postgres: 14/main: walwriter
1151681 postgres   20   0  213M  6108  5248 S  0.0  0.3  0:05.60 postgres: 14/main: autovacuu
1151682 postgres   20   0 73060  3448  2652 S  0.0  0.2  0:09.84 postgres: 14/main: stats col
1151683 postgres   20   0  213M  4552  3800 S  0.0  0.2  0:00.19 postgres: 14/main: logical r
1151687 postgres   20   0  214M 16904 15812 S  0.0  0.8  0:00.31 postgres: 14/main: postgres
1151688 postgres   20   0  214M 18480 17344 S  0.0  0.9  0:00.46 postgres: 14/main: postgres
```

Рисунок 3 – Процессы PostgreSQL

Просмотр процессов осуществляется через программу Putty с использованием команды htop. Среди процессов видим Postgresql, это означает, что сервер обслуживает базу данных. Она и станет одним из следующих объектов атаки.

На шестом этапе происходит запуск стороннего скрипта. Среди полученных процессов нас интересует serv.py (рисунок 4).

```
2152 root      20   0  333M 34228  7448 S  0.0  1.7  1h34:47 python3 ./castle/serv.py
2153 root      20   0  333M 34228  7448 S  0.0  1.7 23:38.61 python3 ./castle/serv.py
2154 root      20   0  333M 34228  7448 S  0.0  1.7 23:22.80 python3 ./castle/serv.py
```

Рисунок 4 – Найденный процесс serv.py

Судя по названию, данный скрипт принимает участие в управлении сервером. Исходя из информации, становится известно и его местоположение. Изучив файл, понимаем, что он позволяет запускать произвольные скрипты с правами администратора, используя скомпрометированный Telegram аккаунт. Отправляем на сервер исполняемый файл с параметрами для подключения к базе данных из рисунка 3, а также программно прописываем sig kill 9 всем процессам которые могут отвечать за защиту базы данных. Эти процессы будут принудительно завершены.

На седьмом этапе происходит получение данных с базы данных и их сохранение сторонним скриптом, рассмотренным ранее. В конечном результате получается получить 35 таблиц (рисунок 5).

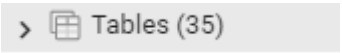


Рисунок 5 – Полученные Tables с данными

Благодаря запуску нашего скрипта с правами админа, процесс просмотра/замены/добавления записей не составляет труда. Этот процесс выделен в модели угроз Stride как «Изменение данных». В случае разглашения данных это также будет и «разглашением данных» в соответствии с моделью угроз. На текущем этапе были применены все её звенья.

В случае если поиск ключей авторизации к сервисам не принёс результатов в виду защиты сервера, злоумышленнику доступен поиск в данных аккаунта таких данных как закрытые ключи к криптовалютным кошелькам, их seed-фразы, или файлы доступа (как правило имеющие название wallet.dat). Либо такого рода информация может содержаться на заражённом сервере, рассмотренном ранее (рисунок 6).

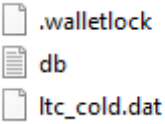


Рисунок 6 – Файл ltc_cold.dat

В рассматриваемом случае найден файл wallet.dat, переименованный в ltc_cold.dat. Его не составило труда найти, используя поиск файла по расширению. При этом название файла сразу же раскрывает используемый блокчейн – Litecoin и способ его хранения – cold, сокращённо от cold wallet. Среди дальнейших действий предстоит найти такое хранилище. Выбор невелик, среди сегмента такие

характеристики имеет только Litecoin Core. Подгружаем ltc_cold.dat в эту программу, получаем полноценный доступ к холодному хранилищу LTC. Среди прав имеем в том числе просмотр балансов, отправка транзакций (рисунок 7). Некоторые данные были закрашены для соблюдения конфиденциальности.

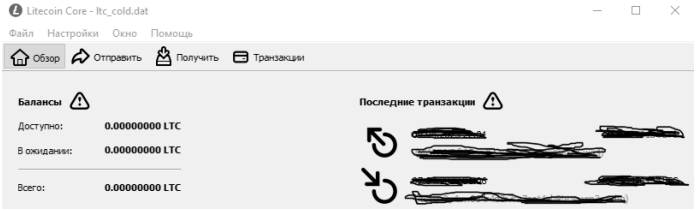


Рисунок 7 – Полученный доступ к холодному хранилищу.

В работе продемонстрирована опасность хранения паролей в социальных сетях. В рассмотренной ситуации это привело к атаке на VPS-сервер и получению его данных со всеми вытекающими. Стоит добавить, что аналогичным способом можно получить доступ к MEW, Bitoin Core, Trust Wallet кошелькам, используя seed-фразы/закрытые ключи/файлы авторизации [4]. Среди мер защиты можно выделить добавление подписи второго аккаунта при совершении транзакций. Все демонстрируемые действия были произведены на собственном оборудовании.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Ионов, Д.Н. Анализ атак, совершаемых с помощью социальной инженерии / Д.Н. Ионов, П.И. Карасев // Кибербезопасность: технические и правовые аспекты защиты информации: Сборник научных трудов I Национальной научно-практической конференции, Москва, 24-26 мая 2023 года. – М.: МИРЭА - Российский технологический университет, 2023. – С. 188-191.

2. Козлов, Ф.С. Анализ технологий для защищенного хранения учетных записей / Ф.С. Козлов, П.И. Карасев // Кибербезопасность: технические и правовые аспекты защиты информации: Сборник научных трудов I Национальной научно-практической конференции, Москва, 24-26 мая 2023 года. – М.: МИРЭА – Российский технологический университет, 2023. – С. 222-224.

3. Автаев, М.С. Анализ безопасности мессенджера Telegram / М.С. Автаев // Трибуна ученого. – 2022. – № 11. – С. 6-10.

4. Семьянов, П.В. Анализ криптографической защиты криптокошелька Bitcoin Core / П.В. Семьянов, С.В. Грезина // Проблемы информационной безопасности. Компьютерные системы. – 2023. – № 2(54). – С. 82-91.

УДК 004.627

М. Д. Соколовский, К. А. Майков, А. Н. Пылькин

ФОРМАТ ХРАНЕНИЯ ВЕЩЕСТВЕННЫХ КОЭФФИЦИЕНТОВ СЖИМАЮЩИХ ОТОБРАЖЕНИЙ ПРИ ФРАКТАЛЬНОМ СЖАТИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань
Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный технический университет имени Н.Э. Баумана», Москва

Рассматривается формат записи вещественных коэффициентов отображений, используемых при фрактальном сжатии изображений с потерями. Предложенный формат основан на раздельном хранении компонент экспоненциального представления чисел с плавающей точкой и кодировании Хаффмана.
Ключевые слова: сжатие изображений, фрактальное сжатие, сжатие данных с потерями, кодирование Хаффмана, числа с плавающей точкой.

Необходимость сжатия изображений возникает в связи с возрастающим объёмом обрабатываемой графической информации в современных высокопроизводительных информационных системах. Использование сжатия позволяет снизить требования к вычислительным ресурсам в таких системах. При этом некоторые решаемые данными системами задачи допускают внесение искажений (потерь) в изображения для достижения лучших характеристик сжатия.

Метод фрактального сжатия изображений предназначен для сжатия одноканальных изображений с потерями. Он основан на применении сжимающих отображений. В изображении выделяются множества ранговых и доменных квадратных блоков, после чего формируется множество отображений из множества доменных блоков во множество ранговых. Данное множество отображений является результатом алгоритма сжатия и может быть использовано для восстановления исходного изображения [1].

В рассматриваемом методе сжатия используются сжимающие отображения, образованные путём комбинации сжатия размерности, углового преобразования, умножения пикселей на вещественный коэффициент контраста c и сложения с вещественным коэффициентом яркости b. Такие отображения описываются как показано на формуле (1), где S – матрица сжатия, R – матрица углового преобразования [1].

$$f(D) = cRSD + b \quad (1)$$

При записи массива отображений в файл для каждого отображения записывается: индекс доменного блока (прообраза), ориентация блока (3 бита) и два вещественных коэффициента *c* и *b*.

Число отображений в файле равно числу используемых ранговых блоков. В работе используется структура выделения квадратных ранговых блоков «quadtree» [2], блоки в которой имеют размеры сторон от 4 до 16 пикселей. В ходе исследования число ранговых блоков для изображений размером 512x512 пикселей варьировалось от 2 до 10 тысяч в зависимости от особенностей исходного изображения и параметров алгоритма сжатия.

Использование компактного формата хранения параметров отображения необходимо для достижения высокого коэффициента сжатия. В данной работе рассматриваются только особенности записи коэффициентов контраста и яркости.

В программе сжатия коэффициенты представлены в формате IEEE 754 float [4] и занимают 32 бита. Однако обеспечиваемая данным форматом точность является избыточной для хранения. Был проведён эксперимент, в ходе которого для записи использовался формат IEEE 754 half [4], занимающий 16 бит. В результате эксперимента не было обнаружено заметных изменений в объёме потерь метода сжатия, в связи с чем данный формат также рассматривается, как избыточный.

Одним из известных подходов к снижению избыточности является использование квантования значений коэффициентов с равным шагом с использованием 5-8 бит для записи коэффициентов [3].

В работе предлагается модифицированный метод хранения вещественных коэффициентов, основанный на экспоненциальном представлении стандарта IEEE 754 [4] и кодировании Хаффмана[5]. Предлагаемый метод состоит из двух этапов.

На первом этапе метода выполняется вычитание из всех коэффициентов контраста их среднего арифметического. Исходное среднее арифметическое записывается в файл без изменений и используется при восстановлении сжатого изображения. Данный шаг позволяет уменьшить абсолютное значение коэффициентов, что приводит к уменьшению потерь точности при их записи в экспоненциальном представлении.

На втором этапе выполняется раздельная запись компонент экспоненциального представления вещественного коэффициента: знака, мантиссы и экспоненты. Каждая компонента записывается в отдельный массив, индекс в котором соответствует индексу описываемого отображения.

Знаки коэффициентов записываются по одному биту подряд без дополнительных преобразований. Были проведены исследования по использованию RLE кодирования, однако они не привели к заметному положительному результату.

Мантиссы коэффициентов записываются напрямую с округлением до 2-4 значащих бит. Для определения размеров мантисс был проведён ряд экспериментов, направленных на измерение объёма потерь метода. Для численной оценки объёма потерь метода использовался показатель SSIM [6]. Результаты одного из экспериментов приведены в таблице 1. Рассматривалось одноканальное тестовое изображение Lena размером 512x512 пикселей. При использовании неокруглённой мантиссы показатель SSIM был равен 0.819.

Таблица 1 – Зависимость SSIM от размеров мантисс коэффициентов

Коэфф. яркости		**2**	**3**	**4**
Коэфф. контраста	**2**	0.703	0.723	0.729
	3	0.751	0.776	0.784
	4	0.770	0.797	0.807

Исходя из результатов экспериментов и в целях достижения высокого коэффициента сжатия в дальнейших исследованиях для мантисс обоих коэффициентов используется размер 3 бита.

Для записи экспонент коэффициентов используется каноническое кодирование Хаффмана [5]. В начале записывается словарь, состоящий из двух массивов, необходимых для создания таблицы декодирования. В первом массиве хранится число кодов каждой длинны: индекс в массиве соответствует длине кода, уменьшенной на 1. Во втором массиве хранятся значения экспонент, отсортированные по длине их кода. Далее записывается массив закодированных экспонент коэффициентов.

При использовании предложенного метода кодирования для хранения коэффициентов в среднем используется 5-7 бит. При этом массивы коэффициентов занимают 25-28% от полного размера файла со сжатыми данными каждый. Несмотря на дополнительные вносимые искажения, предложенный метод позволяет увеличить коэффициент сжатия метода в 1.5-2 раза по сравнению с использованием стандартного формата хранения IEEE 754 half.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Торжкова А.О., Серов А.П., Майков К.А., Соколовский М.Д. Алгоритм сжатия изображений с потерями на основе сжимающих отображений // Актуальные проблемы современной науки и производства: Материалы VIII Всероссийской научно-технической конференции, Рязань, 27–30 ноября 2023 года. – Рязань: Рязанский государственный радиотехнический университет им.В.Ф.Уткина, 2023. – С. 406-411.

2. Yung-Ching Chang, Bin-Kai Shyu and Jia-Shung Wang Region-based fractal image compression with quadtree segmentation // 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing. Vol. 4. Munich, Germany. 1997. pp. 3125-3128.

3. 754-2019 — IEEE Standard for Floating-Point Arithmetic. Revision of IEEE Std 754—2008.

5. Shree Ram Khaitu, Sanjeeb Prasad Panday Fractal Image Compression Using Canonical Huffman Coding // Journal of the Institute of Engineering. vol. 15. 2020. pp. 91-105.

6. M. Vranješ, S. Rimac-Drlje, D. Žagar Objective Video Quality Metrics // Proceedings Elmar – International Symposium Electronics im Marine 45-49. 2007.

УДК 004.056.5(075.8)

М.О. Мещанинов, А.В. Крошилин

ВНЕДРЕНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ УЧЕТА РЕГЛАМЕНТНОГО ОБСЛУЖИВАНИЯ КОНТРОЛЬНО-ИЗМЕРИТЕЛЬНЫХ ПРИБОРОВ НА ПРЕДПРИЯТИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

Представлены подходы к внедрению разработанной информационной системе на предприятии, перечислены основные модули разработанной системы, этапы внедрения системы и преимущества использования.

Ключевые слова: нефтепереработка, внедрение, контрольно-измерительные приборы (КИП), проектирование, тестирование, анализ.

Нефтеперерабатывающие предприятия характеризуются высокой степенью автоматизации и сложностью технологических процессов. Контрольно-измерительные приборы (КИП) играют ключевую роль в обеспечении надежности и безопасности работы оборудования. Эффективное управление регламентным обслуживанием КИП требует использования современных информационных технологий [1, 3].

На сегодняшний день существует множество программных продуктов, предназначенных для учета и планирования обслуживания оборудования. Однако, многие из них не полностью удовлетворяют специфическим требованиям нефтеперерабатывающей отрасли. В этом контексте, разработка и внедрение индивидуальной информационной системы на базе 1С представляется наиболее оптимальным решением [2, 3].

Информационная система для учета регламентного обслуживания КИП была разработана с использованием платформы 1С. Система включает в себя следующие модули [3, 4]:

1. Учет КИП и их характеристик;
2. Планирование регламентных работ;
3. Выполнение и контроль выполнения работ;
4. Анализ и отчетность.

Внедрение системы проходило в несколько этапов [2, 5]. Анализ потребностей предприятия и формирование требований к системе является первым и одним из самых важных этапов внедрения информационной системы. Он позволяет определить ключевые проблемы и потребности предприятия в области учета регламентного обслуживания контрольно-измерительных приборов и сформировать требования к разрабатываемой системе. На этом этапе проводится детальное исследование существующих процессов учета регламентного обслуживания КИП на предприятии. Оценивается эффективность и точность существующих методов учета, выявляются проблемы и недостатки, а также определяются ключевые потребности предприятия в области автоматизации учета регламентного обслуживания КИП. Проектирование архитектуры системы и разработка программного обеспечения являются ключевыми этапами в разработке информационной системы для учета регламентного обслуживания контрольно-измерительных приборов (КИП) на предприятии нефтеперерабатывающей промышленности. Тестирование системы и ее настройка под специфику предприятия. Тестирование системы включает в себя комплексное проверку работоспособности и соответствия требованиям, проверку корректности работы всех функций и процессов системы в соответствии с техническим заданием, проверку способности системы обрабатывать большие объемы данных и обеспечивать быстрый доступ к информации, проверку защищенности системы от несанкционированного доступа и взлома. На этапе обучения персонала и ввод системы в эксплуатацию происходит проведение тренингов и семинаров для пользователей системы, включая инструктаж по использованию функций и процессов системы, а также демонстрацию примеров использования системы в реальных ситуациях, проведение обучения для администраторов системы, включая инструктаж по настройке и управлению системой, а также демонстрацию примеров администрирования системы, проведение обучения для технического персонала предприятия, включая инструктаж по установке и настройке системы, а также демонстрацию примеров технического обслуживания системы.

Внедрение разработанной системы позволило предприятию достичь следующих преимуществ:

1. Улучшение организации регламентного обслуживания КИП, что повысило надежность и безопасность производства.

2. Автоматизация процессов планирования и контроля за выполнением работ, что сократило время и затраты на их организацию.

3. Увеличение прозрачности и доступности информации о состоянии КИП и проведенных ремонтных работах, что облегчает принятие управленческих решений.

4. Возможность проведения аналитики и формирования отчетов о состоянии оборудования и эффективности регламентных работ, что способствует оптимизации ресурсного обеспечения процессов обслуживания.

5. Интеграция с другими системами предприятия, такими как системы управления производством и складского учета, что позволяет создать единую информационную среду и улучшить взаимодействие между подразделениями.

Внедрение информационной системы для учета регламентного обслуживания контрольно-измерительных приборов на предприятии нефтеперерабатывающей промышленности с использованием платформы 1С позволит значительно улучшить организацию и эффективность процессов обслуживания оборудования. Разработанная система обеспечит автоматизацию и контроль за выполнением регламентных работ, повысит надежность и безопасность производственных процессов, а также способствует оптимизации затрат и ресурсов предприятия.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Булашев А.А. Автоматизация управления техническим обслуживанием и ремонтом оборудования на предприятиях нефтегазовой отрасли. // Нефтегазовое дело. – 2019. – № 2. – С. 34-42.

2. Крошилин А.В. Предметно-ориентированные информационные системы: учебное пособие / А.В. Крошилин, С.В. Крошилина, Г.В. Овечкин. — Москва: Курс, 2023. — 176 с. — (Естественные науки).

3. Коротков А.В., Григорьев А.В. Информационные системы в управлении техническим обслуживанием и ремонтом оборудования на предприятиях нефтегазовой отрасли. // Инженерный журнал: наука и инновации. – 2020. – № 3. – С. 76-84.

4. Макаров А.В., Сергеев А.В. Информационные системы в нефтегазовой промышленности: состояние и перспективы развития. // Нефтегазовая геология, геофизика и геотехнологии. – 2019. – Т. 14. – № 3. – С. 76-84.

5. Мещанинов М.О., Крошилина С.В. Автоматизация процесс учета регламентного обслуживания контрольно-измерительных приборов на предприятии // Математическое и программное обеспечение вычислительных систем: Межвуз. сб. науч. тр. / Под ред. Г.В. Овечкина - Рязань: РГРТУ им. В.Ф. Уткина, январь 2024 - 202 с. (58-60).

УДК 004.891.2

А. А. Милованов, С. В. Крошилина

РАЗРАБОТКА ИНФОРМАЦИОННОЙ МОДЕЛИ ДЛЯ АВТОМАТИЗАЦИИ РАСЧЕТА МОТИВАЦИИ ПЕРСОНАЛА ПРОЕКТНОЙ ДЕЯТЕЛЬНОСТИ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

В статье рассматривается проблема автоматизации расчёта мотивации персонала в проектной деятельности. Предлагается разработка подсистемы для программы «1С ЗУП КОРП», которая позволит отслеживать этапы проекта и мотивировать сотрудников на основе оценки их вклада.

Ключевые слова: автоматизированный расчет мотивации, проектная деятельность, 1С, KPI, оценка вклада сотрудника.

На сегодняшний день основным программным инструментом, автоматизирующим процесс расчета мотивации персонала является программа «1С ЗУП КОРП».

Основными возможностями при расчете мотивации персонала в данном программном обеспечении являются:

- Автоматизированная работа с грейдами.
- Автоматизированная работа и расчет показателей эффективности деятельности персонала KPI [1, 3].

В соответствии с тем, что основным направлением деятельности организации является работа в команде, или другими словами проектная деятельность, то следующим рассмотренным программным обеспечением является 1С «Управление проектами» [2, 4].

- Основные автоматизируемые процессы.
- Работы, сроки и содержание проекта**.**
- Проектное бюджетирование.
- Ресурсы проектов.
- Работа подразделений.
- Сводный план (тематический план, производственная программа).
- ТМЦ.
- Субподрядчики.

На современном рынке программного обеспечения присутствует достаточно большое количество облачных решений, направленных на автоматизацию расчета мотивации сотрудников как проектной деятельности так и любых других направлений деятельности. Одним из данных примеров является motiviti.

Сравнительная характеристика анализируемых программных продуктов представлена в (таблице 1).

Таблица 1 - Сравнительная характеристика

	1С ЗУП КОРП	1С Управление проектами	motiviti
Возможность работы с грейдами	+	—	+
Возможность расчета показателей эффективности деятельности сотрудников	+	—	+
Возможность отслеживания выполнения этапов проекта	—	+	—
Ввод индивидуальных методов мотивации	—	—	—
Обеспечение информационной безопасности	+	+	—

По рассмотренным характеристикам можно отметить, что во всех представленных программах отсутствует возможность организации расчета и присвоение определенных методов мотивации персонала. Кроме того, в определенных программах отсутствует возможность отслеживания этапов реализации проекта. По результатам сравнительной характеристики можно сделать вывод о необходимости в разработке подсистемы, направленной на внедрение в используемое программное обеспечение в «1С ЗУП КОРП». Основной целью разрабатываемой подсистемы является расчет мотивации сотрудников проектной деятельности в соответствии с оценкой по результатам реализации определенных проектов, а также присвоение сотрудникам установленных мер мотивации.

На (рисунке 1) представлена информационная модель подсистемы для автоматизации расчета мотивации персонала проектной деятельности

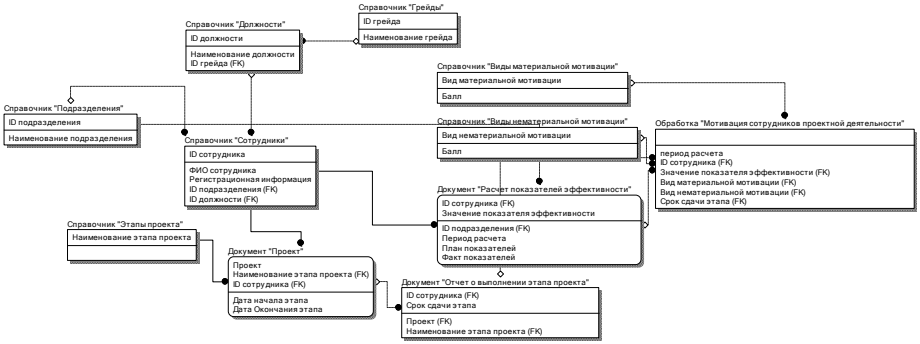


Рисунок 1 - Модель подсистемы для автоматизации расчета мотивации персонала проектной деятельности

По представленной информационной модели можно отметить, что подсистема должна состоять из следующих элементов:

Справочники:

- Этапы проекта.
- Виды материальной мотивации.
- Виды нематериальной мотивации сотрудников.
- Сотрудники.
- Должности.
- Грейды.

Документы:

- Проект.
- Отчет о выполнении этапа проекта.
- Расчет показателей эффективности деятельности сотрудников (KPI).

Обработка:

Мотивация сотрудников проектной деятельности.

Данную модель можно применять для последующей реализации на базе платформы «1С ЗУП КОРП» и дальнейшего внедрения.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Ермакова Я. М. Система мотивации и способы управления ей в команде проекта //Вестник науки. – 2023. – Т. 1. – №. 12 (69). – С. 73-79.

2. Абросимов И. П. Проблемы повышения эффективности работы компаний на базе автоматизации / И.П. Абросимов. – Тренды развития современного общества: управленческие, правовые, экономические и социальные аспекты. – 2022. – 132 с.

3. Милованов А.А., Крошилин А.В., Проблемы внедрения информационной системы мотивации сотрудников // Математическое и программное обеспечение вычислительных систем: Межвуз. сб. науч. тр. / Под ред. Г.В. Овечкина - Рязань: РГРТУ им. В.Ф. Уткина, январь 2024 - 202 с. (61-63)

4. Крошилина С.В. Предметно-ориентированные информационные системы: учебное пособие / А.В. Крошилин, С.В. Крошилина, Г.В. Овечкин. — Москва: КУРС, 2023. — 176 с. — (Естественные науки).

УДК 004.891.2

О. И. Никитов, С. В. Крошилина

ИССЛЕДОВАНИЕ ВИДОВ НЕЙРОСЕТЕЙ ДЛЯ ФОРМИРОВАНИЯ ЗАДАНИЙ НА ТЕСТИРОВАНИЕ ОБУЧАЮЩИХСЯ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

В статье рассматриваются различные виды нейросетей для формирования заданий на тестирование обучающихся.

Ключевые слова: нейросети, перцептрон, сверточные, рекуррентные, генеративно-состязательные.

Существует множество видов нейросетей, которые различаются по структуре, потокам данных, типам нейронов и их плотности, количеству слоёв и фильтрам их активации, а также по другим параметрам [1].

Перцептрон

Перцептрон — это обучаемая с учителем модель, которая делит данные на две категории, поэтому она является двоичным классификатором. Перцептрон был впервые описан Фрэнком Розенблаттом в 1957 году. Он основан на искусственном нейроне с настраиваемыми весами и порогом. Перцептрон разделяет входное пространство на две категории с помощью гиперплоскости, уравнение (1) которой выглядит так:

$$y = F\left(\sum_{i=1}^{n} \omega_i x_i\right) = F(WX) \qquad (1)$$

где

$X=(x1,x2,...,xn)^T$ – вектор входного сигнала;

$W = (\omega 1, \omega 2, ..., \omega n)$ – весовой вектор;

F – оператор нелинейного преобразования.

Персептроны

Персептроны представляют собой тип искусственных нейронных сетей, способных выполнять логические операции, такие, как И, ИЛИ или НЕ. Однако они могут обучаться только на линейно разделимых задачах, например, логическом операторе И. Для решения нелинейных задач, таких как XOR (исключающее ИЛИ), персептроны неэффективны.

Сверточные нейронные сети (CNN)

Сверточные нейронные сети представляют собой концепцию, в которой нейроны организованы в трехмерном пространстве, в отличие от стандартной двумерной модели. Первый слой в такой сети — сверточный,

где каждый нейрон обрабатывает информацию из небольшой области входных данных. Эти нейроны действуют как фильтры.

Архитектура CNN, предложенная Яном Лекуном в 1988 году, эффективно применяется для распознавания образов. Сеть способна интерпретировать части изображения и последовательно выполнять операции для полной обработки.

CNN широко используются в области обработки изображений, компьютерного зрения, распознавания речи и машинного перевода. Например, они могут преобразовывать цветные изображения в черно-белые, выделяя края, что помогает классифицировать изображения по категориям [2].

Рекуррентные нейронные сети (RNN)

Рекуррентные нейронные сети широко применяются в области обработки естественного языка (NLP). Они оценивают предложения на основе частоты встречаемости слов в тексте, что позволяет оценить грамматическую и семантическую правильность предложений.

RNN используются для машинного перевода и генерации новых текстов. Например, обученная на текстах Ремарка, нейронная сеть может создавать новые тексты в стиле автора. RNN используют предыдущие выходные данные в качестве входов, храня скрытые состояния.

Эта архитектура позволяет обработать широкий спектр задач, включая проверку грамматики текста, преобразование текста в речь и анализ тональности. Основное преимущество RNN заключается в работе с последовательными данными, где каждый элемент зависит от предыдущих, но процесс обучения может быть сложным.

Генеративно-состязательные нейросети (GAN)

GAN представляют собой подход к генерации данных на основе обучающего набора. GAN состоит из генератора и дискриминатора, которые соревнуются между собой, что побуждает их создавать всё более реалистичные данные. Чтобы лучше понять, что такое GAN, можно разделить этот термин на три составляющие:

Генеративная – изучение генеративной модели, которая описывает, как данные создаются с точки зрения вероятностной модели.

Состязательная – обучение модели в условиях соревнования.

Сеть – использование глубоких нейронных сетей для обучения.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Ротштейн А.П. Интеллектуальные технологии идентификации: нечеткая логика, генетические алгоритмы, нейронные сети. — Винница: УНИВЕРСУМ–Винница, 1999. — 320 с.

2. Ясвиндер Синг Бамра "Искусственный интеллект — алгоритмы, машины, психология". // – Питер, 2020.

УДК 004.932.72

М.И. Пасынков

МЕТОДЫ ОЦЕНКИ ПОЗЫ ЧЕЛОВЕКА НА ИЗОБРАЖЕНИИ НА ОСНОВЕ ГЛУБОКОГО ОБУЧЕНИЯ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

В статье рассматриваются методы оценки позы человека на изображении, их классификация, проводится сравнительный анализ методов с целью выбора наиболее подходящего для решения задачи оценки позы игроков на видеозаписи игры в волейбол.

Ключевые слова: оценка позы, ключевая точка.

Методы компьютерного зрения применяются во многих сферах деятельности, в том числе в спорте. Так, компьютерное зрение позволяет отслеживать правильность выполнения тех или иных спортивных упражнений, чтобы достичь лучшего результата и предотвратить травмы. В игровых видах спорта в сочетании с технологиями виртуальной и дополненной реальности компьютерное зрение повышает вовлеченность зрителя в просмотр игры.

При анализе игры в волейбол ключевое значение имеют действия игроков: подача, прием, передача, атака, блок. Такие действия выполняются игроками в определенных позах. В данной статье предлагается рассмотреть методы глубокого обучения, применяемые для оценки позы человека на изображении (Human Pose Estimation, HPE), определить наиболее подходящий метод для оценки позы игроков на кадрах видеозаписи игры в волейбол с целью дальнейшей классификации действий игроков.

Методы глубокого обучения стали применяться в задачах оценки позы с 2010-х гг. Эти методы используют возможности нейронных сетей для нахождения прямой связи между данными изображения и оценкой позы, что позволяет достичь лучшей точности и скорости работы в сравнении с другими методами (Iterative Closest Point, Active Shape Model и др.).

Классификация методов HPE на основе глубокого обучения представлена на рисунке 1. В зависимости от количества обрабатываемых объектов на изображении методы на основе глубокого обучения принято делить на:

1) методы оценки позы одного объекта;

2) методы оценки позы множества объектов.

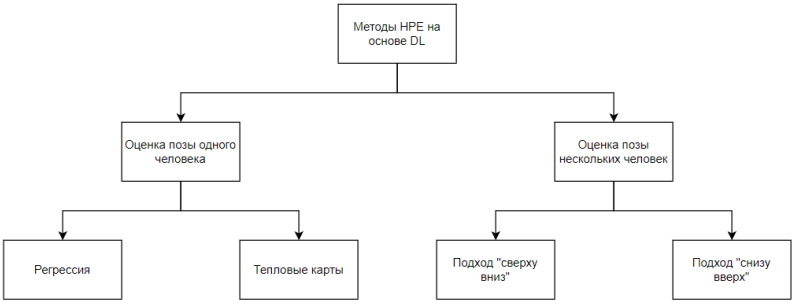


Рисунок 1 - Классификация методов HPE на основе глубокого обучения

Методы оценки позы одного объекта предполагают наличие на изображении ровно одного объекта, позу которого необходимо оценить. Оценка позы в этом случае заключается в предсказании положения ключевых точек объекта на изображении. Эта задача решается либо методами регрессии, либо методами, основанными на тепловых картах.

Регрессионные методы предсказывают координаты ключевых точек напрямую по данным изображения. Так, в алгоритме DeepPose [6] для предсказания координат ключевых точек используется нейронная сеть AlexNet; в алгоритме «Compositional pose regression» [4] – нейронная сеть ResNet-50.

Методы на основе тепловых карт, в отличие от регрессионных, строят тепловые карты для каждой из ключевых точек объекта и предсказывают положение ключевых точек по этим тепловым картам. Значение пикселя с координатами (x, y) на тепловой карте отображает вероятность нахождения ключевой точки в позиции (x, y) на исходном изображении. Такие нейронные сети обучаются минимизировать расхождение между предсказанными и целевыми тепловыми картами. Показывают большую точность в сравнении с регрессионными методами, однако процесс обучения нейронной сети оказывается более трудоемким, т.к. требуется обучить нейронную сеть предсказанию тепловой карты, а не координат ключевой точки. К тому же, результат работы такой нейронной сети необходимо дополнительно обрабатывать (определять координаты ключевых точек на основе тепловых карт), что снижает производительность системы [9].

В случае нахождения нескольких объектов на изображении помимо детектирования ключевых точек объектов, необходимо определять границы каждого из объектов на изображении. В зависимости от порядка решения этих задач методы оценки позы нескольких объектов принято разделять на два типа (рисунок 2).

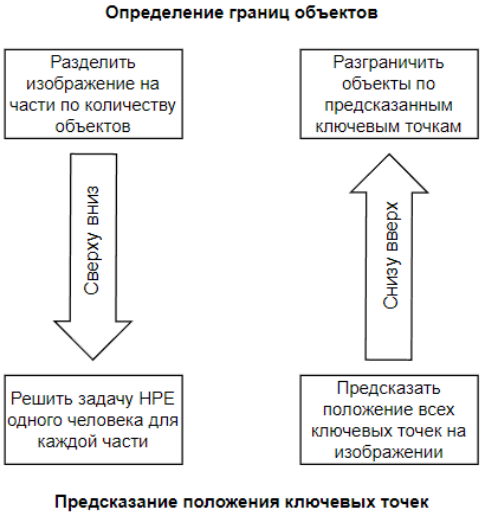


Рисунок 2 - Подходы к решению задачи HPE нескольких человек

К первому типу относятся методы, решающие задачу «сверху вниз» [2, 7]. Сначала на изображении проводятся границы между объектами, после чего для каждого выделенного объекта определяются ключевые точки. Для разграничения объектов на изображении используется детектор объектов, для определения ключевых точек – рассмотренные ранее методы оценки позы одного объекта на изображении. Сложность вычислений растет линейно при увеличении количества объектов на изображении, поэтому такой подход не может быть использован при решении задач в режиме реального времени.

Другим недостатком подхода «сверху вниз» является зависимость от качества детектора объектов, которые зачастую не могут распознать объекты, наложенные друг на друга на изображении.

Ко второму типу относятся методы, решающие задачу «снизу вверх» [1, 3, 10]. В этом случае, наоборот, сначала происходит поиск ключевых точек, а затем точки распределяются между найденными на изображении объектами. Поиск ключевых точек чаще всего производится с помощью тепловых карт, благодаря чему задачу удается решить за константное время.

Однако тепловые карты требуют сложной постобработки, которая выполняется вне сверточной нейронной сети. Поэтому такой подход не поддерживает сквозное (end-to-end) обучение, что усложняет достижение высокой точности работы системы. К тому же, тепловые карты не позволяют точно распределять ключевые точки между объектами, когда точки накладываются друг на друга.

Поэтому в [5] был описан метод оценки позы человека на основе нейронной сети YOLO без использования тепловых карт. Нейронная сеть предсказывает положение 17 ключевых точек человека. Сложная постобработка в этом случае не требуется, нейронная сеть может быть обучена «end-to-end», в связи с чем процесс обучения нейронной сети упрощается, удается достичь лучшего результата оценки ключевых точек.

Таблица 1 – Результаты сравнения методов HPE

Метод	Размер входного изображения	GMACs	AP	AP50
OpenPose [8]	-	-	61.8	84.9
Hourglass [1]	512	413.8	56.6	81.8
PersonLab [3]	1401	911	66.5	88.0
HRNet [4]	512	77.8	64.1	86.3
HigherHRNet	640	1620	70.5	89.3
DEKR [10]	512	90.8	67.3	87.9
YOLOv5m6-pose	960	66.3	66.6	89.8
YOLOv6l6-pose	960	145.6	68.5	90.3

В таблице 1 представлены результаты сравнения методов HPE по метрикам GMACs, AP и AR. Наибольшего значения метрики AP (70.5) удалось достичь при использовании метода HigherHRNet, однако вычислительная сложность при этом составила 1620 GMACs.

Лучший результат по метрике AP50 достигает модификация YOLOv5l6-pose с вычислительной сложностью 145.6 GMACs, что в 2 раза ниже, чем у ближайшего по точности конкурента.

Таким образом, метод YOLO-pose за счет подхода «снизу вверх» без использования тепловых карт позволяет достигать точности на уровне лучших алгоритмов HPE, однако требует значительно меньших вычислительных затрат. Поэтому для решения задачи оценки позы игрока на видеозаписи игры в волейбол был выбран метод YOLO-pose.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In ECCV, 2016.

2. Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In ECCV, 2018.

3. George Papandreou et al. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In ECCV, pp. 282–299, 2018.

4. Ke Sun et al. Deep high-resolution representation learning for human pose estimation. In CVPR, 2019.

5. Maji, Debapriya et al. “YOLO-Pose: Enhancing YOLO for Multi Person Pose Estimation Using Object Keypoint Similarity Loss.” 2022

IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2022): pp. 2636-2645.

6. Toshev, Alexander and Christian Szegedy. “DeepPose: Human Pose Estimation via Deep Neural Networks.” 2014 IEEE Conference on Computer Vision and Pattern Recognition (2013): pp. 1653-1660.

7. Yilun Chen et al. Cascaded pyramid network for multi-person pose estimation. In CVPR, 2018.

8. Zhe Cao et al. Realtime multi-person 2d pose estimation using part affinity fields. In CVPR, 2017.

9. Zheng, Ce et al. «Deep Learning-based Human Pose Estimation: A Survey». ACM Computing Surveys 56 (2020): pp. 1 - 37.

10. Zigang Geng et al. Bottom-up human pose estimation via disentangled keypoint regression. arXiv preprint arXiv:2104.02300, 2021.

УДК 004.891.2

Б.Д. Плешков, А.В. Крошилин

РАЗРАБОТКА МОДУЛЯ ПОДБОРА ИНДИВИДУАЛЬНЫХ МАРШРУТОВ ДЛЯ ТУРИЗМА В РЕКОМЕНДАТЕЛЬНОЙ СИСТЕМЕ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

В работе спроектирована часть рекомендательной системы подбора индивидуальных туристических маршрутов, которая на основе предпочтений пользователя позволяет создавать более точные и персонализированные рекомендации, улучшая удовлетворенность пользователей и повышая эффективность системы. Учитываются данные по основным темам, интересующие пользователя, а также оценка негативного или положительного отношения к ним.

Ключевые слова: рекомендательная система, машинное обучение, информационные технологии и анализ данных, персонализация, алгоритмы, большие объемы данных.

В рамках цифровизации, актуально проведение исследований в области больших данных и рекомендательных систем. Цель исследования - разработать рекомендательную систему для индивидуальных туристических маршрутов, основанную на анализе данных о пользовательской активности в социальных сетях и приложениях. Это позволит определить предпочтения и интересы каждого отдельного туриста, создавая более точные и персонализированные рекомендации [1, 6].

Основная задача заключается в разработке ключевого компонента рекомендательной системы, способного эффективно анализировать данные о предпочтениях пользователя. Это позволит выявить основные

темы, которые вызывают интерес у каждого туриста, и оценить степень его заинтересованности в каждой конкретной теме [2, 7].

Путешествия представляют собой не просто время для отдыха и получения новых впечатлений; они также являются отличным способом расширения горизонтов и изучения различных культур. В связи с этим, мы планируем создать и проверить важную часть рекомендательной системы, которая будет умело собирать и обрабатывать информацию о предпочтениях пользователей. Это позволит определить ключевые области интереса каждого путешественника и оценить их уровень заинтересованности в каждом аспекте.

Таким образом, основное направление исследования заключается не только в разработке новых и инновационных методов, но и в обосновании нового понимания пользовательского поведения и предпочтений, для создания высококачественных персонализированных рекомендаций, способных обеспечить удовлетворение пользователя и развитие эффективности рекомендательной системы.

Проектирование модуля для анализа персональных данных. Анализ данных о пользователе является важным шагом в разработке рекомендательной системы для индивидуальных туристических маршрутов. В данной работе выбраны несколько основных алгоритмов для анализа информации такие как Коллаборативная фильтрация, Контент-ориентированная фильтрация, Гибридные методы рекомендаций, Косинусная схожесть, TF-IDF. Эти методы позволяют проанализировать разнообразную информацию о предпочтениях пользователя и его активности [2].

Для сбора и анализа данных о пользователе система производит анализ профиля пользователя, его постов, комментариев и «лайков». Это позволяет получить информацию о темах, которые интересуют пользователя, а также его мнение касательно тем. Методы анализа включают обработку текста и выявление ключевых слов и фраз, а также анализ тональности текстовых сообщений.

Помимо социальных сетей данные собираются и внутри самого приложения пользователя. Например, система может анализировать посещенные места, историю поиска, отзывы о посещенных объектах. Эти данные предоставляют ценную информацию о предпочтениях пользователя в контексте туристических маршрутов [3].

Для хранения информации о пользователях и их активности, можно использовать следующую модель базы данных приведенной на рисунке 1.

Важно отметить, что в данной схеме используется нормализация, что обеспечивает эффективное использование пространства и уменьшает вероятность дублирования данных.

Для анализа персональных данных пользователя используются следующие алгоритмы и методы.

Коллаборативная фильтрация использует взаимодействие пользователей для предсказания предпочтений одного пользователя на основе поведения других пользователей. Она может быть разделена на две основные категории:

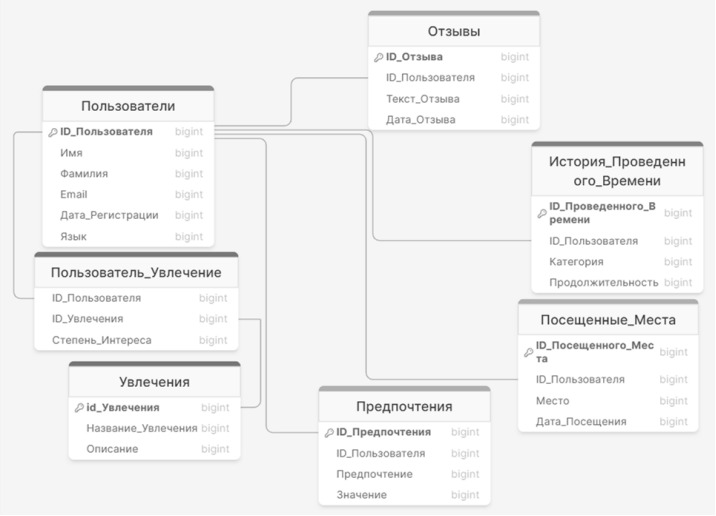


Рисунок 1 – Архитектура БД для хранения данных

Базовая CF: использует только взаимодействие пользователей для предсказаний. Например, если пользователь A часто посещает место B, и пользователь B часто посещает место C, то система может предположить, что пользователь A также захочет посетить место C.

Сглаженная CF: использует матричную факторизацию для уменьшения размерности данных и выявления скрытых структур в данных пользователей.

Контент-ориентированная фильтрация использует характеристики объектов (в нашем случае, мест) для предсказания предпочтений пользователя. Она может включать в себя анализ текстовых описаний мест, их категорий, характеристик и т.д., чтобы найти наиболее подходящие места для конкретного пользователя.

Гибридные методы рекомендаций объединяют коллаборативную и контент-ориентированную фильтрацию. Например, система может сначала применять коллаборативную фильтрацию для генерации списка потенциально интересующих мест, а затем использовать контент-ориентированную фильтрацию для уточнения этих предложений на основе личных предпочтений пользователя.

Косинусная схожесть используется для измерения сходства между двумя векторами. В контексте рекомендательных систем, она может быть

применена для определения сходства между профилами пользователей или характеристиками мест. Например, если у нас есть векторы, представляющие интересы пользователей или характеристики мест, косинусная схожесть позволяет нам понять, насколько близки они друг к другу.

Сходство корреляции Пирсона двух пользователей x, y определяется как

$$\text{simil}(x,y) = \frac{\sum_{i \in I_{xy}} (r_{x,i} - \bar{r_x})(r_{y,i} - \bar{r_y})}{\sqrt{\sum_{i \in I_{xy}} (r_{x,i} - \bar{r_x})^2} \sqrt{\sum_{i \in I_{xy}} (r_{y,i} - \bar{r_y})^2}} \quad (1)$$

где Ixy - это набор элементов, оцененных как пользователем x, так и пользователем y.

Подход, основанный на косинусах, определяет косинусное сходство между двумя пользователями x и y как:

$$\text{simil}(x,y) = \cos(\vec{x},\vec{y}) = \frac{\vec{x} \cdot \vec{y}}{||\vec{x}|| \times ||\vec{y}||} = \frac{\sum_{i \in I_{xy}} r_{x,i} r_{y,i}}{\sqrt{\sum_{i \in I_x} r_{x,i}^2} \sqrt{\sum_{i \in I_y} r_{y,i}^2}} \quad (2)$$

Алгоритм рекомендаций top-N на основе пользователей использует векторную модель, основанную на сходстве, для идентификации k пользователей, наиболее похожих на активного пользователя.

$$\text{simil}(x,y) = \cos(\vec{x},\vec{y}) = \frac{\vec{x} \cdot \vec{y}}{||\vec{x}|| \times ||\vec{y}||} = \frac{\sum_{i \in I_{xy}} r_{x,i} r_{y,i}}{\sqrt{\sum_{i \in I_x} r_{x,i}^2} \sqrt{\sum_{i \in I_y} r_{y,i}^2}} \quad (3)$$

TF-IDF — это статистическая мера, используемая для взвешивания слов в документе. Она учитывает частоту появления слова в документе (term frequency) и обратную частоту его появления во всем корпусе документов (inverse document frequency). В контексте рекомендательных систем, TF-IDF может быть использована для обработки текстовой информации о местах или пользователях, позволяя лучше понять и сравнить их характеристики.

TF (term frequency — частота слова) — отношение числа вхождений некоторого слова к общему числу слов документа. Таким образом, оценивается важность слова

$$\mathrm{tf}(t,d) = \frac{n_t}{\sum_k n_k} \quad (4)$$

где nt есть число вхождений слова в документ, а в знаменателе – общее число слов в данном документе.

IDF (inverse document frequency — обратная частота документа) – инверсия частоты, с которой некоторое слово встречается в документах коллекции. Основоположником данной концепции является Карен Спарк Джонс. Учёт IDF уменьшает вес широкоупотребительных слов. Для каждого уникального слова в пределах конкретной коллекции документов существует только одно значение IDF.

$$\mathrm{idf}(t,D) = \log \frac{|D|}{|\{\, d_i \in D \mid t \in d_i \,\}|} \quad (5)$$

где |D| — число документов в коллекции;

$|\{\, d_i \in D \mid t \in d_i \,\}|$ — число документов из коллекции D в которых встречается (когда $nt \neq 0$)

Выбор основания логарифма в формуле не имеет значения, поскольку изменение основания приводит к изменению веса каждого слова на постоянный множитель, что не влияет на соотношение весов. Таким образом, мера TF-IDF является произведением двух сомножителей: Большой вес в TF-IDF получат слова с высокой частотой в пределах конкретного документа и с низкой частотой употреблений в других документах.

В целом, комбинация этих методов позволяет создавать более точные и персонализированные рекомендации, улучшая удовлетворенность пользователей и повышая эффективность системы.

Выбор методов решения задач обусловлен целями и требованиями к системе. В данном случае, цель состоит в том, чтобы получить максимально полную и достоверную информацию о предпочтениях пользователя. Это включает в себя анализ данных из социальных сетей, данных из приложения пользователя и данных браузера.

Обоснование выбранных алгоритмов и методов. Основные методы, выбранные для решения этой задачи, включают обработку естественного языка (NLP), кластеризацию, машинное обучение и применение принципов защиты данных.

Коллаборативная фильтрация (CF): Этот метод идеально подходит для анализа поведения пользователей и предсказания их будущего поведения на основе взаимодействия других пользователей. В контексте

туризма, это позволяет выявлять популярные маршруты среди пользователей с похожими интересами и предпочтениями, что способствует созданию персонализированных маршрутов для новых пользователей.

Контент-ориентированная фильтрация (Cf): Этот метод позволяет анализировать характеристики мест, таких как исторические достопримечательности, природные красоты, культурные мероприятия и т.д., чтобы предложить пользователям маршруты, соответствующие их интересам и предпочтениям. Это обеспечивает высокую степень персонализации и удовлетворенности пользователей.

Гибридные методы: Гибридные методы, сочетающие в себе преимущества коллаборативной и контент-ориентированной фильтрации, позволяют создавать более точные и релевантные рекомендации. Они могут начинаться с широкого поиска возможных маршрутов на основе поведения пользователей (коллаборативная фильтрация), а затем уточнять эти предложения, используя детальную информацию о местах (контент-ориентированная фильтрация).

Косинусная схожесть: этот метод эффективно используется для измерения сходства между различными данными, такими как профили пользователей или характеристики мест. Это позволяет определить, какие маршруты будут наиболее интересны пользователю на основе его прошлых путешествий и предпочтений.

TF-IDF: этот метод используется для обработки текстовой информации о местах и пользователях, позволяя лучше понять и сравнивать их характеристики. Это особенно важно для контент-ориентированной фильтрации, где анализ текстовых описаний мест может помочь в выявлении ключевых аспектов, интересующих пользователя.

Применение принципов защиты данных: этот метод обеспечивает конфиденциальность персональных данных пользователя. Это важно для соблюдения законов о защите данных и для уважения к приватности пользователя.

Полученные результаты анализа могут включать в себя различные виды информации, такие как предпочтения пользователя, его активность в социальных сетях и в приложении, а также интересы, выявленные на основе его поведения в Интернете. Эта информация может быть использована для создания персонализированных рекомендаций для пользователя.

Важно отметить, что результаты анализа должны быть интерпретированы и использованы для формулирования выводов, которые могут влиять на дальнейшие шаги в развитии системы. Это может включать в себя проведение дополнительных экспериментов или изменение методов сбора данных, если результаты анализа указывают на необходимость этого.

Для реализации данной задачи, был использован следующий набор технологий, предпочитая российские аналоги:

Языки программирования: Python, Java или JavaScript могут быть использованы для разработки модуля. Python является популярным выбором из-за его простоты и богатого набора библиотек для анализа данных и машинного обучения.

Библиотеки для обработки естественного языка (NLP): Natural Language Toolkit (NLTK) или spaCy в Python, CoreNLP в Java, или Natural в JavaScript могут быть использованы для анализа текстовых данных и выявления ключевых слов и фраз.

Библиотеки для кластеризации: scikit-learn в Python, Weka в Java, или ML.js в JavaScript могут быть использованы для группировки мест по типам и интересам пользователя на основе его истории посещений.

Библиотеки для машинного обучения: scikit-learn в Python, Weka в Java, или ML.js в JavaScript могут быть использованы для прогнозирования интересов пользователя на основе его поведения в Интернете.

API социальных сетей и браузеров: Для сбора данных из социальных сетей и браузеров можно использовать API, такие как VK API для социальных сетей и Яндекс Браузер API для браузеров.

Модуль для перевода текста: Модуль, такой как Яндекс.Переводчик, может быть использован для перевода текста на английский язык перед анализом, что обеспечивает более адаптивный анализ.

Библиотеки для работы с базами данных: Для хранения и обработки собранных данных можно использовать библиотеки, такие как SQLAlchemy в Python, Hibernate в Java, или Sequelize в JavaScript.

Фреймворки для веб-разработки: Django или Flask в Python, Spring в Java, или Express.js в JavaScript могут быть использованы для создания веб-приложения, которое будет использовать модуль для сбора данных.

Сводная информация по выбранным технологиям представлена в таблице 1.

Таблица 1 — Основные наборы технологий

Технология	Описание	Пример использования
Python	Язык программирования	Разработка модуля, анализ текстовых данных, кластеризация, машинное обучение
NLTK или spaCy	Библиотеки для обработки естественного языка (NLP)	Анализ текстовых данных для выявления ключевых слов и фраз, анализ тональности текстовых сообщений
scikit-learn	Библиотека для кластеризации и машинного обучения	Группировка мест по типам и интересам пользователя, прогнозирование интересов пользователя
VK API	API социальных сетей	Сбор данных из социальных сетей
Яндекс Браузер API	API браузеров	Сбор данных из браузера
Яндекс.Переводчик	Модуль для перевода текста	Перевод текста на английский язык перед анализом
SQLAlchemy, Hibernate, Sequelize	Библиотеки для работы с базами данных	Хранение и обработка собранных данных
Django или Flask, Spring, Express.js	Фреймворки для веб-разработки	Создание веб-приложения, которое будет использовать модуль для сбора данных

Также необходимо учесть, что итоговое приложение должно быть удобно для людей, которые говорят на разных языках. благодаря мощному инструменту API Яндекс.Переводчик.

Этот сервис автоматического перевода слов и выражений, текстов с фотографий и картинок, сайтов и мобильных приложений, разработан в Яндексе и позволит хранить в базе информацию более структурированно на одном языке о всех пользователях что ускорит работу сервиса.

В результате проведенного исследования была разработана часть рекомендательной системы для индивидуальных туристических маршрутов, которая анализирует собранную информацию о предпочтениях пользователя на основе его активности в социальных сетях и внутри приложения. Это включает анализ и обработку данных, чтобы оценить основные темы, интересующие пользователя, а также степень интереса к каждой теме, включая оценку негативного или положительного отношения к ним.

В ходе исследования были выбраны несколько основных алгоритмов анализа данных: Коллаборативная фильтраци, Контент-ориентированная фильтрация, Гибридные методы, Косинусная схожесть, TF-IDF.

Полученные результаты анализа могут включать в себя различные виды информации, такие как предпочтения пользователя, его активность в социальных сетях и в приложении, а также интересы, выявленные на основе его поведения в Интернете. Эта информация может быть использована для создания персонализированных рекомендаций для пользователя.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Крошилин А. В., Крошилина С. В. Интеллектуальные поисковые системы на основе нечеткой логики. Учебное пособие для вузов. - М.: Горячая линия – Телеком, 2023. – 140 с.: ил.

2. Крошилина С. В., Жулев В. И., Крошилин А. В., Проектирование систем поддержки принятия решений. Учебное пособие для вузов. -М.: Горячая линия– Телеком, 2023. – 180 с.: ил.

3. “Аналитика поведения пользователей: как организовать сбор данныхинформации” [Электронный ресурс]. – https://spark.ru/startup/carrot-quest/blog/68390/analitika-povedeniya-polzovatelej-kak-organizovat-sbor-dannih.

4. “Обзор алгоритмов кластеризации данных” [Электронный ресурс]. – https://habr.com/ru/articles/101338.

5. “GC-NLDP: алгоритм кластеризации графов с локальной дифференциальной конфиденциальностью” [Электронный ресурс]. – https://www.sciencedirect.com/science/article/abs/pii/S0167404822003595.

6. Плешков Б.Д., Крошилин А.В. Исследование в сфере персонализации и адаптации рекомендаций в контексте индивидуальных потребностей туристов на основе искуственного интелекта // Новые информационные технологии в научных исследованиях: материалы XXVIII Всероссийской научно-технической конференции студентов, молодых ученых и специалистов. РГРТУ им. В.Ф. Уткина. т.1, Рязань: 2023. 197с.(180-181)

7. Плешков Б.Д., Крошилина С.В. Разработка модуля сбора и анализа информации в рекомендательной системе индивидуальных туристических маршрутов // Математическое и программное обеспечение вычислительных систем: Межвуз. сб. науч. тр. / Под ред. Г.В. Овечкина - Рязань: РГРТУ им. В.Ф. Уткина, январь 2024 - 202 с. (88-95)

УДК 004.82

М. Д. Привар, З. А. Кононова

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ НА СЕГОДНЯШНИЙ ДЕНЬ

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Липецкий государственный педагогический университет имени П. П. Семенова-Тян-Шанского», Липецк

В статье описана история разработок в сфере искусственного интеллекта, рассматривается обзор состояния различных исследований и разработок систем на текущее время, представлены возможности применения ИИ в различных отраслях человеческой деятельности.

Ключевые слова: Искусственный интеллект, технологии, нейронные системы, машинное обучение

Человеческий разум представляет собой сложную и многогранную систему. Вопрос его изучения и воспроизведения всегда был актуален, а в наше время он стал особенно насущным. Развитие современных компьютерных технологий породило множество задач, связанных с: человеческой речью, созданием систем технического зрения, разработкой автомобилей, которые могут самостоятельно передвигаться без участия человека, и многими другими аспектами.

Системы, имитирующие и воспроизводящие поведение человека, называются искусственным интеллектом (ИИ). Исследование ИИ является ключевым направлением современной науки.

Что такое искусственный интеллект? Одно из определений утверждает, что ИИ — это технология, включающая в себя набор инструментов, позволяющих компьютеру на основе изученных данных отвечать на вопросы и делать выводы, т.е. обрабатывать информацию, которая не была заложена в него изначально. Раздел науки под названием «искусственный интеллект» является частью компьютерных наук, а создаваемые на его основе технологии относятся к IT-сфере.

Системы искусственного интеллекта представляют собой различные устройства или комплексы, использующие технологии ИИ в своей работе. Во многих случаях алгоритм решения задачи остается неизвестным до получения результата. Современный ИИ может искать информацию в Интернете, диагностировать болезни и выполнять другие задачи, что делает жизнь комфортнее и работу эффективнее. Такие системы ИИ с течением времени становятся все более развитыми, и уже сейчас некоторые виды работ они выполняют лучше, чем человек.

Важно отметить, что разработка систем ИИ требует значительной подготовительной работы. Машины обучаются поиску информации, распознаванию и обработке речи, работе с естественным языком, распознаванию лиц и многим другим навыкам. В настоящее время ИИ не способен выполнять множество задач одновременно, но технологии

стремительно развиваются, и со временем искусственный интеллект сможет достичь уровня развития, сравнимого с человеческим.

Для развития искусственного интеллекта ученые начали исследовать методы представления знаний. Это позволило создать экспертные системы (ЭС) — системы, которые помогают принимать решения, извлекая знания из баз данных. Важной целью стало также развитие методов самообучения машин и эксперименты с моделированием работы нервной системы человека, что привело к созданию искусственных нейронных сетей (ИНС). Таким образом, основой всех исследований и разработок в области искусственного интеллекта является принцип имитации процессов человеческого разума с помощью компьютеров. Искусственный интеллект, как научная дисциплина, относится к когнитивным наукам, то есть к тем, которые связаны с усвоением знаний.

Ожидается, что искусственный интеллект, достигший уровня, сопоставимого с человеческим, найдет широкое применение и существенно изменит жизнь людей.

Развитие искусственного интеллекта (ИИ) можно разделить на три этапа:

1. В 1950-х годах началась работа по созданию искусственного интеллекта, сосредоточенная на решении двух конкретных задач. Первая — разработка шахматной программы. В 1954 году корпорация REND при участии Алана Тьюринга и Клода Шеннона приступила к созданию шахматной программы, завершив её в 1957 году. В основе работы программы лежала эвристика — метод выбора решения без теоретических оснований.

Вторая задача касалась машинного перевода с одного языка на другой. С 1954 по 1957 годы в СССР под руководством Л.Н. Королёва проводились первые эксперименты по переводу с английского и китайского языков. В 1954 году корпорация IBM под руководством профессора Л. Достерта перевела около шестидесяти фраз с русского на английский, используя словарь из 250 пар слов и шесть грамматических правил. Однако результаты оказались менее впечатляющими, чем ожидалось, так как задача оказалась сложнее, чем предполагалось. Машины требовали обучения не только правилам, но и исключениям, а вычислительные мощности того времени не позволяли реализовать эту цель. Тем не менее, эти эксперименты дали значительный импульс развитию математической лингвистики.

Среди ранних достижений в области искусственного интеллекта стоит отметить создание в 1963 году Джоном Маккарти первого языка программирования для задач ИИ — ЛИСП. Этот язык стал основой для функционального программирования, тогда как первые языки высокого уровня были исключительно процедурными.

2. Второй этап развития искусственного интеллекта, начавшийся в конце 1960-х годов, включал разработку логического программирования и создание экспертных систем (ЭС). Хотя это были лишь начальные шаги в развитии ИИ, экспертные системы уже позволяли автоматизировать логические выводы на основе знаний, внесенных вручную специалистами. В этих системах эксперты по управлению знаниями опрашивали специалистов и заполняли базы данных вручную, а машина делала логические выводы в пределах заложенной информации, без способности к самообучению.

Одной из значительных проблем этого этапа было сопротивление специалистов, не желавших делиться своими знаниями из-за опасений, что развитие экспертных систем может снизить их профессиональный статус. Экспертные системы позволяли даже начинающим специалистам достигать высоких результатов, что вызывало обеспокоенность у опытных профессионалов.

Несмотря на эти сложности, разработка экспертных систем привлекла большой интерес к вопросу представления знаний в компьютерных системах. Это привело к созданию различных методов, таких как фреймы, семантические сети, продукционные системы и их комбинации, которые стали основой для дальнейшего развития искусственного интеллекта.

Со вторым этапом развития искусственного интеллекта также связано усовершенствование приложений для шашек и шахмат. В этот период прошел первый чемпионат машин в шахматы, где они соревновались друг с другом. Особое внимание следует уделить достижению советской шахматной программы "Каисса" в 1974 году. Ее победа стала мировым событием, поскольку эксперты ожидали, что американская программа займет первое место. М.В. Донской заявил: "Каисса была на уровне второго разряда шахматистов, то есть до того уровня, на котором программы должны были побеждать гроссмейстеров, ей еще далеко".

Устройства второго этапа иногда называют "символьным ИИ". Они в основном основаны на формальной логике, которая хорошо подходит для задач, таких как логические игры, но затрудняет создание систем, применимых в реальном мире.

3. На сегодняшний день интерес к искусственному интеллекту вновь оживает, что является третьим этапом его развития. Этот этап отличается как размахом, так и объемом, поскольку на данный момент у нас есть как технические возможности, так и значительные продвижения в этой области. Началом третьего этапа считается победа машины "Дип Блю" над чемпионом мира по шахматам Г. Каспаровым.

Характерной особенностью текущего этапа является стремительное развитие искусственных нейронных сетей (ИНС), которые моделируют

работу биологических нейронов. Простейшая искусственная нейронная сеть имеет три слоя нейронов: первый слой получает сигналы из окружающего мира, внутренний слой обрабатывает эти сигналы, а выходной слой формирует и выдает результат. Однако в таких сетях, может быть, множество внутренних или скрытых слоев.

Что происходит в области искусственного интеллекта в настоящее время?

В настоящее время значительное количество научных исследований посвящено компьютерному зрению, причем особое внимание уделяется развитию глубинного обучения. Первый раз в истории машины начали выполнять отдельные визуальные задачи лучше, чем человек. Например, точность определения методов лечения раковых заболеваний, продемонстрированная компьютером IBM Watson, составляет 90%, что на 40% выше точности диагностики, проводимой врачами.

Одной из основных концепций в сфере искусственного интеллекта является "машинное обучение", или, как его еще называют, "статистическое обучение". Корни этой технологии уходят в прошлое, к концу 1950-х годов, когда Артур Самюэль предложил обучать машины без использования заданных алгоритмов. Суть этой концепции заключается в том, чтобы программа училась на основе изменений, происходящих в её среде, и благодаря этому улучшала свои навыки в выполнении конкретных задач.

Машинное обучение представляет собой технологию, в которой сначала создается база обучающих примеров, на основе которых машина обучается и способна правильно анализировать и систематизировать поступающую информацию. Другими словами, это комбинация алгоритмов и методов, позволяющих машинам принимать решения на основе имеющихся данных. Это также включает в себя процесс самообучения программы. Благодаря машинному обучению машины могут распознавать лица на основе огромного количества фотографий, и делать это даже более точно, чем человек.

Почему такие высокие ожидания связаны с искусственным интеллектом?

1. Более чем за полвека компьютеризации, охватившей практически все сферы человеческой деятельности, мы столкнулись с некоторыми ограничениями, особенно в обработке постоянно поступающей информации. Это привело к разработке различных инструментов, таких как банки данных, оперативный анализ данных, облачные хранилища и концепция Больших данных. В настоящее время ведущие мировые компании в области информационных технологий участвуют в гонке за разработкой специализированных процессоров и суперкомпьютеров, предназначенных для обучения искусственных нейронных сетей.

2. Эпоха компьютеров, которая когда-то являлась главной движущей силой нашего времени, подходит к концу. Сейчас все внимание сосредоточено вокруг искусственного интеллекта и робототехники. Эти области уже сформировали огромные сегменты, включая промышленную, медицинскую, военную робототехнику, а также транспортные средства с беспилотным управлением. Однако без применения искусственного интеллекта полноценное развитие этих сегментов невозможно.

3. Предполагается, что успехи в разработке искусственного интеллекта принесут огромную прибыль тем странам, которые активно инвестируют в исследования в этой области. Именно по этой причине множество стран включили развитие искусственного интеллекта в число своих приоритетных задач.

Заключение. Ранее одним из основных вопросов в области искусственного интеллекта был спор о возможности моделирования человеческого разума и появления сознания у искусственного интеллекта. Сегодня эта проблема уже не кажется такой гипотетической и, скорее всего, играет важную роль в предсказании ближайшего будущего человечества. История развития искусственного интеллекта, протянувшаяся более полувека, доказывает, что нет существенных и серьезных препятствий для достижения этой цели. Вероятно, искусственный интеллект может быть создан не только на основе искусственных нейронных сетей. Однако именно нейронные сети представляют собой наиболее очевидное и доступное решение, вдохновленное природой.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Бабич, Н. А Анализ эффективности применения интерференционной нейронной сети для решения задачи распознавания образов / Н.А. Бабич // Вестник современных исследований. - 2019. - № 2.3 (29). - С. 5-8. –https://elibrary.ru/item.asp?id=37037590

2. Байнов, А.М. Роль и место робототехники в современном мире / А.М. Байнов, Р.С. Зарипова // Наука и образование: новое время. - 2019. - № 1 (30). - С. 93-95. – https://elibrary.ru/item.asp?id=37106314

3. Вознюк, П.А.Влияние искусственного интеллекта на мировую экономику / П.А. Вознюк // Тенденции развития науки и образования: рецензируемый научный журнал. - 2019. - 2019 г. №48, Часть 3. - С. 14-17. –http://ljournal.ru/wpcontent/uploads/2019/05/lj03.2019_p3.pdf

4. Головенко, А.П. Использование искусственного интеллекта в инновационных системах / А.П. Головенко // Вестник современных исследований. - 2018. - № 12.5 (27). - С. 67-68. – https://elibrary.ru/item.asp?id=36708991

УДК 004.021

А.И. Бракаренко, И.В. Богатырев, Е.Н. Проказникова

НЕОБХОДИМОСТЬ МОДЕЛИРОВАНИЯ КВАНТОВЫХ ВЫЧИСЛЕНИЙ СРЕДСТВАМИ КЛАССИЧЕСКИХ ЭВМ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

В данной статье рассматривается проблема необходимости моделирования квантовых вычислений в учебном процессе.

Ключевые слова: Квантовые вычисления, кубит, квантовый компьютер.

Технологический процесс дошёл до массового производства электронных вычислительных машин (ЭВМ), работающих благодаря дискретным сигналам и бинарной логике, основанной на операциях конъюнкции, дизъюнкции, инверсии и «исключающего или» (AND, OR, NOT, XOR соответственно). Базовой единицей в компьютере является транзистор, на базе него собирают элементы арифметико-логического устройства (АЛУ), такие как: сумматор, пирамидальный сумматор, конвейерный умножитель. Из-за этого есть прямо пропорциональная зависимость между количеством транзисторов и производительностью ЭВМ. Все силы технических производств направлены на уменьшение размеров транзисторов.

В настоящий момент технологический прогресс заходит в тупик. Данную тенденцию можно наблюдать благодаря перовому закону Мура. Первый закон Мура — эмпирическое наблюдение, изначально сделанное Гордоном Муром, согласно которому количество транзисторов, размещаемых на кристалле интегральной схемы, удваивается каждые 24 месяца. В производстве процессоров мы достигли физического предела в уменьшении размеров транзистора, сейчас минимально производимый размер транзистора равен 5 нм (для сравнения: толщина человеческого волоса равна 10 000 нанометров). Помимо физической сложности производства, есть экономическая сторона вопроса. По второму закону Мура стоимость фабрик по производству микросхем экспоненциально возрастает с усложнением производимых микросхем.

Проблема производительности и более экономичного производства лежит в изменении логики работы компьютеров. В 1959 году предпринималась попытка перейти на троичную систему счисления, вместо первичных бит и байт она оперировала тритами и трайтами. Но серьезный технологический прыжок был совершен в 1980 году – именно тогда Юрий Манин, а годом позже Ричард Фейнман предположили модель квантового компьютера. С 1982 по 1984 годы были предположены теоретическая схема квантового компьютера и первый квантовый вентиль – аналог логического элемента. Ключевой особенностью квантовых

компьютеров стало использование так называемых квантовых объектов – кубитов.

Квантовый компьютер — это вычислительное устройство, работа которого строится на квантовомеханических эффектах, в частности на принципе квантовой запутанности, позволяющем реализовать параллелизм вычислений [1-5].

Главные технологии для квантового компьютера:

1. Твердотельные квантовые точки на полупроводниках: в качестве логических кубитов используются либо зарядовые состояния (нахождение или отсутствие электрона в определённой точке), либо направление электронного и/или ядерного спина в данной квантовой точке. Управление через внешние потенциалы или лазерным импульсом.
2. Сверхпроводящие элементы (джозефсоновские переходы, СКВИДы и др.). В качестве логических кубитов используются присутствие/отсутствие куперовской пары в определённой пространственной области. Управление: внешний потенциал/магнитный поток.
3. Ионы в вакуумных ловушках Пауля (или атомы в оптических ловушках). В качестве логических кубитов используются основное/возбуждённое состояния внешнего электрона в ионе. Управление: классические лазерные импульсы вдоль оси ловушки или направленные на индивидуальные ионы + колебательные моды ионного ансамбля. Эту схему предложили в 1994 году Петер Цоллер и Хуан Игнасио Сирак.
4. Смешанные технологии: использование заранее приготовленных запутанных состояний фотонов для управления атомными ансамблями или как элементы управления классическими вычислительными сетями.
5. Оптические технологии: использование генерации квантовых состояний света, быстрого и перенастраиваемого управления этими состояниями и их детектирование.

Основные проблемы, связанные с созданием и применением квантовых компьютеров:

- необходимо обеспечить высокую точность измерений;
- внешние воздействия (включая передачу полученных результатов) могут разрушить квантовую систему или внести в неё искажения.

Кроме этого особенно актуальной становится проблема разработки вычислительных алгоритмов для подобных компьютеров и обучение программистов, работающих с квантовой логикой.

В настоящее время существуют разработки эмуляции работы квантовых вычислений с помощью суперкомпьютеров (например, на суперкомпьютере Ломоносов было достигнуто моделирование 32

кубитов). Однако, количество таких разработок относительно мало. Возможности моделирования квантового компьютера на классических устройствах весьма ограничено. Считается, что на современных персональных компьютерах доступное число кубитов составляет порядка 25-28, а для моделирования 35 идеальных кубитов уже требуется более 1 терабайта оперативной памяти [2].

Все это делает актуальным внедрение в образовательный процесс профильных специальностей дисциплин, направленных на изучение работы квантового процессора и алгоритмов с использованием квантовых вычислений. Для образовательных целей могут использоваться либо дорогостоящие суперкомпьютеры, либо классические компьютеры с установленным специализированным программным обеспечением, эмулирующим квантовые вычисления и работу квантового процессора.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Ю.И.Ожигов Квантовые вычисления. – Москва МГУ, 2003 – 104 с.

2. Корж О.В., Андреев Д.Ю., Корж А.А., Коробков С.В., Чернявский А.Ю. Моделирование работы идеального квантового компьютера на суперкомпьютере "Ломоносов" – Москва МГУ, «Вычислительные методы и программирование: новые вычислительные технологии», Том: 14 Номер: 2 (28), 2013 – 24-34 с.

3. Васюков В.Л. Квантовая логика. – М.: ПЕР СЭ, 2005. – 192 с.

4. Кронберг Д.А., Ожигов Ю.И., Чернявский А.Ю. Квантовая информатика и квантовый компьютер – М.: МГУ.

5. Валиев К.А., Квантовые компьютеры: смена парадигмы вычислений, Вестн. МГУ,сер. 15: Выч.мат.и киберн. 2005 , №2, 3-16 с.

УДК 004.021

С.А. Рябинин, И.Д. Попов, Е.Н. Проказникова

ИССЛЕДОВАНИЕ ЧАСТОТНОЙ ДЕКОМПОЗИЦИИ ИЗОБРАЖЕНИЯ ПОСРЕДСТВОМ ПРЕОБРАЗОВАНИЯ ВЕЙВЛЕТА

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

В данной статье рассматриваются сравнительный анализ частотной декомпозиции изображений

Ключевые слова: Частотная декомпозиция изображений, вейвлет преобразования.

В последние десятилетия развитие цифровых технологий и обработки изображений привело к возникновению различных методов анализа и обработки изображений, обеспечивающих более эффективное использование графической информации. Одним из таких методов является разложение изображения по частотам с использованием вейвлет-

преобразования, это широко изученным методом в области обработки сигналов и изображений, это подтверждается широким спектром литературных источников, посвященных вейвлет-преобразованию [2] и [3]. Вейвлет-анализ представляет собой мощный инструмент для анализа изображений, позволяющий выделять и интерпретировать различные частотные составляющие изображения с высокой точностью и эффективностью. Изображение рассматривается как двумерный массив пикселей, где каждый пиксель содержит информацию о яркости или цвете определенной точки на изображении.

Разложение изображения по частотам методом вейвлета является важным инструментом в обработке изображений, поскольку позволяет выделять как низкочастотные, так и высокочастотные компоненты изображения. Этот подход особенно полезен для обнаружения и удаления помех и шумов, что делает его неотъемлемым в различных сферах применения. Согласно [1], применение вейвлет-преобразований в обработке изображений представляет собой мощный метод анализа и обработки изображений с широким спектром применений.

Например, разложение изображений по частотам вейвлета находит широкое применение в обработке и анализе спутниковых снимков. Низкочастотные компоненты могут использоваться для выделения географических объектов и территорий, в то время как высокочастотные компоненты позволяют выявлять мелкие детали и структуры, такие как дороги, здания и растительность. Это помогает улучшить качество обработки снимков, повысить точность картографических данных и обнаружить изменения в природных и географических объектах.

Также данный алгоритм используется в медицинской диагностике. Он применяется для анализа медицинских изображений, таких как рентгеновские снимки и снимки магнитно-резонансной томографии. Низкочастотные компоненты могут помочь выделить области интереса, в то время как высокочастотные компоненты могут помочь выявить аномалии и патологии.

Данный алгоритм востребован в настоящее время в оборонной сфере, так как разложение изображений по частотам вейвлета используется для обработки данных с различных видов наблюдения, таких как аэрофотосъемка, видеонаблюдение и радиолокационное зондирование. Низкочастотные компоненты позволяют выделить объекты интереса и территории, в то время как высокочастотные компоненты помогают обнаружить скрытые объекты и аномалии, такие как скрытые объекты или изменения в обстановке.

В эпоху цифровых технологий применение данного преобразования позволяет выделять ключевые объекты и структуры на изображениях, что улучшает работу систем распознавания и

классификации содержания, таких как лица людей, автомобили, здания и т.д.

Таким образом, разложение изображения по частотам методом вейвлета является важным инструментом в обработке и анализе изображений, обладающим широким спектром применения и играющим ключевую роль в обработке данных различных областей, таких как спутниковая дистанционная съемка, медицинская диагностика и оборонная сфера.

Для сравнительной оценки были взяты следующие методы: ядро Габора, преобразование Фурье, косинус-преобразование, преобразование Вейвлета. Для оценки частотной декомпозиции каждого метода существует такой параметр как, коэффициент выделения высокочастотных компонентов изображения, отвечающих за детали. Графики показывают (по оси OX – количество шумов, в бит (от 0 до 1024); по оси OУ – коэффициент выделения высоких частот (от 0 до 1)), как этот коэффициент меняется при различных уровнях шума на изображении, что позволяет сравнивать эффективность методов в различных условиях (рисунок 1).

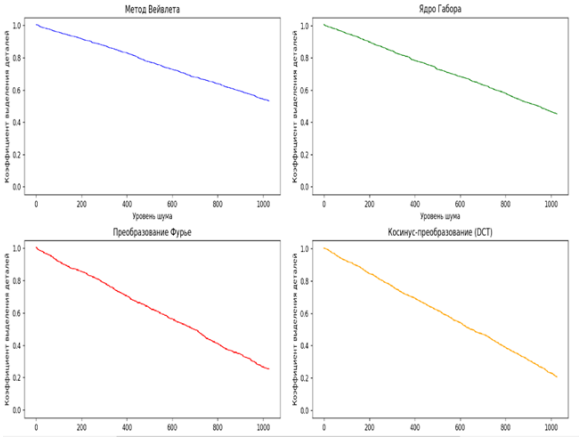


Рисунок 1 – Графическая оценка эффективности методов

Из графика метода Вейвлета видно, что скорость уменьшения коэффициента выделения ниже, чем на других графиках, что может указывать на преобладание этого метода перед остальными.

По результатам моделирования наименьшее количество шумов и пикселей в результате разности наблюдается при использовании метода вейвлет-преобразования. Затем следует метод преобразования Фурье, который также продемонстрировал хорошие результаты. Метод преобразования Косинуса показал немного большее количество артефактов в результате разности. Наконец, метод ядра Габора оказался

наименее эффективным в деле декомпозиции и реконструкции изображений, что отражается в большем количестве шумов и искажений в результате разности. Таким образом, преобразование вейвлета демонстрирует наилучшую производительность, справляясь с задачей декомпозиции и реконструкции изображений наиболее эффективно и точно.

Таким образом, вейвлет-преобразование является мощным инструментом в обработке изображений и находит широкое применение в различных областях, включая медицину, геоинформационные системы, видеообработку и многое другое. Его возможности в анализе и обработке изображений делают его неотъемлемой частью современных технологий.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Кукин, В.А. (2005). Применение вейвлет-преобразований в обработке изображений. Москва: Издательство "Техносфера". ISBN: 978-5-94836-388-92.

2. Добеши, И. (1992). Десять лекций о вейвлетах. Общество промышленности и прикладной математики. ISBN: 978-0898712742

3. Стрэнг, Г., & Нгуен, Т. (1996). Вейвлеты и фильтровые банки. Wellesley-Cambridge Press. ISBN: 978-0961408879

УДК 004.627

С.А. Рябинин, И.Д. Попов, Е.Н. Проказникова

УДАЛЕНИЕ ИМПУЛЬСНЫХ ПОМЕХ С ИЗОБРАЖЕНИЙ, С ПОМОЩЬЮ ВЕЙВЛЕТ-ПРЕОБРАЗОВАНИЯ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

В работе рассматривается использование вейвлет-преобразования для удаления импульсных помех с изображений. Приводятся основные особенности данного метода, его достоинства и недостатки. Подробно обсуждается процесс декомпозиции изображения по частотам с помощью вейвлет-преобразования и роль этого метода в удалении высокочастотных помех. В заключение делается вывод о целесообразности применения вейвлет-преобразования для повышения качества изображений за счет эффективного удаления импульсных помех.

Ключевые слова: вейвлет-преобразование, удаление импульсных помех, декомпозиция изображений по частотам.

Одним из наиболее перспективных направлений в сфере цифровой обработки изображений является обработка изображений методом вейвлет-преобразований. Вейвлет-преобразование в обработке изображений имеет широкий спектр применений: улучшение качества фотографий путем коррекции шумов и артефактов; обработка

спутниковых изображений для выделения различных объектов на поверхности Земли; эффективный анализ медицинских снимков, таких как МРТ или КТ, с целью выявления патологий и улучшения качества диагностики. Метод вейвлет-преобразований является хорошо изученным, однако существует необходимость в поиске новых подходов к его применению. Одним из возможных способов применения является использование этого метода для удаления импульсных помех с изображений. Для этого необходимо сделать декомпозицию изображения на частотные компоненты, что позволяет выделить и удалить шумы и артефакты.

Вейвлет-преобразование разлагает изображение на несколько частотных компонентов: низкочастотные и высокочастотные. Низкочастотные компоненты представляют собой сглаженную версию исходного изображения, получаемую из среднего арифметического соседних пикселей. Высокочастотные компоненты содержат информацию, необходимую для восстановления исходного изображения, представляя собой детали и резкие изменения яркости, которые вычисляются как разности между соседними пикселями.

Для достижения этого разделения на частотные компоненты используется умножение на коэффициент, что соответствует применению фильтра, сглаживающего низкочастотные компоненты и выделяющего высокочастотные.

Математическая формулировка вейвлет-преобразования:

Для одномерного сигнала $\mathrm{f} = (\mathrm{f}_1, \mathrm{f}_2, \dots, \mathrm{f}_\mathrm{N})$, вейвлет-преобразование на первом уровне вычисляется следующим образом:

Низкочастотные компоненты $\mathrm{a} = (\mathrm{a}_1, \mathrm{a}_2, \dots, \mathrm{a}_{\mathrm{N}/2})$:

$$\mathrm{a}_n = \frac{\mathrm{f}_{2\mathrm{n}-1} + \mathrm{f}_{2\mathrm{n}}}{\sqrt{2}}, \qquad n = 1, 2, 3, \dots, N/2$$

Высокочастотные компоненты $\mathrm{d} = (\mathrm{d}_1, \mathrm{d}_2, \dots, \mathrm{d}_{\mathrm{N}/2})$::

$$\mathrm{d}_n = \frac{\mathrm{f}_{2\mathrm{n}-1} - \mathrm{f}_{2\mathrm{n}}}{\sqrt{2}}, \qquad n = 1, 2, 3, \dots, N/2$$

Применение вейвлет-преобразования для обработки изображений:

Для обработки изображений вейвлет-преобразование применяется сначала к строкам, а затем к столбцам изображения. Это позволяет выделить низкочастотные и высокочастотные компоненты для четырех областей изображения: низкочастотная область – А1, горизонтально высокочастотная область – Б1, вертикально высокочастотная область – Б2, диагонально высокочастотная область – Б3. На рис.1 изображено, как эти компоненты разделяются на частотные области: на левом изображении показано разделение на частотные области, на правом изображении показано применение непосредственно на изображении.

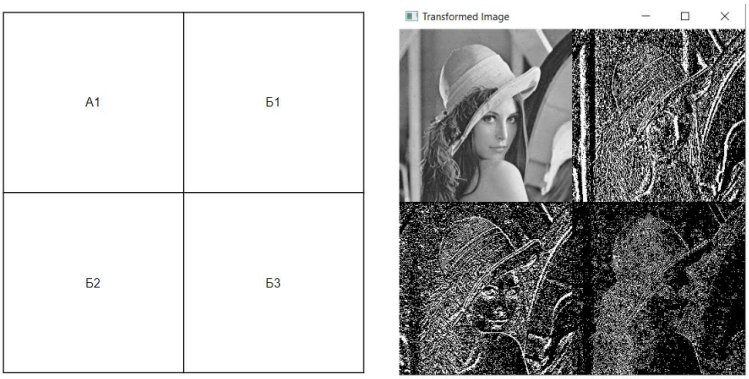


Рисунок 1 – Разделение компонент на частотные области

Удаление импульсных помех достигается путем обнуления диагональных высокочастотных компонентов – Б3, что эффективно устраняет резкие шумы, не влияя на основную структуру изображения. В результате получается более чистое изображение, так как диагональная компонента обнулена, и она не учитывается при реконструкции, что улучшает общее качество изображения.

На рис. 2 можно наблюдать обнуление диагональной высокочастотной составляющей декомпозиции исходного изображения. На декомпозиции справа нижняя правая четверть обнулена.

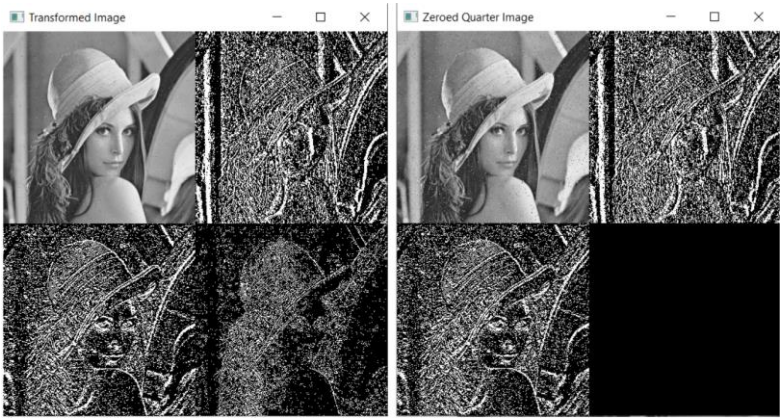


Рисунок 2 – Обнуление высоких частот

На рис. 3 можно наблюдать результат удаления импульсных помех с исходного изображения слева путем обнуления диагональной высокочастотной составляющей, в результате получилось изображение справа. Как видно, некоторая часть помех удалилась.

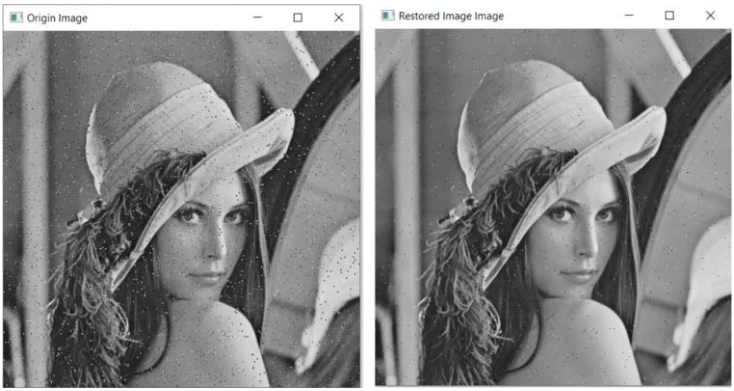


Рисунок 3 – Удаление импульсных помех с изображения

Таким образом, вейвлет-преобразование является мощным инструментом в обработке изображений, особенно для удаления импульсных помех. Его применение находит широкое признание в различных областях, включая медицину, геоинформационные системы, видеообработку и многие другие. Возможности вейвлет-преобразования в анализе и обработке изображений делают его неотъемлемой частью современных технологий.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Малла, С. Г. (1989). Теория мультирезолюционного разложения сигналов: вейвлет-представление. IEEE Transactions on Pattern Analysis and Machine Intelligence, 11(7), 674-693. DOI: [10.1109/34.192463](https://doi.org/10.1109/34.192463)

2. Добеши, И. (1992). Десять лекций о вейвлетах. Общество промышленности и прикладной математики. ISBN: 978-0898712742

3. Стрэнг, Г., & Нгуен, Т. (1996). Вейвлеты и фильтровые банки. Wellesley-Cambridge Press. ISBN: 978-0961408879

УДК 004.021

Е.А. Соколов, Д.А. Кузнецов, Е.Н. Проказникова

ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ АЛГОРИТМОВ СОРТИРОВКИ ДЛЯ ПОЧТИ УПОРЯДОЧЕННЫХ МАССИВОВ СЕТЕВЫХ ПАКЕТОВ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

В данной статье рассматриваются алгоритмы сортировки применительно к почти упорядоченным массивам и предлагается модификация одного из них для повышения эффективности сортировки

Ключевые слова: Алгоритмы сортировки, почти упорядоченные массивы.

Эффективность работы алгоритмов сортировки характеризуется минимальными затратами процессорного времени и памяти. На сегодняшний день существует большое количество алгоритмов сортировки, некоторые из которых позволяют эффективно упорядочивать элементы в массивах в общем случае, когда данные не отсортированы. Однако часто возникает проблема сортировки почти упорядоченных массивов, когда один или несколько элементов расположены не на своем месте. Данная ситуация встречается, например, в следующих случаях: сортировка микропроцессорной системой данных, которые постоянно поступают с различных датчиков; упорядочивание пакетов, принятых по сети, по времени отправки, с учетом того, что информация, хотя чаще всего и приходит в том порядке, в каком она была отправлена, но это не всегда гарантируется [1]; сортировка процессов в очереди заданий в операционных системах, когда пользователь может изменять приоритет существующих задач, снимать и запускать новые процессы, что приводит к постоянному нарушению порядка сортировки данных.

Кроме того, в некоторых системах объем памяти и/или вычислительные ресурсы процессора могут быть серьезно ограничены, либо требуется обрабатывать данные за ограниченное время, поэтому необходимо использовать наиболее эффективные для сортировки методы.

Таким образом, проблема поиска эффективных алгоритмов для сортировки почти упорядоченных массивов сетевых пакетов актуальна.

Для почти упорядоченных массивов сетевых пакетов был проведен сравнительный анализ работы алгоритмов сортировки. Время работы алгоритма для массива конкретной длины вычислялось как среднее арифметическое 100 замеров. Длины массивов, используемые в замерах: 10, 20, 40, 80, 160, 320, 640, 1280, 2560, 5120, 10240.

По результатам моделирования были сделаны выводы алгоритм пузырьковой сортировки самый эффективный для случая обработки отсортированного массива, поскольку во время сортировки происходит проверка упорядоченности массива. Если массив оказывается

упорядоченным, то работа алгоритма завершается. Однако, уже при небольшом нарушении упорядоченности пузырьковая сортировка становится самой неэффективной в большинстве случаев входных данных (кроме массивов, длина которых не превышает десятков элементов).

Сортировка слиянием на всех тестовых данных неэффективна относительно других сортировок (она выполняется быстрее только пузырьковой сортировки в случае обработки почти упорядоченного массива с длиной не менее сотен элементов).

Пирамидальная сортировка эффективнее сортировки слиянием на всех тестовых данных.

Самая эффективная сортировка для почти упорядоченных массивов – Timsort. Данный алгоритм выигрывает в скорости у всех методов сортировки в этом случае. При обработке отсортированных массивов Timsort проигрывает лишь пузырьковой сортировке.

На основании результатов моделирования может быть предложена модификация алгоритма сортировки Timsort.

Поскольку вероятность того, что исходные данные будут представлять уже упорядоченный массив, велика, то для повышения эффективности работы Timsort предлагается в начале его работы проверять упорядоченности исходных данных: если массив уже отсортирован, то завершать работу подпрограммы. Данную проверку можно выполнить за O(n), тогда время работы метода сравняется с таковым у пузырьковой сортировки. В случае обработки почти упорядоченных массивов эффективность работы Timsort несколько ухудшится, но незначительно, поскольку теоретически затраты на проверку исходных данных за O(n) много меньше затрат непосредственно на сортировку массивов, обычно выполняемую за O(n log(n)).

Таким образом, эффективность модифицированного алгоритма Timsort при обработке упорядоченных массивов в пределах погрешности совпадает с эффективностью пузырьковой сортировки, а в случае почти упорядоченных массивов – приближенно равна не модифицированному алгоритму Timsort, поэтому для упорядочивания сетевого буфера рекомендуется применять модифицированный алгоритм Timsort с предварительной проверкой упорядоченности исходных данных.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Столяров А. В. Программирование: введение в профессию. Т.2: Системы и сети. — М.: ДМК Пресс. 2021 — 656 с.

2. Интернет-ресурс neerc.ifmo.ru URL: https://neerc.ifmo.ru/wiki/index.php?title=Сортировки (дата обращения: 14.03.2024)

3. Статья On the Worst-Case Complexity of TimSort Nicolas Auger, Vincent Jugé, Cyril Nicaud, and Carine Pivoteau Université Paris-Est, LIGM (UMR 8049), UPEM, F77454 Marne-la-Vallée, France

УДК 004.4

К. С. Рогов

ИССЛЕДОВАНИЕ АЛГОРИТМА ШИНГЛОВ ДЛЯ СРАВНЕНИЯ ТЕКСТОВ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

В статье рассматривается метод шинглов и его примениние для сравнения текстов. Приводится алгоритмическое описание и пример реализации метода шинглов в контексте сравнения двух текстов между собой.

Ключевые слова: уникальность текста, шингл, длина шингла, контрольные суммы, алгоритм шинглов.

В настоящее время актуальной задачей при создании и публикации каких-либо серьёзных работ ставится задача проверки текста на уникальность. Целью проверки является нахождение сходства или заимствований с другими материалами на аналогичную тематику, а также оценки сходства с этими материалами. В этом контексте предлагается рассмотреть метод шинглов для сравнения текстов между собой.

Говоря простым языком, шингл[1] – это текстовые фрагменты определённой длины, участвующие в процессе определения уникальности текста. Данный метод позволяет с высокой достоверностью определить неуникальные участки, последовательно сверяя их с результатами из интернета или подключаемых модулей Антиплагиата.

Классические приложения и сервисы проверки уникальности используют длину шингла в 5-6 слов. За счёт этого уникальным окажется даже лёгкий рерайт. Чем меньше длина, тем больше придётся трудиться над текстом, переделывая его «от и до».

Можно выделить следующие этапы, через которые проходит текст при его сравнении:

1. Канонизация текста. Это приведение оригинального текста к единой нормальной форме через очищение его от всех вспомогательных единиц текста (предлогов, союзов, знаков препинания, тегов и прочее), которые не должны участвовать в сравнении. Часто предполагается также удаление имен прилагательных, поскольку они, как правильно, несут эмоциональную, а не смысловую нагрузку.

2. Разбиения текста на шинглы. Под длиной шингла понимается количество слов, по которым производится проверка. Сначала берутся первые несколько слов, например, пусть в нашем случае длинна шингла будет составлять 3 слова, начиная с первого по третье. Затем они проверяются на наличие аналогичных последовательностей. На следующем шаге берётся следующий шингл – со второго по четвёртое слово. Следующий шингл включает слова с третьего по пятой, следующий – с четвёртого по шестой.

Таким образом, чтобы шингл считался уникальным, уникальным должно быть каждое n-е слово в проверяемом документе, где n- длина шинга. В нашем примере уникальным должно быть каждое 3-е слово.

3. Вычисления, через статические функции, 84-х хэшей шинглов. Текст представляется в виде таблиц с набором контрольных сумм, рассчитанных для каждого шингла по 84-м статическим хэш-функциям.

Из обоих наборов случайным образом отбираются 84 значения – для каждого из документов – и сравниваются в соответствии с функциями своей контрольной суммы. Иными словами, потребуется 84 операции, чтобы сравнить тексты.

Важное примечание: завышенное число снижает производительность, значительно увеличивая число операций.

4. Случайной выборки значений 84 контрольных сумм. Для увеличения производительности при сравнении элементов каждого из 84-х выбранных массивов нужно произвести случайную выборку контрольных сумм для каждой из строк. Выбор минимального значения из каждой строки в итоге даст набор наименьших значений контрольных сумм шинглов для каждой из хэш функций.

5. Сравнения и определения результата. Сравнение каждого из 84 элементов обоих документов выявляет соотношение одинаковых значений, что позволяет определить уровень идентичности, или уникальности каждого из текстов. Таким образом, если количество неуникальных шинглов мало, то уникальность считается высокой.

Ниже приведён фрагмент кода, соответствующий реализованному методу Шинглов. Реализация состоит из создания отдельного класса с методами, которые используются для сравнения текстов.

Класс ShingleMethod содержит константы STOP_SYMBOLS и STOP_WORDS_RU, они необходимы для канонизации текста, которая выполняется методом canonize. Метод получает на вход строку и на выходе уже выдаёт приведённую к каноническому виду строку. Метод genShingle выполняет разбивает текст на шинглы, а затем вычисляет их контрольные суммы. На выходе мы получает массив контрольных сумм шинглов. Метод compare предназначен для сравнения двух массивов контрольных сумм, которые получаются как раз посредством применения к двум текстам метода genShingle.

```
import java.util.ArrayList;

public class Shingle {
    private static final String STOP_SYMBOLS[] = {".", ",", "!", "?", ":", ";", "-", "\\", "/", "*", "(", ")", "+", "@",
            "#", "$", "%", "^", "&", "=", "'", "\"", "[", "]", "{", "}", "|"};
    private static final String STOP_WORDS_RU[] = {"это", "как", "так", "и", "в", "над", "к", "до", "не", "на", "но", "за",
            "то", "с", "ли", "а", "во", "от", "со", "для", "о", "же", "ну", "вы",
            "бы", "что", "кто", "он", "она"};

    private String canonize(String str) {
        for (String stopSymbol : STOP_SYMBOLS) {
            str = str.replace(stopSymbol, "");
        }

        for (String stopWord : STOP_WORDS_RU) {
            str = str.replace(" " + stopWord + " ", " ");
        }

        return str;
    }

    public ArrayList<Integer> genShingle(String strNew, int ShingleLen) {
        ArrayList<Integer> shingles = new ArrayList<Integer>();
        String str = canonize(strNew.toLowerCase());
        String words[] = str.split(" ");
        int shinglesNumber = words.length - ShingleLen;

        //Create all shingles
        for (int i = 0; i <= shinglesNumber; i++) {
            String shingle = "";

            //Create one shingle
            for (int j = 0; j < ShingleLen; j++) {
                shingle = shingle + words[i+j] + " ";
            }

            shingles.add(shingle.hashCode());
        }

        return shingles;
    }

    public double compare(ArrayList<Integer> textShingles1New, ArrayList<Integer> textShingles2New) {
        if (textShingles1New == null || textShingles2New == null) return 0.0;

        int textShingles1Number = textShingles1New.size();
        int textShingles2Number = textShingles2New.size();

        double similarShinglesNumber = 0;

        for (int i = 0; i < textShingles1Number; i++) {
            for (int j = 0; j < textShingles2Number; j++) {
                if (textShingles1New.get(i).equals(textShingles2New.get(j))) similarShinglesNumber++;
            }
        }

        return ((similarShinglesNumber / ((textShingles1Number + textShingles2Number) / 2.0)) * 100);
    }
}
```

Рисунок 1 – Пример реализации метода шинглов

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Бабаков А., Исследование и разработка методов автоматического поиска заимствований в русскоязычных текстах, 04.06.2014. http://sv-journal.org/2015-2/05/ru/index.php?lang=ru (Дата обращения 15.12.2022);

2. Шингл в Антиплагиате - что это такое? https://xn----8sbaagj5acc1auhmhyrdh0nxa.xn--p1ai/shingl-v-antipagiate (Дата обращения 15.12.2022);

УДК 004.021

К. С. Рогов

СУЩЕСТВУЮЩИЕ СПОСОБЫ ПОИСКА ЗАИМСТВОВАНИЙ В ТЕКСТЕ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

В статье рассматриваются существующие способы обработки текстов и поиска заимствований в тексте. Даётся краткое описание каждого способа и его преимущества и недостатки в сравнении с другими способами.

Ключевые слова: заимствования в тексте, плагиат, метод шинглов, метод супершинглов, метод опорных слов, метод I-Match.

В современное время поиск заимствований в текстах применяется для следующих задач:

- определение оригинальности текста;
- проверка на наличие плагиата;
- поиск источников, использованных для создания материала.

Поиск заимствований в текстах сопряжён с непосредственной обработкой данных текстов для определения процента совпадений. Данная тема довольно актуальна в современном времени, методы обработки текстов совершенствуются, в частности и для определения заимствований в текстах. Далее будут рассмотерны существующие способы обработки текстов.

Метод шинглов

Говоря простым языком, шингл[1] – это текстовые фрагменты определённой длины (обычно 5-6 слов), участвующие в процессе определения уникальности текста. Данная поисковая технология используется во всех сервисах и приложениях. Метод Шинглов позволяет с высокой достоверностью определить неуникальные участки, последовательно сверяя их с результатами из интернета или подключаемых модулей Антиплагиата. Чем меньше длина, тем больше придётся трудиться над текстом, переделывая его «от и до».

Алгоритм шинглов неумолим, из-за чего студентам приходится выжимать из себя все соки, чтобы сделать курсовую, диплом, реферат или диссертацию максимально уникальной – в среднем требуется не менее 80%, для рефератов планка снижается до 50-70%. Зато от диссертаций требуется уникальность не менее 80-90%.

При сравнении текст проходит через следующие этапы:

1. Канонизация текста.
2. Разбиения текста на шинглы.
3. Вычисления, через статические функции, 84-х хэшей шинглов.

4. Случайной выборки значений 84 контрольных сумм.

5. Сравнения и определения результата.

Метод супершинглов

Данный метод является модификацией метода шинглов. Строится ограниченный набор контрольных сумм. Над 84 шинглами строится 6 супершинглов (по 14 шинглов). Тексты считаются совпавшими при совпадении хотя бы двух супершинглов из 6.

На практике метод, что для определения нечетких дубликатов хватает минимум двух совпадений супершинглов. То есть для наилучшего поиска совпадений как минимум двух супершинглов, документ должен быть представлен различными попарными соединениями из 6 супершинглов. Такие группы называют мегашинглами. Число таких мегашинглов равно 15 (число сочетаний из 6 по 2). Два документа сходны по содержанию, если у них совпадает хотя бы один мегашингл.

Описанный метод применим для отбора документов для поиска, если критерием отбора документов является масштабные заимствования, для поиска конкретных заимствованных фрагментов алгоритм не применим, так как мегашингл содержит лишь небольшое подмножество шинглов документа, и, соответственно, не может содержать информацию о всех возможных заимствованиях.

Метод I-Match

В методе I-Match [2] создаѐтся словарь L, который включает слова со средними значениями IDF, поскольку такие слова обеспечивают, как правило, более точные результаты при обнаружении заимствований. Слова с большими и маленькими значениями IDF, которые представляют собой очень редкие, или же наоборот, часто встречающиеся слова, отбрасываются. Затем для каждого документа формируется множество U различных слов, входящих в него, и определяется пересечение U и словаря L. Если размер этого пересечения больше некоторого минимального порога (определяемого экспериментально), то список слов, входящих в пересечение упорядочивается, и для него вычисляется I-Match сигнатура (хеш-функция SHA1).

Два документа считаются похожими, если у них совпадают I-Match сигнатуры (имеет место коллизия хеш кодов). Алгоритм имеет высокую вычислительную эффективность. Основной недостаток — неустойчивость к небольшим изменениям содержания документа.

Алгоритм не применим для определения конкретных заимствований, а также при модификации слов, входящих в словари, алгоритм не является устойчивым, что приводит к уменьшению точности.

Метод опорных слов

Метод "Опорных слов" [3] является модификацией предыдущего метода. Сначала из множества документов по выбирается множество из N слов (N — определяется экспериментально), называемых "опорными".

Затем каждый документ представляется N-мерным двоичным вектором, где i-я координата равна 1, если i-е "опорное" слово имеет в документе относительную частоту выше определенного порога (устанавливаемого отдельно для каждого "опорного" слова), и равна 0 в противном случае. Этот двоичный вектор называется сигнатурой документа. Соответственно, два документа считаются идентичными при совпадении сигнатур (или очень близки, близость считается по расстоянию Хэмминга).

При построении множества «опорных» слов используются следующие соображения:

1. Множество слов должно охватывать максимально возможное число документов.

2. Число слов в наборе должно быть минимальным.

3. При этом «качество» слова должно быть максимально высоким.

Благодаря этому алгоритму, можно отфильтровать несколько тысяч слов и оставить только 3-5 тысяч.

Алгоритм не применим для определения конкретных заимствований, но позволяет выделить похожие документы в случае их существенного совпадения. При модификации "опорных" слов также не является устойчивым, что приводит к уменьшению полноты.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Бабаков А., Исследование и разработка методов автоматического поиска заимствований в русскоязычных текстах, 04.06.2014. http://sv-journal.org/2015-2/05/ru/index.php?lang=ru (Дата обращения 15.12.2022);

2. Чеусов А.В. АЛГОРИТМ СИНТАКСИЧЕСКОГО АНАЛИЗА ТЕКСТА ДЛЯ ОБРАБОТКИ БОЛЬШИХ ОБЪЕМОВ ДАННЫХ. Информатика. 2007;(1(13)):с. 98-105.

3. Метод «опорных» слов, https://studbooks.net/2253122/informatika/match_metod (Дата обращения 18.12.2022)

УДК 004.78

М.А. Садовников, Г.Д. Рукоделов

РАЗРАБОТКА И СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ МОНИТОРИНГА И ЛОГИРОВАНИЯ ИНФОРМАЦИОННО – ТЕЛЕКОММУНИКАЦИОННЫХ СЕТЕЙ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

Рассмотрены современные методы эффективного анализа сетевых пакетов информационно-телекоммуникационных сетей.

Ключевые слова: мониторинг, сетевой пакет, сетевой трафик, протокол.

Исследование методов мониторинга компьютерной сети является актуальной задачей в условиях быстрого развития информационных технологий и увеличения угроз информационной безопасности. В данной статье рассматривается необходимость разработки эффективных методов мониторинга сетевой активности с целью обнаружения аномалий, определения уязвимостей и обеспечения бесперебойной работы IT-инфраструктуры. Суть работы заключается в анализе современных подходов к мониторингу компьютерных сетей с учетом их применимости в условиях реальных бизнес-процессов и конкретных сценариев использования.

Данное исследование также предполагает обзор известных методов мониторинга компьютерных сетей и анализ их недостатков, таких как ограниченная масштабируемость, недостаточная точность обнаружения угроз и сложность внедрения в сложные корпоративные среды. Понимание данных недостатков позволит выявить потенциальные области улучшения существующих методов и разработать более эффективные подходы к мониторингу компьютерных сетей.

Методы мониторинга сети

В современных технологиях используются различные методы мониторинга сети. Самыми популярными считаются: метод мониторинга по протоколу SNMP, метод мониторинга по протоколу ICMP и пакетный анализ.

SNMP (Simple Network Management Protocol) является протоколом сетевого управления, который позволяет удаленно анализировать состояние сетевых устройств, управлять ими и собирать информацию о их работе. По сути, SNMP-мониторинг представляет собой сбор и анализ данных сетевой активности с использованием протокола SNMP.

Функционал SNMP-мониторинга включает в себя:

1. Сбор информации о сетевых устройствах: (мониторинг состояния устройств путем считывания параметров).
2. Обнаружение и реагирование на события: (реагирование на изменения и события в сети).
3. Удаленное управление: (возможность управления сетевыми устройствами через SNMP).
4. Автоматизация мониторинга: (использование SNMP для автоматического сбора данных).

Преимущества данного метода заключаются в его использовании широким спектром устройств, способностью анализировать различные параметры, такие как нагрузка процессора, использование памяти и сетевой трафик, а также доступе к данным, который происходит через агентов, установленных на сетевых устройствах.

Недостаток данного метода заключается потреблении большого количества ресурсов сети при активном мониторинге большого количества устройств.

SNMP-мониторинг обеспечивает ценные данные для администраторов сети, позволяя им получать информацию о состоянии сетевых устройств и оперативно реагировать на различные события и проблемы в сети.

Активный мониторинг с использованием протокола ICMP (Internet Control Message Protocol) — это метод непрерывного контроля сетевых узлов и оборудования путем отправки тестовых запросов и анализа ответов. Этот подход позволяет администраторам сети отслеживать доступность узлов, оценивать качество соединения, обнаруживать потенциальные проблемы и отслеживать производительность сети.

Функционал ICMP-мониторинга включает в себя:

1. Проверка доступности узлов: ICMP используется для проверки доступности сетевых узлов путем отправки эхо-запросов (ping) и получения эхо-ответов. Это позволяет определить, работает ли удаленный узел и доступен ли он для обмена данными.

2. Измерение времени отклика: ICMP позволяет измерять время, требуемое для отправки и получения пакетов между узлами. Это помогает оценить задержку (ping) и качество соединения между устройствами в сети.

3. Передача сообщений об ошибках и уведомления: ICMP используется для отправки сообщений об ошибках, таких как недоступность узла или невозможность доставки пакета. Это помогает в диагностике сетевых проблем и уведомляет об изменениях состояния узлов в сети.

4. Диагностика сетевых проблем: ICMP обеспечивает возможность отслеживания и мониторинга состояния сети путем анализа сообщений об ошибке и установлении связи с удаленными узлами.

Преимущества данного метода заключаются в способности проверки доступности сетевых узлов и определении задержек и потерь пакетов.

Недостатки данного метода - ограничение функциональности и предоставление подробной информации о ресурсах устройств.

ICMP является важной частью сетевых протоколов и обеспечивает ключевые функции для обеспечения работы и мониторинга сетевой инфраструктуры.

Пакетный анализ – это метод мониторинга компьютерной сети, который включает в себя анализ сетевых пакетов данных для изучения, контроля и оптимизации работы сети. В научной статье, пакетный анализ рассматривается как важнейший инструмент для обеспечения

безопасности, производительности и эффективного управления современными компьютерными сетями.

Функционал пакетного анализа включает в себя:

1. Захват сетевых пакетов: пакетный анализатор записывает и анализирует сетевые пакеты, проходящие через сетевое устройство или точку доступа. Эти пакеты могут содержать информацию о трафике, коммуникации между устройствами и различные протоколы.

2. Анализ протоколов: пакетный анализатор позволяет интерпретировать данные сетевых пакетов и анализировать протоколы, используемые в сети. Это включает распознавание протоколов передачи данных, структуру пакетов и информацию о заголовках.

3. Отслеживание трафика: с помощью пакетного анализатора можно отслеживать и анализировать сетевой трафик, определять объем передаваемых данных, идентифицировать пакеты с определенными характеристиками и выявлять паттерны поведения.

4. Инспекция безопасности: пакетный анализатор позволяет обнаруживать потенциальные угрозы безопасности в сети, такие как вредоносные программы, атаки, аномальный трафик и уязвимости в системах.

5. Диагностика сетевых проблем: путем анализа сетевых пакетов можно быстро выявлять и определять причины сетевых проблем, таких как низкая производительность, задержки, потери пакетов и конфликты в сети.

Преимущества данного метода заключаются в обеспечении подробной информации о трафике, протоколах и параметрах сети. Также он позволяет проводить анализ и отладку сложных проблем сетевой инфраструктуры.

Недостаток данного метода – требование специализированных средств и навыков для анализа и интерпретации данных.

Пакетный анализ является мощным инструментом для мониторинга и анализа сетевой активности, обеспечивая подробное понимание работы сетевой инфраструктуры и выявляя потенциальные проблемы.

Исследование методов мониторинга

Исследование выполнялось на одной и той же сети, без перебоев и ошибок.

В процессе исследования методов мониторинга сети было выделено три важных параметра, от которых зависит эффективность мониторинга.

Пропускная способность – параметр, отражающий объем данных, который может быть передан через сеть за определенный период времени. В нашем случае данные - это сетевые пакеты. Чем выше пропускная способность выше, тем больший объем данных метод может обработать.

Основные параметры для подсчета пропускной способности: промежуток времени, количество пакетов, средний размер пакета.

Для подсчета пропускной способности будем использовать следующую формулу:

$$Tp = \frac{\sum_{i=1}^{n} size_i}{t}, \quad (1)$$

где Tp – пропускная способность, size – размер одного пакета (бит), n – количество пакетов, t – время (с).

При исследовании системы мониторинга, использующей протокол SNMP, были получены следующие данные:

Промежуток времени: 20 с;

Количество пакетов: 7;

Сумма размера каждого пакета: 10840 бит.

Тогда по формуле (1) получаем значение пропускной способности Tp = 542 б/с.

При исследовании системы мониторинга, использующей протокол ICMP, были получены следующие данные:

Промежуток времени: 3600 с;

Количество пакетов: 2812;

Сумма размера каждого пакета: 1 760 400 бит.

Тогда по формуле (1) получаем значение пропускной способности Tp = 489 б/с

При исследовании системы мониторинга, использующей Пакетный анализ, были получены следующие данные:

Программа вывела значение пропускной способности Tp = 532 б/с.

По результатам оценки видно, что мониторинг по протоколу SNMP работает лучше, чем остальные методы.

Задержка - представляет собой время, необходимое для передачи данных от отправителя к получателю через сеть. Она измеряется в миллисекундах и влияет на производительность сетевых приложений и общее восприятие скорости Интернет-соединения.

Для подсчета пропускной способности будем использовать следующую формулу:

$$del_{avg} = \frac{1}{n}\sum_{i=1}^{n} del_{i,} \quad (2)$$

где del_{avg} – среднее время задержки (ms), del – время задержки в конкретный момент времени, n – количество временных участков.

При исследовании системы мониторинга, использующей протокол SNMP, были получены следующие графические данные (рисунок 1).

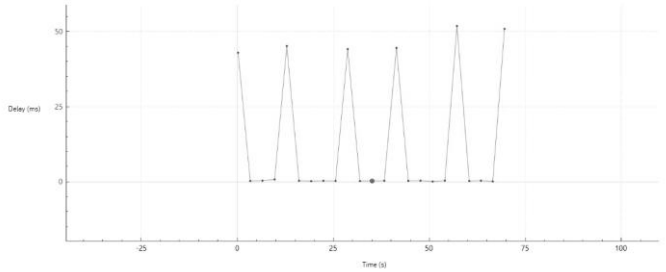


Рисунок 1 – Задержка, протокол SNMP

По формуле (2) вычислим среднее значение задержки за 70 с:
del_{avg} = 46.7 (ms)

При исследовании системы мониторинга, использующей протокол ICMP, были получены следующие графические данные (рисунок 2).

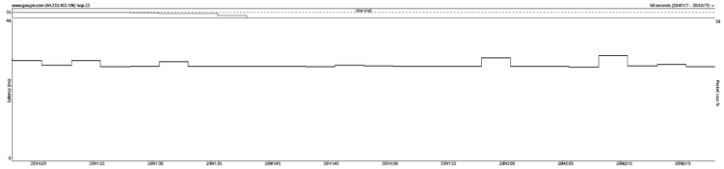


Рисунок 2 – Задержка, протокол ICMP

По формуле (2) вычислим среднее значение задержки за 55 с:
del_{avg} = 27,1 (ms)

При исследовании системы мониторинга, использующей Пакетный анализ, были получены следующие графические данные (рисунок 3).

Рисунок 3 – Задержка, Пакетный анализ

По формуле (2) вычислим среднее значение задержки за 55 с:
del_{avg} = 24.34 (ms)

По результатам оценки видно, что задержка при мониторинге Пакетным анализом меньше, чем остальные методы. Это значит, что сетевые пакеты передаются быстрее чем в остальных методах.

Потоковая передача - непрерывное поступление данных к мониторинговой системе без значительных задержек или прерываний. Потоковая передача позволяет эффективно и непрерывно анализировать сеть, обеспечивая оперативный анализ состояния сети и выявление потенциальных проблем.

Для подсчета потоковой передачи будем использовать следующую формулу:

$$Str = 8 * \sum_{i=1}^{n} \frac{len_i}{t_i}, \quad (3)$$

где Str – потоковая передача (байт/с), len_i – длина i-го пакета (байт), t_i – время доставки пакета.

При исследовании системы мониторинга, использующей протокол SNMP, были получены следующие данные:

Общее время: 291 (ms)

Сумма длин всех пакетов: 1092 (байт).

Тогда по формуле (3) получаем значение потоковой передачи Str = 30 б/с.

Протокол ICMP не предусматривает сохранение и демонстрацию пакетов, что является минусом метода.

При исследовании системы мониторинга, использующей Пакетный анализ, были получены следующие данные:

Общее время: 217 (ms)

Сумма длин всех пакетов: 1229 (байт).

Тогда по формуле (3) получаем значение потоковой передачи Str = 45.3 б/с.

По результатам оценки (таблица 1) видно, что при мониторинге с помощью Пакетного анализа данные передаются быстрее и более непрерывно, чем в остальных методах.

Таблица 1 – Результаты оценки методов

	SNMP	ICMP	Пакетный анализ
Пропускная способность (б/с)	542	489	532
Задержка (ms.)	46.7	27,1	24.34
Потоковая передача (б/с)	30	-	45.3

Заключение

Проведенное исследование методов мониторинга SNMP, ICMP и пакетного анализа позволяет понять, что каждый из них представляет ценный инструмент для обеспечения надежности, производительности и безопасности сетевых инфраструктур. Комбинированное использование этих методов может обеспечить комплексное покрытие сети и повысить эффективность мониторинга. Дальнейшие исследования могут быть

направлены на оптимизацию и совместное применение этих инструментов для повышения качества управления и поддержки сетевых систем.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Груздев Д. А., Закалкин П.В., Кузнецов С.И., Тесля С.П. Мониторинг информационно-коммуникационных сетей /Груздев Д. А., Закалкин П.В., Кузнецов С.И., Тесля С.П – М.: 2014. № 3. С. 80–86

2. Злобина Н.В., Волжанкин Н.В., Пособилов Н.Е. Обеспечение централизованного мониторинга для систем сложной архитектуры с большим объёмом данных / Злобина Н.В., Волжанкин Н.В., Пособилов Н.Е. – М.: 2014. № 1 (5). С. 95–101

3. Ениватов А.А. Мониторинг трафика локальных сетей / Ениватов А.А. – М.: № 23. 13 с.

УДК 519.688

О. Д. Саморукова, А. В. Крошилин

ИСПОЛЬЗОВАНИЕ РЕГУЛЯРНЫХ ВЫРАЖЕНИЙ ПРИ РЕШЕНИИ ЗАДАЧИ ИЗВЛЕЧЕНИЯ ИНФОРМАЦИИ ИЗ НЕСТРУКТУРИРОВАННЫХ ТЕКСТОВЫХ ДАННЫХ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

В рамках разработки системы поддержки принятия решений при подборе схем медикаментозного лечение решается задача автоматического извлечения ключевой информации о лекарственных средствах из инструкций по их применению. В данной статье рассматривается возможность использования регулярных выражений для решения описанных задач.

Ключевые слова: системы поддержки принятия решений, алгоритмы извлечения текстовой информации, регулярные выражения.

Корректное извлечение требуемой информации является важнейшим вопросом построения системы поддержки принятия решений [1, 3]. В виду большого количества слабоструктурированных исходных данных необходимо разработать систему, способную быстро и качественно извлекать требуемую для дальнейшей работы информацию. Учитывая, что разрабатываемая система направлена на решение задач в области медицины, алгоритмы ее работы должны быть прозрачны, понятны и иметь возможность оперативного внесения корректировок, при необходимости. В связи с этим, выбор сделан в пользу инженерных методов, основанных на правилах [2, 4].

Существует достаточное количество способов, позволяющих осуществлять различные манипуляции, в том числе и извлечение данных из текстового контента. Регулярные выражения можно назвать фундаментальным и незаменимым инструментом для решения подобных задач. Регулярные выражения, называемые RegEx, предлагают универсальное эффективное решение для различных программных приложений. Они представляют собой надежные шаблоны, которые могут использоваться для поиска, сопоставления текстовых данных и манипулирования ими. Регулярные выражения играют жизненно важную роль при выполнении операций проверки текста и его извлечении, предлагая мощные функциональные возможности для эффективного поиска и замены определенных шаблонов в текстовой строке. С их помощью можно обрабатывать файлы журналов, извлекать отдельные определенные данные из текстовых документов, перенаправлять URL-адреса, проверять вводимый пользователем текст в веб-формах и др. [7]

Регулярное выражение работает путем определения шаблона, представляющего определенный набор символов или последовательностей. Созданный шаблон применяется к целевому тексту с целью выявления совпадений или выполнения каких-либо преобразований.

Выполним краткий обзор синтаксиса регулярных выражений с указанием основных компонентов и наиболее часто используемых операций:

Литералы: Регулярные выражения могут состоять из буквенных символов, которые точно соответствуют самим себе. Например, шаблон “hello" будет соответствовать строке “hello” в целевом тексте.

Модификаторы: модификаторы определяют дополнительные правила сопоставления. Распространенные модификаторы включают:

i: Сопоставление без учета регистра.

g: Глобальное сопоставление (сопоставляет все вхождения, а не останавливается на первом совпадении).

m: Многострочное сопоставление.

Привязки: привязки используются для указания положения совпадения в тексте. Например: ^ (курсор): соответствует началу строки, $ (знак доллара): соответствует концу строки.

Метасимволы: специальные символы с особым значением в регулярном выражении.

Таблица 1 – Примеры метасимволов

Метасимвол	Описание	Пример
[]	Любой набор символов, например, a-z или a-f	[a-z]
\	Сигнализирует о специальной последовательности (например, \d) или используется для экранирования специального символа	\ d
^	Начинается с	^www.
$	Заканчивается на	.com$
*	Подстановочный знак, соответствующий нулю или более вхождений	
+	Одно или несколько вхождений	
{3}	Определенное количество вхождений, например, три цифры	\d{3}
|	Оператор Или	

Специальные последовательности: предназначены для поиска одного или нескольких символов определенного типа.

Таблица 2 – Примеры специальных последовательностей

Специальная последовательность	Описание
\ d	Любая десятичная цифра. Эквивалентно [0-9].
\ D	Любая десятичная цифра. Эквивалентно [^ 0-9]
\ s	Любой пробельный символ, такой как пробел, табуляция или перенос строки. Эквивалентно [\t \ n \ r \ f \v]
\ S	Любой символ, не содержащий пробелов, такой как 1, J или £. Эквивалентно [^ \t \ n\ r \ f \v]
\ w	Любой буквенно-цифровой символ, такой как 1 или L. Эквивалентно [a-zA-Z0-9_]
\ W	Любой не буквенно-цифровой символ, такой как ; или !. Эквивалентно [^a-zA-Z0-9_]

Наборы: предназначены для поиска определенных комбинаций символов.

Таблица 3 – Примеры наборов

Установить	Описание
[a-zA-Z]	Любые заглавные или строчные буквы
[0-9]	Любая десятичная цифра. Эквивалентно [^ 0-9 Любые цифры от 0 до 9]

Объединение этих компонентов и операций позволяет создавать сложные и мощные шаблоны для поиска, проверки достоверности и преобразования текстовых данных с использованием регулярных выражений [6].

Рассмотри какие задачи могут быть решены с применением регулярных выражений и приведем некоторые примеры.

Проверка (валидация) данных – процесс проверки соответствия входных строковых данных установленному формату или набору критериев. Проведение валидации гарантирует, что введенные, либо импортированные из каких-либо источников данные являются точными, целостными и соответствуют заданным требованиям.

Поиск и замена – процесс, помогающий находить в тексте шаблоны и заменять их иными необходимыми данными.

Извлечение данных – один из распространенных вариантов использования регулярных выражений, который позволяет с использованием шаблонов извлекать нужную информацию из больших текстов.

Частными и наиболее простыми примерами извлечения данных является извлечение хештэгов #, упоминаний @ и чисел. Рассмотрим пример, на языке Phyton (рисунок 1)

```
import re


def extract_hashtags(text):
    regex = "#(\w+)"
    return re.findall(regex, str(text))


string = 'Data science is really interesting #datascience #python'
hashtags = extract_hashtags(string)
hashtags
```

Рисунок 1 – фрагмент кода (извлечение хештэга)

В приведенном примере загружен модуль re и написана простая функция с именем extract_hashtags(), которая принимает текстовую строку в качестве аргумента и возвращает список Python, содержащий любые хэштеги, которые он находит с помощью findall() функции re.

Чтобы упростить просмотр самого регулярного выражения, регулярное выражение #(\w+) присвоено переменной и передано в качестве аргумента. Это выражение, сообщает re для поиска непрерывных

строк, которые начинаются с # и за которыми следует строка unicode (\w+), которая может содержать буквы от A до Z, цифры от 0 до 9 или знак подчеркивания. Для поиска упоминаний используется регулярное выражение @(\w+), а для поиска чисел [0-9]+.

Рассмотрим более сложный пример: извлечение даты и времени из журнала (рисунок 2).

```
import re

log = '2022-12-24T12:34:56: This is a log message.'

# define a pattern to match the date and time
dt_pattern = r'(\d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2})'

# search for the date and time
match = re.search(dt_pattern, log)
if match:
    dt = match.group(1)  # get the date and time from the match

print(dt)  # Output: 2022-12-24T12:34:56
```

Рисунок 2 – фрагмент кода (извлечение даты и времени)

В этом примере dt_pattern - это регулярное выражение, которое определяет шаблон для сопоставления даты и времени в log строке. search() Функция выполняет поиск в log строке первого вхождения шаблона и возвращает Match объект, если найдено совпадение. Можно использовать group() метод Match объекта, чтобы получить дату и время из совпадения.

Парсинг логов – по сути, это также процесс извлечения информации, но в качестве исходных данных используются логи, из которых могут извлекаться данные об ошибках, действиях пользователей и др. Данная функция полезна для устранения проблем и угроз безопасности.

Очистка данных – процесс предварительной обработки данных, используемый для удаления ненужных для дальнейшей работы символов, обеспечения согласованности и форматирования текста.

Рассмотри пример очистки данных, реализованный на языке JavaScript (рисунок 3,4)

Пусть имеется набор данных с описанием каких-то продуктов, содержащий ненужные символы, которые необходимо удалить.

```
const dataset = [
 "Product A - $99.99",
 "Product B: 50% off!",
 "Product C - *Limited Stock*",
 "Product D (New Arrival)"
];

const cleanPattern = /[^a-zA-Z0-9\s]/g;
const cleanedData = dataset.map(description => description.replace(cleanPattern, ""));

console.log("Cleaned Data:");
console.log(cleanedData);
```

Рисунок 3 – Фрагмент кода (очистка данных)

Входящий массив данных – dataset, содержит лишние для дальнейшей работы символы. Объявляется переменная cleanPattern как шаблон регулярного выражения, который соответствует любым символам, кроме буквенных, цифровых и пробелов, для очистки данных. Любой нежелательный специальный символ (любой символ, который не является буквенно-цифровым символом) будет захвачен шаблоном [^a-zA-Z0-9s].

Каждое описание повторяется с использованием map() метод в массиве “dataset”. Затем используется replace() метод и cleanPattern заменить нежелательные символы пустой строкой.

В итоге, получается следующий массив данных.

```
Cleaned Data:
[
 "Product A  99.99",
 "Product B 50 off",
 "Product C Limited Stock",
 "Product D New Arrival"
]
```

Рисунок 4 – Результат «очистки данных»

Итак, регулярные выражения – это универсальный инструмент, доступный к использованию во всех современных языках программирования, позволяющий решать практически все задачи, которые могут возникнуть при работе с текстовыми данными. Необходимо иметь в виду, что разработка хорошего регулярного выражения, как правило, требует итераций, а качество и надежность повышаются по мере того, как в него добавляются новые интересные

данные, включающие крайние случаи. Использование регулярных выражений как часть широкой системы или для ненадежных наборов данных возможно при условии уверенности в его надежности, отказоустойчивости и производительности. Однако, при условии построения корректных шаблонов с привлечения такого инструмента, как регулярные выражения, формируется логичная и предсказуемая система обработки информации, позволяющая существенно экономить время на проведение требуемых манипуляций и получать гарантированный результат [5].

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Жулева С.Ю., Крошилин А.В., Крошилина С.В. Разработка системы поддержки принятия решений для организации рабочего времени медицинского работника на основе методов искусственного интеллекта // Биомедицинская радиоэлектроника. 2023. Т. 26. № 3. С. 55-60.

2. Крошилин А.В. Предметно-ориентированные информационные системы: учебное пособие / А.В. Крошилин, С.В. Крошилина, Г.В. Овечкин. — Москва: КУРС, 2023. — 176 с. — (Естественные науки).

3. Саморукова О.Д., Крошилин А.В., Крошилина С.В., Ключевые аспекты разработки системы поддержки принятия решений при подборе схемы медикаментозного лечения // Биотехнические, медицинские и экологические системы, измерительные устройства и роботехнические комплексы - Биомедсистемы-2023 [текст]: сб. тр. XXXVI Всерос. науч.-техн. конф. студ., мол. ученых и спец., 6-8 декабря 2023 г. / под общ. ред. В.И. Жулева. - Рязань: ИП Коняхин А.В. (Book Jet), 2023. – 332с., ил. (181-184)

4. О.Д. Саморукова, А.В. Крошилин. Способы автоматической обработки входящей информации при подборе схемы медикаментозного лечения // Современные технологии в науке и образовании – СТНО-2024 [текст]: сб. тр. VII междунар. науч.-техн. форума: в 10 т. Т.4./ под общ. ред. О.В. Миловзорова. – Рязань: Рязан. гос. радиотехн. ун-т, 2024

5. James Buchanan. Regex parsing: Using regular expressions to extract data from your logs // URL https://newrelic.com/blog/how-to-relic/extracting-log-data-with-regex 2023

6. Matt Clarke. How to use Python regular expressions to extract information // URL: https://practicaldatascience.co.uk/data-science/how-to-use-python-regular-expressions-to-extract-information 2021

7. Emmanuel Oluwole. Five Practical Use Cases For Regular Expressions // URL: https://blog.openreplay.com/five-practical-use-cases-for-regular-expressions/ 2023

УДК 004.622

И. А. Семиков, Г. В. Овечкин

ПРОБЛЕМАТИКА НАБОРОВ БОЛЬШИХ ДАННЫХ ПРИ РАЗРАБОТКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

В статье рассматривается понятие больших данных, их значимость. Представлены основные характеристики больших данных. Также в работе рассмотрены методы обработки данных. И представлен разбор некоторых основных проблем в обработке больших данных.

Ключевые слова: большие данные, рост объемов данных, методы обработки больших данных, проблемы анализа больших данных.

Большие данные – наборы структурированных и неструктурированных данных огромных объемов. Также данные наборы обычно обладают значительным многообразием по виду данных [1].

Появление термина «большие данные» связано с огромным ростом объемов информации в современном мире. Данные могут быть получены из любой сферы деятельности человека. Анализ различной информации способен привести к различным улучшениям. [2]. Исследование потребностей и действий человека способно выявить его потребности и нужно, а также как лучше это сделать. Большие данные предоставляют возможность лучше понимать современный мир.

К основным характеристикам больших данных можно отнести следующие критерии (рисунок 1).

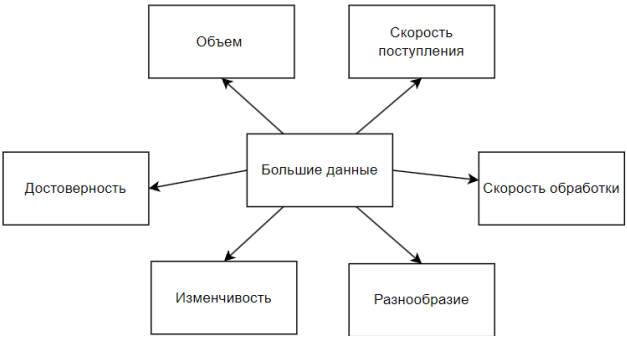


Рисунок 1 – Характеристики больших данных

1. Объем данных. Размерность поступивших новых данных за определенный период времени. Эти наборы данных требуется хранить, они занимают достаточно много места.

2. Скорость поступления и обработки. Критерий характеризует скорости накопления, обработки данных. Обычно, хранимые наборы данных обновляются на регулярной основе, поэтому требуются системы и технологии, способные обрабатывать такие данные в режиме реального времени.

3. Разнообразие данных. Сами наборы данных могут быть структурированными и неструктурированными или иметь частичную структурированность. Данные также могут отличаться и по типу: фотографии, текст, звуки и другие.

4. Изменчивость. Большие наборы данных, обычно, анализируется на постоянной основе. Но потоки поступления данных могут иметь пики и спады под влиянием множества факторов: сезон, социальные явления и других причин. Чем изменчивее и не стабильнее поступление данных, тем труднее производить анализ этих данных.

5. Достоверность. Применяется как к самому набору данных, так и к результатам анализа. Данная характеристика отображает соответствие анализируемых данных с реальностью.

К основным этапам обработки больших данных относятся следующие аспекты.

1. Поиск и сбор данных. Данные необходимо где-то получать. Это могут быть различные датчики, социальные сети, веб-источники и многое другое.

2. Хранение данных. Собранные данные необходимо хранить. Для этого используются базы данных. Проектирование и обслуживание эффективных систем хранения данных – один из ключевых аспектов обработки больших данных.

3. Обработка данных. Из-за больших объемов и высокой скорости поступления наборы данных необходимо быстро и эффективно обрабатывать. Применяются распределенные вычисления и обработка в реальном времени.

Одной из проблем является объем. В современном мире все больше и больше сфер применяют различные информационные технологии. Также больше людей используют интернет технологии и используют различные информационные сервисы. Рост данных в информационной сфере с каждым годом возрастает. На рисунке 2 представлено прогнозируемое количество данных в информационной сфере в зеттабайтах (1 зеттабайт ≈ 1 млрд терабайтов).

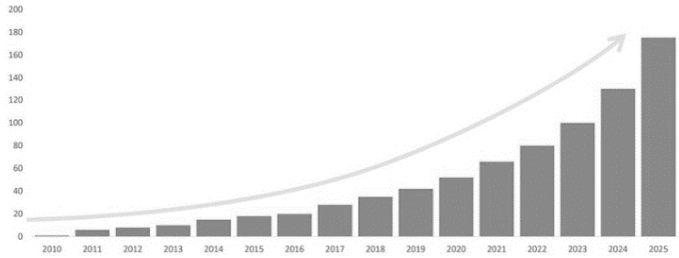


Рисунок 2 – Рост объемов данных

Огромные наборы данных требуется хранить и обрабатывать, что приводит к применению более эффективных и мощных технологий, это приводит к увеличению затрат (ресурсов и финансов). В некоторых случаях данные требуется обрабатывать быстро и в режиме реального времени, так как в некоторых случаях информация может устаревать, прежде чем принесет практическую пользу.

Также из одной проблем является классификация данных. Существует множество способов классифицировать данных, но специфика и объем информации не позволяют создать единые направления и методы работы с ними [3]. Каждая конкретная область применения больших данных требует индивидуального в процессе своей обработки. Эта проблема вытекает из того, что большая часть информации не структурирована, имеет разный вид.

Эффективная работа с большими наборами данных сводится к эффективному сбору, хранению и обработке. Это в свою очередь требует использования специальных инструментов, технологий и подходов, которые позволяют обрабатывать большие объемы данных в кратчайшие сроки и с наименьшими затратами.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Майер-Шенбергер В., Кукьер К. Большие данные: Революция, которая изменит то, как мы живем, работаем и мыслим. 2015.

2. Романенко Е.В. Место Big Data в современной социально-экономической жизни общества // Инновационная наука. 2016. №4-3 (16). URL: https://cyberleninka.ru/article/n/mesto-big-data-v-sovremennoy-sotsialno-ekonomicheskoy-zhizni-obschestva (дата обращения: 05.06.2024).

3. Менщиков А.А., Перфильев В.Э., Федосенко М.Ю., Фабзиев И.Р. Основные проблемы использования больших данных в современных информационных системах // Столыпинский вестник. 2022. №1. URL: https://cyberleninka.ru/article/n/osnovnye-problemy-ispolzovaniya-bolshih-dannyh-v-sovremennyh-informatsionnyh-sistemah (дата обращения: 03.06.2024).

УДК 000.000

Е. А. Соколов

ИССЛЕДОВАНИЕ МЕТОДОВ И РАЗРАБОТКА ПО ДЛЯ РАСПОЗНАВАНИЯ ДВИЖУЩИХСЯ ОБЪЕКТОВ С ИСПОЛЬЗОВАНИЕМ СРЕДСТВ БИБЛИОТЕКИ OPENCV

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

В данной работе исследуются современные методы распознавания движущихся объектов и разработано ПО для решения данной проблемы. Для выявления и классификации движущихся объектов используются средства библиотеки OpenCV и нейросети YOLO. Результаты показывают баланс скорости работы и точности распознавания.

Ключевые слова: компьютерное зрение, распознавание движения, OpenCV, фоновое вычитание, YOLO, распознавание объектов, машинное обучение.

Проблема распознавания движущихся объектов в настоящее время становится все более актуальной: необходимость ее решения встречается, например, в логистике, на производстве, в сельском хозяйстве, военном деле. Цели работы — исследование современных методов распознавания движущихся объектов и готовое программное обеспечение, способное эффективно решать данную задачу.

Поскольку задача по распознаванию объектов достаточно сложна, часто применяются готовые библиотеки и модели. Также стоит отметить, что движущихся объектов может быть весьма много, а количество пикселей на обрабатываемых кадрах исчисляется обычно сотнями тысяч или миллионами, поэтому для ускорения обработки предпочтительно использовать методы определения движения, не основанные на машинном обучении, а нейросети применять для дополнительной обработки уже выделенных частей изображений при необходимости. В данной работе использована популярная кроссплатформенная библиотека OpenCV, в которой реализованы алгоритмы компьютерного зрения с открытым исходным кодом. Для выделения движущихся областей рассматриваются методы фонового вычитания, предоставляемые OpenCV, а непосредственно для распознавания объектов – средства нейросети YOLO.

Исследование методов определения движения

Вычитание фона – это метод для создания маски переднего плана путем вычитания текущего кадра из фоновой модели. Фоновое моделирование включает в себя два основных этапа: инициализацию фона и его обновление (модель корректируется с учетом изменений в сцене). В простейшем случае маска определяется как разность между

фиксированным фоном и текущим изображением [1]. Данный метод самый быстрый, но требует статичного положения камеры и отсутствия движения фона. Более современные методы фонового вычитания позволяют динамически обновлять фоновое изображение, что позволяет обрабатывать данные и без статичного фона. Рассмотрим некоторые методы, соответствующие этому требованию: MOG и MOG2.

Идея алгоритма MOG заключается в том, что каждый пиксель фона представляется смесью нескольких гауссовских распределений, отражающих вероятные цвета пикселя фона. Важный аспект алгоритма MOG – это определение временных пропорций (весов) для каждой компоненты смеси, обновляющиеся при каждом кадре. Эти пропорции указывают, как долго цвета пикселя остаются на изображении. Таким образом, вероятные цвета, которые остаются на изображении дольше и более статичны, имеют больший вес в смеси гауссовских распределений.

Алгоритм MOG2 похож на MOG, но динамически выбирает количество гауссовских распределений для каждого пикселя, используя модель адаптивной смеси гауссовских распределений. MOG2 также добавляет третий компонент в матрицу маски, который обозначает пиксель тени объекта, что позволяет лучше адаптироваться к изменениям в сцене и более точно отделять объекты от их теней [2].

Исследование методов распознавания объектов, основанных на машинном обучении

Ранее для распознавания объектов на изображениях широко использовались каскады Хаара, но в настоящее время они считаются устаревшими, поскольку появились новые нейронные сетей, которые работают быстрее и выдают лучшие результаты распознавания. Одной из таких систем является YOLO — архитектура нейронных сетей, предназначенная для обнаружения объектов на изображении и видеопотоке, использующая следующий принцип: исходное изображение рассматривается как квадратная матрица размерности N, в каждой клетке которой записана информация о наличии объекта, его типе (классе), наличии ограничивающих рамок на соответствующей части картинки. Таким образом, нейросеть просматривает картинку только один раз, что существенно увеличивает скорость обработки. Также YOLO использует функцию потерь, учитывающую точность рамок и уверенность в классах объектов [3].

Разработка ПО для распознавания движущихся объектов

При проведении исследования также разработано ПО для распознавания объектов на видео с веб-камеры или из локального файла с выделением движущихся объектов на видеопотоке и их определением по требованию пользователя, если класс объекта известен системе.

Приложение состоит из серверной и клиентской частей: первая, разделенная на две отдельные программы, отвечает за чтение и обработку

видео, а вторая позволяет отображать готовый видеопоток и передавать команды пользователя обратно на сервер.

Высокоуровневая структура приложения представлена на рисунке 1.

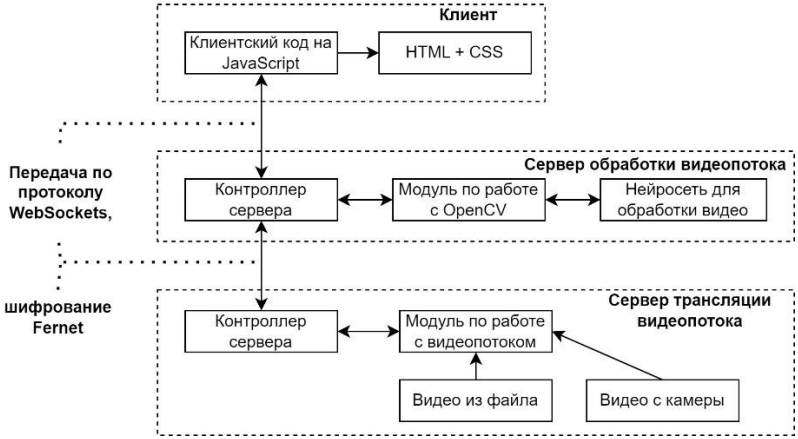


Рисунок 1 – Высокоуровневая структура приложения

Фрагмент интерфейса приложения представлен на рисунке 2.

Рисунок 2 – Фрагмент интерфейса разработанного приложения

Таким образом, исследована часть методов для распознавания движущихся объектов и написано ПО, решающее данную задачу.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1.Интернет-ресурс habr.com – URL: https://habr.com/ru/articles/786436/ (дата обращения: 27.04.2024)

2. Интернет-ресурс opencv.org – URL: https://docs.opencv.org/3.2.0/db/d5c/tutorial_py_bg_subtraction.html (дата обращения: 29.04.2024)

3. Интернет-ресурс leonardoaraujosantos.gitbook.io – URL: https://leonardoaraujosantos.gitbook.io/artificialinteligence/machine_learning/deep_learning/single-shot-detectors/yolo/ (дата обращения: 14.05.2024)

УДК 004.622

Ю. С. Соколова

ОБЗОР ИНСТРУМЕНТОВ PYTHON ДЛЯ ПРОВЕДЕНИЯ РАЗВЕДОЧНОГО АНАЛИЗА ДАННЫХ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

Точность прогнозов, выдаваемых моделью машинного обучения (МО), напрямую зависит от количества и качества передаваемых в нее данных. Поэтому перед подачей в модель данные должны пройти этап подготовки. В работе рассмотрены библиотеки языка Python, ускоряющие процесс подготовки данных для моделей МО.

Ключевые слова: разведочный анализ данных, формирование отчета, библиотеки Python, описательные статистики, визуализация, корреляция.

В современном обществе технологии, использующие искусственный интеллект, становятся все более популярными и востребованными, а их внедрение в различные сферы человеческой деятельности доказало свою эффективность.

Основной целью искусственного интеллекта является разработка инструментария, позволяющего компьютерным системам работать независимо от человека и находить решения сложных профессиональных задач в самых разных областях человеческой деятельности на полноценной интеллектуальной основе. Такие области искусственного интеллекта как машинное обучение, глубокое обучение, робототехника, обработка естественного языка, компьютерное зрение, нечеткая логика, экспертные системы широко применяются в медицине, промышленности, торговле, банках, транспорте и других сферах деятельности.

Машинное обучение представляет собой подраздел искусственного интеллекта, который позволяет компьютерам обучаться на основе данных и опыта, а затем использовать полученные знания для прогнозирования, классификации и принятия решений. Алгоритм обучения будет способен делать качественные предсказания, если он «изучил» много примеров из реальной жизни. При этом эти примеры должны быть корректны. В анализе данных широко известен принцип «Garbage in – garbage out» (GIGO, в пер. с англ. *мусор на входе – мусор на выходе*), означающий, что при неверных входных данных будут получены неверные результаты, даже если сам по себе алгоритм правилен.

На практике данные в «сыром» виде обычно малопригодны для построения модели решения конкретной прикладной задачи [1]. Прежде чем на их основе будет построена модель, данные должны пройти процесс подготовки, который может быть весьма трудоемким и занимать больше времени, чем собственно построение и валидация самой модели. Здесь на помощь аналитикам и исследователям «приходит» разведочный анализ данных (EDA, *Exploratory Data Analysis*), который позволяет понять, с какими данными они имеют дело, выявить основные характеристики, интересные закономерности и потенциальные аномалии в наборе данных, которые могут исказить выводы.

Разведочный анализ данных применяется для:

- *понимания структуры и характеристик набора данных*: прежде чем принимать решения или строить модели, важно понять, что представляют собой данные, каковы их основные характеристики и структура;
- *очистки данных*: EDA помогает выявить ошибки и аномалии в данных, которые могут исказить анализ, например пропущенные значения или выбросы;
- *формулировки гипотез*: EDA позволяет идентифицировать взаимосвязи между переменными, что помогает понять, как одни факторы влияют на другие; на основе наблюдаемых закономерностей в ходе аналитической работы можно сформулировать гипотезы для дальнейшего тестирования;
- *выбора моделей*: понимание данных позволяет выбрать наиболее подходящие статистические модели и методы анализа.

К основным инструментам и методам EDA, помогающим раскрыть смысл в данных относятся: визуализация данных, определение статистик и мер центральной тенденции, корреляционный анализ, анализ выбросов и аномалий.

Визуализация данных позволяет увидеть закономерности, динамику и связи между данными. Для этого используются графические изображения, диаграммы, «ящики с усами», тепловые карты.

Сводные статистики и *меры центральной тенденции* позволяют получить обобщенное представление о распределении данных и количественно оценить их основные характеристики (среднее, моду, медиану и пр.). Это ключевые числовые метрики, которые помогают понять типичные и наиболее значимые значения в наборе данных.

Корреляционный анализ помогает установить взаимосвязь между переменными и определить, насколько сильна эта связь. Коэффициент корреляции измеряет степень линейной зависимости между двумя переменными.

Анализ выбросов и аномалий – это процесс выявления и исследования значений данных, которые существенно отличаются от

остальных наблюдений. Выбросы и аномалии могут возникнуть из-за ошибок в данных, случайных событий или указывать на особенности исследуемого явления.

Сегодня Python является наиболее востребованным языком программирования в анализе данных, машинном и глубоком обучении, обработке естественного языка, поскольку в нем имеется огромное количество специализированных библиотек для конкретных областей науки о данных. Эти библиотеки позволяют специалистам по изучению данных легко применять передовые методы к своим данным, не прибегая к написанию сложных алгоритмов с нуля.

Большой популярностью в EDA пользуется библиотека Pandas, позволяющая исследователю лучше понять особенности набора данных (датасета), взаимосвязи (корреляции) между признаками, а также сделать первые простые выводы на основе данных [1, 2]. Pandas содержит встроенные средства визуализации на основе графики Matplotlib. Обычно встроенная графика Pandas, библиотеки Matplotlib и Seaborn используются совместно [3].

Продемонстрируем средства анализа и визуализации, встроенные в эти библиотеки, на примере набора данных, который содержит подробную информацию о факторах риска сердечно-сосудистых заболеваний [4]. Он включает информацию о возрасте (`age`), поле (`gender`), росте (`height`), весе (`weight`), значениях артериального давления (`ap_hi` и `ap_lo`), уровне холестерина (`cholesterol`), уровне глюкозы (`gluc`), привычках курения (`smoke`) и употреблении алкоголя (`alco`) более чем 70 тысяч человек. Кроме того, в нем указывается, активен ли человек или нет (`active`), есть ли у него какие-либо сердечно-сосудистые заболевания (`cardio`). Этот набор данных предоставляет исследователям отличный ресурс для применения современных методов машинного обучения для изучения потенциальных взаимосвязей между факторами риска и сердечно-сосудистыми заболеваниями, что в конечном итоге может привести к лучшему пониманию этой серьезной проблемы со здоровьем и разработке более эффективных профилактических мер.

При использовании для разведочного анализа библиотеки Pandas для получения информации по каждой гипотезе необходимо применить соответствующий метод. Так, для проверки того, что рост пациентов распределен нормально, необходимо отобразить гистограмму по признаку `height`:

```
df['height'].hist(bins=20);
```

Для проверки наличия в данных аномалий и выбросов, можно построить «ящик с усами»:

```
sns.boxplot(df['height']);
```

Если исследователя интересует вопрос среднего возраста здоровых и больных пациентов, то перед тем, как применить метод расчета среднего значения, данные необходимо сгруппировать.

Получается, что быстро проанализировать данные с использованием библиотеки Pandas не получится: придется вспомнить и применить серию методов и функций, и разведочный анализ данных займет довольно большую часть времени в работе над моделью.

С целью решения подобной проблемы появились инструменты автоматической визуализации и представления датасета. К таким инструментам можно отнести следующие библиотеки Python, которые могут выполнять EDA всего одной строкой кода:

- `ydata-profiling`;
- `sweetviz`;
- `d-tale`.

Библиотека ydata_profiling

Библиотека `ydata_profiling` – это библиотека с открытым исходным кодом, которая генерирует подробный отчет по данным одной строкой кода и может быть легко использована для больших наборов данных [5]. Для формирования отчета достаточно импортировать библиотеки, загрузить набор данных, сгенерировать отчет и, используя метод `to_file()`, сохранить отчет в виде *html*-файла:

```
import pandas as pd
from ydata_profiling import ProfileReport

df = pd.read_csv('cardio.csv', delimiter=';')
profile = ProfileReport(df, title="Pandas Profiling Report")
profile.to_file("cardio_report.html")
```

Сформированный в результате выполнения представленного кода *html*-файл содержит разделы: *Overview*, *Variables*, *Interactions*, *Correlations*, *Missing values*, *Duplicate rows*, *Sample*.

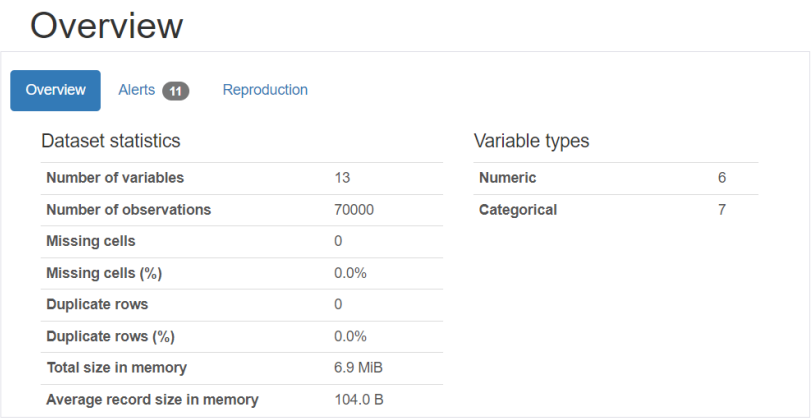


Рисунок 1– Содержимое раздела *Overview* сгенерированного отчета

В разделе *Overview* (рисунок 1) представлена общая статистика по всему датасету: его размер, типы переменных, процентное соотношение пропусков и дублирующих строк.

В разделе *Variables* (рисунок 2) содержится информация о каждой переменной: название, тип, признаки с которыми эта переменная имеет высокую корреляцию, количество уникальных значений, количество пропусков и т.д. Для категориальных и числовых значений информация различается. Подробную статистику по признаку можно посмотреть, нажав на кнопку *More details*. Для категориальных признаков отобразится график с наиболее часто встречающимися значениями, информация по длинам слов, их анализ. Числовые признаки содержат информацию о его распределении, уникальных значениях, пропусках, значения квантилей и пр.

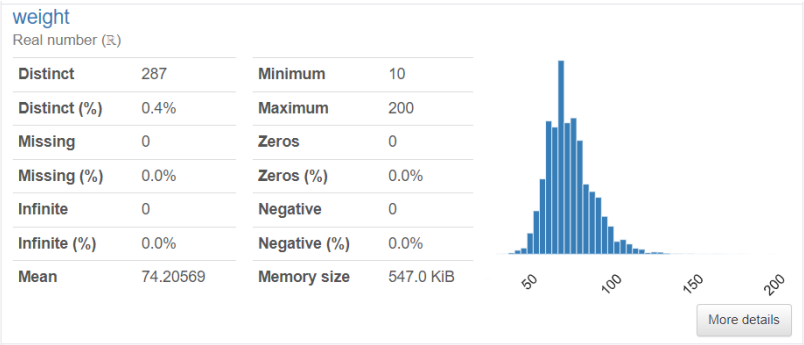


Рисунок 2 – Информация раздела *Variables* о переменной *weight*

Для числовых признаков в разделе *Interactions* представлен график зависимости между двумя признаками. В разделе *Correlations*

представлена информация о корреляции между числовыми признаками в виде тепловой карты и таблицы (рисунок 3).

Heatmap | Table

	active	age	alco	ap_hi
active	1.000	-0.010	0.025	-0.001
age	-0.010	1.000	0.029	0.219
alco	0.025	0.029	1.000	0.033
ap_hi	-0.001	0.219	0.033	1.000

Рисунок 3 – Фрагмент таблицы значений коэффициентов корреляции

В разделах *Missing values* и *Duplicate rows* представлена информация по пропущенным значениям и по дублирующим строкам соответственно. Раздел *Sample* позволяет посмотреть первые и последние строки датасета.

Таким образом, отчет, сгенерированный с помощью `ydata-profiling`, можно использовать для первичной визуализации набора данных, числового и статистического анализа переменных, для определения зависимости между признаками, выявления пропущенных значений.

Библиотека sweetviz

`Sweetviz` – это библиотека автоматического анализа с открытым исходным кодом [6]. `Sweetviz` также можно использовать для сравнения нескольких наборов данных и выводов по ним. Это может быть удобно при сравнении обучающего и тестового наборов данных.

После загрузки данных подробный отчет по ним может быть получен одной строкой кода (аналогично `ydata_profiling`):

```
import pandas as pd
import sweetviz as sv

df = pd.read_csv('cardio.csv', delimiter=';')
report = sv.analyze(df)
report.show_html()
```

В верхней части сформированного отчета (рисунок 4) будет представлена информация о наборе: число строк, дубликатов, число признаков, их тип. Ниже по каждому признаку будет выведена информация о количестве пропусков, количестве уникальных значений, а также его статистические оценки (квантили, среднее, медиана и пр.).

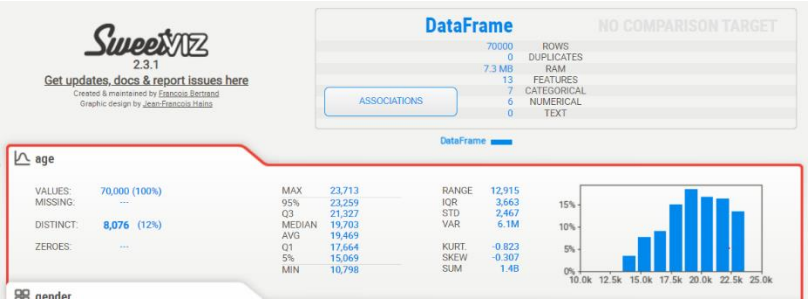


Рисунок 4 – Фрагмент отчета, сгенерированного `sweetviz`

Также в отчете содержится информация о связях переменных в наборе данных, детальная информация о каждом признаке: для категориальных – график часто встречающихся значений, для числовых – число пропусков, статистические показатели, гистограмма распределения признака, абсолютная и относительная частота появления признака в наборе данных.

Таким образом, с помощью `sweetviz` можно провести первичный осмотр набора данных, просмотреть свойства признаков, провести числовой и статистический однофакторный анализ.

Библиотека D-Tale

`D-Tale` – это библиотека с открытым исходным кодом [7]. `D-Tale` делает подробный разведочный анализ набора данных. Отчет в `D-Tale` (как и в вышерассмотренных библиотеках) формируется одной строкой кода:

```
import pandas as pd
import dtale

df = pd.read_csv('cardio.csv', delimiter=';')
report = dtale.show(df)
report
```

В результате выполнения представленного кода отобразится таблица, содержащая несколько первых строк набора данных. Каждый столбец этой таблицы содержит меню с перечнем доступных операций. Здесь можно выбрать следующие действия: анализ пропусков, обзор дубликатов, просмотр корреляции признаков и описательных статистик (среднее, медиану, квантили, выбросы и пр.), отобразить график распределения признака (или «ящик с усами»). На рисунке 5 представлен фрагмент сгенерированного отчета для признака `weight`.

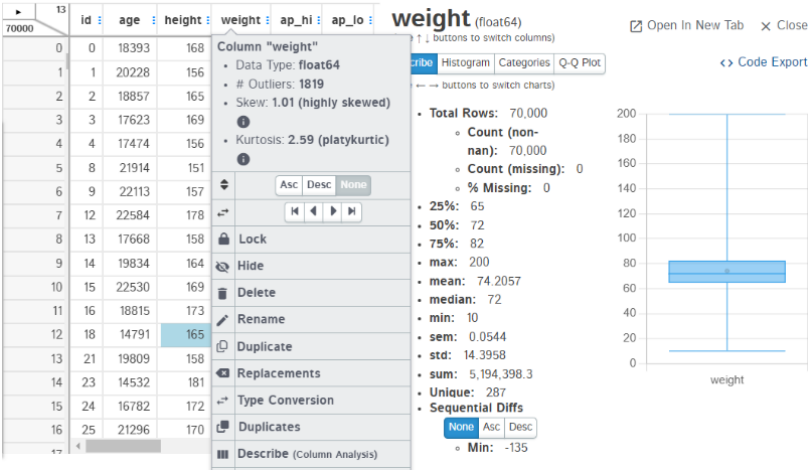


Рисунок 5 – Фрагмент отчета, сгенерированного библиотекой `D-Tale`

Интересная особенность: библиотека предоставляет функцию экспорта кода для каждого графика или элемента анализа в отчете. На рисунке 6 представлен код, сгенерированный `D-Tale` для признака `weight`. Этот код может быть скопирован, вставлен в проект и далее отредактирован в соответствии с целями EDA.

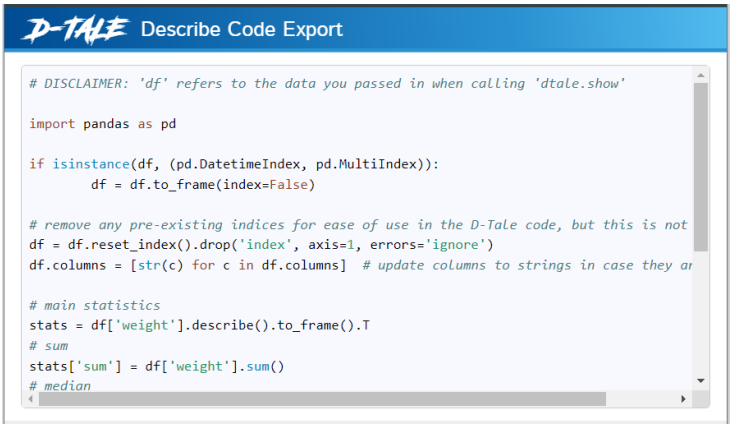


Рисунок 6 – Код, сгенерированный `D-Tale` для признака `weight`

Таким образом, библиотека `D-Tale` позволяет проводить полноценный анализ одного признака, а также многофакторный анализ, смотреть дублирующие строки / колонки, сразу их удалять, переименовывать и производить другие операции.

Сравнение библиотек

Рассмотренные библиотеки можно использовать для ускорения работы, первичной визуализации данных, числового и статистического анализа переменных, определения зависимости между признаками, выявления пропущенных значений. Они позволяют одной строчкой кода сформировать понимание того, с какими данными приходится работать. В таблице 1 представлен их сравнительный анализ по функциональным возможностям.

Таблица 1 – Сравнение основных возможностей библиотек для EDA

Основные возможности	**Название библиотеки**		
	Ydata-profiling	Sweetviz	D-Tale
Отображение информации о типе признаков, повторяющихся строках, пропусках, наиболее частых значениях	+	+	+
Статистический анализ (минимум, максимум, квантили, мода, медиана)	+	+	+
Описательные статистики	+	+	+
Визуализация	+	+	+
Сравнение наборов данных	–	+	+
Экспорт кода	–	–	+

Необходимо понимать, что ни одна из перечисленных выше библиотек не сможет провести детальный анализ так, как это делает специалист по данным. Кроме того, для более точной настройки разведочного анализа рекомендуется проводить его, используя библиотеку Pandas, и писать код на Python, так как важно знать и применять основы EDA, настраивать вывод нужной информации в том виде, который позволяет получить полное понимание как о самом признаке, так и обо всем наборе данных.

Библиотеки автоматического разведочного анализа необходимы для ускорения анализа данных, но никак не заменяют его.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Силен Д., Мейсман А., Али М. Основы Data Science и Big Data, Python и наука о данных. – СПб.: Питер, 2019. – 336 с.

2. Devpractice Team. Pandas. Работа с данными. 2-е изд. – devpractice.ru. 2020. – 170 с.

3. Devpractice Team. Python. Визуализация данных. Matplotlib. Seaborn. Mayavi. – devpractice.ru. 2020. – 412 с.

4. https://www.kaggle.com/competitions/heart-desease-kaggle-dpo-2023

5. https://github.com/ydataai/ydata-profiling

6. https://github.com/fbdesignpro/sweetviz

7. https://github.com/man-group/dtale

УДК 004.623

М. М. Таукенов

КОНСОЛИДАЦИЯ ДАННЫХ КАК СОВРЕМЕННЫЙ МЕТОД ОБРАБОТКИ И ХРАНЕНИЯ ДАННЫХ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

В условиях непрерывного роста объемов данных и усложнения ИТ-инфраструктур, организации сталкиваются с необходимостью оптимизации управления данными и снижения затрат. Проблема решается через консолидацию данных, что позволяет существенно улучшить операционную эффективность, снизить расходы на техническое обслуживание и упростить административные процессы.

Ключевые слова: консолидация баз данных, управление данными, операционная эффективность, снижение затрат, ИТ-инфраструктура, оптимизация процессов, безопасность данных..

В эпоху цифровизации и всевозрастающего потока информации значения цифровых данных для организаций всех масштабов невозможно переоценить. Однако с увеличением объема и разнообразия данных постоянно возрастает и сложность их управления. В этом контексте консолидация данных выступает не просто как технологичное решение, но и как стратегическая необходимость.

Консолидация данных — это процесс объединения данных из различных баз в единое, централизованное хранилище данных. Такая практика позволяет организациям уменьшить излишнее дублирование данных, сократить расходы на хранение и управление данными, а также упростить доступ и анализ информации. В результате реализации процесса консолидации достигается не только оптимизация ресурсов, но и повышение качества данных, что, в свою очередь, ведет к более осмысленным выводам и лучшим бизнес-решениям.

Достижение цели консолидации баз данных требует применения тщательно отобранных подходов и инструментов, способных обеспечить эффективную интеграцию разрозненных наборов данных. Выбор оптимального подхода зависит от специфических потребностей и условий организации. Существует множество технологических подходов к консолидации данных, включая системы управления базами данных (СУБД, такие как MySQL, PostgreSQL,Oracle,и NoSQL СУБД, такие как MongoDB, Cassandra и Redis), инструменты ETL (Извлечение, Трансформация, Загрузка такие как Informatica, Talend и Apache NiFi), системы для управления данными и хранилища данных (Решения Data Warehouse, включая Teradata и Amazon Redshift, платформы для

управления данными, такие как SAP Master Data Governance), множество инструментов работы с data lake (такие как Apache Hadoop и Amazon S3),

распределенные системы и облачные решения (Облачные платформы, такие как Google Cloud Platform, Amazon Web Services и Microsoft Azure), федерация данных (такие как Denodo и Informatica Data Federation), master data management (mdm) системы (IBM InfoSphere и Oracle Hyperion), инструменты автоматизации и интеграции (Celonis и UiPath).

Каждый из этих подходов и инструментов имеет свои особенности и лучше подходит для определенных типов и объемов данных, а также целей и задач консолидации. Выбор конкретных решений должен происходить в соответствии с требованиями к безопасности, производительности, масштабируемости и стоимости, чтобы удовлетворить все бизнес-потребности организации и обеспечить плавную интеграцию данных в единую экосистему.

На рынке одним из мощных инструментов является ETL (Extract, Transform, Load), который обеспечивают сбор и трансформацию данных перед их загрузкой в централизованные системы хранения.

Эти инструменты не только помогают в сборе и трансформации данных, но также в их очистке, стандартизации, объединении и загрузке в централизованные системы, таким образом обеспечивая качественную и эффективную подготовку данных для аналитических и операционных целей.

В условиях постоянно растущего объема данных стратегически важным становится не только их хранение, но и гарантирование их эффективности и управляемости. Консолидация данных играет ключевую роль в достижении этих целей.

Эффективность данных обуславливается их способностью быть доступными и полезными для конкретных задач и решений. Это включает в себя актуальность, целостность, точность и скорость доступа. Консолидация помогает повысить эффективность данных путем устранения дубликатов, уменьшения количества репликаций и обеспечения централизованной точки доступа к данным. Эти действия приводят к более быстрому времени отклика систем и уменьшает время, необходимое для поиска и анализа данных.

Консолидация данных может улучшить **производительность** системы, особенно для организаций, которые имеют множество различных баз данных. Вместо того, чтобы каждый раз искать необходимые данные в разных базах данных, при консолидации все они будут храниться в одном месте, что сокращает время доступа и ускоряет общий процесс поиска и обработки данных.

Управляемость данных подразумевает способность организации управлять данными на протяжении всего их жизненного цикла — от

сбора до удаления. Консолидация баз данных способствует управляемости за счет централизации управленческих функций, таких как безопасность, резервное копирование, аудит и соблюдение правил. Централизованная система упрощает применение политик и стандартов, облегчает мониторинг состояния данных и их использования, а также повышает производительность оперативного восстановления после сбоев.

Практической реализацией этих принципов является создание **единой платформы управления данными**, которая осуществляет организацию, хранение, защиту и обеспечение доступа к данным, не зависимо от их источников. Платформы такого типа, зачастую, основаны на Data Warehouses (хранилищах данных), Data Lakes («озера данных») или облачных решениях, каждое из которых обладает своими преимуществами и может быть адаптировано под нужды конкретной организации.

Данный подход позволяет не только повысить систематический контроль над данными, но и обеспечивает возможность глубокой аналитики и более обоснованного принятия решений на основе описательной, предикативной и прескриптивной аналитики. Современные аналитические инструменты и интеллектуальные системы могут полноценно использовать возможности централизованного управления данными, для выявления тенденций, прогнозирования поведения и оптимизации бизнес-процессов.

В итоге, консолидация баз данных позволяет преобразовать данные из пассивного актива в активный ресурс, который активно содействует продуктивности, инновационности и конкурентоспособности компании.

Процесс консолидации баз данных предполагает объединение данных, разбросанных по множеству физических серверов и систем хранения, в более малое количество мощных и эффективных решений. Эта централизация имеет значительные экономические выгоды.

Снижение затрат на инфраструктуру: Уменьшение числа серверов и другого оборудования напрямую сокращает капитальные затраты (CAPEX). Необходимость в приобретении нового оборудования уменьшается, а с ним и расходы на его обслуживание и замену.

Оптимизация используемого пространства: Меньшее количество физических устройств требуют меньше места в дата-центрах, что приводит к экономии на аренде или обеспечивает дополнительное пространство для другого использования.

Затраты на энергопотребление: Более эффективные, централизованные сервера часто потребляют меньше электроэнергии, что снижает операционные расходы. Это также сокращает стоимость охлаждения, так как меньшее количество оборудования выделяет меньше тепла.

Уменьшение расходов на техническое обслуживание: Консолидированная инфраструктура упрощает процессы поддержки и уменьшает количество потенциальных точек сбоев, что сокращает расходы на техническое обслуживание и ремонт.

Важно отметить, что консолидация данных является шагом к трансформации инфраструктуры предприятия, которая позволяет принимать обоснованные и финансово выгодные решения в управлении данными на долгосрочной основе. Результатом такой трансформации становится не только экономия ресурсов компании, но и освобождение потенциала для инновационного развития и роста.

Хотя консолидация баз данных предлагает значительные преимущества, процесс реализации может сопровождаться рядом сложностей и препятствий. К одним из наиболее распространенных трудностей, с которыми могут столкнуться организации можно отнести:

Сопротивление изменениям. Люди естественным образом склонны сопротивляться изменениям, особенно когда они влияют на укоренившиеся рабочие процессы. Переход к консолидированной системе может вызвать опасения у сотрудников по поводу новых процедур и возможности утраты рабочих мест.

Интеграция разнородных систем. Различные отделы в пределах одной организации могут использовать разные системы управления базами данных, каждая из которых имеет собственные технические характеристики. Интеграция этих систем может быть довольно сложной задачей.

Несоответствие данных. Очищение, стандартизация и дедубликация данных в преддверии консолидации — трудоемкий процесс. Различия в форматах, потерянные или неактуальные данные существенно осложняют задачу.

Безопасность и приватность данных. Консолидация требует пересмотра политик безопасности и часто приводит к необходимости усиления защитных мер в период миграции данных и их повторного размещения.

Затраты и ROI. Оценка затрат на консолидацию и прогнозирование возврата инвестиций (ROI) может быть сложной. Необходимо учитывать не только прямые издержки на приобретение нового оборудования и программного обеспечения, но и косвенные расходы, связанные с обучением персонала и потенциальным простоем в работе.

Сложности миграции. Перемещение данных из нескольких источников в единое хранилище требует тщательного планирования и часто выполняется поэтапно для минимизации рисков потери данных и прерывания рабочих процессов.

Поддержание работы текущих систем. В процессе консолидации важно поддерживать бесперебойную работу существующих систем, что часто представляет собой непростую задачу с технической и организационной точки зрения.

Управление проектом. Консолидация баз данных — это масштабный проект, который требует строгого управления проектами, особенно когда факторов риска и стейкхолдеров много.

Разрешение этих препятствий требует комплексного подхода, включая обширное планирование, привлечение опытных специалистов и участие ключевых лиц принятия решений. Очень важно общение и сотрудничество между всеми заинтересованными сторонами для минимизации потенциальных проблем и обеспечения плавного перехода к новой системе.

Подводя итоги, мы видим, что консолидация баз данных приносит существенные преимущества для организаций: от повышения эффективности и управляемости данных до оптимизации аналитических возможностей и снижения затрат на IT-инфраструктуру. Рассмотрение реальных примеров из практики иллюстрирует, как эти преимущества могут быть реализованы в различных индустриях и отраслях.

Тем не менее, процесс консолидации баз данных часто связан с рядом вызовов и препятствий, начиная от сопротивления изменениям и заканчивая техническими и организационными сложностями. Выявление этих препятствий и разработка стратегий их преодоления являются неотъемлемой частью успешной консолидации данных.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Г. И. Агафонов. DAMA-DMBOK. Свод знаний по управлению данными // DAMA International. - 2017. - С. 830.

2. Паклин Н., Орешков В., Бизнес-аналитика: от данных к знаниям. // Издательство "Питер", 2013. – С. 704.

3. Арьков В. Ю., Бизнес-аналитика. Извлечение, преобразование и загрузка данных. Учебное пособие. // Издательские решения, 2020. – С. 128.

УДК 004.942

С. А. Веркин, А. Р. Пирожков, А. В. Рогатин, А.И. Толоконников

АЛГОРИТМЫ ОПРЕДЕЛЕНИЯ СТОЛКНОВЕНИЯ ГЕОМЕТРИЧЕСКИХ ФИГУР

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

В данной статье производится постановка задачи нахождения столкновения, анализ самых распространённых алгоритмов и по результатам даётся сравнительная характеристика алгоритмов и вывод о сфере и случаях их применения.

Ключевые слова: столкновения, Минковский, AABB, Sphere-Sphere, GJK.

При рассмотрении взаимодействий физических моделей одной из важных задач является определение их столкновений. Для человека задача определения столкновений не является настолько сложной, как для компьютера, поэтому алгоритмы нахождения столкновений на ЭВМ всегда остаются в тени. Тем не менее, алгоритмы, определяющие, произошло ли столкновение объектов, имеют большое прикладное значение во многих областях, таких как робототехника, навигация, медицина и другие. В данной статье мы рассмотрим различные методы определения столкновения между фигурами, их особенности и области применения, которые позволяют компьютеру решать эту, казалось бы, лёгкую задачу эффективными способами.

В начале стоит дать точную постановку задачи нахождения столкновения. Как входные данные задача принимает две геометрические фигуры, представленные в любом виде, а возвращает единственное значение – столкнулись ли данные геометрические фигуры или нет.

С первого взгляда покажется, что объективно верным решением данной задачи будет проверить пересечения граней фигур, заданных n-мерными плоскостями второго порядка, но в большинстве случаев это будет очень затратно и потребует отдельных алгоритмов для разных представлений объекта.

На практике при нахождении пересечений используется **сумма Минковского**. Сумма Минковского – фигура, получаемая путём сложения каждой точки одной выпуклой фигуры с каждой точкой другой выпуклой фигуры. Графически сумму Минковского можно представить, если одну фигуру обвести по контуру другой фигурой.

Графическое представление суммы Минковского можно увидеть далее (рисунок 1) [2]. На рисунке представлены фигуры A и B и соответственно фигура C – сумма Минковского A и B.

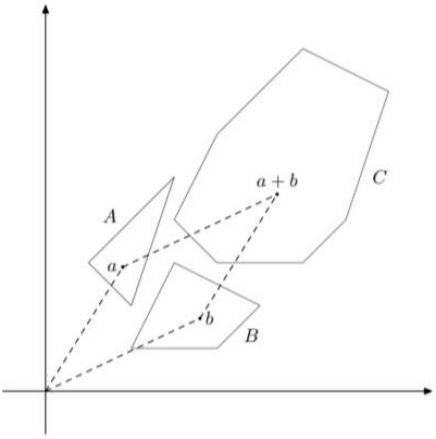


Рисунок 1 – Графическое представление суммы Минковского

Таким образом, мы можем заменить задачу нахождения столкновения двух выпуклых фигур задачей нахождения столкновения суммы Минковского двух фигур с единственной точкой.

Задача становится ещё легче, если использовать разность Минковского. Разность Минковского – фигура, получаемая путём вычитания каждой точки одной выпуклой фигуры из каждой точки другой выпуклой фигуры. По сути разность Минковского – это сумма Минковского, где одна из фигур инвертирована относительно начала координат. Тогда мы знаем точно, с какой точкой сравнивать пересечение. Этой точкой будет начало координат.

Из этого вытекает универсальное условие столкновения фигур:

Если разность Минковского двух выпуклых фигур содержит начало координат, то эти фигуры сталкиваются.

Соответственно, каждых из алгоритмов столкновений решает две подзадачи: нахождение разности Минковского фигур и определение принадлежности начала координат данной разности.

Далее мы рассмотрим три самых известных алгоритма нахождения столкновений.

Метод Axis-Aligned Bounding Box (с анг. выровненная по осям ограничивающая коробка) или AABB – это простой и широко используемый алгоритм в компьютерной графике и игровой разработке, где фигуры упрощаются до выровненных по осям n-мерных параллелепипедов. Грани параллелепипеда определяются максимальными и минимальными координатами фигуры, то есть параллелепипед полностью содержит фигуру.

Легко понять, что сумма (а соответственно и разность) Минковского двух параллелепипедов – более большой параллелепипед,

совмещающий в себе габариты двух. Чтобы проверить, что фигура содержит начало координат, достаточно сравнить координаты каждой грани разности Минковского с началом координат. Если у каждой пары противоположных граней разные знаки, то происходит столкновение.

AABB является простым в реализации и вычислительно эффективным методом для быстрого отбрасывания объектов, что делает его хорошим выбором для проверки коллизий в играх и при работе с трехмерной графикой.

Метод Sphere-sphere – это простой и эффективный алгоритм для проверки коллизий между сферами в компьютерной графике и физических симуляциях. Часто в физических симуляциях используется такое понятие, как материальная точка, что, по сути, означает тело, представленное в пространстве лишь точкой. Тем не менее для того, чтобы эта точка могла взаимодействовать и сталкиваться с другими точками, то часто эффективно представить её, как небольшой сферический объём.

Для данного метода сумма Минковского будет находиться так же просто. Если обвести одну сферу вокруг другой, то мы получим сферу с радиусом, равным сумме радиусов изначальных сфер.

То есть для того, чтобы проверить столкновение можно найти расстояние между центрами этих сфер и сравнить, что оно больше суммы радиусов.

Метод Sphere-sphere обладает высокой скоростью работы и простотой реализации, что делает его хорошим выбором для быстрой проверки коллизий между объектами в реальном времени. Однако этот метод подходит для мелких простейших объектов, ведь он не учитывает форму изначального моделируемого объекта.

Все предыдущие алгоритмы пытались упростить представляемую форму для того, чтобы быстро найти столкновения фигур, но зачастую это будет выливаться в случаи ложного срабатывания. Для того, чтобы сравнить точно необходимы более продвинутые алгоритмы, такие как **алгоритм GJK** или **Алгоритм Гилберта Джонсона Керти**.

Этот алгоритм требует особого представления фигур с помощью вспомогательной функции, возвращающей точку для каждого переданного ей направляющего вектора. Если фигура выпуклая, то её точно можно представить в таком виде, а если невыпуклая, то фигуру можно представить, как набор таких функций, разбив фигуру на несколько более простых [1].

Для того, чтобы найти граничные точки разности Минковского находятся две точки: одна точка первой фигуры в одном направлении поиска и вторая точка второй фигуры в противоположном направлении поиска. При сложении или вычитании одной точки из другой мы получаем граничные точки, соответствующие сумме или разности

Минковского. Функцию, возвращающую граничную точку на разности Минковского мы назовём функцией Configurational Space Obstacle (с анг. препятствие конфигурационного пространства) или F_{CSO}.

Направляющих векторов бесконечное множество, поэтому необходимо ограничится каким-то набором точек. Первое, что можно придумать, так это выбрать дискретное множество таких точек, что значительно повредит точности алгоритма.

Но для GJK используется собственный алгоритм приближения к началу координат. Этот алгоритм состоит из нескольких этапов (рисунок 2) [3].

На этапе 1 берётся точка в произвольном направлении поиска. На рисунке эта произвольная точка – точка A. Из этой точки проводится вектор-перпендикуляр к началу координат. Далее находится следующая точка B в направлении поиска данного перпендикуляра.

На этапе 2 точка B добавляется во множество точек Q и уже от данного множества строится перпендикуляр и находится точка D.

На этапе 3 точка D добавляется в набор. Множество проверяется на содержание в своём объёме начала координат. Если начало координат содержится в данном симплексе (n-мерном треугольнике) Q, алгоритм завершается обнаружением столкновения. Если нет, то точка A убирается из него, так как она находится не в направлении начала координат.

Этап 3 продолжается до момента, как симплекс окружит начало координат, или в симплексе начнут появляться точки, находящиеся по противоположную от начала координат сторону, что означает, что столкновения между фигурами нет.

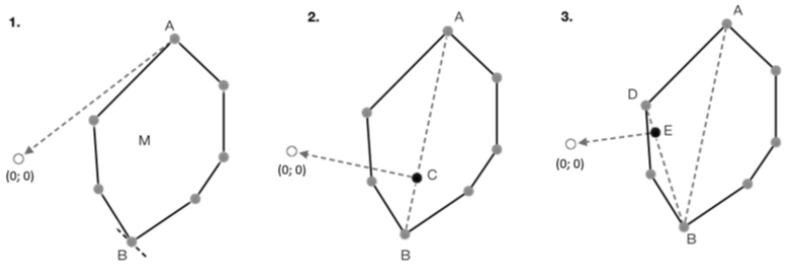


Рисунок 2 – Метод итеративных приближений GJK

Реализацию алгоритма на алгоритмическом языке можно представить следующим образом:

1. Инициализировать вспомогательную точку $S \leftarrow F_{CSO}$ (случайное направление)

2. Инициализировать множество точек симплекса $Q \leftarrow \{S\}$

3. Инициализировать направление $D \leftarrow -S$

4. Цикл:
   $S \leftarrow F_{CSO}(D)$
   Если $S \cdot D < 0$ тогда Вернуть ложь
   $Q \leftarrow Q \cup \{S\}$
   Если СконструироватьСимплекс (Q, D) тогда Вернуть истина

S – вспомогательная точка, добавляемая в симплекс

Q – набор точек симплекса

D – направление поиска

CSO – разность Минковского

СконструироватьСимплекс – функция, обновляющая симплексный набор и проверяющая на принадлежность начала координат симплексу [4].

Алгоритм GJK особенно эффективен для объектов с высокой степенью симметрии, таких как сферы, кубы, цилиндры и другие геометрические формы, но иногда при несимметричных фигурах может вылиться в значительное увеличение количества итераций и уменьшению быстродействия.

На следующем этапе исследования мы создали программные реализации всех описанных ранее алгоритмов для двумерного пространства и замерили их быстродействие на большой выборке данных. В таблице 1 представлено среднее время сравнения каждого типа фигур с каждой, а в таблице 2 представлено среднее время работы каждого алгоритма.

Таблица 1 – Время нахождения столкновения разных типов фигур

	Прямоугольник	**Круг**	**Выпуклая фигура**
Прямоугольник	0,00057 мс	0,00104 мс	0,00119 мс
Круг	0,00104 мс	0,00055 мс	0,00119 мс
Выпуклая фигура	0,00125 мс	0,00121 мс	0,00138 мс

Таблица 2 – Время нахождения столкновения разными методами

	AABB	**Sphere-Sphere**	**GJK**
Время	0,00015 мс	0,00018 мс	0,00112 мс

Результаты метрик для разных алгоритмов существенно варьируются. Проверка пересечений двух прямоугольников с помощью AABB оказывается самой простой операцией. Несколько дольше проводится проверка пересечений для окружностей методом Sphere-Sphere. В других же случаях, когда использовался GJK, время работы программы увеличивалось многократно.

После анализа результатов было сделано предположение, что алгоритм GJK выполняется гораздо дольше остальных, и поэтому можно использовать более простые методы (например AABB) для того, чтобы отбрасывать заведомо не сталкивающиеся фигуры перед тем, как

проверять их GJK. Была выдвинута гипотеза, что это будет работать особенно хорошо для систем, в которых фигуры рассеяны в пространстве.

Был проведён эксперимент, где замерялось среднее время работы алгоритма GJK без начальной фильтрации и с начальной фильтрации. Был получен следующий график прироста быстродействия алгоритма с фильтрацией по сравнению с обычным GJK (рисунок 3).

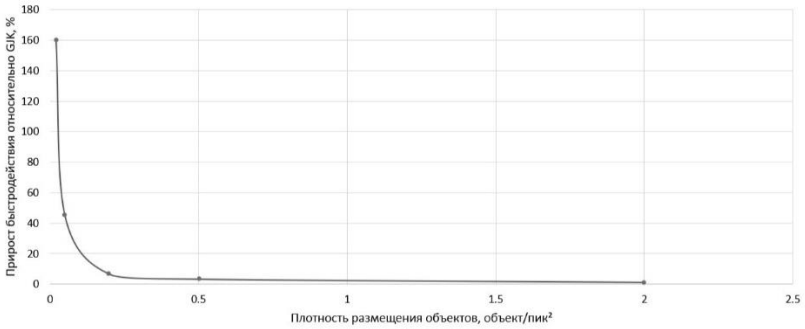


Рисунок 3 – График зависимости прироста быстродействия от плотности

При небольшой плотности размещения объектов производительность алгоритма с начальной фильтрацией значительно превосходит производительность алгоритма без фильтрации, что доказывает нашу гипотезу.

Из полученных окончательных результатов можно сделать вывод, что одной из главных задач при рассмотрении фигур является проверка возможности применения более простых методов – во многих случаях это поможет существенно улучшить быстродействие программы.

Поскольку алгоритм ГЖК выполняет примерно в 8 раз дольше, чем другие, то можно сделать вывод, что для моделей, в которых объекты большую часть времени находятся на расстоянии друг от друга, можно использовать метод AABB для первоначальной (фильтрующей) проверки столкновений.

В сферах физического моделирования, в которых необходимо моделировать движения материальных точек, удобнее использовать Sphere-Sphere.

В качестве основного вывода можем сказать, что каждый из рассмотренных методов имеет свои плюсы и минусы, и каждый выделяется в своей сфере применения, поэтому, прежде чем моделировать систему, важно определить, насколько вам важна точность измерений и как часто объекты будут контактировать. В данной работе был проведён анализ этих методов, их реализации и сделаны выводы о сферах их применения.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. https://arxiv.org/ftp/arxiv/papers/1505/1505.07873.pdf
A Geometric Interpretation of the
Boolean Gilbert-Johnson-Keerthi Algorithm
Jeff Linahan

2. Визуализация алгоритма обнаружения столкновений GJK [Электронный ресурс] : https://www.haroldserrano.com/blog/visualizing-the-gjk-collision-algorithm

3. Задача на нахождение суммы Минковского [Электронный ресурс] : https://codeforces.com/problemset/problem/1195/F?locale=ru

4. Реализация GJK – 2006 – [Электронный ресурс] : https://caseymuratori.com/blog_0003

УДК 004.942

Е. Д. Фирсова

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ФОРМИРОВАНИЯ КОМАНД РАЗРАБОТЧИКОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И УПРАВЛЕНИЯ ИХ РАБОТОЙ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

В IT-сфере проблема формирования эффективных команд осложнена сложностью подбора сотрудников и недостатками методов оценки. Разработка ПО, использующего тесты Белбина и тесты на компетенции, а также упрощающего анализ результатов тестирования, поможет создавать эффективные команды, улучшая сотрудничество и снижая риски.

Ключевые слова: коллективная работа, эффективное сотрудничество, программное обеспечение, soft skill, компетенции.

В IT-сфере уже много лет особое внимание уделяется коллективной работе. Сплочённая и хорошо подобранная команда способна достигать значительных успехов в любой области. Эффективное сотрудничество позволяет ускорить разработку, снизить риски и изменить подход к выполнению задач. Однако успешная работа в команде требует баланса и разнообразного состава.

Создание успешной команды — сложная задача. Мы часто слышим, что «группа разработчиков из Boston Dynamics представила новый продукт» или «команда Яндекса решает такую-то проблему», и привыкаем к этим знакомым коллективам, ожидая от них новых достижений. Но сложность подбора сотрудников, внутренняя иерархия, взаимодействие различных групп разработчиков с разными ролями

остаются вне поля зрения пользователей. Очевидно, что взаимодействие внутри команды может либо объединить таланты, либо ослабить их.

Для формирования сбалансированных и разносторонних команд необходимо выявлять сильные и слабые стороны каждого участника, а также учитывать их личностные качества. Например, при выборе лидера команды важно оценивать не только soft skills, но и множество других качеств.

Существует множество методов оценки, таких как тесты и опросы. Однако у этих методов есть недостатки: неправильная интерпретация результатов, отсутствие аналитики командного результата, и отсутствие единой шкалы оценивания. Например, популярные тесты на компетенции, предрасположенности и общепрофессиональные навыки размещены на разных сайтах и не позволяют проводить комплексный анализ и сохранять результаты.

Разработка программного обеспечения для формирования и управления командами необходимо, чтобы решить вышеуказанные проблемы за счёт:

- Использования теста Белбина[1] для определения предпочтений и склонностей каждого члена команды.

- Использования теста на компетенции, включающего оценку общепрофессиональных навыков и управленческих компетенций[2].

- Использования конструктора команд на основе теста Белбина, который поможет формировать сбалансированные и эффективные команды.

- Использования функционала для анализа командного результата.

Проект основан на передовом технологическом стеке:

- Язык программирования Java 17 и фреймворк Spring обеспечивает надёжность и масштабируемость backend-части приложения.

- Фронтенд будет создан с использованием HTML, CSS и фреймворка Vue.js, что обеспечивает современный интерфейс и удобство использования.

- Для разработки будут использоваться среды IntelliJ IDEA CE (для backend) и WebStorm (для frontend).

- Для хранения данных будет использоваться база данных MySQL, что обеспечит надёжность и эффективность операций с данными.

Общая схема клиент-серверного приложения с использованием базы данных mySql, фреймворка Java Spring для серверной части (backend), фреймворка Vue.js для клиентской части (frontend).

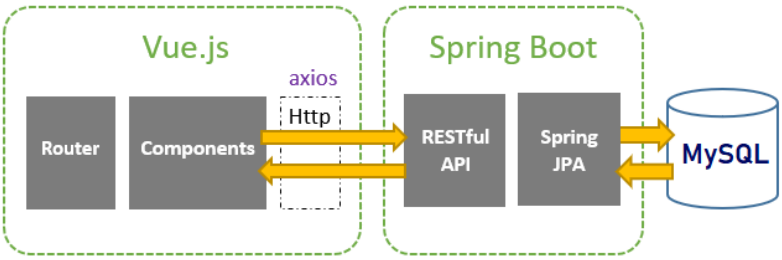


Рисунок 1 — Схема архитектуры приложения

1. База данных (MySQL):

Реляционная система управления базами данных.

2. Серверная часть (backend):

Java Spring — фреймворк для создания надёжных и масштабируемых приложений на Java.

RESTful API — архитектурный стиль для создания веб-сервисов, использующих HTTP запросы.

Spring JPA (Java Persistence API) — это модуль фреймворка Spring, который упрощает работу с объектно-реляционными базами данных, предоставляя удобные методы для управления сущностями и выполнения запросов.

3. Клиентская часть (frontend):

Vue.js - это JavaScript-фреймворк для создания пользовательских интерфейсов.

Components (компоненты) - это независимые и повторно используемые элементы пользовательского интерфейса, созданные на основе Vue.js. Они представляют собой многократно используемые блоки кода, которые можно использовать в разных частях приложения.

Router (маршрутизатор) - это механизм навигации в одностраничном приложении (SPA), который перенаправляет пользователя на разные страницы приложения, в зависимости от URL-адреса. В Vue.js реализуется с помощью пакета Vue Router.

Axios - библиотека (фреймворк) на языке JavaScript, которая позволяет делать запросы на сервер и получать ответы. Она обеспечивает удобный и простой интерфейс для взаимодействия с API.

Таким образом, разрабатываемая информационная система предоставит удобный и эффективный механизм для формирования команд.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1.ОПРЕДЕЛИ СВОЮ РОЛЬ В КОМАНДЕ! ТЕСТ БЕЛБИНА // GitHub URL: [https://test-belbina.github.io/test_belbina/block/1] (дата обращения: 15.05.2024).

2. Как проверить личные качества кандидата на собеседовании // Поток. Блог URL: https://potok.io/blog/hr-howto/lichnye-kachestva-soft-skills-kandidata/ (дата обращения: 15.05.2024).

УДК 004.032.26

Д.С. Коротков, Н.И. Цуканова

ОБ ОДНОМ ИЗ СПОСОБОВ ДЕТЕКТИРОВАНИЯ АНОМАЛИЙ РАСХОДА ТОПЛИВА В АВТОМОБИЛЕ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

Рассматривается способ детектирования аномалий расхода топлива в автомобиле, основанный на машинном обучении нейронных сетей автокодировщика и классификатора.

Ключевые слова: обучающая и тестовая выборки, аномалии расхода топлива, автокодировщик, машинное обучение, нейронная сеть, ошибка реконструкции

Постановка задачи. Многие транспортные компании терпят ежегодные убытки от недобросовестного поведения сотрудников – водителей транспортных средств. Например, водитель использует транспортное средство компании в личных целях без договоренности с начальством или сливает топливо из своей машины в канистры для дальнейшей перепродажи. Поэтому компании вынуждены обращаться к поставщикам оборудования, осуществляющего мониторинг движения автомобилей. Такой мониторинг необходим, чтобы круглосуточно в любой момент времени получать качественные и точные данные о состоянии и местоположении автомобиля для определения аномальных ситуаций расхода топлива во времени (таких, как слив топлива).

Но, если данные GPS и так показывают местоположение машины во времени, то с данными по топливу необходимы дополнительные настройки, которые выполняются сотрудниками техподдержки вручную.

Целью работы является автоматизация выявления аномальных ситуаций в данных по расходу топлива для сокращения затрат организации, занимающейся спутниковым мониторингом транспортных средств.

В данной работе рассматривается способ детектирования аномалий расхода топлива в автомобиле, основанный на использовании машинного обучения автокодировщика и классификатора.

В последнее время появилось много работ [1-2], посвященных использованию автокодировщиков для обнаружения аномалий.

Большинство из них основаны на следующей идее. Автокодировщик – это нейронная сеть [3-5], целью обучения которой является получение на ее выходе как можно более точной копии данных, поступающих на вход нейронной сети. Разность между входом и выходом автокодировщика называется ошибкой реконструкции[2]. При небольшом количестве аномальных событий в выборке ошибка реконструкции более выпукло покажет аномальные события. Тогда на ней можно обучить любой классификатор [3,5], и он покажет более высокую точность детектирования аномальных событий.

Методика и алгоритм решения задачи

Имеется размеченная выборка X->Y [4], каждая запись которой содержит следующие данные о состоянии автомобиля и расходе топлива в нем (X): 1) дата и время записи с датчика уровня топлива; 2) уровень топлива во всех баках автомобиля; 3) заведен ли двигатель машины в этот момент; 4) двигается ли автомобиль: если скорость больше 0 – то равняется 1, иначе 0; 5) разность уровня топлива в текущей записи по сравнению с предыдущей; 6) разность в минутах между датами и временем текущей и предыдущей записи. Всего 6 признаков. Для каждой записи по признакам X эксперт устанавливает 7) метку Y (аномальная (1) или нормальная (0) ситуация). Предполагается, что аномальных примеров значительно меньше (10%-15%), чем нормальных (90% - 85%)

1. Исходную выборку делим на обучающую и две тестовые выборки в следующей пропорции: *Хобуч->Yобуч* - 50%; *Xтест1->Yтест1* - 40%; *Xтест2->Yтест2* - 10%.

2. Из Хобуч->Yобуч удаляем аномальные примеры и получаем Хоб_норм ->Yоб_норм, которая немного меньше. Она состоит только из нормальных примеров.

3. На этой выборке Хоб_норм->Хоб_норм обучаем автокодировщик, находим его весовые коэффициенты [3,5].

```
# From training data, select only normal examples (91%)
X_train_norm = X_train[y_train == 0]

# Define the autoencoder
input_dim = X_train_norm.shape[1]
encoding_dim = 16  # This can be tuned

input_layer = Input(shape=(input_dim,))
encoder = Dense(64, activation="relu")(input_layer)
encoder = Dense(32, activation="relu")(encoder)
encoder = Dense(encoding_dim, activation="relu")(encoder)
decoder = Dense(32, activation="relu")(encoder)
decoder = Dense(64, activation="relu")(decoder)
decoder = Dense(input_dim, activation="sigmoid")(decoder)
autoencoder = Model(inputs=input_layer, outputs=decoder)

# Compile the autoencoder
autoencoder.compile(optimizer='adam', loss='mse',metrics='accuracy')

# Train the autoencoder
autoencoder.fit(X_train_norm, X_train_norm, epochs=50, batch_size=32, validation_split=0.2)
```

Рисунок 1 – Архитектура автокодировщика

4. Выборку Хтест1 подаем на вход обученного автокодировщика. Примеры этой выборки содержат как нормальные, так и аномальные ситуации, поэтому на выходе получим значения Хтест1_вых, отличающиеся от входных.

5. Вычисляем ошибку реконструкции – разность между входом и выходом автокодировщика Err=(Хтест1 - Хтест1_вых). Ошибка реконструкции содержит данные, которые усиливают различия между аномальной и нормальной ситуацией.

6. Поэтому на выборке (Хтест1 - Хтест1_вых) -> Ytest обучаем любой классификатор, например, нейронную сеть (НС) или SVM [4].

```
# Define the fully connected neural network
classifier = Sequential()
classifier.add(Dense(64, input_dim=difference_flat.shape[1], activation='relu'))
classifier.add(Dense(32, activation='relu'))
classifier.add(Dense(16, activation='relu'))
classifier.add(Dense(1, activation='sigmoid'))

# Compile the classifier
classifier.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Train the classifier
classifier.fit(difference, y_test1, epochs=30, batch_size=32, validation_split=0.2)
```

Рисунок 2 – Классификатор ошибки реконструкции (детектор аномалий расхода топлива в автомобиле)

7. Оцениваем с использованием различных метрик [3,4] качество классификатора на выборке Xтест2-> Yтест2.

На рисунке 1 приведено описание на языке Python архитектуры автокодировщика, на рисунке 2 архитектуры нейронной сети – детектора аномалий расхода топлива в автомобиле.

С помощью разработанной системы нейронных сетей были получены результаты, приведенные на рисунках 3 и 4. На рисунке 3 – классификационный отчет о качестве работы всей системы автокодировщик – классификатор. На рисунке 4 среднеквадратическая ошибка автокодировщика на обучающей и тестовой выборке.

Выводы. Полученные результаты экспериментов при различных выборках показали целесообразность использования предложенной в работе методики.

Эта методика была реализована в программном продукте. На рисунке 5 приведен пример интерфейса пользователя, где показан график расхода топлива во время движения автомобиля. На нем разными цветами выделены нормальные и аномальные ситуации.

```
# Evaluate the classifier
print("Classification Report:")
print(classification_report(y_test2, y_pred))

print("Confusion Matrix:")
print(confusion_matrix(y_test2, y_pred))
```

```
157/157 [==============================] - 1s 5ms/step
157/157 [==============================] - 1s 4ms/step
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      4178
           1       1.00      0.99      0.99       822

    accuracy                           1.00      5000
   macro avg       1.00      0.99      1.00      5000
weighted avg       1.00      1.00      1.00      5000

Confusion Matrix:
[[4178    0]
 [  12  810]]
```

Рисунок 3 – Оценка качества работы системы автокодировщик – классификатор с помощью классификационного отчета

```
from sklearn.metrics import mean_squared_error

# Calculate MSE for the training set
train_mse = mean_squared_error(X_train_norm, autoencoder.predict(X_train_norm))

# Calculate MSE for the test set
test_mse = mean_squared_error(X_test1, autoencoder.predict(X_test1))

print(f'Training MSE: {train_mse}')
print(f'Test MSE: {test_mse}')
```

```
659/659 [==============================] - 2s 3ms/step
625/625 [==============================] - 1s 2ms/step
Training MSE: 1.0676915564379076e-05
Test MSE: 0.013623861466003062
```

Рисунок 4 – Среднеквадратическая ошибка автокодировщика на обучающей и тестовой выборке

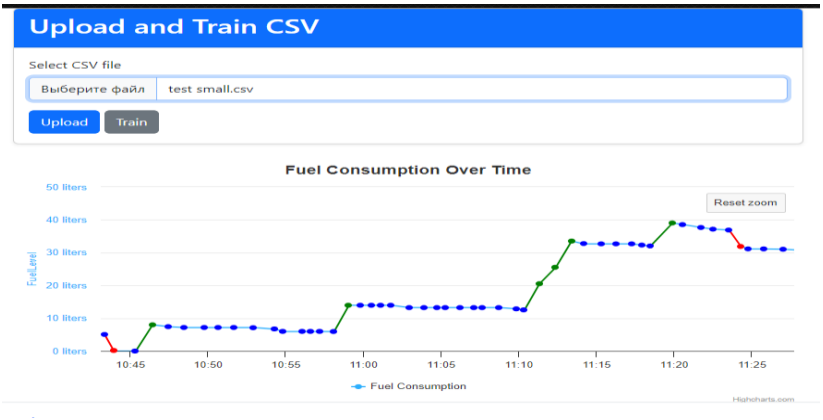


Рисунок 5 – Расход топлива во время движения автомобиля

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1.Детектирование аномалий с помощью автоенкодеров на Python. https://habr.com/ru/articles/491552/

2.Autoencoders and anomaly detection https://fraud-detection-handbook.github.io/fraud-detection-handbook/Chapter_7_DeepLearning/Autoencoders.html

3. Цуканова Н. И., Александров В. В., Головкин Н. В., Шурыгина О.В. Применение искусственных нейронных сетей и машинного обучения к оценке качества Коллективно-договорных актов в сфере образования// Вестник РГРТУ. 2023.№ 86. С.122-132

4. Демидова Л.А., Морошкин Н.А. Аспекты разработки архитектуры вопросно-ответной системы для обработки больших данных на основе нейросетевого моделирования// Вестник Рязанского государственного радиотехнического университета. 2023. № 86. С.145-155

5. Цуканова Н.И., Шитова К.Г. Поиск допустимых значений параметров заявки на кредитование с помощью нейронной сети. // Вестник Рязанского государственного радиотехнического университета. 2021. № 78. С.89-101

УДК 004.891.3

П. А. Баранчиков, К. А. Чадакин

ВЫЯВЛЕНИЕ АНОМАЛИЙ В ПОТОКАХ ЧИСЛОВЫХ ДАННЫХ С ПОМОЩЬЮ АЛГОРИТМОВ ИЕРАРХИЧЕСКОЙ ВРЕМЕННОЙ ПАМЯТИ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

В статье описывается способ выявления аномалий в потоках числовых данных с применением иерархической памяти, рассматривается алгоритм обучения иерархической памяти, а также ее преимущества и недостатки.

Ключевые слова: выявление аномалий, потоки числовых данных, алгоритм обучения иерархической памяти.

В области анализа данных и машинного обучения важным вопросом является выявление аномалий в числовых данных, особенно при их быстром и объемном производстве. Алгоритмы иерархической временной памяти (HTM), моделирующие неокортекс головного мозга, предлагают эффективное решение этой проблемы, особенно в анализе временных последовательностей. Применение HTM для мониторинга работы трансформаторов демонстрирует его потенциал в обнаружении аномалий.

В мониторинге и выявлении сбоев выделяют два основных подхода машинного обучения: классификацию и кластеризацию. Классификация предсказывает класс объекта на основе его характеристик и требует размеченных данных. Кластеризация разделяет данные на заранее неизвестные классы. Для обучения моделей решению задач классификации разметка данных не требуется.

Основные методы кластеризации включают

- Метод K-средних минимизирует сумму квадратов расстояний между объектами и центроидами их кластеров. Метод чувствителен к выбросам и выбору начальных центроидов.

- Иерархическая кластеризация строит дендрограмму, не требуя задания числа кластеров заранее. Метод имеет высокую вычислительную сложность.

- DBSCAN определяет кластеры как области высокой плотности объектов и эффективно выявляет кластеры произвольной формы. Производительность метода зависит от выбора параметров ε и minPts.

- Спектральная кластеризация основывается на спектральных свойствах матрицы смежности графа, хорошо работает для сложных структур данных. Метод требует вычислительно дорогого вычисления собственных значений.

Иерархическая временная память (HTM) воспроизводит свойства коры головного мозга, обучаясь на изменяющихся входных данных и запоминая паттерны и их последовательности. Основная концепция HTM – разреженное распределенное представление, где активна лишь небольшая доля нейронов, что позволяет эффективно представлять информацию через комплексные паттерны нейрональной активности и улучшает устойчивость к шуму.

Математическое описание разряженного представления (SDR, Sparse Distributed Representation) следующее. Пусть имеется популяция из N нейронов, их мгновенная активность представляется как SDR, то есть N-мерный вектор из бинарных компонентов, например, $x = [b_0, \dots, b_{n-1}]$. Обычно эти векторы являются сильно разреженными, то есть небольшой процент компонентов равен 1. w_x обозначает количество компонентов в x, которые равны 1, то есть $w_x = \|x\|_1$.

Над разряженными представлениями определены следующие операции и свойства.

Перекрытие (Overlap). Определяет сходство между двумя кодировками SDR. Показатель перекрытия — это количество битов, которые включены в одних и тех же позициях в обоих векторах. Если x и y — это два бинарных SDR, то перекрытие можно вычислить как скалярное произведение:

$$overlap(x, y) \equiv x \cdot y$$

Обратите внимание, что типичные метрики расстояния, такие как Хэммингово или Евклидово расстояние, не используются для количественной оценки сходства. С помощью перекрытия можно вывести некоторые полезные свойства, которые не будут соблюдаться при использовании этих метрик расстояния.

Два SDR совпадают (matching), если результат перекрытия превышает заданное пороговое значение θ.

$$match(x, y) \equiv overlap(x, y) \geq \theta$$

Значение θ подбирается таким образом, что $\theta \leq w_x$ и $\theta \leq w_y$.

Рассмотрим пример двух SDR векторов.

$x = [0100000000000000000010000000000000110000000]$

$y = [1000000000000000000010000000000000110000000]$

Оба вектора имеют размер $n = 40$ и $w = 4$. Перекрытие между x и y равно 3, и таким образом, вектора совпадают при $\theta = 3$.

При известных n и w, количество уникальных SDR-кодировок можно вычислить по формуле:

$$\binom{n}{w} = \frac{n!}{w!\,(n-w)!}$$

Далее описывается как операций SDR реализованы в **процессе пространственного объединения** (SP, Spatial Pooling process), с векторными операциями. Пусть X – бинарный входной вектор длины n, W

– матрица соединений синапсов, Y – выходной вектор, а inh – функция торможения. Тогда процесс пространственного объединения можно выразить следующим уравнением.

$$Y = \mathrm{inh}(X \times W)$$

Процесс SP принимает на вход X. На практике этот вектор обычно разреженный, но это не обязательно. Каждая колонка в SP представляет собой проксимальные сегменты в клетке. Для представления набора соединенных синапсов в SP можно использовать бинарную матрицу W размером $n \times c$. Эта матрица не обязательно должна быть разреженной, но на практике почти всегда таковой является. Результатом умножения вектора на матрицу является вектор перекрытий. На этапе торможения выбираются выигравшие колонки для формирования выхода таким образом, что индексы, соответствующие k наибольшим перекрытиям, соответствуют активным битам в выходном SDR. В итоге выходной вектор Y (SDR) имеет размер $1 \times c$, в котором k компонент равны одному.

При отсутствии обучения процесс SP анализирует перекрытие между набором случайно инициализированных колонок и отдельными бинарными входными векторами. Колонки с k наибольшими перекрытиями выигрывают и формируют активные биты в результирующем SDR. Этот SDR затем используется в качестве входа для последующих процессов TM (Temporal Memory).

Процесс построения **временной памяти** (TM) рассматривается в двух фазах. Первая фаза вычисляет активные состояния на временном шаге t, а вторая предсказывает, какие состояния будут активными на следующем временном шаге $t + 1$.

Первая фаза TM следует за процессом SP и определяет текущее активное временное состояние. На этом этапе рассматриваются клетки в каждой колонке, предсказанные на предыдущем временном шаге, и фактически выигравшие колонки в выходном SDR SP. Клетки в выигравших колонках, которые были правильно предсказаны, остаются активными, чтобы представлять активное временное состояние системы. Опишем уравнение процесса построения временной памяти. Пусть Y' – выходная матрица результатов временной памяти; Y – вектор размера n, представляющий результат выполнения операции пространственного объединения (SP); P – матрица предсказания состояний размером $l \times c$. Построчное поэлементное умножение (обозначенное символом $\circ$) этих двух матриц используется для определения активного состояния, которое также имеет размер $l \times c$. Все три структуры являются разреженными. Уравнение процесса временной памяти имеет следующий вид.

$$Y \circ p = Y'$$

Во второй фазе TM определяются предсказания на следующий временной шаг путём сопоставления сегментов отдельных клеток с текущим активным состоянием. Рисунок 6 показывает схему этой

операции. Каждая клетка содержит список сегментов, где каждый сегмент представлен SDR размером $l \times c$. Для каждого сегмента TM вычисляет совпадение между этим представлением и текущим активным состоянием. Если совпадение любого сегмента превышает порог, предсказанное состояние для этой клетки становится активным.

Для исследования алгоритма обучения иерархической временной памяти был использован набор данных телеметрии, собранных с трансформаторов [4]. Данные, собранные с помощью IoT-устройств в период с 25 июня 2019 года по 14 апреля 2020 года с обновлением каждые 15 минут, включают множество параметров, среди которых напряжение, сила тока и сопротивление. Для исследования была выбрана величина сопротивления. График участка данных для этой величины представлен на графике (Рисунок).

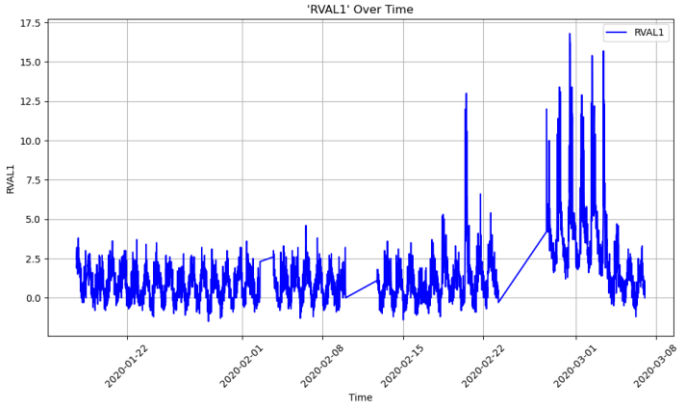


Рисунок 1 – График изменения сопротивления трансформатора

С помощью фреймворка nupic на языке Python была построена модель временной памяти, способная анализировать текущее состояние и выявлять процент «похожести» на предыдущие состояния. Эта величина в контексте фреймворка называется anomaly score. На вход моделе подается изменяемая величина – RVAL1 и метка времени. RVAL1 была закодирована с помощью ScalarEncoder с параметрами n = 50 и w = 21. Метка времени закодирована с помощью DateEncoder.

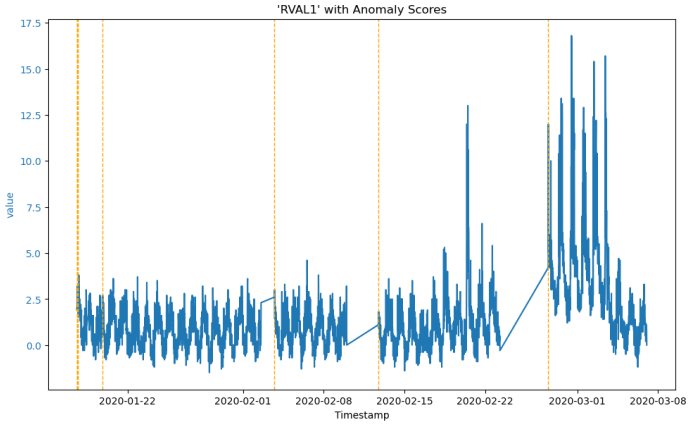


Рисунок 2 – График величины RVAL1 с метками аномалий

Результаты представленные на графике Рисунок отображает исходные данные и вертикальные метки, в которых модель определила аномалию с вероятностью более 50%.

В проведенном исследовании был использован набор данных телеметрии, собранных с трансформаторов, для анализа и оценки производительности алгоритма обучения иерархической временной памяти (HTM). Этот алгоритм, реализованный с использованием фреймворка nupic на языке Python, позволил анализировать изменения величины сопротивления трансформаторов и выявлять аномальные ситуации.

Графики, представленные в статье, демонстрируют результаты выявления аномалий на основе алгоритма HTM. Использование anomaly score позволило определить точки, в которых наблюдались необычные изменения в данных, что может быть важным для диагностики и предотвращения возможных аварийных ситуаций.

Таким образом, результаты исследования подтверждают эффективность и потенциал алгоритма HTM в анализе и мониторинге временных данных, что открывает перспективы его применения в различных областях, включая промышленность, энергетику и многие другие.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Предсказания от математиков. Разбираем основные методы обнаружения аномалий. URL: https://habr.com/ru/companies/lanit/articles/447190/

2. И. Чубукова. Задачи Data Mining. Классификация и кластеризация. URL: https://intuit.ru/studies/courses/6/6/lecture/166?page=4

3. Subutai, Ahmad. Properties of Sparse Distributed Representations and their Application to Hierarchical Temporal Memory. Redwood City, 2015. – стр. 18

4.Distributed Transformer Monitoring. URL: https://www.kaggle.com/datasets/sreshta140/ai-transformer-monitoring?select=Power.csv

УДК 004.65

Е.С. Щенёв, А.Н. Пылькин, Ю.Б. Щенёва, Е.В. Гавзова

ОСОБЕННОСТИ РАЗРАБОТКИ ИНТЕЛЛЕКТУАЛЬНЫХ ПРОГРАММНЫХ СРЕДСТВ АНАЛИЗА РЕЗУЛЬТАТОВ ОСВОЕНИЯ КОМПЕТЕНЦИЙ

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань
Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный университет имени С.А. Есенина»

В работе рассматриваются особенности разработки интеллектуальных программных средств анализа результатов освоения компетенций. Рассмотрена технология изменения эмоций на фотографиях с помощью нейронных сетей.

Ключевые слова: технология параллельного визуального отображения, адаптация метода «Лица Чернова», нейронные сети, алгоритмы машинного обучения.

Повсеместная цифровизация общества диктует необходимость в написании и поддержке высококачественного программного обеспечения. С ростом количества функций и методов автоматизированной системы освоения знаний повышается уровень использования ее компонентов. Отображение визуальных компонентов, демонстрирующих определенные результаты или процессы, является немаловажным фактором для наглядности и взаимодействия пользователя с используемой системой. Ярким примером таких компонентов являются элементы искусственного интеллекта, представляющие самообучаемые алгоритмы, отражающие предельную ясность предмета или субъекта исследуемой системы. Алгоритмы искусственного интеллекта могут превосходить способности человека во многих сферах, использующих обработку данных [1].

Зачастую студенты во время прохождения тестирования испытывают дискомфорт и волнение, возникающие в результате отсутствия уверенности в правильности промежуточных результатов. После проведения большого числа апробаций систем тестирования, транслирующих результаты прохождения экзаменационных тестов параллельно, было выявлено, что такой опыт способствует снижению

уровня стресса студентов, а также улучшению тенденции правильности прохождения тестирования в целом.

В результате этого было принято решение о внедрении технологии параллельного визуального отображения прогресса прохождения в существующую систему тестирования. Метод «Лица Чернова» должен быть адаптирован под современные реалии, а именно – с помощью применения искусственного интеллекта для задач распознавания эмоций на фотографии, а также дальнейшего преобразования их в положительную или отрицательную сторону.

Адаптация указанного метода заключается в изменении начальных, промежуточных и итоговых средств демонстрации результатов прохождения тестирования студентом. Вместо привычных для метода «смайлов», включающих в себя определенные состояния образующих лица, используются реальные фотографии студентов. В зависимости от позитивной или, наоборот, негативной прогрессии, строящейся в зависимости от ответов на каждый вопрос тестирования, изменяется и фотография студента. Применение такого метода позволяет оптимизировать процесс оценивания статистических данных, полученных во время прохождения тестирования.

Распознавание эмоций на фотографиях – это сложный процесс, который включает в себя анализ изображений для определения эмоционального состояния человека на фото. Для этого используются различные методы машинного обучения и компьютерного зрения. Один из таких методов – выделение ключевых точек на лице, которые могут указывать на определённую эмоцию [2].

Процесс начинается с того, что алгоритм обнаруживает лицо на фотографии. Затем он определяет координаты ключевых точек на лице, таких как глаза, нос, рот и брови. Эти точки помогают определить выражение лица и его эмоциональное состояние, что демонстрирует рисунок 1.

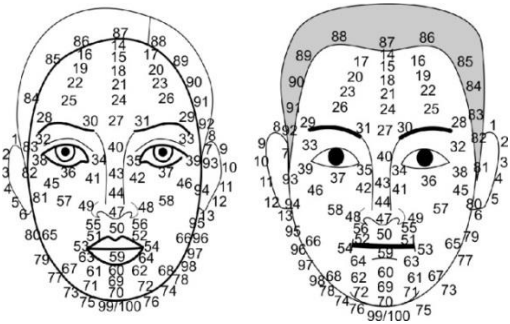


Рисунок 1 – Представление ключевых точек на изображении и определение их координат

После того как ключевые точки выделены, алгоритм анализирует их расположение и форму для определения типа эмоции. Например, если глаза широко открыты и рот слегка приоткрыт, то это может указывать на удивление или страх. Если же глаза сужены, и рот плотно закрыт, то это может указывать на гнев или отвращение.

Технология распознавания эмоций на фотографиях с помощью выделения ключевых точек является мощным инструментом для анализа эмоционального состояния людей. Она может использоваться в различных областях, включая маркетинг, психологию и медицину.

Например, в маркетинге такая технология может использоваться с целью помочь продавцам понимать, какие продукты или услуги вызывают положительные эмоции у потребителей. В психологии может помочь выявить пациентов с депрессией или другими психическими расстройствами. А в медицине может помочь врачам лучше понимать эмоциональное состояние своих пациентов.

Технология *изменения эмоций на фотографиях* с помощью нейронных сетей представляет собой процесс, в котором алгоритмы машинного обучения анализируют изображение лица, определяют его эмоциональное состояние, а затем изменяют это состояние, чтобы создать новую эмоцию.

Один из способов, которым нейронные сети могут изменить эмоцию на фотографии – использование описанной выше технологии распознавания ключевых точек на лице. Метод предполагает определение координат глаз, носа, рта и бровей, которые являются важными индикаторами эмоционального состояния человека. После того как ключевые точки выделены, алгоритм может анализировать их расположение и форму для определения типа эмоции, а также изменить эту эмоцию, перемещая ключевые точки в новые позиции. Технология изменения эмоций на фотографиях с помощью нейронных сетей может быть использована для создания новых эмоций или изменения существующих на основе первоначально определенной эмоции на фотографии.

Существует несколько популярных нейронных сетей, которые способны использовать технологию изменения эмоций на фотографиях с помощью выделения ключевых точек:

1. FaceApp –приложение для редактирования фотографий, которое использует нейронные сети для изменения возраста, пола и эмоций на лицах людей.

2. DeepFaceLab – инструмент для создания синтетических изображений лиц, который использует глубокое обучение для генерации реалистичных изображений.

3. DALL-E – модель искусственного интеллекта, которая может создавать изображения на основе текстовых описаний и изменять эмоции на лицах.

4. Nvidia GauGAN –инструмент для создания изображений, который использует генеративно-состязательные нейросети (GAN) для генерации реалистичных пейзажей и фотопортретов.

В качестве основы при разработке адаптированного метода «Лица Чернова» была использована модель для редактирования изображений в реальном времени. Модель работает на основе алгоритмов машинного обучения, которые анализируют изображение и определяют эмоциональное состояние человека на фото. Исходный код позволяет преобразовать модель, способную только определять эмоции на фотографии, к модели, изменяющей эти эмоции к нужным.

Исходная модель включает в себя функционал распознавания эмоции на фотографии, выделения ключевых точек на ней, а также координат каждой из этих точек. Для того чтобы изменять эмоции в «положительную» или «отрицательную» сторону необходимы не только координаты изначальных точек, но и векторы для последующего перемещения этих точек в определенные направления, зависящие от выбранной эмоции.

В качестве основной определяющей для построения каждого вектора к определенной ключевой точки являются параметры, которые подаются на вход модели помимо изначального изображения лица. Эти параметры необходимы для той математической модели, что позволяет рассчитывать все векторы для каждой точки.

Подводя итог вышесказанному, можно заключить следующее, разработанный программный продукт и методы отображения визуальных компонентов, реализованные с помощью элементов искусственного интеллекта, демонстрирующие результаты системы тестирования, наглядно отображают результат взаимодействия пользователя с сетевой информационной системой.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Гавзова Е.В., Щенёв Е.С., Щенёва Ю.Б. Разработка программного и методического обеспечения электронного дистанционного учебного курса с применением элементов искусственного интеллекта контроля знаний // Информатика и прикладная математика. 2022. № 28. С. 27-32.

2. Щенёва Ю.Б., Бодров О.А., Бубнов С.А., Майков К.А. Повышение эффективности управления организационным процессом подготовки IT-специалистов в вузе // Математическое и программное обеспечение вычислительных систем. Межвузовский сборник научных трудов. Рязанский государственный радиотехнический университет. Рязань, 2022. С. 10-13.

СОДЕРЖАНИЕ

МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Межвузовский сборник научных трудов

Подписано в печать 28.06.2024.
Формат 60x84 1/16. Бумага офсетная.
Печать цифровая. Гарнитура «Liberation Serif».
Печ. л. 12,875. Тираж 100 экз. Заказ № 7293.

Рязанский государственный радиотехнический
университет имени В.Ф. Уткина
390005, г. Рязань, ул. Гагарина, д. 59/1

Издательство ИП Коняхин А.В.

Отпечатано в типографии Book Jet
390005, г. Рязань, ул. Пушкина, д. 18
Сайт: http: //bookjet.ru
Почта: info@ bookjet.ru
Тел.: +7(4912) 466-151

ISBN 978-5-907811-43-0