Российская Академия Наук

Автоматика и телемеханика

Журнал основан в 1936 году
Выходит 12 раз в год

3
МАРТ
Москва 2024

Учредители журнала:
Отделение энергетики, машиностроения, механики и процессов управления РАН,
Институт проблем управления им. В.А. Трапезникова РАН (ИПУ РАН),
Институт проблем передачи информации им. А.А. Харкевича РАН (ИППИ РАН)

Адрес редакции: 117997, Москва, Профсоюзная ул., 65
Тел./факс: 8 (495) 198-17-20, доб. 1443
Электронная почта: redacsia@ipu.ru

Зав. редакцией *Е.А. Мартехина*

Москва
«Издательство «Наука»

Тематический выпуск

Вступительное слово к специальному выпуску по материалам XXV Международной конференции DAMDID/RCDL-2023

Специальный выпуск журнала “Автоматика и телемеханика” Российской академии наук посвящен научным результатам, представленным на XXV Международной конференции «Аналитика и управление данными в областях с интенсивным использованием данных» (DAMDID/RCDL-2023), прошедшей 24–27 октября 2023 г. в НИУ ВШЭ (Москва, Россия). Читателю предлагаются полные тексты избранных статей, соответствующих тематике журнала и представленных на DAMDID/RCDL-2023.

В 2015 г. в результате трансформации конференции RCDL (“Электронные библиотеки: перспективные методы и технологии, электронные коллекции”) была образована конференция DAMDID для создания форума, посвященного насущным проблемам анализа и управления данными в ходе исследований в различных областях с интенсивным использованием данных. Данная реорганизация была связана с расширением сферы использования электронных библиотек, в частности с активизацией их применения в научных исследованиях, возникновением потребности в технологиях эффективного хранения, обработки и анализа научных данных, интеграции неоднородных данных из множества источников. В итоге преобразования была обеспечена не только тематическая преемственность конференции по отношению к RCDL, но также сохранено RCDL-сообщество, сформировавшееся в течение 16 лет успешной работы данной конференции.

Конференция DAMDID известна как мультидисциплинарный форум исследователей и практиков из различных областей науки и промышленности, содействующий сотрудничеству и обмену идеями и методами в сфере анализа и управления данными, развиваемыми в конкретных областях X-информатики ($X \in \{$*астро, био, гео, нейро, мед, физика, химия, материаловедение, социо, бизнес, финансы,* $\ldots\}$).

Тематика конференции включает различные треки группы анализа данных, решения задач и организации экспериментов:

- постановки и решение задач в областях с интенсивным использованием данных (методы, процессы, инструменты);
- организация экспериментов (теории, гипотезы, модели, имитационное моделирование, инфраструктуры, потоки работ);
- методы и процедуры анализа данных (разведочный анализ, статистика и машинное обучение, мета-анализ, эффективность и масштабируемость методов);

- концептуальное моделирование предметных областей и представление знаний (семантика, онтологии, схемы баз данных, концептуализация алгоритмов и потоков работ, семантическая интероперабельность);
- поддержка исследований в инфраструктурах данных (функции и архитектуры виртуальных лабораторий и обсерваторий, совместное использование данных в междисциплинарных исследованиях).

В программу конференции также традиционно включена тематика управления данными:

- методы, инструменты и инфраструктуры сбора, хранения, обработки данных (данные и метаданные, их семантика, качество и происхождение, очистка, обнаружение аномалий);
- интеграция данных (модели данных, схемы, разрешение сущностей и слияние данных, виртуальная и материализованная интеграция, хранилища данных, ETL-процессы, многомерные модели данных, разноструктурированные данные);
- извлечение информации из данных наблюдений (полнота и актуальность информации в астрономии, спектроскопии, материаловедении, медицине и т.д.);
- извлечение информации из текстов;
- исследовательские инфраструктуры и их применение (облака, гриды, распределенные кластеры, суперкомпьютеры; реализация, масштабирование и оценка производительности инфраструктур; новые модели программирования, виртуализация);
- роль семантического веба в областях с интенсивным использованием данных.

Программа включала три ключевых и три приглашенных пленарных доклада, из них четыре — научных и два — индустриальных.

Доклады были распределены по следующим секциям и направлениям:

- алгоритмы решения задач;
- анализ данных в медицине и когнитивных науках;
- анализ изображений;
- анализ социальных сетей;
- анализ данных в астрономии;
- базы данных в материаловедении;
- извлечение информации из текстов;
- искусственный интеллект и машинное обучение в материаловедении;
- концептуальные модели, онтологии и семантический веб;
- методы машинного обучения и их приложения;
- моделирование в науках о Земле.

Программный комитет конференции, включающий ученых из Великобритании, Германии, Индии, Италии, Испании, Китая, Латвии, России, Сербии, США, Японии, рассмотрел 79 заявок на конференцию. В результате одностороннего слепого рецензирования, проводившегося в два этапа, включая обсуждение итогов рецензирования членами программного комитета, 53 ра-

боты были приняты как полные статьи, 15 — как короткие, 11 — отклонены или сняты.

В конференции приняли участие более 120 человек. Были проведены три пленарных заседания и заседания 18 секций. Заслушано 6 пленарных докладов и 61 секционный доклад. Участники представляли научные и учебные заведения из 11 регионов России: Казани, Москвы, Мурома, Новосибирска, Обнинска, Самары, Санкт-Петербурга, Томска, Тюмени, Челябинска, а также зарубежных стран: Армении, Великобритании, Германии, Индии, Ирана, Канады, Китая, Пакистана, ОАЭ, США и Японии.

Как и в предыдущие годы, доклады, не вошедшие в специальные выпуски журналов, будут опубликованы в серии Communications in Computer and Information Science издательством Springer.

На основе рекомендаций программного комитета по итогам рецензирования в соответствии с тематикой журнала и с учетом предпочтений авторов для публикации в специальном выпуске было отобрано 8 статей:

- *Б.Г. Миркин, А.А. Паринов* “Агломеративный консенсусный кластер-анализ с автоматическим выбором числа кластеров”;
- *М. Сохраби, А. Фатоллахи-Фард, В.А. Громов* “Алгоритм генной инженерии (GEA): эффективный метаэвристический алгоритм для решения задач комбинаторной оптимизации”;
- *Т.М. Биджиев, Д.Е. Намиот* “Атаки на модели машинного обучения, основанные на фреймворке PyTorch”;
- *М.М. Зуева, С.О. Кузнецов* “Индексы интересности как инструмент отбора формальных понятий для построения нейронной сети на основе решетки формальных понятий”;
- *Е.Д. Вязилов, Д.А. Мельников* “Об использовании данных цифровых двойников в моделях, связанных с учетом воздействия окружающей среды на предприятия”;
- *К.А. Найденова, В.А. Пархоменко* “Правдоподобные рассуждения в алгоритме генерации хороших классификационных тестов”;
- *Д.А. Люткин, Д.В. Поздняков, А.А. Соловьев, Д.В. Жуков, М.Ш.И. Малик, Д.И. Игнатов* “Применение трансформеров для определения профильного врача на основе запросов пользователей”;
- *А.Г. Сорока, Г.В. Михельсон, А.В. Мещеряков, С.В Герасимов* “Smart Routes: система для разработки и сравнения алгоритмов решения задачи оптимизации маршрутов с реалистичными ограничениями”.

В центре внимания авторов отобранных статей лежат такие темы, как методы машинного обучения и анализа данных, правдоподобные рассуждения и нейросети, алгоритмы управления и оптимизации, анализ текстов и моделирование на основе цифровых двойников.

Ж. Башариас, Барселона
Д.И. Игнатов, Москва
С.О. Кузнецов, Москва
С.А. Ступников, Москва

© 2024 г. Б.Г. МИРКИН, д-р техн. наук (bmirkin@hse.ru)
(Национальный исследовательский университет
“Высшая школа экономики”, Москва;
университет Лондона, Биркбек),
А.А. ПАРИНОВ, (aparinov@hse.ru)
(Национальный исследовательский университет
“Высшая школа экономики”, Москва)

АГЛОМЕРАТИВНЫЙ КОНСЕНСУСНЫЙ КЛАСТЕР-АНАЛИЗ С АВТОМАТИЧЕСКИМ ВЫБОРОМ ЧИСЛА КЛАСТЕРОВ[1]

Представлены теоретические и вычислительные результаты, связанные с оригинальной моделью консенсусного кластерного анализа, основанной на так называемом проективном расстоянии между разбиениями. Это расстояние определяется как сумма квадратов элементов разности бинарной матрицы инциденций одного разбиения и ее ортогональной проекции на подпространство, порождаемое столбцами матрицы инциденций другого разбиения. Оказывается, при достаточном количестве разбиений предлагаемый метод агломеративного кластеринга правильно вычисляет не только консенсусное разбиение, но число кластеров в нем.

Ключевые слова: консенсусный кластер-анализ, проективное расстояние, консенсусная матрица, агломеративный кластер-анализ, средне-взвешенный критерий.

DOI: 10.31857/S0005231024030014, **EDN:** UCGYKT

1. Введение

Проблема консенсусного кластерного анализа состоит в следующем. Задана некоторая совокупность разбиений данного множества объектов, иногда называемая кластерным ансамблем. Требуется сформировать некое “усредненное” разбиение, наиболее согласованное с имеющейся совокупностью. Впервые эта проблема была сформулирована как математическая задача аппроксимации в работе [1] с использованием введенного им расстояния между разбиениями в связи с предложенным им общим подходом к анализу данных неколичественного вида. Аксиоматическая характеризация расстояния Миркина была опубликована в данном журнале в статье [2]. Оказалось, что использование этого расстояния в задаче аппроксимации не совсем адекватно, так как не удовлетворяет так называемому тесту Мучника (см. раздел 7.6.4 в [3]). Поэтому в [3, 4] предложена более адекватная мера близости между

[1] Статья выполнена при поддержке Российского научного фонда (проект № 22-11-00323) в НИУ ВШЭ, Москва, Россия. Исследование выполнено с использованием суперкомпьютерного комплекса НИУ ВШЭ.

разбиениями, называемая в данной работе проективным расстоянием, которая до сего времени не подвергалась эмпирической верификации. Интерес к данной проблематике со стороны международного сообщества пробудился уже в новом веке с публикацией статей [5, 6]. В этих и последующих статьях по консенсусному кластер-анализу (см., например, обзоры в [7, 8]) мотивацией послужила вполне практическая и насущная проблематика. Дело в том, что кластер-анализ широко используется во многих практических приложениях — маркетинге, банковском деле, биоинформатике, искусственном интеллекте и пр. Между тем, результаты применения алгоритмов зависят от параметров, задаваемых пользователем "на глазок", таких как количество кластеров или порог значимости расстояния. Поэтому возникает совокупность более или менее равноважных кластерных разбиений, полученных при различных параметризациях, и, следовательно, задача консенсусного кластерного анализа.

Цель данной статьи — представить и обосновать метод формирования консенсусного разбиения, основанный на использовании проективного расстояния между разбиениями. Метод использует иерархическую схему агломеративного кластер-анализа на основе так называемой консенсусной матрицы связей между объектами со следующими особенностями: (1) сдвиг величин связи так, чтобы сумма всех связей стала нулевой; (2) использование средневзвешенной внутренней связи в качестве максимизируемого критерия; (3) обнуление диагональных элементов матрицы связи после каждого объединения. Критерий средневзвешенной внутренней связи реализует цель минимизации суммарного проективного расстояния между консенсусным разбиением и заданным ансамблем. Проективное расстояние между двумя разбиениями определяется как сумма квадратов элементов разности двух матриц: матрицы инциденций разбиения из данного ансамбля и его ортогональной проекции на линейное подпространство, порожденное матрицей инциденций искомого консенсусного разбиения [3, 9]. Вычислительный эксперимент основан на новом генераторе ансамбля "синтетических" разбиений. Генератор включает вероятность "мутации", которая определяет и разнообразие генерируемых разбиений и их близость к исходному "истинному" разбиению. В качестве конкурентных алгоритмов используются наиболее популярные схемы максимизации суммарных внутренних связей, включая так называемый критерий модулярности [10] и алгоритм Лувен [4].

2. Консенсусная матрица и методы ее сдвига

При заданном ансамбле разбиений $R_1, \ldots, R_M$ на множестве N объектов I консенсусное разбиение обычно формируется с использованием так называемой консенсусной матрицы размерности $N \times N$, $A = (a_{ij})$, (i, j)-й элемент которой, a_{ij}, определяется как количество таких разбиений ансамбля, R_m $(m = 1, \ldots, M)$, в которых и i, и j принадлежат одному и тому же классу $(i, j \in I)$. Любое разбиение R множества I можно взаимнооднозначно пред-

ставить через ее бинарную $N \times N$ матрицу смежности $r = (r_{ij})$, в которой $r_{ij} = 1$, если i и j находятся в одном и том же классе R, и $r_{ij} = 0$ в противном случае. Как известно, $A = \sum_m r_m$, где r_m – матрица смежности R_m $(m = 1, \ldots, M)$.

Элементы консенсусной матрицы выражают степень сходства между объектами согласно заданным разбиениям R_m $(m = 1, \ldots, M)$. Наиболее сходными являются те объекты, которые входят в один и тот же класс во всех разбиениях ансамбля. Для таких объектов консенсусная связь равна $a_{ij} = M$. Напротив, самые отличающиеся объекты — те, которые входят в разные классы во всех разбиениях ансамбля без исключения. Для них $a_{ij} = 0$. Элементы матрицы A неотрицательны. Для дальнейшего анализа полезно преобразовать эту матрицу так, чтобы часть элементов стала отрицательной, а среднее значение связи стало равным нулю.

В литературе предложено два способа такого преобразования: сдвиг модулярности [10] и сдвиг шкалы [3], определяемые следующим образом.

- **Сдвиг модулярности.** Это преобразование использует понятие случайного взаимодействия. Суммарные связи $a_{i.} = \sum_j a_{ij}$ и $a_{.j} = \sum_i a_{ij}$ рассматриваются как "энергетические заряды" строки i и столбца j соответственно. Случайное взаимодействие зарядов выражается их произведением, точнее величиной $a_{i.}a_{.j}/a_{..}$, где $a_{..} = \sum_i a_{i.} = sum_j a_{.j}$ — суммарный заряд, так чтобы взаимодействие также выражалось в единицах заряда. Это приводит к следующему преобразованию модулярности:

  $$a_{ij} \leftarrow a_{ij} - a_{i.}a_{.j}/a_{..}, \tag{1}$$

  поэтому нетрудно доказать, что после применения сдвига модулярности сумма всех связей, а значит, и их средняя величина, становится равной нулю.

- **Сдвиг шкалы**

  Это преобразование состоит в вычитании из всех элемнтов матрицы одной и той же пороговой величины, равной величине средней связи $\overline{a} = \frac{\sum\limits_{i,j} a_{ij}}{N^2} = \frac{a_{..}}{N^2}$:

  $$a_{ij} \leftarrow a_{ij} - \overline{a}. \tag{2}$$

  Конечно, после этого сдвига нуля шкалы в точку среднего и само среднее, и суммарная связь становятся тоже нулевыми.

Пример. Рассмотрим множество, состоящее из шести объектов $I = \{1, 2, 3, 4, 5, 6\}$ и пять разбиений на этом множестве, представленных в табл. 1 цифровыми метками классов.

Консенсусная матрица для этих данных представлена в табл. 2.

Таблица 1. Пять разбиений на множестве шести объектов, представленные цифровыми метками классов в соответствующих столбцах

№	R1	R2	R3	R4	R5
1	1	1	2	3	3
2	1	1	1	3	2
3	1	2	1	2	2
4	2	2	2	2	3
5	2	2	2	1	1
6	2	3	2	1	1

Таблица 2. Консенсусная матрица пяти разбений, представленных в табл. 1

	1	2	3	4	5	6
1	5	3	1	2	1	1
2	3	5	3	0	0	0
3	1	3	5	2	1	0
4	2	0	2	5	3	2
5	1	0	1	3	5	4
6	1	0	0	2	4	5

Таблица 3. Консенсусная матрица (слева), матрица случайных взаимодействий (в середине) и результат ее модулярного сдвига (справа)

	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
1	5	3	1	2	1	1	2,22	1,88	2,05	2,39	2,39	2,05	2,78	1,12	–1,05	–0,39	–1,39	–1,05
2	3	5	3	0	0	0	1,88	1,59	1,74	2,03	2,03	1,74	1,12	3,41	1,26	–2,03	–2,03	–1,74
3	1	3	5	2	1	0	2,05	1,74	1,89	2,21	2,21	1,89	–1,05	1,26	3,11	–0,21	–1,21	–1,89
4	2	0	2	5	3	2	2,39	2,03	2,21	2,58	2,58	2,20	–0,39	–2,03	–0,21	2,42	0,42	–0,21
5	1	0	1	3	5	4	2,39	2,03	2,21	2,58	2,58	2,21	–1,39	–2,03	–1,21	0,42	2,42	1,79
6	1	0	0	2	4	5	2,05	1,74	1,89	2,21	2,21	1,89	–1,05	–1,74	–1,89	–0,21	1,79	3,11

Таблица 4. Консенсусная матрица после сдвига шкалы

	1	2	3	4	5	6
1	1,96	–0,04	–2,04	–1,04	–2,04	–2,04
2	–0,04	1,96	–0,04	–3,04	–3,04	–3,04
3	–2,04	–0,04	1,96	–1,04	–2,04	–3,04
4	–1,04	–3,04	–1,04	1,96	–0,04	–1,04
5	–2,04	–3,04	–2,04	–0,04	1,96	0,96
6	–2,04	–3,04	–3,04	–1,04	0,96	1,96

В следующей табл. 3 представлена и сама эта матрица, и рассчитанная на ее основе матрица случайных взаимодействий, и результат ее модулярного преобразования.

Для получения матрицы связей со сдвигом шкалы, нужно подсчитать среднюю связь, 3.04, между объектами и вычесть ее из всех элементов консенсусной матрицы (см. табл. 4).

Сравнивая табл. 3 и 4, нельзя не заметить, что модулярный сдвиг оставляет положительными значительно больше связей, чем сдвиг шкалы. В правой части табл. 3 имеются две строки с тремя положительными элементами каждая — это строки 2 и 5. Напротив, в матрице после сдвига шкалы остается только один внедиагональный положительный элемент (см. табл. 4). Это важно потому, что только положительные связи могут “заставить” объекты объединяться в одном кластере.

3. Критерии разбиения

Из различных критериев кластер-анализа, опубликованных в литературе, рассмотрим два критерия, использующих внутрикластерные связи. В качестве связей рассматриваются элементы консенсусной матрицы, полученной сдвигом шкалы или модулярности. Один из этих критериев — это сумма связей внутри классов разбиения. Иными словами, для всякого разбиения $R = \{R_1, \ldots, R_K\}$ объектов множества I на K классов/кластеров внутренняя связь в кластере R_k $(k = 1, \ldots, K)$ рассчитывается как сумма всех связей a_{ij} для таких пар (i, j), в которых оба объекта, и i, и j, принадлежат R_k. Таким образом, суммарная внутренняя связь в R выражается формулой

$$f(R) = \sum_{k=1}^{K} \sum_{i,j \in R_k} a_{ij}. \tag{3}$$

Задача состоит в том, чтобы найти разбиение R, максимизирующее эту величину. Очевидно, что при неотрицательных связях a_{ij} данный критерий ведет к тривиальному решению: максимум суммы внутренних связей достигается на разбиении, состоящем из единственного универсального кластера, включающего все объекты. Именно поэтому предварительное преобразование связей путем сдвига шкалы или модулярности оказывается существенным. Следует отметить, что популярный критерий модулярности разбиения [10, 13] есть не что иное, как суммарная внутренняя связь (3) после сдвига модулярности [3].

Другой рассматриваемый здесь критерий — это средневзвешенная внутренняя связь [3]:

$$g(R) = \sum_{k=1}^{K} \frac{\sum\limits_{i,j \in R_k} a_{ij}}{N_k}, \tag{4}$$

где N_k – это количество объектов в кластере R_k.

Критерий (4) похож на критерий (3). Разница в том, что суммы связей внутри кластеров делятся здесь на их численности. По-другому говоря, в качестве оценки внутрикластерных связей здесь выступает произведение средней внутренней связи и его численности, что и объясняет название критерия. Критерий средневзвешенной внутренней связи возникает в контексте аппроксимирующего кластер-анализа [12]. Хотя он и не обязательно ведет к универсальному кластеру как оптимальному решению при неотрицательных связях, все равно полезно его применять после сдвига связей к нулевому среднему, как видно далее. Именно этот критерий является наиболее адекватным в задаче консенсусного кластер-анализа, как будет объяснено в разделе 5.

4. Агломеративные алгоритмы

Имеется несколько подходов, обычно применяемых для получения локально-оптимальных решений: объединение (агломерация) малых кластеров в большие, разделение больших кластеров на меньшие части, последовательное формирование кластеров по одному, обмен объектов из разных кластеров и т.п. Здесь рассматриваются только агломеративные алгоритмы, поскольку предполагается показать, что этот подход, примененный к средневзвешенному критерию (4), вполне эффективен для достижения цели консенсусного кластер-анализа.

Ниже показаны три агломеративных алгоритма для кластер-анализа.

Первый из них — обычный агломеративный алгоритм применительно к критерию суммарной связи (3).

Алгоритм AgSu:

1. Инициализация. В качестве начального принимается разбиение на синглтоны – кластеры, состоящие из одного объекта каждый, $S = \{S_1, \ldots, S_N\}$, $S_k = \{k\}$, $k = 1, \ldots, N$. Определяем матрицу связей между ними, $B = (b_{ij})$, как равную $A = (a_{ij})$, так что связь b_{ij} между синглтонами $\{i\}$ и $\{j\}$ равна a_{ij}.

2. Общий шаг. При заданном разбиении $S = \{S_1, \ldots, S_m\}$ множества объектов I и $m \times m$ матрице $B = (b_{st})$ суммарных связей между кластерами $(s, t = 1, \ldots, m)$ найти максимальную величину $b_{s*t*} = \max_{s,t} b_{st}$. Если $b_{s*t*} > 0$, следует соединить кластеры S_{s*} и S_{t*} в объединенный кластер $S_{s*} \leftarrow S_{s*} \cup S_{t*}$ и пересчитать величины связей путем арифметического прибавления строки $t*$ к строке $s*$: $b_{s*t} \leftarrow b_{s*t} + b_{t*t}$ для всех $t = 1, \ldots, m$, после чего подобным же образом сложить столбцы: $b_{ss*} \leftarrow b_{ss*} + b_{st*}$ для всех $s = 1, \ldots, m$. После этого удалить строку и столбец $t*$ из матрицы B и уменьшить m на 1. Если $b_{s*t*} < 0$, вычисления прекращаются. Если $m < 3$, тоже стоп.

Этот алгоритм действительно локально-оптимальный: на каждом шаге объединение производится оптимальным образом, поскольку увеличивает значение критерия максимально возможным образом. Действительно, разность между значениями критерия (3) после объединения кластеров s и t и до этого равна $2b_{st}$.

Сформулируем агломеративный алгоритм, предназначенный для максимизации средневзвешенного критерия (4). Единственное отличие от алгоритма AgSu состоит в том, что теперь объединение кластеров должно максимизировать критерий (4), а не (3).

Рассмотрим какое-нибудь разбиение $S = \{S_1, \ldots, S_m\}$ и разбиение $S(s,t) = \{S_1, \ldots, S_{s-1}, S_s \cup S_t, \ldots, S_m\}$, полученное объединением классов S_s и S_t в этом разбиении. Разность между $g(S(s,t))$ и $g(S)$ может быть выражена следующим образом:

$$\Delta(s,t) = g(S(s,t)) - g(S) = \frac{2b_{st} - N_t b_s s/N_s - N_s b_t t/N_t}{N_s + N_t}, \tag{5}$$

где b_{st}, b_{ss}, b_{tt} – элементы матрицы B суммарных связей между кластерами или внутри кластеров. Эта формула доказывается элементарными преобразованиями формулы для разности $g(S(s,t)) - g(S)$.

Алгоритм AgSa:

1. Инициализация. В качестве начального принимается разбиение на синглтоны – кластеры, состоящие из одного объекта каждый, $S = \{S_1, \ldots, S_N\}$, $S_k = \{k\}$, $k = 1, 2, \ldots, N$. Определяем матрицу связей между ними, $B = (b_{ij})$, как равную $A = (a_{ij})$, так что связь b_{ij} между синглтонами $\{i\}$ и $\{j\}$ равна a_{ij}.

2. Общий шаг. При заданном разбиении $S = \{S_1, \ldots, S_m\}$ множества объектов I и $m \times m$ матрице $B = (b_{st})$ суммарных связей между кластерами $(s, t = 1, \ldots, m)$, найти максимальную величину $\Delta(s*, t*) = \max_{s,t} \Delta(s,t)$ согласно (5). Если $\Delta(s*, t*) > 0$, следует соединить кластеры S_{s*} и S_{t*} в объединенный кластер $S_{s*} \leftarrow S_{s*} \cup S_{t*}$ и пересчитать величины связей путем арифметического прибавления строки $t*$ к строке $s*$: $b_{s*t} \leftarrow b_{s*t} + b_{t*t}$ для всех $t = 1, \ldots, m$, после чего подобным же образом сложить столбцы: $b_{ss*} \leftarrow b_{ss*} + b_{st*}$ для всех $s = 1, \ldots, m$. После этого удалить строку и столбец $t*$ из матрицы B и уменьшить m на 1. Если $\Delta(s*, t*) < 0$, вычисления прекращаются. Если $m < 3$, тоже стоп.

Алгоритмы AgSu и AgSa имеют то преимущество, что после сдвига число кластеров определяется автоматически — объединение кластеров прекращается, как только все внедиагональные величины b_{st} для алгоритма AgSu, или $\Delta(s,t)$ для алгоритма AgSa, становятся отрицательными. В этом случае никакое дальнейшее объединение кластеров не может увеличить значения критерия.

Как хорошо известно, агломеративные алгоритмы имеют тот недостаток, что используют сравнительно медленные вычисления — число шагов при поиске максимального элемента матрицы имеет порядок N^2, особенно на первых шагах. Некоторые усилия были предприняты по сокращению вычислений за счет использования таких свойств критериев кластеризации, которые позволяют применить результаты предыдущих вычислений [18]. Позже появилась работа [5] с революционной идеей, что вовсе нет никакой нужды в изнуряющем поиске максимума в матрице. Авторы использовали суммарный критерий (3) после сдвига модулярности, чтобы сформулировать и обосновать свою идею, названную ими Лувенский алгоритм: давайте возьмем один элемент s пары (s,t) случайным образом, так что максимум b_{st} определяется только перебором t порядка N, не N^2. Конечно, эту идею можно использовать с любым критерием, не только (3). Далее сформулируем Лувенский алгоритм применительно к произвольному критерию $c(R)$, который надо максимизировать на множестве всех разбиений множества I. Будем считать, что вычисление разности $\Delta c(s,t) = c(S(s,t)) - c(S)$ — несложная операция.

Лувенский алгоритм GAL.

1. Инициализация. Рассматривай тривиальное разбиение на N одиночных кластеров $R = \{\{1\}, \ldots, \{N\}\}$ в качестве начального разбиения.

2. Общий шаг. При заданном разбиении $R = \{R_1, \ldots, R_K\}$ организуй последовательный просмотр кластеров в произвольном порядке $1, \ldots, K$.

2.1. Для каждого конкретного s найти $t*$, максимизирующий разность $\Delta c(s,t)$ по всем $t = 1, \ldots, K$.

2.2. Объедини кластеры R_s и R_{t*}, в качестве R рассматривай разбиение $R(s,t*)$ и уменьши K на 1.

2.3. Проверка: Если $\Delta c(s,t*) < 0$ или $K < 3$, то останов. В противном случае переходи к следующему кластеру.

2.4. Если все кластеры пройдены, начинай шаг 2 с текущим R.

Определим одну дополнительную операцию, которую можно выполнять перед началом работы агломеративного алгоритма и/или на любом его шаге:

ZD: Обнуление диагональных элементов матрицы связи.

Согласно этой операции каждый диагональный элемент текущей матрицы связей B заменяется нулем.

Было проверено, стоит ли выполнять ZD перед каждым агломеративным шагом. Оказалось, что это помогает в алгоритме AgSa и не помогает — точнее, ухудшает результат — при других рассматривавшихся алгоритмах. Поэтому далее применяем ZD перед каждым шагом объединения в алгоритме AgSa, а в других алгоритмах — только вначале.

5. Вычислительный эксперимент

В настоящее время эксперименты с методами консенсусного кластер-анализа носят, если так можно выразиться, опосредованный характер. Берется таблица данных, либо реальная, скажем из хранилища данных в Ирвайнском кампусе Университета Калифорнии, либо "синтетическая", содержащая кластерную структуру, порожденную тем или иным генератором кластерных структур. К этой таблице повторно применяется один кластерный алгоритм или несколько алгоритмов при различных, как правило случайных, значениях параметров алгоритма. Например, это может быть метод к-средних при случайных инициализациях и постоянном числе кластеров, равно одному в сгенерированной таблице данных. Результаты этих применений и образуют исходный ансамбль разбиений. Таким образом, проблематика консенсусного кластер-анализа здесь комбинируется со спецификой взятой таблицы данных и алгоритма или алгоритмов получения разбиений. Это порождает вопросы, связанные с качеством формируемых кластерных ансамблей, их репрезентативностью, их разнообразием, полнотой и прочее [11, 14, 19]. Тематика консенсусного кластер-анализа не имеет никакого отношения ни к качеству данных, ни к качеству алгоритмов кластеризации, используемых для получения ансамблей разбиений. Эксперименты должны быть организованы таким образом, чтобы генератор данных непосредственно генерировал ансамбль разбиений множества I так, чтобы и разноообразие ансамбля и его репрезентативность было легко контролировать.

Здесь предлагаем именно такую организацию вычислительного эксперимента для консенсусного кластер-анализа. Генератор "синтетических" данных начинает с генерации "истинного" разбиения. Этим процессом управляют три параметра: численность множества объектов N, количество классов в разбиении K, минимальный размер класса m. Этот последний параметр особенно полезен, когда в применяемых алгоритмах кластер-анализа используются вероятностные соображения. Для оценки параметров такого алгоритма может понадобиться не менее m элементов. Для выполнения условия распределяем mK объектов по классам, помещая в каждый ровно m объектов. Остающиеся $N - mK$ объектов случайно распределяются по K кластерам. Это можно сделать в Матлаб с помощью команды $randi(K, T)$, которая приписывает каждый из $T = N - Km$ объектов какому-то из K разных классов.

После того, как получено истинное разбиение S, генерируется ансамбль разбиений $R_1, \ldots, R_M$ для его представления. Для этого задается вероятность "мутации", p, $0 < p < 1$. Для генерации R_1 переназначаем $100p\%$ сущностей в любой случайно выбранный кластер. Другие разделы генерируются аналогично. Увеличивая p, увеличиваем разнообразие ансамбля. То, что такой ансамбль является репрезентативным для истинного раздела, следует из **генерации**.

Кому-то такой механизм мутации может показаться слишком упрощенным. Например, все разделы, созданные с его помощью, имеют то же количество кластеров, что и истинные. Действительно, можно предложить более сложные схемы мутации, например со случайными слияниями и разделениями кластеров исходных данных. Отметим, однако, хорошие свойства предложенного генератора данных. Во-первых, увеличивая вероятность мутации p, действительно можно создавать достаточно разнообразные разделы. Во-вторых, уменьшая число разделов в ансамблях, можно создавать действительно сложные ситуации для консенсусных алгоритмов кластеризации, например делая их число меньше, чем число частей в истинном разделе, $M < K$.

В последующих расчетах были использованы два значения количества объектов $N = 1000$ и $N = 3000$, три значения количества кластеров $K = 4$, $K = 9$ и $K = 15$, и два значения размера ансамбля разбиений $M = 40$ и $M = 10$. Они сведены в табл. 5, в которой также перечислены рассматриваемые алгоритмы.

Качество результатов оценивается по двум характеристикам: количеству полученных кластеров и ARI (Adjusted Rand index), индексу сходства между

Таблица 5. Параметры экспериментов

N	K	M	m	p	Типа сдвига данных	Алгоритм
1000	4	10	2	0,8	Модульный сдвиг	Агломеративный
3000	9	40			Сдвиг масштаба	Лувена
	15					

истинным и полученным алгоритмом разбиением [6]. ARI основан на количестве пар объектов, которые совпадают в сравниваемых разделах, т.е. либо принадлежат к одному кластеру, либо к разным кластерам в обоих разделах:

$$(6) \quad ARI\,(A,B) = \frac{\binom{N}{2} * \sum_{s=1}^{K_A}\sum_{t=1}^{K_B}\binom{n_{st}}{2} - \sum_{s=1}^{K_A}\binom{a_s}{2}\sum_{t=1}^{K_B}\binom{b_t}{2}}{\frac{1}{2}\binom{N}{2}\left[\sum_{s=1}^{K_A}\binom{a_s}{2} + \sum_{t=1}^{K_B}\binom{b_t}{2}\right] - \sum_{s=1}^{K_A}\binom{a_s}{2}\sum_{t=1}^{K_B}\binom{b_t}{2}}.$$

В (6) A и B – два разбиения множества сущностей с частями K_A и K_B соответственно; a_s и b_t – кардинальности частей в A и B соответственно; n_{st} – частоты в совместном распределении AB; $\binom{n}{2}$ – биномиальный член, равный $n(n-1)/2$.

Чем ближе значение ARI к единице, тем более похожи разделы; $ARI = 1,0$ показывает, что $A = B$. Если один из разделов состоит только из одной части, самого множества I, то $ARI = 0$. ARI может быть и отрицательным, что случается довольно редко, как, скажем, при специально определенных "дуальных" парах разделов [7].

После создания ансамбля разбиений и вычисления соответствующей консенсусной матрицы происходят вычисления одним из восьми вариантов обработки в зависимости от варианта преобразования матрицы (модульный сдвиг или сдвиг масштаба), используемого критерия (суммарный или средневзвешенный) и применяемого алгоритма (агломерация или Лувен). Результаты представлены в табл. 6–9 в зависимости от размера данных N, и ансамблей разбиений M.

Эти таблицы наглядно демонстрируют следующее.

1. Результаты при 1000 и 3000 объектах практически совпадают, это означает, что количество рассматриваемых объектов мало влияет на консенсусные решения.

Таблица 6. Результаты применения алгоритма консенсус кластеризации при $N = 1000$, $M = 40$

		Суммарный критерий				Средневзвешенный критерий			
		Лувен		Агломерация		Лувен		Агломерация	
		Масштаб	Модуль	Масштаб	Модуль	Масштаб	Модуль	Масштаб	Модуль
4	ARI	0,84/0,09	0,88/0,03	0,89/0,03	0,90/0,01	0,98/0,01	0,98/0,01	0,99/0,0	0,99/0,0
	#	3,8/0,44	4/0	4/0	4/0	8,6/1,1	9/1,4	4/0	4/0
9	ARI	0,44/0,03	0,43/0,04	0,45/0,01	0,50/0,03	0,99/0,0	0,99/0,00	1,0/0,0	1,0/0,0
	#	4,2/0,45	4,2/0,45	4/0	4,8/0,45	11,6/1,5	11,8/1,3	9/0	9/0
15	ARI	0,29/0,01	0,28/0,01	0,33/0,02	0,34/0,01	0,99/0,0	0,99/0,0	1/0	1/0
	#	4/0	4/0	4,8/0,45	5/0	17,4/0,5	17,6/0,89	15/0	15/0

Таблица 7. Результаты применения алгоритма консенсус кластеризации при $N = 3000$, $M = 40$

		Суммарный критерий				Средневзвешенный критерий			
		Лувен		Агломерация		Лувен		Агломерация	
		Масштаб	Модуль	Масштаб	Модуль	Масштаб	Модуль	Масштаб	Модуль
4	ARI	0,88/0,01	0,87/0,01	0,88/0,01	0,88/0,01	0,98/0,00	0,98/0,00	0,99/0,0	0,99/0,0
	#	4/0	4/0	4/0	4/0	12,2/1,1	13/1	4/0	4/0
9	ARI	0,40/0,02	0,40/0,03	0,43/0,05	0,41/0,02	0,99/0,0	0,99/0,00	1,0/0,0	1,0/0,0
	#	4,2/0,45	3,8/0,45	4,2/0,45	4/0	14,6/0,55	13,8/0,45	9/0	9/0
15	ARI	0,26/0,01	0,27/0,01	0,29/0,02	0,28/0,02	1,0/0,0	0,99/0,0	1/0	1/0
	#	4/0	4/0	4,4/0,55	4,4/0,55	19,4/1,1	19,8/1,3	15/0	15/0

Таблица 8. Результаты применения алгоритма консенсус кластеризации при $N = 1000$, $M = 10$

		Суммарный критерий				Средневзвешенный критерий			
		Лувен		Агломерация		Лувен		Агломерация	
		Масштаб	Модуль	Масштаб	Модуль	Масштаб	Модуль	Масштаб	Модуль
4	ARI	0,44/0,02	0,43/0,02	0,41/0,03	0,40/0,02	0,70/0,02	0,70/0,02	0,67/0,03	0,66/0,03
	#	3/0	4/0	4/0	3,2/0,45	13,8/0,84	14/1	4/0	4/0
9	ARI	0,18/0,02	0,16/0,01	0,21/0,05	0,21/0,02	0,73/0,01	0,79/0,01	0,73/0,01	0,74/0,01
	#	3/0	4/1	3/0	3,8/0,84	20,2/1,1	20,0/1,2	9/0	9/0
15	ARI	0,12/0,01	0,11/0,01	0,14/0,01	0,13/0,01	0,81/0,01	0,81/0,01	0,76/0,03	0,76/0,03
	#	3/0	4/0,71	3,4/0,55	4,2/0,45	26,2/1,9	26,2/1,3	14,2/0,84	14,4/0,55

Таблица 9. Результаты применения алгоритма консенсус кластеризации при $N = 3000$, $M = 10$

		Суммарный критерий				Средневзвешенный критерий			
		Лувен		Агломерация		Лувен		Агломерация	
		Масштаб	Модуль	Масштаб	Модуль	Масштаб	Модуль	Масштаб	Модуль
4	ARI	0,39/0,03	0,40/0,01	0,41/0,01	0,40/0,02	0,71/0,01	0,71/0,01	0,69/0,01	0,68/0,01
	#	3/0	3/0	3/0	3,2/0,45	12,2/3,63	13,4/2,88	4/0	4/0
9	ARI	0,16/0,01	0,16/0,02	0,17/0,01	0,17/0,02	0,79/0,01	0,79/0,01	0,72/0,01	0,72/0,01
	#	3/0	3,6/0,55	3/0	3,6/0,55	23,4/0,89	23,6/0,89	9/0	9/0
15	ARI	0,10/0,01	0,10/0,01	0,10/0,01	0,11/0,01	0,83/0,01	0,83/0,01	0,77/0,01	0,77/0,01
	#	3/0	4,4/0,89	3,4/0,55	4,6/0,55	30,0/1,9	31,6/2,7	15/0	15/0

2. Вопреки ожиданиям, результаты при двух разных нормализациях — модульной и со сдвигом масштаба — во многом схожи, так что вопрос о том, какую из них использовать, становится нерелевантным при консенсусной кластеризации.

3. Восстановление кластеров всегда лучше при использовании полусреднего критерия, а не суммарного. Чем больше число кластеров, тем больше разница.

4. Чем больше размер ансамбля, тем лучше: результаты восстановления данных при $M = 10$ значительно хуже, чем при $M = 40$. Особенно разрушительным является эффект при суммарном критерии, при котором уровень восстановления кластеров сохраняется в среднем на уровне ARI, равном 0,4 при $K = 4$, 0,2 при $K = 9$ и 0,1 при $K = 15$.

5. При $M = 40$ агломерация по полусреднему критерию приводит к идеальным результатам при $K = 9$, 15 и почти идеальным при $K = 4$. Лувенский алгоритм при полусреднем критерии достигает почти таких же хороших результатов в восстановлении кластеров. Однако он терпит неудачу в отношении количества кластеров. Напротив, при $M = 10$ алгоритм Лувена всегда выигрывает в восстановлении кластеров, хотя по-прежнему переоценивает их количество.

6. Среди рассматриваемых методов есть один, который всегда правильно восстанавливает количество кластеров: агломерация по критерию полусреднего. Он работает даже при уменьшении ARI до величины порядка 0,7. Единственный случай, в котором он может потерпеть неудачу, пусть и незначительную, — это случай $K = 15$, $M = 10$, т.е. $M < K$, при меньшем количестве объектов ($N = 1000$, но не при $N = 3000$).

6. Обсуждение

Некоторые из приведенных выше эмпирических результатов могут быть объяснены теоретическими соображениями, связанными с общей концепцией консенсусной кластеризации, основанной на индексах расстояния между разделами. Учитывая индекс $d(R, S)$, оценивающий несходство между любыми разделами R и S из I, можно определить понятие консенсусного раздела следующим образом. Если задан ансамбль разделов $R_1, R_2, \ldots, R_M$ из I, то консенсусным разделом является любой раздел R из I, который минимизирует суммарное расстояние $D(R) = \sum_{m=1}^{M} d(R_m, R)$. Обычно расстояние $d(R_m, R)$ определяется как расстояние несовпадения, или расстояние Миркина, между соответствующими $N \times N$ двоичными матрицами r_m и r, элементы которых $r_m(i, j)$ или $r(i, j)$ равны 1, если i и j находятся в одной части Rm или R соответственно; в противном случае они равны 0. Расстояние Миркина — это количество несовместимых пар (i, j), таких, что i и j находятся в одной части одного из разделов, а в другом разделе i и j принадлежат разным частям [17]. Очевидно, что это половина расстояния $L1$ между бинарными матрицами разделов. Нетрудно доказать, что консенсусным разделом с расстоянием Миркина является тот, который максимизирует суммарный критерий

$$F(R) = \sum_{k=1}^{K} \sum_{i,j \in R_k} \left(a_{ij} - \frac{M}{2} \right), \tag{7}$$

где a_{ij} элементы консенсусной матрицы для ансамбля $R_1, \ldots, R_M$. Действительно,

$$D(R) = \sum_{m=1}^{M} d(R_m,\ R) = \sum_{m=1}^{M} \sum_{i,j=1}^{N} |r_m(i,j) - \ r(i,j)|/2 =$$

$$= \sum_{i,j=1}^{N} \sum_{m=1}^{M} |r_m(i,j) - \ r(i,j)|/2. \tag{8}$$

очевидно, что внутренняя сумма равна

$$\sum_{m=1}^{M} |r_m(i,j) - r(i,j)| = \sum_{m=1}^{M} |r_m(i,j) - r(i,j)|^2 =$$

$$= \sum_{m=1}^{M} (r_m(i,j) + r(i,j) - 2r(i,j)\, r_m(i,j)) = a_{ij} + Mr(i,j) - 2a_{ij} r(i,j), \tag{9}$$

тогда как $\sum_{m=1}^{M} r_m(i,j) = \ a_{ij}$.

Эти манипуляции корректны, поскольку значения элементов $r(i,j)$ и $r_m(i,j)$ здесь равны нулю или единице, так что квадратичная операция оставляет их инвариантными. Отбросив первый элемент, a_{ij}, который здесь постоянен, и умножив остаток на $-1/2$, можно увидеть, что задача минимизации $D(R)$ действительно эквивалентна задаче максимизации $F(R)$ в (7).

Видно, что критерий (7) действительно является внутрикластерным суммарным критерием (3) с предварительным сдвигом сходства на $M/2$. К сожалению, у определенного таким образом консенсусного разбиения есть недостаток: оно не проходит так называемый тест Мучника [9]. Этот тест требует проверить для любого разбиения $T = T_1, \ldots, T_K$ на I является ли T консенсусным разбиением для ансамбля его K дихотомических представлений $T_k = T_k, I - T_k$ $(k = 1, 2, \ldots, K)$. Если да, то расстояние проходит тест; если нет, то расстояние не проходит тест.

Посмотрим, удовлетворяет ли консенсус расстояний Миркина этому тесту. Возьмем разбиение $T = T_1, \ldots, T_7$ с $K = 7$ частями, так что $K/2 = 3,5$. Рассмотрим записи матрицы консенсуса a_{ij} в этом случае. Предположим сначала, что i и j принадлежат одной и той же части T. Тогда они должны принадлежать одной и той же части в каждом T^m, поскольку каждая часть T содержится в любой части T^m. Это означает, что $a_{ij} = 7$ для таких i и j. Рассмотрим теперь, что i принадлежит, скажем, T_1, а j — другой части, скажем T_2. Тогда i и j принадлежат разным частям как в T_1, так и в T_2. Однако они принадлежат одной и той же части $I - T_3$ в T_3, потому что $I - T_3$ содержит и T_1, и T_2. Аналогично, эти i и j принадлежат одной и той же части в T^m для всех остальных $m = 4, 5, 6, 7$. Таким образом, $a_{ij} = 5$ для этих i

и j. Следовательно, все записи в матрице консенсуса здесь равны либо 5, либо 7, причем обе больше, чем $K/2 = 3,5$. Таким образом, для максимизации критерия (7) выгодно собрать все сущности в универсальном разбиении I, состоящем из единственной части самого I, но не из разбиения T. Следовательно, тест Мучника действительно провален.

Существует и другая мера расстояния, которая называется проективным расстоянием [11, 12]. Рассмотрим номинальный признак над множеством сущностей I, представленный разбиением $S = S_l$, и другой номинальный признак, представленный разбиением $R = R_k$. Определим $N \times L$ фиктивную матрицу Y, соответствующую разделу S, матрицу инцидентности раздела, приписав каждой категории S_l в S бинарную переменную y_l, фиктивную, которая является просто $1/0$ N-мерным вектором, элементы которого $y_{il} = 1$, если $i \in Sl$ и $y_{il} = 0$, в противном случае $(l = 1, \ldots, L)$. Аналогично определим $N \times K$ матрицу инцидентности X, столбцы которой x_k — $0/1$-векторы, соответствующие категориям S_k из S. Проективное расстояние определяется как суммарная квадратичная разность между Y и его ортогональной проекцией на линейное пространство, охватывающее столбцы X [12, 11]. Используя символ $\| \|^2$ для обозначения суммы квадратов (квадратичной нормы), проективное расстояние между R и S определяется по формуле $\delta(X, Y) = \|Y - P_X Y\|^2$, где P_X – ортогональный проектор $P_X = X(X^T X)^{-1} X_T$ на линейное пространство, охватывающее столбцы X. Заметим, что эта мера расстояния несимметрична. Точный смысл расстояния $\delta(X, Y)$ разобран в [11, с. 319]. Здесь сосредоточимся на суммарном расстоянии $\triangle(R) = \sum_{m=1}^{M} \delta(X, Y_m)$, которое должно быть минимизировано относительно неизвестного разбиения R, представленного матрицей X, для получения проективной дистанционной консенсусной кластеризации. Матрицы Y_m представляют здесь разделы R_m заданного ансамбля разделов.

Оказывается, эта задача эквивалентна задаче максимизации полусреднего критерия $g(R)$ в (4). Чтобы доказать это, рассмотрим матрицы инцидентности X и Y_m разделов R и R_m соответственно. Эти бинарные матрицы обозначают через 1 принадлежность объектов (строк) к кластерам. Обозначим общее число кластеров во всех разбиениях ансамбля $(m = 1, 2, \ldots, M)$ через L и сформируем $N \times L$ матрицу $Y = (Y_1, Y_2, \ldots, Y_M)$, состоящую из всех L столбцов этих матриц. Столбцы Y соответствуют всем кластерам в разбиениях $R_1, R_2, \ldots, R_M$. Тогда критерий $\triangle(R) = \sum_{m=1}^{M} \delta(X, Y_m)$ можно переформулировать как $\triangle(X) = \|Y - P_X Y\|^2$, или, что эквивалентно, как $\triangle(X) = Tr((Y - P_X Y)(Y - P_X Y)^T)$, где Tr обозначает след квадратной матрицы, что есть сумма ее диагональных элементов. Раскрывая скобки в последнем выражении, получается $\triangle(Y) = = Tr(YY^T - P_X YY^T - YY^T P_X + P_X YY^T P_X) = Tr(YY^T - P_X YY^T)$. Действительно, операция Tr коммутативна, так что $Tr(P_X YY^T) = Tr(YY^T P_X)$ и $Tr(P_X YY^T P_X) = Tr(P_X P_X YY^T) = Tr(P_X YY^T)$. Последнее уравнение следует из того, что $P_X P_X = P_X$, что легко доказать непосредственно. Заметим теперь, что матрица YY^T равна консенсусной матрице A. Очевидно,

что $a_{ii} = L$ для всех $i \in I$, так что $Tr(YY^T) = NL$. С другой стороны, (i,i)-й диагональный элемент матрицы $P_X A$ равен сумме произведений $p_{ij}a_{ij}$, где p_{ij} — либо 0, если i и j находятся в разных кластерах, либо $1/N_k$, если i и j принадлежат одному кластеру S_k. На этом доказательство завершено.

Теперь можно доказать, что консенсусное разбиение, определенное с помощью проективного расстояния, действительно проходит тест Мучника. Рассмотрим снова разбиение $T = T_1, \ldots, T_K$ на I и ансамбль его K дихотомических представлений $T^k = T_k, I - T_k$ $(k = 1, 2, \ldots, K)$. Матрица консенсуса A здесь состоит из $a_{ij} = K$, если i и j принадлежат некоторому T_k $(k = 1, 2, \ldots, K)$, или $a_{ij} = K - 2$, если i и j принадлежат разным частям T. Рассмотрим средневзвешенный критерий (4) для разбиения $R = R_1, R_2, \ldots, R_m$. Обозначим среднее сходство внутри R_k через a_k. Тогда значение (4), очевидно, равно сумме $N_k a_k$, где N_k — количество объектов в R_k. Максимальное значение a_k в этом случае равно K, и оно достигается, когда R_k входит в часть T, поскольку в этом случае все внутрикластерные значения $a_{ij} = K$. Если же, напротив, R_k пересекает несколько частей T, то некоторые внутрикластерные значения a_{ij} будут равны $K - 2$, так что $a_k < K$. Это доказывает, что максимальное значение критерия (4) в рассматриваемом случае равно NK (как сумма всех значений $N_k K$), и оно достигается при любом R, либо совпадающем с T, либо являющимся более гранулированной версией T, полученной путем деления некоторых его частей. Подтверждением полученных результатов можно считать доказанные факты: средневзвешенный критерий (4) воплощает хорошую концепцию консенсусной кластеризации с использованием проективного расстояния между разделами, тогда как суммарный критерий (3) относится к плохой концепции консенсуса кластеризации с использованием расстояния Миркина. Именно поэтому критерий (4) в экспериментах в подавляющем большинстве случаев превосходит критерий (3).

Также предстоит объяснить два других эмпирически наблюдаемых факта:

1. Почему столь разные преобразования данных, как сдвиг модульности и сдвиг масштаба, приводят к очень похожим результатам консенсусной кластеризации?

2. Почему эвристика постоянного обнуления диагональных записей настолько эффективна при определении нужного числа кластеров с помощью критерия полусреднего?

Следует отметить, что видимое "противоречие" между высокими значениями ARI и неправильным количеством кластеров (см. результаты Лувена для полусреднего критерия в табл. 6 и 7 выше) легко объясняется нечувствительностью индекса ARI к лишним мелким кластерам. Возьмем, например, разбиение R множества из 1000 человек на две равные по размеру части. Сделаем из одной из частей 20 одиночных кластеров и обозначим полученное таким образом разбиение I на 22 кластера через S. Индекс ARI между R и S равен 0,96, что не так уж далеко от единицы.

7. Заключение

Основной целью данной работы является выдвижение полусреднего критерия консенсусной кластеризации (4), модифицированного постоянным обнулением главной диагональной эвристики, в качестве критерия, который должен использоваться при консенсусной кластеризации для восстановления как скрытого разбиения, так и количества кластеров в нем. Отметим, что этот критерий возникает при консенсусной кластеризации со специально разработанной системой оценки несходства между разделами — проективным расстоянием. В отличие от традиционно используемых расстояния несовпадения или расстояния Миркина между разделами (см., например, в [5]), проективное расстояние, как показано, проходит естественный тест на валидность (тест Мучника). В представленных экспериментах агломеративная кластеризация с критерием (4) демонстрирует очень сильную тенденцию к восстановлению как скрытого разбиения, так и количества кластеров в нем. Сравнивается производительность этого метода с наиболее популярным методом кластеризации — кластеризацией по модулю. К сожалению, кластеризация по модульному принципу оказывается менее чем удовлетворительной и не должна применяться в качестве инструмента кластеризации по консенсусу. Другим вкладом данной работы является новый дизайн вычислительных экспериментов с консенсусными методами кластеризации. Вместо традиционных подходов к созданию ансамблей разделов, опосредованных наборами данных и применяемыми методами кластеризации, предлагается простой вероятностный механизм мутации для создания репрезентативного ансамбля разделов, разнообразие которого контролируется значением вероятности мутации. В эксперименты не включаются реальные наборы данных, такие как те, что находятся в знаменитом репозитории UC Irvine Machine Learning, поскольку нет прямых доказательств того, что признаки в этих наборах действительно связаны с истинными разделами. Будущая работа должна включать объяснение наблюдаемых странностей, разработку более реалистичных механизмов мутации и адаптацию подхода к большим наборам данных. Интересным направлением могут стать подходы, связанные с методами анализа формальных понятий (FCA) [16].

СПИСОК ЛИТЕРАТУРЫ

1. *Миркин Б.Г.* Об одном подходе к обработке нечисловых данных / Математические методы моделирования и решения экономических задач (Ред. К.А. Багриновский). Новосибирск, ИЭиОПП СО АН СССР, 1969. С. 141–150.
2. *Миркин Б.Г., Черный Л.Б.* Об измерении близости между различными разбиениями конечного множества объектов // АиТ. 1970. № 5. С. 120–127.
3. *Mirkin B.* Clustering: A Data Recovery Approach // Chapman and Hall, 2012. V. 19. https://doi.org/10.1201/9781420034912
4. *Миркин Б.Г., Мучник И.Б.* Геометрическая интерпретация показателей качества классификации / Методы анализа многомерной экономической информации (Ред. Б.Г. Миркин). Новосибирск. Наука, Сибирское отделение. 1981. С. 3–11.

5. *Strehl A., Ghosh J.* Cluster Ensembles — A Knowledge Reuse Framework for Combining Multiple Partitions // J. Machin. Learning Res. 2002. P. 583–617. https://doi.org/10.1162/153244303321897735
6. *Monti S., Tamayo P., Mesirov J., et al.* Consensus Clustering: A Resampling-Based Method for Class Discovery and Visualization of Gene Expression Microarray Data // Machine Learning. 2003. P. 91–118. https://doi.org/10.1023/A:1023949509487
7. *Ünlü R., Xanthopoulos P.* Estimating the number of clusters in a dataset via consensus clustering // Expert Syst. Appl. 2019. https://doi.org/10.1016/j.eswa.2019.01.074
8. *Alguliyev R., Aliguliyev R., Sukhostat L.* An efficient algorithm for big data clustering on a single machine // CAAI Transactions on Intelligence Technology. 2020. https://doi.org/10.1049/trit.2019.0048
9. *Liu P., Zhang K., Wang P., et al.* A clustering-and maximum consensus-based model for social network large-scale group decision making with linguistic distribution // Inform. Sci. 2022. P. 269–297.
10. *Newman M.E.* Modularity and community structure in networks // Proc. Nation. Acad. Sci. 2006. P. 8577–8582.
11. *de Amorim R.C., Shestakov A., Mirkin B., et al.* The Minkowski central partition as a pointer to a suitable distance exponent and consensus partitioning // Patt. Recognit. 2017. P. 62–72.
12. *Blondel V.D., Guillaume J.L., Lambiotte R., et al.* Fast unfolding of communities in large networks // J. Statist. Mechan.:Theory Experiment. 2008. No. 10. P. 10008–10016.
13. *Brandes U., Delling D., Gaertler M., et al.* On modularity clustering // IEEE Transaction. Knowledge. 2007. P. 172–188.
14. *Fern X., Lin W.* Cluster ensemble selection // Statist. Anal. Data Mining: The ASA Data Sci. J. 2008. No. 1. P. 128–141. https://doi.org/10.1002/sam.10008
15. *Guénoche A.* Consensus of partitions: a constructive approach // Advances in Data Analysis and Classification. 2011. No. 5(3). P. 215–229.
16. *Hubert L.J., Arabie P.* Comparing partitions // J. Classifikat. 1985. No. 2. P. 193–218.
17. *Kovaleva E.V., Mirkin B.G.* Bisecting K-means and 1D projection divisive clustering: A unified framework and experimental comparison // J. Classifikat. 2015. P. 414–442.
18. *Murtagh F., Contreras P.* Algorithms for hierarchical clustering: an overview // Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. 2012. No. 32. P. 86–97.
19. *Pividori M., Stegmayer G., Milone D.H.* Diversity control for improving the analysis of consensus clustering // Inform. Sci. 2016. No. 361. P. 120–134.
20. *Gnatyshak D., Ignatov D.I., Mirkin B.G., et al.* A Lattice-based Consensus Clustering Algorithm // CLA. CEUR Workshop Proceedings. 2016. V. 1624. P. 45–56.

Статья представлена к публикации членом редколлегии А.А. Галяевым.

Поступила в редакцию 08.07.2023

После доработки 21.10.2023

Принята к публикации 20.01.2024

© 2024 г. М. СОХРАБИ (msohrabi@hse.ru)
(Национальный исследовательский университет
“Высшая школа экономики”, Москва)
А.М. ФАТХОЛЛАХИ-ФАРД (fathollahifard.amirmohammad@courrier.uqam.ca)
(Университет Квебека в Монреале, Канада),
В.А. ГРОМОВ (stroller@rambler.ru)
(Национальный исследовательский университет
“Высшая школа экономики”, Москва)

АЛГОРИТМ ГЕНЕТИЧЕСКОЙ ИНЖЕНЕРИИ (GEA): ЭФФЕКТИВНЫЙ МЕТАЭВРИСТИЧЕСКИЙ АЛГОРИТМ ДЛЯ РЕШЕНИЯ ЗАДАЧ КОМБИНАТОРНОЙ ОПТИМИЗАЦИИ[1]

Генетические алгоритмы (ГА) известны своей эффективностью в решении задач комбинаторной оптимизации благодаря их способности исследовать разнообразные пространства решений, обрабатывать различные представления, использовать параллелизм, сохранять хорошие решения, адаптироваться к изменяющимся условиям, управлять комбинаторным разнообразием и проводить эвристический поиск. Тем не менее такие ограничения, как преждевременная сходимость, неспецифичность и стохастичность операторов кроссовера и мутации, делают ГА не всегда эффективными при нахождении глобального оптимума. Чтобы преодолеть эти недостатки, в данной статье предлагается новый метаэвристический алгоритм, названный алгоритмом генетической инженерии (GEA), вдохновленный концепциями генной инженерии. GEA модифицирует традиционный ГА, включая новые методы поиска для выделения, коррекции, вставки и экспрессии новых генов на основе существующих, что способствует появлению желаемых признаков и производству хромосом на основе выбранных генов. Сравнение с результатами работы других алгоритмов на стандартных примерах демонстрирует эффективность GEA.

Ключевые слова: генетический алгоритм, метаэвристические алгоритмы, генная инженерия, комбинаторная оптимизация.

DOI: 10.31857/S0005231024030027, **EDN:** UAGNKK

1. Введение

Задачи комбинаторной оптимизации, относящиеся к классу NP-сложных, требуют эффективных алгоритмов для поиска оптимальных или близких к оптимальным решений. Генетические алгоритмы (ГА) [1] здесь популярны

[1] Данная работа является результатом исследовательского проекта, реализованного в рамках программы фундаментальных исследований Национального исследовательского университета “Высшая школа экономики” (НИУ ВШЭ). Исследование выполнено с использованием суперкомпьютерного комплекса НИУ ВШЭ [27]. Хотелось бы отметить, что публикации профессора Мостафа Хаджиагайи-Кештели вдохновили авторов на разработку нового метаэвристического алгоритма.

благодаря их способности исследовать разные области пространства решений и адаптироваться к изменяющейся динамике. Однако ограничения ГА, включающие вычислительную сложность, преждевременную сходимость, недостаточный учет информации о конкретной задаче и необходимость тонкой настройки параметров, предполагают поиск новых подходов [2]. Эти недостатки стимулируют попытки авторов перепроектировать ГА, используя концепцию генной инженерии, чтобы сделать его более эффективным для решения задач комбинаторной оптимизации.

ГА являются одними из самых первых алгоритмов, предназначенных для нахождения удовлетворительных решений за разумные сроки вычислений, а не только на нахождение оптимальности [3]. Несмотря на наличие множества других метаэвристических алгоритмов [4–18], следует отметить, что литература по ГА весьма обширна, она включает множество исследований, представляющих разнообразные варианты ГА с различными техниками генетического программирования и инженерии [19–27]. Например, в [28] проведено исследование, нацеленное на выявление полезного генетического материала и минимизацию присутствия вредных генетических компонентов, что привело к разработке нового варианта ГА. В [29] предложен подход, основанный на идее защиты "хороших" хромосом от нежелательных кроссоверов. В [30] исследовалась интеграция нейронной сети обратного распространения (ВР) с ГА. В [31] разработана серия адаптивных элитарных популяционных стратегий, которые нашли применение в рамках генетических алгоритмов.

В [32] интегрирована стратегия балансировки нагрузки для облачных вычислений с ГА. В [33] проведено усовершенствование ГА путем внедрения нового многородительского оператора кроссовера. В [34] предложен улучшенный алгоритм DV-Hop на основе ГА, в то время как в [35] исследованы ГА, основанные на памяти. В [36] разработан гибридный ГА, сочетающийся с нечеткой логикой, а в [37] предложен гибрид ГА с другими новыми метаэвристическими алгоритмами. Кроме того, в [38] разработан модифицируемый генетический алгоритм с недоминируемой сортировкой, основанный на новых операторах поиска. В [39] представлен генетический алгоритм недоминируемой сортировки на основе локального поиска, адаптированный для решения задач маршрутизации в туристической индустрии. Однако ни одно из рассмотренных до сих пор исследований не предложило введение новых операторов поиска, основанных на наборе методов, направленных на изоляцию, очистку, вставку и экспрессию новых генов в существующие ГА-хромосомы, как в настоящем исследовании.

В последнее время для решения сложных задач оптимизации был предложен ряд новых популяционных алгоритмов. Здесь можно указать на алгоритмы поиска кукушки (Cuckoo Search, CS) [4], алгоритм оптимизации китов (Whale Optimization Algorithm, WOA) [5], синусно-косинусный алгоритм (Sine Cosine Algorithm, SCA) [6], оптимизацию ястребов Харриса (Harris Hawks Optimization, HHO) [7], алгоритм поиска белки (Squirrel Search Algorithm, SSA) [8], алгоритм красного оленя (Red Deer Algorithm,

RDA) [9], алгоритм поиска воробьев (Sparrow Search Algorithm, SSA) [10], алгоритм поиска капуцина (Capuchin Search Algorithm) [11], оптимизатор Aquila (Aquila Optimizer, AO) [12], алгоритм группы хамелеонов (Chameleon Swarm Algorithm, CSA) [13], оптимизация Aptenodytes Forsteri (AFO) [14], оптимизатор навозного жука (Dung Beetle Optimizer, DBO) [15], оптимизация белухи (Beluga Whale Optimization, BWO) [16] и др. Однако стоит отметить, что согласно теореме "Нет бесплатного обеда" [17] ни один метаэвристический алгоритм не может превзойти другие для всех задач оптимизации. Следовательно, существует постоянный спрос на разработку новых метаэвристических алгоритмов, способных увеличить производительность в различных проблемных областях [18].

В настоящей статье предложен новый метаэвристический алгоритм генетической инженерии (GEA), коррелирующий с принципами генной инженерии. Генная инженерия включает множество методов, применяемых для изоляции, очистки, вставки и экспрессии новых генов на базе существующих, что приводит к появлению желаемых признаков и хромосом на основе данного набора генов. На основе результатов из этой области в статье ставится цель переосмыслить процесс оптимизации и преодолеть ограничения, связанные с традиционными генетическими алгоритмами. Методы, применяемые в GEA, обеспечивают более точное управление процессом оптимизации, учитывают специфику рассматриваемой задачи и сокращают влияние случайности в операциях мутации и скрещивания. Введение концепции "манипуляция генами" в популяции направлено на более эффективный поиск в пространстве решений, что способствует улучшению сходимости и качества получаемых решений. Для оценки эффективности GEA были проведены широкомасштабные вычислительные эксперименты на ряде эталонных примеров; его производительность сравнивалась с другими метаэвристическими алгоритмами. Результаты указывают на высокую скорость сходимости алгоритма, качество получаемых им решений и его робастность, что подчеркивает его потенциал как нового и эффективного подхода к решению задач комбинаторной оптимизации.

Дальнейшее изложение организовано следующим образом: в разделе 2 представлены основные принципы алгоритма GEA. Раздел 3 посвящен деталям реализации GEA, включая описание операторов генной инженерии. В разделе 4 представлены описание проведенных вычислительных экспериментов и сравнительный анализ результатов работы GEA и других алгоритмов. Заключительный раздел 5 подводит итоги исследования и указывает на возможные направления будущих исследований.

2. Биологические основания алгоритма

В настоящее время генная инженерия (ГИ) перешла от теоретических исследований к практическим применениям. Этот подход продемонстрировал значительный потенциал в лечении различных заболеваний, например в им-

мунотерапии рака [19] и в технологии CRISPR для удаления вируса ВИЧ из геномов инфицированных клеток [20]. Методы генной инженерии оказывают влияние на генетику человека, меняя характеристики новорожденных [21], и обещают прорыв в сельском хозяйстве в создании высокоурожайных культур [22]. Разработка золотого риса, направленного на борьбу с дефицитом витамина А и предотвращение слепоты по всему миру, демонстрирует потенциал генной инженерии [23]. Точность в идентификации доминантных хромосом, отвечающих за урожайность растений, привела к созданию высокоурожайных видов, которые способствуют решению глобальных проблем, включая изменение климата, загрязнение окружающей среды и продовольственный дефицит [24].

Направленные мутации предполагают точные изменения ДНК в организмах с целью появления благоприятных изменений. Здесь специфические мутации вводятся в гены для стимулирования развития положительных признаков или подавления негативных. Разработка методов управления мутациями в генах, вызывающих болезни, позволяет создавать инновационные методы лечения генетических расстройств, таких как муковисцидоз и мышечная дистрофия. Для достижения этой цели необходимо выполнить два ключевых шага: идентификация ключевых генов и точное управление неинформативными генами [23, 25, 26].

Генная инъекция представляет собой новый подход, при котором терапевтические гены доставляются непосредственно в организм для лечения или профилактики заболеваний. Этот метод обладает перспективами в лечении таких заболеваний, как рак и сердечно-сосудистые расстройства. Введение генов, производящих терапевтические белки, позволяет ученым усиливать естественные защитные механизмы организма, стимулировать регенерацию тканей и оказывать целенаправленное воздействие на раковые клетки. Генная инъекционная терапия является значимым инструментом в области персонализированной медицины [25].

В заключение раздела отметим, что генная инженерия представляет собой революционное явление с многообразными приложениями в различных областях медицины, сельского хозяйства и охраны окружающей среды. Использование доминантных хромосом, направленных мутаций и идентификации необходимых генов позволяет сформировать генетические основы для выведения высокоурожайных сельскохозяйственных культур и разработки методов лечения заболеваний. Методы точного редактирование генов и генной инъекционной терапии позволяют достигать беспрецедентной точности и индивидуализации.

3. Предложенный метаэвристический алгоритм

ГА [1] представляет собой эволюционный алгоритм, использующий классические операторы мутации и скрещивания. В рамках данного исследования предложен новый подход, использующий методы генной инженерии (GE).

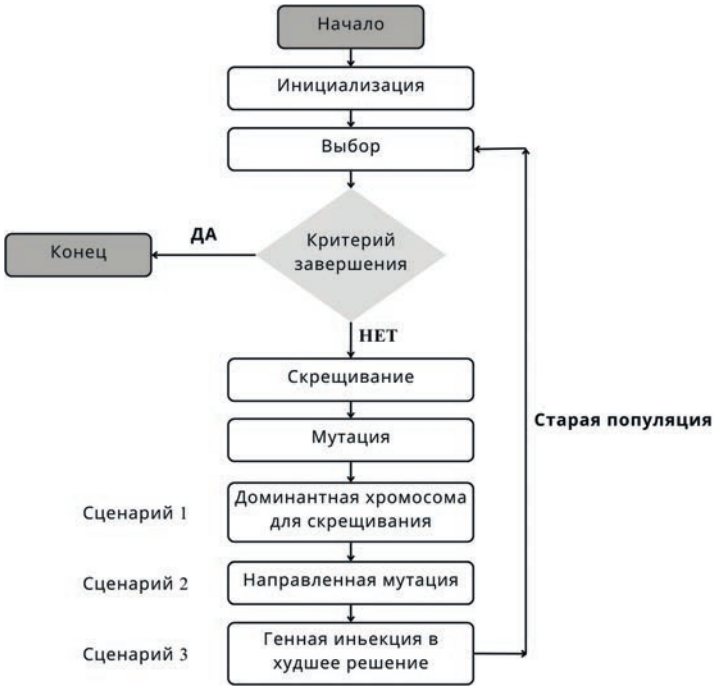
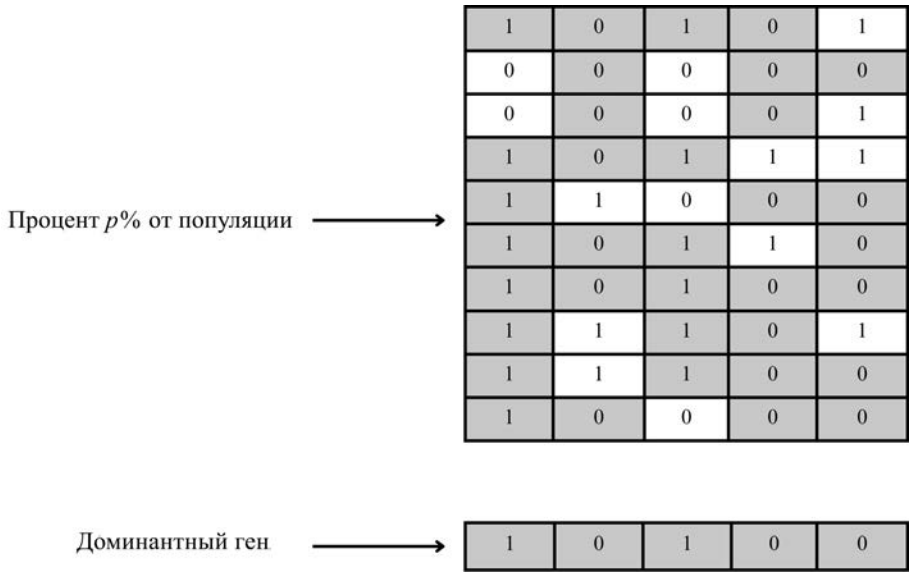


Рис. 1. Блок-схема предлагаемого ГЭА.

Общая блок-схема предлагаемого алгоритма (GEA) представлена на рис. 1. На приведенной схеме можно опустить любой оператор, от скрещивания до инъекции гена, с целью адаптации алгоритма к различным задачам и оценки эффективности метода при использовании части операторов. Подобно другим метаэвристикам GEA начинается с формирования начальной популяции. После формирования начальной популяции все особи оцениваются на основе специфичной для задачи функции приспособленности. Важно подчеркнуть, что каждая задача характеризуется уникальной функцией приспособленности, необходимой для представления решения в рамках эволюционного алгоритма. Например, в задаче маршрутизации хромосома определяется как последовательность посещений [2], в задачах планирования производственных процессов — как последовательность работ на машинах [18], в задаче о рюкзаке — с помощью бинарных переменных [3]. В GEA также предполагается, что хромосома бинарна, т.е. каждый ген может принимать значения ноль или один. В этом алгоритме, помимо операторов мутации и скрещивания, используются три оператора генной инженерии, представленные ниже в трех сценариях.

3.1. Сценарий 1

Поиск доминантной хромосомы для кроссовера (наиболее часто повторяющихся генов). Первый сценарий GEA направлен на выявление доминантной хромосомы, здесь учитываются только $p\%$ лучших особей в популяции. Зна-



1	0	1	0	1
0	0	0	0	0
0	0	0	0	1
1	0	1	1	1
1	1	0	0	0
1	0	1	1	0
1	0	1	0	0
1	1	1	0	1
1	1	1	0	0
1	0	0	0	0

1	0	1	0	0

Рис. 2. Поиск доминантных генов в популяции p%.

чение p задается пользователем до начала работы алгоритма и может быть оптимизировано с учетом специфики задачи. Хромосома считается доминантной, если она содержит наибольшее количество повторяющихся генов среди лучших p% особей. Процесс идентификации доминантных генов и формирования доминантной хромосомы описывается уравнениями (1) и (2). На рис. 2 представлен пример определения доминантной хромосомы; псевдокод данной операции представлен в алгоритме 1.

$$RM_i = \left[\Sigma_{j=1}^{M} gene_j\right], \tag{1}$$

$$DC = \max(RM), \tag{2}$$

где M, RM и DC обозначают количество особей в p% популяции, матрицу повторений и доминантную хромосому соответственно.

3.2. Сценарий 2

Направленная мутация. Второй сценарий в GEA направлен на увеличение эффективности оператора мутации, что предельно важно для предотвращения застревания алгоритма в локальном оптимуме. В традиционных ГА часто применяется случайная мутация, что, вероятно, является их недостатком; в GEA оператор мутации был модифицирован с целью управления процессом мутации. Здесь возможен поиск важных генов, с тем чтобы алгоритм “сосредоточился” на генах, способствующих развитию желаемых признаков или результатов. Направление процесса мутации и устранение случайности, связанной с традиционными мутациями, значительно повышает производительность генетического алгоритма. Подход позволяет алгоритму приоритизировать гены и вносить полезные изменения контролируемым образом. Тем

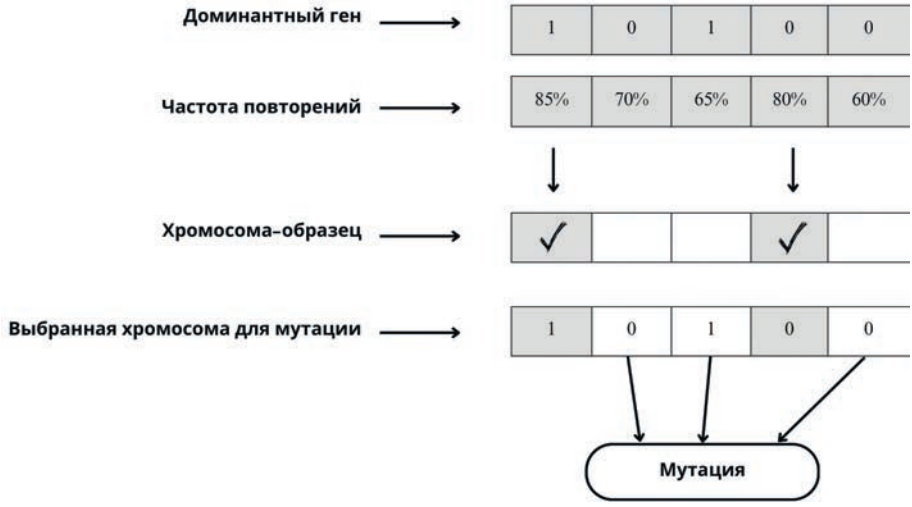


Рис. 3. Направленная мутация путем фиксации информативных генов.

самым, GEA преодолевает ограничения случайной мутации и повышает свою способность эффективно исследовать пространство решений. На рис. 3 представлен пример процесса направленной мутации.

$$f(x) = \begin{cases} 1, & \text{если } M_{ij}, \text{желаемая хромосома} \\ 0, & \text{в противном случае.} \end{cases} \tag{3}$$

- **Желательный ген.** Первый этап применения данного оператора заключается в поиске наиболее часто повторяющихся генов среди $p\%$ лучших хромосом, которые считаются желательными генами. Цель этого этапа — выделение части генов, являющихся наиболее информативными элементами для формирования элитной части популяции. Наличие этих генов в решении способствует удержанию популяции в элитной области и, при незначительных изменениях, может привести к приближению к глобальному оптимуму в ближайшем будущем. Здесь формируется матрица шаблонов размером $n * m$, где n — количество хромосом в элитной части популяции (лучшие $p\%$), а m — количество генов в хромосоме (число переменных в задаче). Матрица шаблонов состоит из бинарных элементов: 1 отвечает специфической части хромосомы, которая считается желаемой и должна быть зафиксирована, а 0 представляет неинформативные гены, подлежащие мутации. Уравнение (3) демонстрирует процесс формирования матрицы шаблонов. Если число повторений достигает определенного порога, ген становится желаемым. Пороговое значение устанавливается пользователем до начала работы алгоритма.
- **Мутация поиска желательных генов.** После формирования шаблона для $p\%$ хромосом с наивысшими значениями приспособленности кандидат выбирается с помощью метода колеса рулетки. Мутация применяется исключительно к неинформативным генам, которые обозначены нулем в соответ-

ствующей матрице шаблонов. Целью применения целевой мутации является эффективный поиск в пространстве решений для нахождения глобального оптимума. Специально разработанная мутация повторяется в рамках общего числа мутаций, но только в отношении неинформативных генов, что способствует инвестициям в элитную часть популяции и обеспечивает более быструю конвергенцию.

3.3. Сценарий 3

Инъекция генов. Третий сценарий в GEA подчеркивает значимость учета всей популяции, включая особи с наименьшими показателями пригодности. Здесь, в отличие от первых двух сценариев, ориентированных на элитную часть популяции, возможна ситуация, когда даже наименее успешные решения могут способствовать общему улучшению алгоритма. В алгоритмах оптимизации важно учитывать особи с наихудшими показателями, поскольку они также обладают потенциалом для положительных изменений. В данном сценарии осуществляется "иньекция" в наихудшие особи с использованием инженерного подхода для улучшения их производительности. Минимальные изменения в наихудших особях могут способствовать их движению к глобальному оптимуму и в конечном итоге включению в число элитных решений в последующих итерациях алгоритма.

Для реализации этой стратегии создается матрица паттернов для элитной части популяции. Затем из неэлитной части популяции (представляющей $1 - p\%$) отбираются особи. Исходя из матрицы паттернов, в выбранную хромосому инъектируются гены из хромосомы с наибольшим количеством повторений. Этот оператор инъекции генов способствует передаче полезной генетической информации от доминирующей хромосомы другим особям в популяции, позволяя им улучшиться и внести свой вклад в общий процесс оптимизации. На рис. 4 представлен пример, демонстрирующий механизм работы

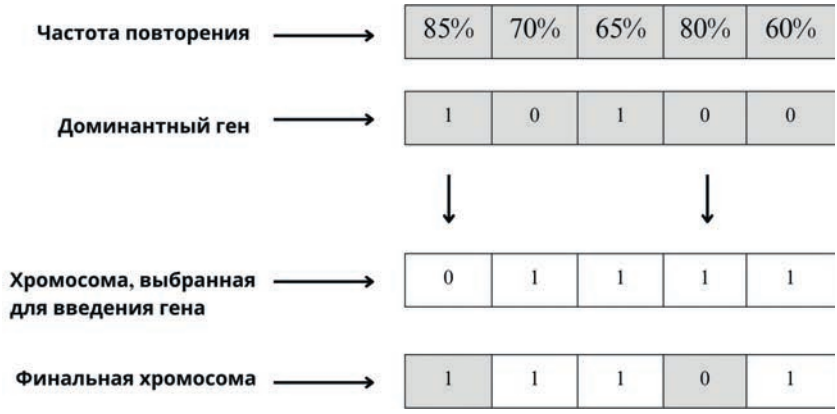


Рис. 4. Инъекция информативных генов в сухие особи популяции.

предложенного оператора инъекции генов. Он показывает процесс передачи генетической информации для улучшения выбранной хромосомы.

Включение этого третьего сценария в GEA позволяет использовать потенциал даже наименее успешных решений для продвижения алгоритма к глобальному оптимуму. Такой подход обеспечивает всестороннее исследование пространства решений и способствует улучшению всей популяции в ходе последовательных итераций.

Алгоритм 1. Доминантная хромосома

```
Данные: Поп., Проб. Инфо.
Результат: ДоминантныйГен, Маска, МаскаПеревернутая

while i меньше длины хромосомы do
    while j меньше, чем количество Поп. do
        Гены <– [Гены, Поп_j(i)]
    end
    while в генах есть элемент do
        темп <– сумма(Гены == Гены(1))
        if размер ДоминантныйГен == 0 then
            ДоминантныйГен <– Гены(1)
            СчетчикДоминантныхГенов <– темп
        else
            if темп > СчетчикДоминантныхГенов then
                ДоминантныйГен <– Гены(1)
                СчетчикДоминантныхГенов <– темп
            else
                ДоминантныйГен <– [ДоминантныйГен, Гены(1)]
            end
        end
    end
end
Маска <– нули(1, размер(хромосома))
while i меньше длины хромосомы do
    if (СчетчикДоминантныхГенов > порог) и (порог не 0) then
        Маска(i) <– 1
    end
end
МаскаПеревернутая <– !Маска
```

4. Результаты вычислительного эксперимента

В рамках данного исследования была проведена оценка эффективности алгоритма GEA при решении задач комбинаторной оптимизации на примере задачи отыскания оптимальных маршрутов транспортных средств. Эта задача заключается в определении оптимальных маршрутов для транспортных средств, направленных на обслуживание ряда точек спроса с минимальными транспортными издержками. Алгоритм GEA был сопоставлен не только с традиционным алгоритмом генетической оптимизации (GA), но и с тремя

его модификациями: GEA1, GEA2 и GEA3. Каждая из этих модификаций использовала свой сценарий (как было показано в разделе 3). В алгоритме GEA на каждой итерации основного цикла случайным образом выбирался один из этих сценариев.

В исследовании были использованы шесть эталонных примеров (см. [9, 18]). Для всех рассматриваемых алгоритмов были установлены одинаковые параметры: максимальное число итераций — 1000, размер популяции — 100. Вероятность скрещивания и мутации были унифицированы для всех алгоритмов и составили 0,8 и 0,1 соответственно. Для GEA были установлены доли учета сценариев в размере 0,5, 0,5 и 0,2 для первого, второго и третьего сценариев соответственно.

Для оценки производительности алгоритмов было проведено 10 независимых запусков каждого алгоритма на каждом тестовом наборе данных. Результаты представлены в табл. 1, включающей лучшие, худшие, средние значения и стандартные отклонения полученных решений для каждого алгорит-

Таблица 1. Результаты работы алгоритмов по критериям Лучшее=Л, Худший=Х, Среднее=С и Стандартное отклонение=Стд. (Лучшие значения по каждому критерию и тестовому экземпляру выделены жирным шрифтом.)

Тестовый экземпляр		Ф1	Ф2	Ф3	Ф4	Ф5	Ф6
Точки спроса × Количество автомобилей		8 × 3	10 × 3	14 × 4	20 × 4	25 × 5	30 × 5
GA	Л	**257,3492**	**268,1687**	**301,6661**	317,6503	326,5457	308,8542
	Х	291,6624	269,0742	316,0882	351,2298	363,0131	343,9097
	С	260,7805	268,7120	305,8615	333,5178	338,1742	321,1532
	Стд	10,8507	0,4675	5,4002	12,3733	11,3976	11,5798
GEA_1	Л	**257,3492**	**268,1687**	**301,6661**	319,3303	319,5602	307,1991
	Х	**257,3492**	269,0742	318,8057	342,2278	359,0854	370,7047
	С	**257,3492**	268,2593	304,7409	324,0378	330,8722	328,0479
	Стд	**5,99E-14**	0,2863	5,4787	6,8149	10,8851	21,2861
GEA_2	Л	**257,3492**	**268,1687**	**301,6661**	**317,1235**	321,5556	**302,5377**
	Х	**257,3492**	269,0742	306,3834	353,7992	359,0854	**322,7266**
	С	**257,3492**	268,3498	302,8296	327,4803	333,1713	**311,4745**
	Стд	**5,99E-14**	0,3817	1,6555	12,1645	13,4915	**7,3910**
GEA_3	Л	**257,3492**	**268,1687**	**301,6661**	317,6503	319,0169	308,8834
	Х	**257,3492**	269,0742	306,3834	**331,8416**	**331,3571**	346,2497
	С	**257,3492**	268,4404	302,3684	323,3476	326,245	323,5826
	Стд	**5,99E-14**	0,4373	1,58597	5,45505	**4,1059**	13,7657
GEA	Л	**257,3492**	**268,1687**	**301,6661**	317,6503	**317,7347**	304,4598
	Х	**257,3492**	**268,1687**	**301,6661**	**331,8416**	331,4877	343,6004
	С	**257,3492**	**268,1687**	**301,6661**	**321,4611**	**323,1822**	313,4242
	Стд	**5,99E-14**	**5,99E-14**	**0**	**4,7726**	6,0097	11,8294

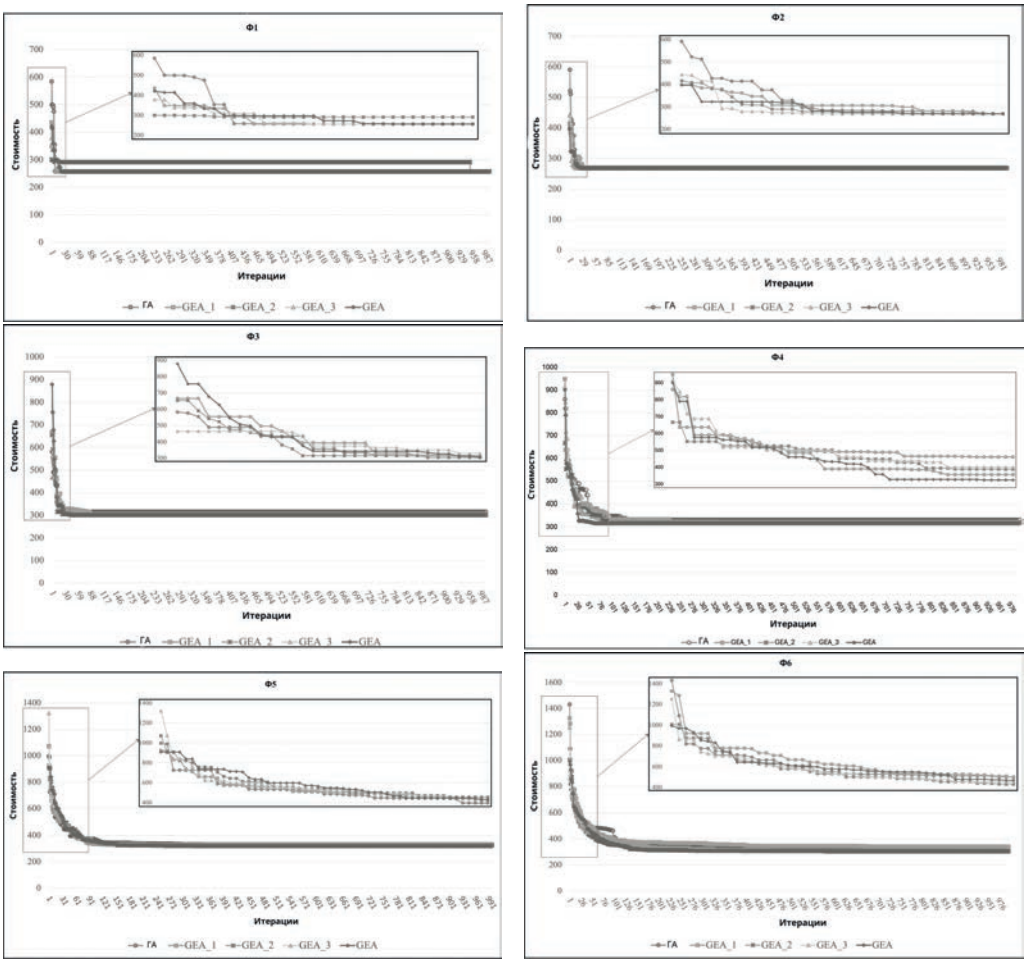


Рис. 5. Скорость сходимости метаэвристических алгоритмов во всех эталонных случаях.

ма. Это позволило проанализировать устойчивость и надежность применяемых метаэвристических алгоритмов. Рисунок 5 демонстрирует скорость сходимости алгоритмов к их наилучшей производительности. Также был проведен статистический анализ с уровнем достоверности 0,95, включающий нормализованные стандартные отклонения по всем алгоритмам. Результаты этого анализа представлены на рис. 6.

Результаты, представленные в табл. 1, свидетельствуют о том, что GEA, использующий все сценарии, превосходит другие алгоритмы. В большинстве случаев он последовательно находит решения, близкие к оптимальным, которые превосходят результаты, полученные с помощью GA и других модификаций GEA. Среди вариантов GEA особо выделяется GEA2, подтверждая эффективность второго сценария в поиске улучшенных решений. Рисунок 5 показывает, что все алгоритмы демонстрируют приемлемую скорость сходи-

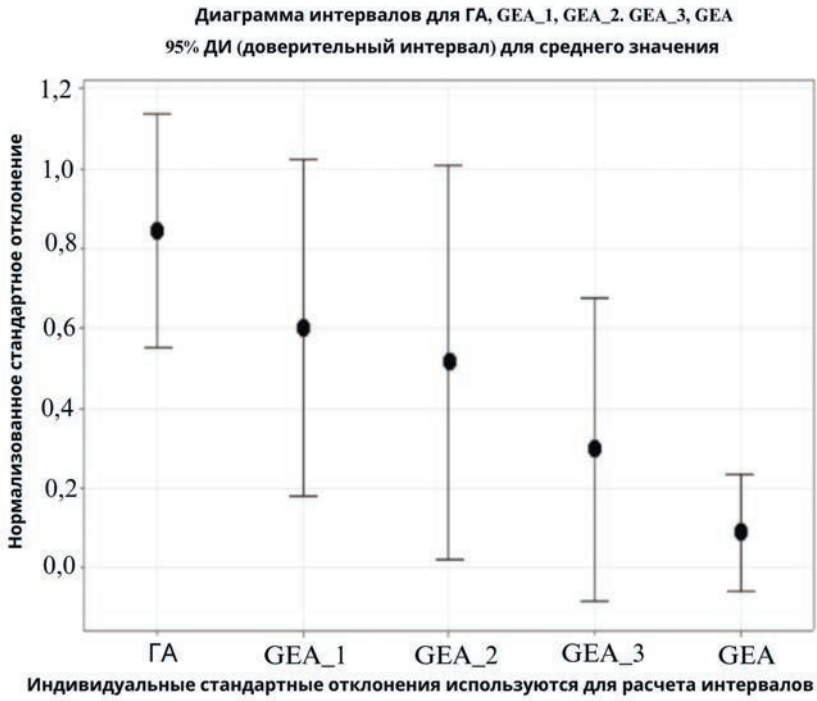


Рис. 6. Интервальный график на основе доверительного уровня 95% для анализа робастности метаэвристических алгоритмов.

мости на тестовых наборах данных, обеспечивая схожее качество решений. Статистический анализ, представленный на рис. 6, наглядно подтверждает высокую точность GEA по сравнению с другими алгоритмами.

Таким образом, проведенная оценка подтвердила эффективность GEA в решении задач комбинаторной оптимизации, в частности в задаче маршрутизации транспортных средств. Результаты, изложенные в табл. 1 и на рис. 5 и 6, выявляют превосходную производительность и точность GEA, особенно при использовании всех сценариев. Полученные данные свидетельствуют о потенциале GEA как надежного метаэвристического алгоритма для решения оптимизационных задач.

5. Выводы и возможные направления будущих исследований

В данной статье проведено исследование алгоритма GEA и его эффективности при решении задач комбинаторной оптимизации, в частности задачи о маршрутизации транспортных средств. Результаты, полученные в ходе сравнительного анализа с традиционным ГА и различными вариантами GEA, демонстрируют превосходство GEA, особенно при использовании всех сценариев. GEA превосходит другие алгоритмы, давая лучшие близкие к оптимальным решения в большинстве случаев.

Эффективность GEA в решении задачи маршрутизации транспортных средств демонстрирует его потенциал в реальных задачах, где эффективная маршрутизация транспорта имеет решающее значение. Эти результаты дают представление об эффективности методов генной инженерии для решения задач комбинаторной оптимизации и указывают на необходимость учета различных сценариев при разработке алгоритмов.

Можно выделить несколько направлений будущих исследований. Во-первых, можно провести дальнейшие исследования, чтобы изучить влияние различных параметров на производительность GEA и его вариаций. Тонкая настройка параметров алгоритма может повысить его эффективность и привести к получению более качественных решений. Во-вторых, применение подхода к другим оптимизационным задачам позволит получить представление о его эффективности и конкурентоспособности. Кроме того, интеграция GEA с другими методами оптимизации или гибридизация его с подходами машинного обучения может расширить его возможности. Наконец, проведение экспериментов на более высокоразмерных задачах позволит установить границы масштабируемости и эффективности GEA [40, 41].

СПИСОК ЛИТЕРАТУРЫ

1. *Holland J.* Adaptation in natural and artificial systems. Ann Arbor: University of Michigan Press, 1975.
2. *Elshaer R., Awad H.* A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants // Computers Indust. Engin. 2020. V. 140. P. 106242.
3. *Katoch S., Chauhan S.S., Kumar V.* A review on genetic algorithm: past, present, and future // Multimedia Tools Appli. 2021. V. 80. P. 8091–8126.
4. *Yang X.S., Deb S.* Engineering optimisation by cuckoo search // Int. J. Math. Modell. Numer. Optim. 2010. V. 1. No. 4. P. 330–343.
5. *Mirjalili S., Lewis A.* The whale optimization algorithm // Advanc. Engin. Software. 2016. V. 95. P. 51–67.
6. *Mirjalili S.* SCA: a sine cosine algorithm for solving optimization problems // Knowledge-Based Syst. 2016. V. 96. P. 120–133.
7. *Heidari A.A., Mirjalili S., Faris H., et al.* Harris hawks optimization: Algorithm and applications // Future Generat. Comput. Syst. 2019. V. 97. P. 849–872.
8. *Jain M., Singh V., Rani A.* A novel nature-inspired algorithm for optimization: Squirrel search algorithm // Swarm Evoluti. Comput. 2019. V. 44. P. 148–175.
9. *Fathollahi-Fard A.M., Hajiaghaei-Keshteli M., Tavakkoli-Moghaddam R.* Red deer algorithm (RDA): a new nature-inspired meta-heuristic // Soft Comput. 2020. V. 24. P. 14637–14665.
10. *Xue J., Shen B.* A novel swarm intelligence optimization approach: sparrow search algorithm // Syst. Sci. Control Engine. 2020. V. 8. No. 1. P. 22–34.
11. *Braik M., Sheta A., Al-Hiary H.* A novel meta-heuristic search algorithm for solving optimization problems: capuchin search algorithm // Neural Comput. Appli. 2021. V. 33. P. 2515–2547.

12. *Abualigah L., Yousri D., Abd Elaziz M., et al.* Aquila optimizer: a novel meta-heuristic optimization algorithm // Comput. Indust. Engin. 2021. V. 157. P. 107250.
13. *Braik M.S.* Chameleon Swarm Algorithm: A bio-inspired optimizer for solving engineering design problems // Expert Syst. Appl. 2021. V. 174. P. 114685.
14. *Yang Z., Deng L., Wang Y., et al.* Aptenodytes forsteri optimization: Algorithm and applications // Knowledge-Based Syst. 2021. V. 232. P. 107483.
15. *Xue J., Shen B.* Dung beetle optimizer: A new meta-heuristic algorithm for global optimization // J. Supercomput. 2023. V. 79. No. 7. P. 7305–7336.
16. *Zhong C., Li G., Meng Z.* Beluga whale optimization: A novel nature-inspired metaheuristic algorithm // Knowledge-Based Syst. 2022. V. 251. P. 109215.
17. *Wolpert D.H., Macready W.G.* No free lunch theorems for optimization // IEEE Transactions on Evoluti. Comput. 1997. V. 1. No. 1. P. 67–82.
18. *Fathollahi-Fard A.M., Hajiaghaei-Keshteli M., Tavakkoli-Moghaddam R.* The social engineering optimizer (SEO) // Engin. Appli. Artific. Intellig. 2018. V. 72. P. 267–293.
19. *Li D., Li X., Zhou W.L., et al.* Genetically engineered T cells for cancer immunotherapy // Signal Transduct. Targeted Therapy. 2019. V. 4. No. 1. P. 35.
20. *Xiao Q., Guo D., Chen S.* Application of CRISPR/Cas9-based gene editing in HIV-1/AIDS therapy // Frontiers Cellul. Infect. Microbiol. 2019. V. 9. P. 69.
21. *Raposo V.L.* The first Chinese edited babies: a leap of faith in science // JBRA Assist. Reproduct. 2019. V. 23. No. 3. P. 197.
22. *Li C.* Breeding crops by design for future agriculture // J. Zhejiang Univer. Sci. B. 2020. V. 21. No. 6. P. 423.
23. *Dubock A.* Golden rice: to combat vitamin A deficiency for public health. Vitamin A. 2019. V. 1.
24. *Huang T.K., Puchta H.* Novel CRISPR/Cas applications in plants: from prime editing to chromosome engineering // Transgen. Res. 2021. V. 30. P. 529–549.
25. *Shahryari A., Saghaeian Jazi M., Mohammadi S., et al.* Development and clinical translation of approved gene therapy products for genetic disorders // Front. Genet. 2019. V. 10. P. 868.
26. *Zhuo C., Zhang J., Lee J.H., et al.* Spatiotemporal control of CRISPR/Cas9 gene editing // Signal Transduct. and Targeted Therapy. 2021. V. 6. No. 1. P. 238.
27. *Kostenetskiy P.S., Chulkevich R.A., Kozyrev V.I.* HPC Resources of the Higher School of Economics // J. Phys. Conf. Seri. 2021. V. 1740. No. 1. P. 012050. https://doi.org/10.1088/1742-6596/1740/1/012050
28. *Gero J.S., Kazakov V.* A genetic engineering approach to genetic algorithms // Evoluti. Comput. 2001. V. 9. No. 1. P. 71–92.
29. *Kameya Y., Prayoonsri C.* Pattern-based preservation of building blocks in genetic algorithms // IEEE Congre. Evolut. Comput. (CEC). 2011. P. 2578–2585.
30. *Ding S., Su C., Yu J.* An optimizing BP neural network algorithm based on genetic algorithm // Artific. Intellig. Rev. 2011. V. 36. P. 153–162.
31. *Liang Y., Leung K.S.* Genetic algorithm with adaptive elitist-population strategies for multimodal function optimization // Appl. Soft Comput. 2011. V. 11. No. 2. P. 2017–2034.
32. *Dasgupta K., Mandal B., Dutta P., et al.* A genetic algorithm (ga) based load balancing strategy for cloud computing // Procedia Techn. 2013. V. 10. P. 340–347.

33. *Elsayed S.M., Sarker R.A., Essam D.L.* A new genetic algorithm for solving optimization problems // Engin. Appli. of Artific. Intellig. 2014. V. 27. P. 57–69.

34. *Peng B., Li L.* An improved localization algorithm based on genetic algorithm in wireless sensor networks // Cognitive Neurodynam. 2015. V. 9. P. 249–256.

35. *Askarzadeh A.* A memory-based genetic algorithm for optimization of power generation in a microgrid // IEEE Transact. Sustainable Energy. 2017. V. 9. No. 3. P. 1081–1089.

36. *Reddy G.T., Reddy M.P.K., Lakshmanna, et al.* Hybrid genetic algorithm and a fuzzy logic classifier for heart disease diagnosis // Evolut. Intellig. 2020. V. 13. P. 185–196.

37. *Fathollahi-Fard A.M., Dulebenets M.A., Hajiaghaei-Keshteli M., et al.* Two hybrid meta-heuristic algorithms for a dual-channel closed-loop supply chain network design problem in the tire industry under uncertainty // Adv. Engin. Inform. 2021. V. 50. P. 101418.

38. *Fathollahi-Fard A.M., Tian G., Ke H., et al.* Efficient Multi-objective Metaheuristic Algorithm for Sustainable Harvest Planning Problem // Comput. Oper. Res. 2023. V. 158. P. 106304.

39. *Kolaee M.H., Mirzapour Al-e-Hashem S.M.J, Jabbarzadeh A.* A local search-based non-dominated sorting genetic algorithm for solving a multi-objective medical tourism trip design problem considering the attractiveness of trips // Engin. Appl. Artific. Intellig. 2023. V. 124. P. 106630.

40. *Du D., Pardalos P.M.* Handbook of combinatorial optimization. Springer Science & Business Media. 1998. V. 4.

41. *Mart R., Pardalos P.M., Resende M.G.* Handbook of heuristics. Springer Publishing Company, Incorporated. 2018.

Статья представлена к публикации членом редколлегии А.А. Галяевым.

Поступила в редакцию 08.07.2023

После доработки 09.10.2023

Принята к публикации 20.01.2024

© 2024 г. Т.М. БИДЖИЕВ (temirlanbid@gmail.com),
Д.Е. НАМИОТ, д-р техн. наук (dnamiot@gmail.com)
(Московский государственный университет им. М.В. Ломоносова)

АТАКИ НА МОДЕЛИ МАШИННОГО ОБУЧЕНИЯ, ОСНОВАННЫЕ НА ФРЕЙМВОРКЕ PYTORCH

Рассматриваются последствия использования облачных сервисов для обучения нейронных сетей с точки зрения кибербезопасности. Ресурсоемкость обучения нейронных сетей создает проблемы, что приводит к росту зависимости от облачных сервисов. Однако такая зависимость создает новые риски кибербезопасности. Исследование посвящено новому методу атаки, использующему веса нейронных сетей для незаметного распространения скрытых вредоносных программ. Рассматриваются семь методов встраивания и четыре типа триггеров для активации вредоносного программного обеспечения. Представлен фреймворк с открытым исходным кодом, автоматизирующий внедрение кода в весовые параметры нейронных сетей, что позволяет исследователям изучать и противодействовать этому новому вектору атак.

Ключевые слова: нейронные сети, вредоносное программное обеспечение, стеганография, триггеры.

DOI: 10.31857/S0005231024030038, **EDN:** TZXTPW

1. Введение

Значительный прогресс и широкое применение машинного обучения в различных областях привели к возникновению важных вопросов, касающихся безопасности и надежности моделей машинного обучения [1, 2].

Одной из главных проблем безопасности в машинном обучении является уязвимость моделей к различного рода атакам [3, 4]. Новой тенденцией в современных системах машинного обучения является рост числа атак, направленных на внедрение вредоносного программного обеспечения (ПО) в нейронные сети [5–7]. Такая форма атаки представляет собой серьезную угрозу безопасности и надежности моделей, поскольку злоумышленники могут использовать их для выполнения вредоносных действий, обходя традиционные механизмы защиты. Такие атаки могут привести к скрытой активации вредоносных функций, утечке конфиденциальной информации или неправильной классификации данных, что подрывает точность моделей машинного обучения [8]. Поэтому понимание, обнаружение и защита от атак, связанных с внедрением вредоносных программ в нейронные сети, становятся важными задачами в области кибербезопасности.

Методы внедрения вредоносного ПО используются при поставке готовых моделей конечному пользователю через сервисы MLaaS (Machine Learning

as a Service). Часто потребители, не являющиеся экспертами в области машинного обучения, работают с сериями MLaaS, не имея представления об обучении, тестировании и обработке данных. Как правило, самым важным критерием для таких пользователей является точность модели. Злоумышленник может воспользоваться этим и незаметно для пользователя внедрить в модель глубокой нейронной сети вредоносное ПО.

Обзор методов внедрения вредоносных программ непосредственно в модели машинного обучения представлен в данной статье [9].

2. Постановка задачи

Целью данной работы является изучение и анализ атак на машинное обучение, направленных на внедрение вредоносного кода в нейронные сети, путем разработки специализированного фреймворка [10], автоматизирующего процесс внедрения вредоносного кода в веса нейронных сетей [11].

Этот фреймворк позволяет исследователям и специалистам проводить эксперименты и проверять устойчивость своих моделей машинного обучения, а также разрабатывать и применять контрмеры для защиты от подобных атак. Фреймворк обеспечивает гибкость и масштабируемость, позволяя настраивать и адаптировать методы внедрения вредоносного ПО в зависимости от конкретных требований и сценария использования.

Был проведен глубокий анализ существующих методологий и изучены подходы к реализации программного обеспечения. Данная работа вносит вклад в область безопасности машинного обучения и дает практическое представление о защите моделей машинного обучения от атак с внедрением кода на весовые коэффициенты моделей.

В следующих разделах рассмотрим теоретические основы, методологию исследования и экспериментальную оценку предлагаемого фреймворка, конечной целью которого является повышение безопасности и надежности систем машинного обучения.

3. Методология

На рис. 1 показан типичный сценарий взаимодействия между пользователем и злоумышленником. В этом сценарии пользователь, намеревающийся использовать нейросетевую модель в бизнес-целях, начинает процесс с выбора желаемой архитектуры модели. Затем пользователь приступает к обучению модели с помощью провайдеров MLaaS или получает предварительно обученную модель из различных источников.

В конкретном сценарии, где злоумышленник берет на себя роль сервиса MLaaS, предполагается, что он не обладает способностью изменять архитектуру полученной нейронной сети. Однако злоумышленник сохраняет возможность манипулировать весовыми параметрами сети. Это позволяет злоумышленнику вносить вредоносные модификации в модель, не изменяя ее фундаментальной структуры.

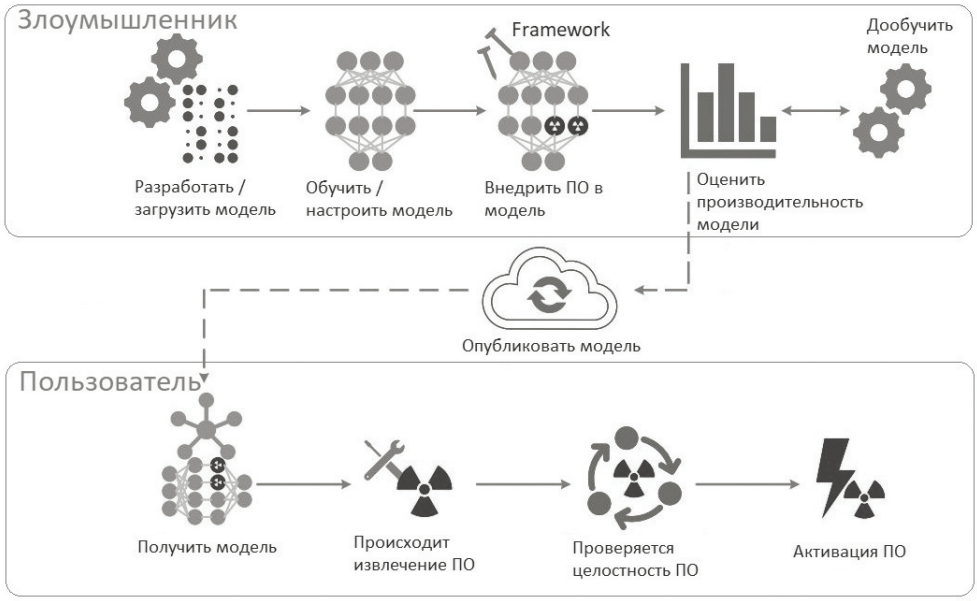


Рис. 1. Общий сценарий взаимодействия пользователя и злоумышленника.

Для того чтобы реализовать все цели, злоумышленнику необходимо выполнить действия в соответствии с рис. 1. В этом процессе злоумышленник начинает с получения модели нейронной сети, которая может быть предоставлена пользователем или разработана им самим. Затем он приступает к обучению модели до желаемого уровня точности, причем этот этап может быть выполнен как самостоятельно, так и передан на аутсорсинг провайдерам MLaaS. После успешного обучения злоумышленник подготавливает и внедряет в модель вредоносное ПО, одновременно отслеживая и контролируя потерю точности модели с помощью методов, подробно описанных в данной статье. Для развертывания и активации вредоносного ПО злоумышленник создает триггер, используя методы, рассмотренные в этом же исследовании. Когда модель полностью подготовлена, злоумышленник может использовать дополнительные техники, такие как “загрязнение цепочки” [12], чтобы распространить эту модель в публичных хранилищах или других местах.

Далее пользователь получает нейросеть со встроенным вредоносным ПО, которое будет извлечено и запущено при активации триггера.

4. Методы внедрения вредоносного ПО

4.1. Внедрение вредоносных байтов в нейроны

Согласно стандарту IEEE [13], число с плавающей точкой имеет размер 32 бита, где первый бит отводится под знак числа, следующие 8 бит – под экспоненту, а последние 23 бита – под мантиссу. В результате получается число $\pm 1.m \times 2^n$ в двоичной форме, где m – мантисса числа, n – экспонента.

Такое число принадлежит диапазону от 2^{-127} до $2^{127}-1$. Экспонента отвечает за его величину, т.е., сохранив несколько первых байт числа, оставшуюся часть можно заменить на вредоносные байты, сохранив небольшую разницу с исходным числом.

4.2. Методы StegoNet

В статье StegoNet [5] предлагается четыре метода внедрения вредоносного ПО. Давайте рассмотрим их.

LSB замена

Этот подход основан на использовании замены LSB (Least Significant Bits) [7], как это применяется в технике стеганографии [14]. Она включает в себя выбор количества нейронов и параметров, подходящих для вредоносной программы. Двоичный код вредоносной программы делится на сегменты длиной, равной выбранной длине замены LSB, и эти сегменты записываются в параметры модели вместо последних нескольких битов соответствующего параметра.

Это решение неприменимо к сильно сжатым моделям нейронных сетей. Например, размер модели MobileNet [15] составляет всего 4 МБ при 4 миллионах 8-битных параметров. Такие сжатые модели быстро теряют точность даже при незначительных изменениях параметров. Этот метод не подходит для сжатых моделей.

Устойчивое обучение

Удаление некоторого набора нейронов из топологии нейросетевой модели может привести к значительному снижению точности, но параметры, соединяющие оставшиеся нейроны, могут быть скорректированы (переобучены) для достижения первоначальной точности. На основе этой интуиции была предложена методика “Устойчивое обучение”.

Как показано на рис. 2,*а*, метод предполагает прямую замену всех битов выбранных параметров на сегменты вредоносных программ. Такие нейроны (т.е. с измененными параметрами) не будут обновляться в процессе переобучения. Ожидается, что после обучения точность модели будет восстановлена, что позволит скрыть наличие внедренного ПО.

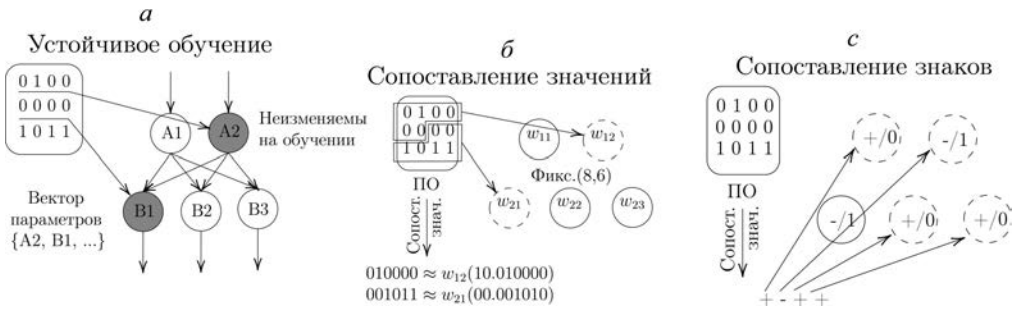


Рис. 2. Методы внедрения вредоносных программ [5].

Сопоставление значений

Предположим, есть модель с 8-битными числами с фиксированной точкой и шестью битами после десятичной точки. Чтобы облегчить сравнение значений, двоичный код вредоносной программы изначально делится на сегменты длиной, соответствующей количеству битов после десятичной точки. Затем для каждого сегмента выполняется полный исчерпывающий поиск параметров модели, чтобы найти (или заменить) такое же (или близкое) значение битов дробного параметра, см. пример на рис. 2,*б*. Наконец, сопоставляем сегмент кода с соответствующим параметром, при необходимости заменяя дробные биты параметра на сегмент кода.

Сопоставление знаков

Метод сопоставления знаков использует аналогичное правило "полного поиска и сопоставления", основанное на бите знака параметров модели. Как показано на рис. 2,*в*, метод сопоставления знаков проходит через параметры модели и сопоставляет бит знака параметра с каждым отдельным битом вредоносного кода. Например, 0 сопоставляется со знаком параметра +, таким образом в конечном итоге код сопоставляется с последовательностью битов знака для соответствующих параметров. Необходимо хранить вектор сопоставленных параметров.

4.3. Методы EvilModel

В статье EvilModel [6] предлагаются еще три метода. Рассмотрим их.

MSB сохранение

Поскольку наиболее важная экспоненциальная часть параметра находится в первом байте, первый байт является наиболее существенным для определения значения параметра. Поэтому его можно оставить без изменений и внедрить вредоносную программу в следующие три байта. Таким образом, значения параметров остаются в разумных пределах.

Быстрая замена

Если заменить параметры тремя байтами вредоносного кода и первым байтом префикса 0x3C или 0xBC в зависимости от знака параметра, большинство значений параметров все равно окажутся в разумных пределах. По сравнению с методом MSB, этот метод может оказать большее влияние на производительность модели, но поскольку ему не нужно разбивать параметры в нейроне, он будет работать быстрее.

Половинная замена

Как и в случае с методом MSB reservation, если оставить неизменными первые два байта, а не один, и изменить два других байта, значение этого числа будет колебаться в меньшем диапазоне. Однако, поскольку в параметре заменяются только два байта, этот метод может внедрить меньше данных, чем два предыдущих метода.

5. Сравнение методов внедрения

Давайте сравним точность различных моделей до и после применения всех описанных выше методов, см. [5, 6] и табл. 1.

Как видно из этой таблицы, наивный метод LSB замены может поддерживать хорошую точность тестирования на средних нейронных сетях, но это не относится к малым нейронным сетям. Например, он приводит к значительному снижению точности (т.е. резкому падению до $\approx 0{,}1\%$) в моделях нейронных сетей с высокой степенью сжатия из-за ограниченной точности данных и уменьшения количества параметров.

Напротив, метод устойчивого обучения может сравнительно лучше поддерживать вредоносные программы на небольших нейронных сетях. Для небольших вредоносных программ, таких как EquationDrug, ZeusVM и Cerber [16], он может поддерживать точность тестирования на уровне оригинала, даже в самых маленьких сетях MobileNet [15] (4,2 МБ) и SqueezeNet [17] (4,6 МБ). Однако точность MobileNet значительно снизилась с 66,7% до 0,7% при увеличении размера вредоносной программы с 0,59 МБ (Cerber) до 3,35 МБ (WannaCry). Можно заметить, что верхняя граница отношения размера вредоносной программы к размеру модели для метода Resilience training составляет $\approx$15% без снижения точности.

Такая проблема была устранена с помощью метода сопоставления значений, основанного на "полном поиске и отображении". Для сильно сжатых моделей нейронных сетей, таких как "Comp.Alexnet" [18], параметры модели сильно сжаты и метод может быть менее эффективным. Аналогичная тенденция прослеживается и в методе сопоставления знаков. В целом, однако, метод сопоставления знаков всегда может сохранить исходную точность тестирования для всех применимых случаев.

Для метода MSB reservation из-за избыточности нейросетевых моделей при внедрении вредоносного ПО точность тестирования не влияет на модели большого размера (>200 Мб). В некоторых случаях (например, Vgg 16 [19] с NSIS) точность немного увеличивается, что также отмечено в статье Stegonet [5]. При реализации вредоносного ПО с использованием MSB сохранения точность снижается по мере увеличения размера встроенного вредоносного ПО для моделей среднего и малого размера. Например, точность снижается на 5% для моделей среднего размера, таких как Resnet50 с Mamba. Теоретически максимальный порядок встраивания (т.е. отношение размера вредоносного кода к размеру модели) для метода MSB сохранения составляет 75%. В эксперименте верхняя граница порядка встраивания без существенного снижения точности составляет 25,73% (Googlenet с Artemis).

Эффективность метода быстрой замены схожа с методом MSB сохранения, но нестабильна для небольших моделей. Когда в модель среднего или малого размера внедряется более крупное вредоносное ПО, точность модели значительно снижается. Например, для Mobilenet с VikingHorde точность тестирования резко падает до 0,108%. Это показывает, что быстрая подстанов-

ка может быть использована вместо метода MSB сохранения, когда модель велика или задача требует много времени. В эксперименте порядок встраивания без существенного снижения точности составил 15,9% (Resnet18 с Viking Horde).

Метод половинной замены превосходит все остальные методы. Благодаря избыточности нейросетевых моделей точность после встраивания кода любого размера практически не снижается, даже если почти половина модели заменена вредоносными байтами, точность колеблется в пределах 0,01% от исходной. В Squeezenet небольшого размера (4,74 МБ) можно внедрить образец вируса Mamba размером 2,3 МБ, при этом точность увеличится на 0,048%. Теоретически максимальный порядок встраивания составляет 50%. В эксперименте было достигнуто близкое к теоретическому значение 48,52% (Squeezenet с Mamba).

6. Методы активации вредоносного ПО

6.1. Триггеры

В статье StegoNet [5] предлагается три различных триггера: Логит триггер, Ранговый триггер, Настроенный ранговый триггер.

Метод Логит триггера предполагает запоминание выходов логитов для заранее выбранных входных данных – триггеров. После того, как в модель подается один экземпляр из входных данных, выбранных в качестве триггера, происходит совпадение выходных значений логитов, и вредоносный код извлекается и активируется. Теоретически это невозможно, поскольку вероятность подачи точно такого же входного образца очень мала, а выходы логитов (числа с плавающей точкой) должны полностью совпадать.

Поэтому метод рангового триггера будет более полезен. Разница заключается в сравнении не самих логитов, а их рангов, т.е. индексов в отсортированном массиве логитов. Например, пусть последний слой имеет размерность 3 и триггером будут логиты $\{p_1, p_2, p_3\} = \{0{,}5, 0{,}2, 0{,}4\}$. Но из-за разнообразия входов получили выход $\{0{,}55, 0{,}13, 0{,}42\}$. Ранги логитов будут равны $r = \{p_1, p_3, p_2\}$, и они будут совпадать.

Метод настроенного рангового триггера включает в себя выбор исходной выборки, дополненной различными вариациями, и дообучение модели на этих дополненных выборках. Однако вместо оригинальной функции потерь, зависящей от значений логитов, используется функция потерь, основанная на рангах логитов. Для этого придется вручную задать целевое значение рангов логита. Пусть x – аугментированные входные данные, h^r – установленная для них метка ранга логита, если логит не рассматривается, то его значение равно 0. Функция потерь будет выглядеть следующим образом:

$$arg\min_{w} \frac{1}{n} \sum_{1}^{n} \mathcal{L}(f_w(x), h^r).$$

После активации триггера вредоносная программа собирается в единый код. Затем его хэш-сумма сравнивается с предварительно сохраненной хэш-суммой несегментированной вредоносной программы. Если они совпадают, вредоносная программа запускается.

6.2. Активация

Если к нейросетевой модели прилагается стороннее программное обеспечение, например программа эксплуатации модели, то в нее можно внедрить весь необходимый код для проверки триггеров и активации вредоносного ПО на целевом устройстве. Это самый простой случай, рассмотрим другие.

Сначала предполагалось, что модель обучается на недоверенном источнике, т.е. на стороне злоумышленника, и у него есть все данные модели – атака "белого ящика". Модель передается по сети в сериализованной форме [20], а затем десериализуется. С помощью атаки типа "insecure deserialization" [21, 22] злоумышленник может изменить функции активации в модели. Например, вместо softmax использовать модифицированную версию с проверкой триггеров и развертыванием вредоносных программ.

Другой подход заключается в использовании уязвимостей в библиотеках и исполняемых средах. Например: CVE-2018-6269, CVE-2017-12852.

7. Фреймворк

7.1. Обзор

Целью предложенного автором фреймворка, см. [10], является разработка готового решения для встраивания вредоносных программ в нейронные сети, позволяющего беспрепятственно интегрировать вредоносный код в архитектуру сети. Используя уникальные характеристики нейронных сетей и их широкое применение в различных приложениях, предложенный фреймворк призван исследовать потенциальные риски и уязвимости, связанные с этими моделями.

Предложенный фреймворк представляет собой комплексное решение, позволяющее исследователям и специалистам по безопасности изучать поведение нейронных сетей при воздействии на них встроенного вредоносного ПО. Это открывает возможности для анализа влияния вредоносных атак на производительность сети, выявления уязвимостей и разработки надежных механизмов защиты.

Ключевые особенности:

1. **Внедрение вредоносных программ**. Фреймворк включает в себя различные методы внедрения вредоносных программ в структуру нейронной сети путем изменения их весовых коэффициентов, обеспечивая беспрепятственную интеграцию без ущерба для общей функциональности сети.

2. **Возможность оценки**. Платформа авторов предоставляет инструменты для оценки влияния встроенного вредоносного ПО на производительность сети, т.е. метрики снижения точности.

3. **Гибкость и совместимость**. Фреймворк разработан как совместимый с фреймворком глубокого обучения PyTorch [23], что позволяет легко интегрировать его в существующие архитектуры нейронных сетей без модификации.

В целом платформа авторов позволяет исследователям и практикам в области кибербезопасности лучше понять последствия внедрения вредоносных программ в нейронные сети. Так как платформа надежная и гибкая, то облегчается поиск эффективных мер противодействия и разработка более устойчивых и безопасных моделей нейронных сетей.

7.2. Архитектура и компоненты

Чтобы проверить падение точности моделей до и после внедрения вредоносного ПО с помощью предложенного фреймворка, была проведена серия экспериментов с различными архитектурами нейронных сетей [24].

1. **Набор данных**. Для тестирования модели и структуры был использован набор данных ImageNet [25]. Этот набор содержит 1,2 миллиона трехканальных изображений размером 224×224 пикселей, представляющих 1000 различных категорий объектов.

2. **Методология**.

— Были проведены эксперименты с использованием нескольких предварительно обученных нейросетевых моделей.

— Для каждого эксперимента с помощью предложенного фреймворка были внедрены вредоносные программы разного размера. Примеры вредоносных программ были взяты из репозиториев [16, 26].

— Производительность модифицированных моделей сравнивалась с оригинальными моделями, а также оценивалось влияние внедренных вредоносных программ на точность модели.

3. **Результаты и анализ**.

В табл. 1 показана точность моделей в результате внедрения вредоносных программ с помощью предложенного фреймворка.

Для сравнения возможностей фреймворка были выбраны наиболее часто используемые модели и образцы вредоносного ПО. Поскольку большие модели имеют большую емкость, результаты реализации были показаны только для метода LSB замены, а для остальных методов сравниваются в основном средние и малые модели. Обратите внимание, что для метода устойчивого обучения дообучение моделей не использовалось.

— Результаты эксперимента продемонстрировали эффективность фреймворка при внедрении вредоносных программ в нейронные сети.

— Модифицированные сети успешно сохраняют точность в большинстве случаев классификации на обычных входных данных, демонстрируя желаемое уклончивое поведение.

— Результаты экспериментов выявили компромисс между емкостью моделей и точностью классификации.

Таблица 1. Точность моделей после внедрения вредоносного ПО в их веса с помощью фреймворка. Случаи со значительным снижением точности моделей выделены жирным шрифтом

Метод	Модель	Базовая точность	EquationDrug 372KB	ZeusVM 405KB	NSIS 1,7MB	Mamba 2,30MB	WannaCry 3,4MB	VikingHorde 7,1MB	Artemis 12,8MB
LSB замена	Alexnet	52,8%	52,8%	52,8%	52,8%	52,8%	52,8%	52,8%	52,8%
	Resnet101	76,7%	76,7%	76,7%	76,4%	76,3%	76,2%	75,7%	74,9%
	Inception	68,0%	67,9%	68,0%	68,1%	68,0%	67,9%	67,8%	67,1%
	Resnet50	76,0%	76,0%	76,1%	76,0%	76,3%	75,9%	76,2%	75,2%
	Googlenet	67,1%	66,8%	66,9%	66,7%	65,9%	65,7%	–	–
	Resnet18	67,8%	67,9%	67,8%	67,3%	68,0%	67,5%	**58,4%**	–
	Mobilenet	70,9%	**0,1%**	**0,1%**	**0,1%**	**0,1%**	–	–	–
	Squeezenet	54,9%	**0,1%**	**0,1%**	–	–	–	–	–
MSB сохранение	Inception	68,0%	68,0%	68,2%	67,7%	67,0%	68,1%	61,1%	62,7%
	Resnet18	67,8%	67,7%	67,4%	67,3%	67,0%	66,3%	66,2%	60,9%
	Mobilenet	70,9%	71,1%	69,2%	68,1%	67,0%	63,8%	**0,7%**	–
Быстрая замена	Inception	68,0%	68,0%	67,7%	68,0%	68,1%	67,2%	67,9%	68,0%
	Resnet18	67,8%	67,7%	67,3%	67,2%	67,0%	67,6%	66,2%	61,2%
	Mobilenet	70,9%	70,8%	70,9%	65,7%	**59,8%**	**40,7%**	**1,6%**	–
Половинная замена	Inception	68,0%	68,0%	68,0%	68,0%	68,0%	68,0%	68,0%	68,0%
	Resnet18	67,8%	67,8%	67,8%	67,8%	67,8%	67,8%	67,8%	67,8%
	Mobilenet	70,9%	70,9%	69,9%	69,3%	66,0%	67,7%	**52,0%**	–
Устойчивое обучение	Inception	68,0%	68,4%	67,6%	67,8%	67,3%	68,3%	67,7%	67,8%
	Resnet18	67,8%	67,5%	67,7%	68,0%	67,9%	67,2%	67,3%	68,0%
	Mobilenet	70,9%	**54,9%**	**20,4%**	**0,4%**	**0,4%**	**0,7%**	–	–
Сопоставление значений	Inception	68,0%	68,0%	68,0%	67,4%	67,7%	67,5%	67,2%	66,2%
	Resnet18	67,8%	67,4%	67,4%	67,2%	67,4%	67,9%	67,4%	67,2%
	Mobilenet	70,9%	70,1%	70,7%	**60,1%**	**53,1%**	–	–	–
Сопоставление знаков	Inception	68,0%	68,0%	68,0%	68,0%	–	–	–	–
	Resnet18	67,8%	67,8%	67,8%	67,8%	–	–	–	–
	Mobilenet	70,9%	–	–	–	–	–	–	–

8. Контрмеры

Защита нейронных сетей от внедрения вредоносного ПО имеет решающее значение для обеспечения их целостности и надежности. Для снижения рисков, связанных с этим типом атак [27], можно применить несколько контрмер. Обычно используются следующие контрмеры.

1. **Изменение архитектуры сети**. Можно изменить структуру модели или ее параметры так, чтобы хэш-значение развернутой модели не совпадало с сохраненным. Это может быть достигнуто путем изменения архитектуры сети, добавления случайных слоев или изменения весовых параметров [28].

2. **Использование надежных каналов поставки моделей**. Выбирая надежных и заслуживающих доверия поставщиков MLaaS, организации могут значительно снизить вероятность приобретения зараженных вредоносным ПО моделей [29].

3. **Регулярное обновление моделей**. Для защиты от возникающих угроз очень важно поддерживать модели нейронных сетей в актуальном состоянии с помощью последних патчей и обновлений безопасности. Регулярные обновления моделей с улучшенной архитектурой, надежными алгоритмами обучения и усиленными мерами безопасности помогут предотвратить и обнаружить попытки проникновения вредоносного ПО.

4. **Мониторинг моделей**. Постоянный мониторинг развернутых моделей необходим для выявления необычного поведения или отклонений от ожидаемых моделей. Такие методы, как самоанализ модели, обнаружение аномалий и анализ времени выполнения, помогут выявить потенциальное внедрение вредоносного ПО и инициировать соответствующие ответные меры [30].

9. Заключение

В данной работе представлен новый фреймворк для встраивания вредоносного кода в нейронные сети, использующий весовые параметры нейронов в качестве носителей вредоносной информации. Интегрируя различные методы и компоненты, фреймворк демонстрирует возможность встраивания вредоносного кода в нейросетевые модели, подчеркивая потенциальные риски безопасности, связанные с развертыванием таких моделей.

Экспериментальная оценка этого фреймворка на различных архитектурах нейронных сетей показала его эффективность в успешном внедрении вредоносного ПО при сохранении функциональности и производительности модели. Полученные результаты подчеркивают необходимость разработки надежных мер безопасности для борьбы с растущим уровнем угроз, связанных с внедрением вредоносного ПО в системы машинного обучения.

Проведено обсуждение архитектуры и компонентов этого фреймворка, включая используемые методы внедрения вредоносных программ, процесс интеграции и тестирование модели.

В заключение можно сказать, что данная платформа доказывает наличие уязвимостей в системах машинного обучения и подчеркивает необходимость принятия мер безопасности для обеспечения целостности, надежности и устойчивости этих систем. Понимая и снижая риски, связанные со встроенным вредоносным ПО, можно повысить безопасность приложений машинного обучения и помочь создать более безопасный цифровой ландшафт.

СПИСОК ЛИТЕРАТУРЫ

1. *Namiot D., Ilyushin E., Pilipenko O.* On trusted AI Platforms // Int. J. Open Inform. Techn. 2022. V. 10. No. 7. P. 119–127.

2. *Kostyumov V.* A survey and systematization of evasion attacks in computer vision // Int. J. Open Inform. Techn. 2022. V. 10. No. 10. P. 11–20.
3. *Stoecklin Ph.M., Kirat D., Jang J.* DeepLocker: How AI Can Power a Stealthy New Breed of Malware // SecurityIntelligence. 2018.
4. *Ilyushin E., Namiot D., Chizhov I.* Attacks on machine learning systems-common problems and methods // Int. J. Open Inform. Techn. 2022. V. 10. No. 3. P. 17–22.
5. *Liu T.* StegoNet: Turn Deep Neural Network into a Stegomalware // Annual Computer Security Applications Conference. ACSAC'20. 2020. P. 928–938.
6. *Wang Z.* EvilModel 2.0: Bringing Neural Network Models into Malware Attacks // arXiv:2109.04344. 2021.
7. *Liu T., Wen W., Jin Y.* SIN2: Stealth infection on neural network – A low-cost agile neural Trojan attack methodology // IEEE Int. Symposium on Hardware Oriented Security and Trust. 2018. P. 227–230.
8. *Stefnisson S.* Evasive Malware Now a Commodity // SecurityWeek. 2018.
9. *Bidzhiev T., Namiot D.* Research of existing approaches to embedding malicious software in artificial neural networks // Int. J. Open Inform. Techn. 2022. V. 10. No. 9. P. 21–31.
10. *Bidzhiev T.* NNMalwareEmbedder. 2023. https://github.com/Temish09/NNMalwareEmbedder
11. *Keita K., Michel P., Neubig G.* Weight poisoning attacks on pretrained models // arXiv preprint arXiv:2004.06660. 2020.
12. *Lakshmanan R.* A Large-Scale Supply Chain Attack Distributed Over 800 Malicious NPM Packages // The Hacker News. 2022.
13. *IEEE Computer Society.* IEEE 754-2019 – IEEE Standard for Floating-Point Arithmetic. 2019.
14. *Snehal K., Neeta D. Jacobs D.* Implementation of lsb steganography and its evaluation for various bits // 1st International Conference on Digital Information Management. 2007. P. 173–178.
15. *Howard G.A.* MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications // arXiv:1704.04861. 2017.
16. *ytisf.* theZoo – A Live Malware Repository. 2021. https://github.com/ytisf/theZoo.
17. *Iandola N.F.* SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size // arXiv preprint arXiv:1602.07360. 2016.
18. *Krizhevsky A., Sutskever I., Hinton E.G.* Imagenet classification with deep convolutional neural networks // Advances in neural information processing systems. 2012. No. 25. P. 1097–1105.
19. *Simonyan K., Zisserman A.* Very deep convolutional networks for largescale image recognition // arXiv preprint arXiv:1409.1556. 2014.
20. *Rossum G. van.* pickle – Python object serialization // Python Software Foundation, Python Documentation. 2021.
21. *Trail of Bits.* Fickling. 2021. https://github.com/trailofbits/fickling.
22. *Acunetix.* What is Insecure Deserialization? // Acunetix. 2017.
23. *Paszke A.* PyTorch: An Imperative Style, High-Performance Deep Learning Library. 2019.

24. *Szegedy C.* Going deeper with convolutions // Proceedings of the IEEE conference on computer vision and pattern recognition. 2015. P. 1–9.
25. *Deng J.* Imagenet: A large-scale hierarchical image database // IEEE conference on computer vision and pattern recognition. 2009. P. 248–255.
26. *InQuest.* malware-samples. 2021. https://github.com/InQuest/malware-samples.
27. *Yansong G.* Strip: A defence against trojan attacks on deep neural networks // Proceedings of the 35th Annual Computer Security Applications Conference. 2019.
28. *Yansong G.* Backdoor attacks and countermeasures on deep learning: A comprehensive review // arXiv preprint arXiv:2007.10760. 2020.
29. *Parker S., Wu Z., Christofides D.P.* Cybersecurity in process control, operations, and supply chain // Computers & Chemical Engineering. 2023. V. 171. P. 108–169.
30. *Costales R.* Live trojan attacks on deep neural networks // arXiv:2004.11370. 2020

Статья представлена к публикации членом редколлегии А.А. Галяевым.

Поступила в редакцию 08.07.2023

После доработки 24.10.2023

Принята к публикации 20.01.2024

© 2024 г. М.М. ЗУЕВА (m.zueva@hse.ru),
С.О. КУЗНЕЦОВ, д-р физ. мат. наук (skuznetsov@hse.ru)
(Научно-исследовательский университет "Высшая школа экономики", Москва)

ИНДЕКСЫ ИНТЕРЕСНОСТИ ДЛЯ ПОСТРОЕНИЯ НЕЙРОННЫХ СЕТЕЙ НА ОСНОВЕ РЕШЕТОК ПОНЯТИЙ[1]

Трудность интерпретации результатов работы нейронных сетей является насущной проблемой, решению которой уделяется много внимания. Нейронные сети, основанные на решетках понятий, представляют собой перспективное направление в данной области. Отбор понятий для построения нейронной сети ключевым образом влияет на качество ее работы. Средством отбора понятий могут являться индексы интересности, когда для построения нейронной сети используются понятия с наибольшими показателями определенного индекса. В статье исследуется влияние выбора индекса интересности как средства отбора формальных понятий на качество работы нейронной сети.

Ключевые слова: архитектура нейронной сети, анализ формальных понятий, индексы интересности, нейронные сети на основе решеток понятий.

DOI: 10.31857/S0005231024030044, **EDN:** TWSZEY

1. Введение

Сложность интерпретации результатов при работе с нейронными сетями является важной проблемой, которой в последнее время посвящено много научных работ. Одним из возможных решений является построение нейронной сети на основе решеток понятий (concept lattice). В [1] была представлена нейронная сеть с архитектурой, построенной на основе решетки понятий для повышения устойчивости классификации. В [2] был предложен метод построения нейронной сети из решеток понятий, которые строились как на основе монотонных, так и антимонотонных соответствий Галуа.

Так как количество понятий растет экспоненциально с размером входных данных, важной задачей является возможность уменьшения количества понятий для построения нейронной сети без потери качества (ее) работы. Это можно сделать двумя способами: за счет отбора наиболее значимых признаков (предобработка) и за счет отбора наиболее важных ("интересных") понятий (постобработка). В [3] были рассмотрены различные методы отбора "интересных" понятий, основанных на "индексах интересности". В [4] меры интересности понятий сравнивались по таким аспектам, как эффективность вычисления и возможность их применимости к зашумленым данным.

В данной работе проведено исследование четырех индексов интересности (*basic level*, *target entropy*, Δ-*stability* и *lift*) в качестве критериев отбора инте-

[1] Работа была выполнена при поддержке Российского научного фонда (проект № 22-11-00323).

ресных понятий для построения нейронной сети и классификации объектов. Статья состоит из следующих разделов:

— в разделе 2 приведены основные определения анализа формальных понятий (АФП);

— раздел 3 посвящен теоретическим сведениям об изучаемых индексах интересности;

— в разделе 4 даны постановка задачи и формальное описание эксперимента;

— в разделе 5 представлен метод разработки архитектуры нейронной сети;

— в разделе 6 обсуждаются результаты экспериментов;

— раздел 7 содержит выводы, полученные по результатам работы.

2. Анализ формальных понятий

Обратимся к главным определениям из анализа формальных понятий (АФП) [5]. В АФП исследуется множество G объектов, множество M признаков и бинарное отношение $I \subseteq G \times M$ такое, что $(g, m) \in I$ тогда и только тогда, когда объект g имеет признак m. Такая тройка $K = (G, M, I)$ называется *формальным контекстом*. Используя *операторы Галуа*, определяемые для $A \subseteq G$, $B \subseteq M$ как

$$A' = \{m \in M \mid gIm \text{ для всех } g \in A\},$$
$$B' = \{g \in G \mid gIm \text{ для всех } m \in B\},$$

формальное понятие контекста K определяется как пара (A, B) такая, что $A \in G$, $B \in M$, $A' = B$, $B' = A$. При этом A называется *объемом*, а B называется *содержанием* понятия (A, B). Формальные понятия частично упорядочены отношением $\geqslant$:

$$(A_1, B_1) \leqslant (A_2, B_2) \iff A_1 \subseteq A_2,$$

которое задает полную (алгебраическую) решетку на множестве понятий, называемую *решеткой понятий* $L = (G, M, I)$.

Отношение покрытия, соответствующее частичному порядку $\leqslant$ (если оно существует), обозначается знаком $\prec$:

$$(A_1, B_1) \prec (A_2, B_2) \iff (A_1, B_1) \leqslant (A_2, B_2),$$

и не существует понятия (A_3, B_3) такого, что $(A_1, B_1) < (A_3, B_3) < (A_2, B_2)$.

3. Индексы интересности

Приведем формальное описание изучаемых в статье индексов интересности.

3.1. Базовый уровень (Basic Level)

Впервые общее определение базового уровня понятия было представлено в [6]. Неформально связностью понятия называется мера сходства всех пар

объектов из содержания понятия. Согласно идее Э. Роша, формализованной в [6], понятие (A, B) принадлежит базовому уровню, если оно удовлетворяет следующим условиям:

— (BL_1) (A, B) обладает высокой связностью;

— (BL_2) (A, B) обладает большей связностью, чем его верхние соседи (т.е. понятия, покрывающие понятие (A, B) в смысле отношения покрытия $\prec$);

— (BL_3) (A, B) обладает лишь чуть меньшей связностью, чем его нижние соседи (т.е. понятия, покрываемые понятием (A, B) в смысле отношения покрытия $\prec$).

В другом виде:

$$BL(A, B) = \mathcal{C}(\alpha_1(A, B), \alpha_2(A, B), \alpha_3(A, B)), \tag{1}$$

где $\mathcal{C}(\alpha_1, \alpha_2, \alpha_3) = \alpha_1 \otimes \alpha_2 \otimes \alpha_3$; $\otimes$ – t-норма.

В расчетах данного индекса предлагается использовать любое из двух следующих известных определений сходства множеств sim_Y:

$$sim_{SMC}(B_1, B_2) = \frac{|B_1 \cap B_2| + |Y - (B_1 \cup B_2)|}{|Y|}, \tag{2}$$

$$sim_J(B_1, B_2) = \frac{|B_1 \cap B_2|}{|B_1 \cup B_2|}. \tag{3}$$

Далее вводятся два индекса связности формального понятия:

$$coh^{\varnothing}(A, B) = \frac{\sum_{\{x_1, x_2\} \subseteq A, x_1 \neq x_2} sim(x_1, x_2)}{|A| \cdot (|A| - 1)/2} \tag{4}$$

– среднее сходство двух обьектов, входящих в объем данного формального понятия;

$$coh^{m}(A, B) = \min_{x_1, x_2 \in A} sim(x_1, x_2) \tag{5}$$

– наименьшая степень сходства двух объектов, входящих в объем данного формального понятия.

Так как в [7] авторы заключают, что показатель на основе индекса связности $coh^{\varnothing}(A, B)$ дает лучшие результаты отбора интересных понятий, в данной работе будут использованы только два вида показателя *базового уровня*, основанных на данном индексе: BL_{ees} – с использованием sim_{SMC} и BL_{eeJ} – с использованием sim_J.

В этих показателях

$$\alpha_1^{\varnothing} = coh^{\varnothing}(A, B), \tag{6}$$

$$\alpha_2^{\varnothing\varnothing} = 1 - \frac{\sum_{c \in \mathcal{UN}(A,B)} coh^{\varnothing}(c)/coh^{\varnothing}(A, B)}{|\mathcal{UN}(A, B)|}, \tag{7}$$

$$\alpha_3^{\varnothing\varnothing} = \frac{\sum_{c \in \mathcal{LN}(A,B)} coh^{\varnothing}(A, B)/coh^{\varnothing}(c)}{|\mathcal{LN}(A, B)|}. \tag{8}$$

3.2. Целевая энтропия (Target Entropy)

Целевая энтропия формального понятия определяется как дисперсия значений целевого признака, соответствующих содержанию данного формального понятия.

3.3. Δ-устойчивость (Δ-stability)

Устойчивость формального понятия является широко применяемой характеристикой, однако сложность алгоритма ее нахождения экспоненциально растет с увеличением количества признаков в содержании понятия. Поэтому в [8] была введена оценка устойчивости – *Δ-устойчивость*.

$$\Delta(p) = \min(\Delta(p, q)), q < p, \tag{9}$$

$\Delta(p, q)$ – оценка устойчивости сверху. Данная величина является минимальной разницей между размером объема понятия и размером объема ближайшего снизу понятия.

3.4. Подъем (Lift)

Согласно [9] *lift* определяется как отношение наблюдаемой совместной вероятности X и Y к их ожидаемой совместной вероятности, если бы они были статистически независимы.

В [10] приводится формула расчета индекса интересности *lift* формального понятия, для этого можно рассматривать только содержание формального понятия и общее множество признаков:

$$lift(A, B) = \frac{\prod\limits_{b \in B} Pr(b)}{Pr(B)}, \text{ где } Pr(\cdot) = \frac{|(\cdot)'|}{|G|}. \tag{10}$$

4. Постановка задачи

Выше были рассмотрены четыре индекса интересности понятий:

1) *Basic Level* (в данной работе были использованы BL_{ees} и BL_{eeJ});
2) *Δ-stability*;
3) *target entropy*;
4) *lift*.

Задача заключается в исследовании влияния выбора индекса для отбора понятий (когда формальные понятия уже получены). Исследование проводилось в следующей последовательности:

— бинаризация и подготовка датасета к обработке;

— построение формального контекста на основе набора данных;

— вычисление множества понятий на основе формального контекста;

— вычисление каждого индекса интересности для каждого формального понятия;

— сортировка понятий на основе величины изучаемого индекса;

— отбор k-лучших понятий для построения нейронной сети.

5. Архитектура нейронной сети

После отбора интересных понятий нейронная сеть строится на основе отношения покрытия на отобранных понятиях. Архитектура нейронной сети на основе решетки понятий выглядит следующим образом [2] (рисунок):

— входной слой (*Input Layer*) состоит из нейронов, связанных с признаками $m \in M$ контекста $K = (G, M, I)$;

— скрытые слои (*Hidden Layer*$_i$). Каждое формальное понятие может быть однозначно представлено своим содержанием. Признаки из множества признаков M итеративно соединяются в скрытых слоях таким образом, чтобы в последнем скрытом слое были получены нейроны, соответствующие отобранным формальным понятиям;

— выходной слой (*Output layer*). Число нейронов в данном слое соответствует числу целевых классов.

Для построения понятий из формального контекста были использованы инструменты библиотеки FCApy (https://pypi.org/project/fcapy/). Функции для расчета индексов BL_{ees} и BL_{eeJ}, $lift$ были написаны согласно определениям и формулам из раздела 3. Для расчета индексов *target entropy* и Δ-*stability* были использованы встроенные возможности библиотеки FCApy.

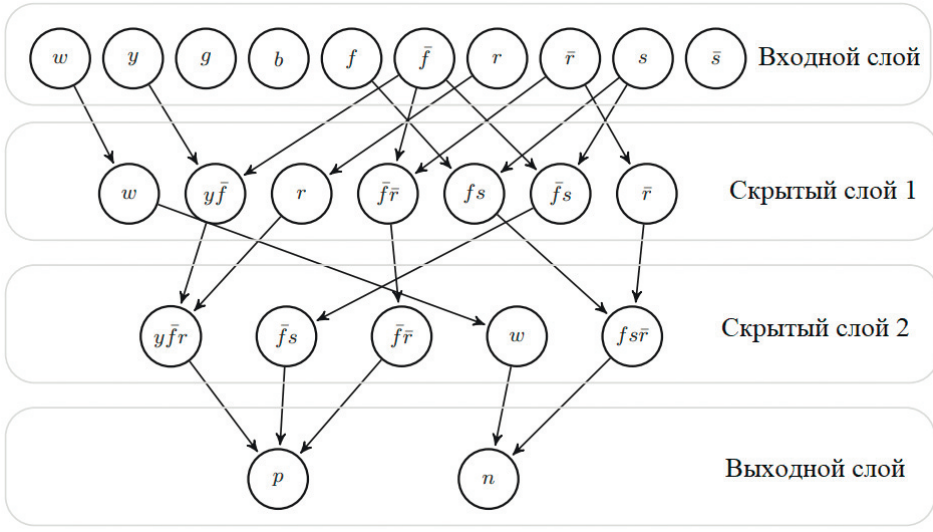


Рис. 1. Схема архитектуры нейронной сети на основе решетки понятий.

При выборе количества понятий использовался следующий критерий: наименьшее подмножество понятий, покрывающее все множество объектов:

$\{(A_1, B_1), (A_2, B_2), \ldots, (A_n, B_n)\} : A_1 \cup A_2 \cup \ldots \cup A_n = G.$

После расчетов индексов интересности для каждого индекса выбиралось k понятий с наибольшим значением данного индекса. Далее на основе данного множества понятий строилась нейронная сеть (использовались возможности библиотеки neural_lib, построенной на основе описания из [2]. Эта библиотека работает на основе пакета PyTorch).

Ее основные параметры: функция активации ReLU; оптимизатор Adam.

Наборы данных были разделены в отношении 70% и 30% на тренировочную и тестовую выборки. Проводились эксперименты с различным количеством генераций, лучшие результаты представлены в табл. 3–6.

5.1. Описание наборов данных

Для анализа были взяты четыре набора данных из библиотеки UCI (http://archive.ics.uci.edu/ml/) и предварительно бинаризованы. Названия и основные характеристики использованных наборов данных приведены в табл. 1.

Таблица 1. Характеристики наборов данных

Название набора	Количество объектов	Количество признаков	Количество классов
Heart Disease	303	33	2
House Votes	232	16	2
Car Evaluation	1727	21	4
Iris	150	16	3

Все используемые в работе наборы данных являются сбалансированными, кроме Car Evaluation.

5.2. Эксперименты с различными методами МО

Перед проведением основных экспериментов ряд базовых моделей были применены для анализа взятых наборов данных (табл. 2). Как видно из таблицы, лучшие результаты модели показывают на наборах данных House Votes и Iris, при этом для всех наборов данных лучшее качество получает модель XGBoost и случайный лес (Random Forest).

Таблица 2. Результаты State Of The Art моделей (метрика – Accuracy)

Название набора	Метод ближайшего соседа	Случайный лес	Наивный Байес	XGBoost	SVM
Heart Disease	0,83	0,85	0,81	0,81	0,79
House Votes	0,96	0,96	0,94	0,96	0,97
Car Evaluation	0,88	0,95	0,81	0,96	0,91
Iris	0,94	0,94	0,94	0,94	0,92

5.3. Сравнение результатов работы нейронной сети для разных индексов интересности

В табл. 3–6 приведены результаты экспериментов с индексами интересности. **Выделенным цветом** показаны результаты, сравнимые с качеством, полученным с использованием базовых моделей для тех же наборов данных.

Таблица 3. Результаты применения индексов интересности для набора данных Heart Disease

	BL_{ees}	BL_{eeJ}	target entropy	Δ-stability	lift
# генераций	8000	6000	8000	6000	7000
Recall	0,88	0,91	0,89	0,96	0,85
F1	0,84	0,80	0,88	0,95	0,84
Accuracy	**0,82**	0,76	0,72	**0,94**	**0,83**
# понятий	7	7	20	7	7

Таблица 4. Результаты применения индексов интересности для набора данных House Votes

	BL_{ees}	BL_{eeJ}	target entropy	Δ-stability	lift
# генераций	5000	2000	3000	2000	3000
Recall	0,85	0,94	0,94	0,97	0,94
F1	0,88	0,91	0,95	0,95	0,95
Accuracy	0,88	0,91	**0,95**	**0,95**	**0,95**
# понятий	7	7	20	7	7

Таблица 5. Результаты применения индексов интересности для набора данных Car Evaluation

	BL_{ees}	BL_{eeJ}	target entropy	Δ-stability	lift
# генераций	5000	5000	5000	5000	5000
Recall	0,44	0,45	0,25	0,47	0,25
F1	0,40	0,41	0,20	0,43	0,20
Accuracy	**0,82**	**0,84**	0,68	**0,87**	0,68
# понятий	7	7	20	7	7

Таблица 6. Результаты применения индексов интересности для набора данных Iris

	BL_{ees}	BL_{eeJ}	target entropy	Δ-stability	lift
# генераций	5000	3000	7000	5000	3000
Recall	0,95	0,95	0,87	0,95	0,95
F1	0,95	0,95	0,86	0,95	0,95
Accuracy	**0,95**	**0,95**	0,86	**0,95**	**0,95**
# понятий	7	7	20	7	7

Стоит отметить:

— Результаты качества с использованием индекса Δ-*устойчивости* в качестве критерия отбора понятий для всех четырех наборов данных оказались сравнимыми с эталонными моделями (метод ближайшего соседа, случайный лес, наивный Байес, XGBoost, SVM), тогда как индекс *target entropy* показал сопоставимые результаты только для набора данных House Votes (табл. 4).

— Индекс *lift* был успешен во всех экспериментах, кроме набора Car Evaluation (табл. 5).

— Индексы BL_{ees} и BL_{eeJ} показали близкие результаты, но для набора Heart Disease (табл. 3) индекс BL_{ees} оказался более успешен и сравним с эталонными моделями в отличие от BL_{eeJ}.

— Наихудшие результаты были получены для набора Car Evaluation (табл. 5), что можно объяснить его несбалансированностью при наличии четырех значений целевого признака.

— Самые высокие показатели качества были получены для наборов House Votes (табл. 4) и Iris (табл. 6). Это сбалансированные наборы данных со сравнительно небольшим количеством признаков в отличие от остальных использованных наборов данных.

— Индекс Δ-*устойчивость* во всех случаях показал более высокие показатели по сравнению с другими индексами интересности для тех же наборов данных.

6. Заключение

По полученным результатам можно сделать следующие выводы:

1) с помощью использования индексов интересности можно получить качество классификации, сравнимое с работой эталонных моделей;

2) индекс интересности *target entropy* показал наихудшие результаты относительно остальных индексов интересности;

3) индекс *lift* показал хорошие результаты, но не справился с классификацией несбалансированного набора данных с несколькими целевыми признаками;

4) индексы интересности *Basic Level* - BL_{ees} и BL_{eeJ} справились с классификацией в наборах данных с небольшим количеством признаков;

5) Индекс Δ-*устойчивости* в качестве средства отбора понятий показал хорошие результаты как на наборах данных с бинарным целевым признаком, так и при классификации с несколькими целевыми классами, в отличие от остальных индексов, исследованных в работе. Соответствующие показатели качества обучения превосходят полученные с помощью других индексов.

В дальнейшем планируется исследование других индексов интересности в качестве критериев отбора интересных понятий для построения нейронных сетей на их основе.

СПИСОК ЛИТЕРАТУРЫ

1. *Tsopze N., Nguifo E.M., Tindo G.* CLANN: Concept lattice-based artificial neural network for supervised classification // The Fifth International Conference on Concept Lattices and Their Applications. 2007. P. 24–26.
2. *Kuznetsov S.O., Makhazhanov N., Ushakov M.* On neural network architecture based on concept lattices // ISMIS 2017. P. 653–663.
3. *Kuznetsov S.O., Makhalova T.P.* Concept interestingness measures: a comparative study // Proceedings of the Twelfth International Conference on Concept Lattices and Their Applications. 2015. P. 59–72.
4. *Kuznetsov S.O., Makhalova T.P.* On interestingness measures of formal concepts // Inf. Sci. 442. 2018. P. 202–219.
5. *Ganter B., Wille R.* Contextual attribute logic / International Conference on Conceptual Structures. 1999. P. 377–388.
6. *Rosch E.* Basic objects in natural categories // Cognitive Psychology 8. 1976. P. 382–439.
7. *Belohlavek R., Trnecka M.* Basic level of concepts in formal concept analysis // ICFCA 2012. P. 28–44.
8. *Buzmakov Al., Kuznetsov S.O., Amedeo Napoli.* Scalable Estimates of Concept Stability // ICFCA 2014. P. 157–172.
9. *Zaki M.J., Meira W., Jr.* Data Mining and Analysis: Fundamental Concepts and Algorithms // Cambridge University Press. 2014. P. 339.
10. *Makhalova T.* Interesting Measures of Closed Patterns for Data Mining and Knowledge Discovery // HSE University, Moscow, Russia. 2020. P. 25.

Статья представлена к публикации членом редколлегии А.А. Галяевым.

Поступила в редакцию 08.07.2023

После доработки 16.10.2023

Принята к публикации 20.01.2024

© 2024 г. Е.Д. ВЯЗИЛОВ, д-р техн. наук (vjaz@meteo.ru),
Д.А. МЕЛЬНИКОВ (melnikov@meteo.ru)
(Всероссийский научно-исследовательский институт гидрометеорологической информации – Мировой центр данных, Обнинск),
О.А. МИНКОВ (oaminkov@gmail.com)
(Обнинский институт атомной энергетики – филиал Национального исследовательского ядерного университета «МИФИ»)

ОБ ИСПОЛЬЗОВАНИИ ДАННЫХ ЦИФРОВЫХ ДВОЙНИКОВ В МОДЕЛЯХ, СВЯЗАННЫХ С УЧЕТОМ ВОЗДЕЙСТВИЯ ОКРУЖАЮЩЕЙ СРЕДЫ НА ПРЕДПРИЯТИЯ

Цифровые двойники объектов отражают состояние окружающей среды и деятельность предприятий, на которые воздействует среда. Предлагается использовать модели для расчета показателей оценки воздействия опасных природных явлений или изменений климата; прогноза этих воздействий; оценки убытков; расчета стоимости мероприятий по защите предприятий; оценки целесообразности проведения превентивных мероприятий с целью их оптимизации. Приведены требования к моделям оценки воздействий, работающим с цифровым двойником. Представлены трудности при использовании таких моделей. Рассматриваются предложения по разработке отдельных моделей воздействий. Показана схема использования цифровых двойников при моделировании воздействий окружающей среды на предприятия.

Ключевые слова: цифровой двойник, модели оценки воздействий, окружающая среда.

DOI: 10.31857/S0005231024030059, **EDN:** TUQHLT

1. Введение

Глобальные проблемы изменения климата нарастают, условия ведения бизнеса становятся трудными из-за усложнения самих предприятий и растущей зависимости от складывающихся гидрометеорологических условий (ГМУ). Информация о состоянии окружающей среды используется для решения бизнес-процессов, зависящих от ГМУ. При огромном объеме гидрометеорологических данных руководителям предприятий сложно понять, какие категории наблюденных, прогнозных или климатических данных следует использовать в бизнес-процессах. Одновременно с гидрометеорологическими данными [1] в некоторых моделях используются экономические, финансовые, технические, социальные и другие сведения о предприятии. Модели прогноза воздействий опираются на интегрированные данные, полученные из различных доменов. Существующие системы интеграции усваивают данные, которые поставщики данных предоставляют в общее пользование.

Это приводит к фрагментарному предоставлению данных. Автоматизация использования гидрометеорологических данных достигла такого уровня, что сразу после измерения значений параметров о состоянии ГМУ получаются комплексные показатели состояния погоды — комфортность, суровость, уровень опасности и др. В строительстве используются строительные нормы и правила [2], пособия типа [3], в которых отражены эмпирические модели, позволяющие рассчитывать снеговые и гололедные нагрузки на кровли домов, определять инженерную защиту территории от затопления. Для экономической оценки адаптации отраслей к изменениям климата разработаны простые программные средства [4]. Развиваются цифровые двойники (ЦД), в том числе и в области окружающей среды [1, 5, 6]. ЦД — это база данных нового типа, предоставляющая данные в стандартизованной структуре для выявления опасных явлений (ОЯ), "моделирования и прогнозирования воздействий ОЯ на" [7] предприятия, оценки ущерба и стоимости превентивных мероприятий, оптимизации и принятия решений [1, 8]. На основе ЦД появляется возможность моделировать условия эксплуатации и предсказать состояние предприятий и бизнес-процессов при воздействии ОЯ. Потенциал ЦД заключается в способности доставлять данные моделям в составе атрибутов, которые необходимы для решения конкретных бизнес-процессов. Целью исследования является определение предварительного состава моделей для оценки воздействий окружающей среды на предприятия, которые используются для повышения эффективности гидрометеорологического обеспечения предприятий с использованием ЦД объектов.

2. Схема использования цифровых двойников при моделировании

Направлениями использования ЦД в области учета ГМУ являются [6]:

1. Моделирование развития новых промышленных районов. При развитии новых промышленных районов, постройке, строительстве и эксплуатации предприятий ЦД позволяет оценить сценарии расположения предприятий с учетом климатических условий, транспортных возможностей и экономических выгод. Это позволяет выбрать сценарий развития промышленных районов с учетом экологической, гидрометеорологической и транспортной безопасности.

2. Расположение предприятий внутри промышленного района и планирование доставки сырья и материалов между ними. С помощью ЦД моделируются логистические операции, которые позволяют повысить скорость доставки материалов и сырья, оптимизировать бизнес-процессы предприятия (например, повысить безопасность доставки грузов и материалов).

3. Аналитика и оптимизация решений. На основе ЦД получается расширенная аналитика и оптимизируются решения за счет выявления аномалий, превышений пороговых значений, расчета тенденций и других показателей состояния ГМУ для выдачи прогноза воздействий и рекомендаций.

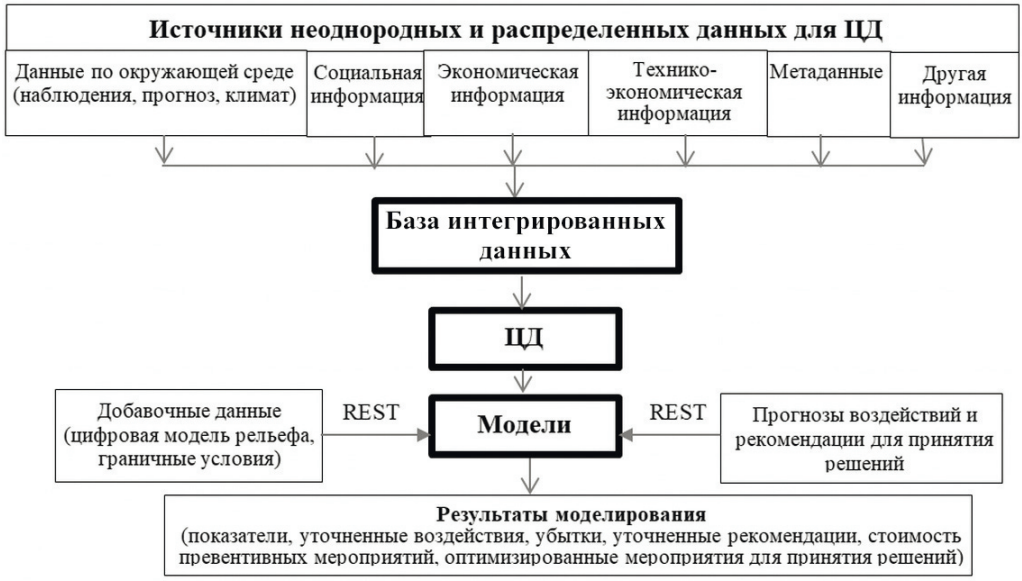


Рис. 1. Схема использования ЦД при моделировании.

4. Управление рисками. ЦД используется для учета уровня опасности ситуаций, связанных с влиянием ГМУ, например на работу транспорта, перевозимые грузы, операции погрузки–разгрузки. Вероятность влияния на эту деятельность отдельных явлений и вероятность убытков от них известна.

Схема использования ЦД при моделировании воздействия ОЯ представлена на рис. 1. ЦД здесь рассматривается как цифровое представление свойств окружающей среды и сопутствующих данных, связанных с социальными, экономическими, технологическими ситуациями, складывающимися на предприятиях. Для интеграции данных в Росгидромете используется Единая государственная система информации об обстановке в Мировом океане (ЕСИМО) [9]. Информационные ресурсы в этой системе хранятся в унифицированном виде и предназначены для применения в режиме самообслуживания на портале по адресу http://esimo.ru. Данные, не интегрированные в ЕСИМО, доставляются до ЦД с помощью REST-сервисов. ЦД отражает как данные по окружающей среде, так и другим сферам и используется при моделировании воздействий окружающей среды.

Сложные модели, например расчетно-модельный комплекс моделирования разливов нефти, включенный в ЕСИМО, работают на удаленном сервере, и для него регулярно готовятся и доставляются интегрированные данные на этот сервер, где автоматически запускается модель [9, 10]. Пользователь, зная место, время разлива и объем разлитой нефти, запускает модель. Результаты моделирования передаются на портал и там визуализируются как анимация пятна распространения нефти на карте. Пользователь проводит анализ на предмет, куда движется разлив и когда достигнет побережья.

ЦД используется для:

— выявления ОЯ для конкретных предприятий на основе локальных пороговых значений показателей состояния ГМУ;

— получения данных по составу только тех, которые нужны предприятию;

— использования не только данных о состоянии окружающей среды, но и «прогноза возможных воздействий ОЯ на деятельность предприятия, оценки ущерба, расчетов стоимости превентивных мероприятий» [5];

— моделирования воздействий ОЯ на бизнес-процессы предприятий;

— повышения осведомленности руководителей о складывающихся ГМУ, воздействиях ОЯ на предприятия и возможных ущербах.

С помощью ЦД проводится анализ воздействий ОЯ и моделирование ситуаций на цифровой модели объекта, при которых происходят разрушения кровли крыш, аварии, порча продукции и т.п. Выполненное с помощью ЦД моделирование воздействий ОЯ на предприятия позволяет увидеть на цифровой модели новые опасные места или недостаточную ветроустойчивость объекта.

Пути, методы и средства для инженерных изысканий в области гидрометеорологии, включающие модели по расчету показателей состояния гидрометеорологической обстановки, оценке уровня опасности ОЯ и моделированию воздействий окружающей среды на морскую деятельность, отражены в пособии [3]. Далее кратко рассмотрим модели, которые используют данные из ЦД.

3. Описание моделей

3.1. Модели прогноза погоды

Модели прогноза погоды (температуры воздуха, влажности, ветра, атмосферного давления по пространству с различными пространственно-временными масштабами разрешения) применяются в России и других странах [11, 12]. Эти модели на основе данных наблюдений, используя классические уравнения динамки и термодинамики атмосферы, интерполируют значения параметров в узлы регулярной сетки. Результаты работы этих моделей представляют объекты ЦД для обслуживания потребителей, выявления ОЯ и прогноза воздействий этих явлений на население и предприятия. Полученные поля распределения наблюденных, расчетных и прогностических параметров используются для прогноза гидрологических, морских и других явлений, переноса загрязняющих веществ. Например, рассчитывается прогноз направления и высоты волн или течений на основе ветра. На основе метеопрогнозов вычисляется распространение вулканического пепла после извержения вулкана [13, 14].

Сегодня потребители используют информационную продукцию, полученную на основе моделей анализа и прогноза. Продукция включает карты пространственного распределения отдельных параметров; таблицы с цифровыми

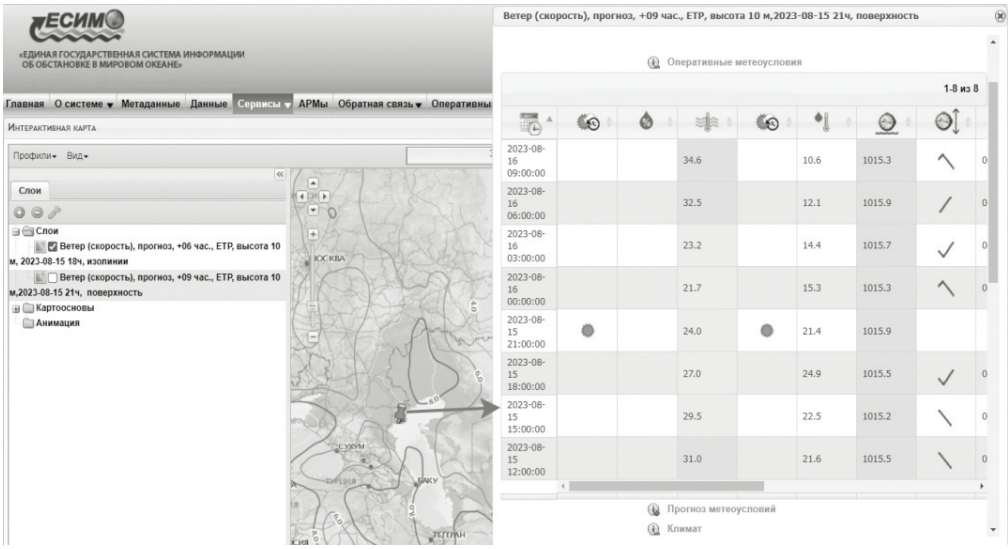


Рис. 2. Результаты оценки уровня опасности для точки на карте.

данными; графики временного хода; описания сложившихся ГМУ, подготовленных на основе наблюденных данных. Детально познакомиться с информационной продукцией в форме результатов визуализации анализов и прогнозов можно на сайте Гидрометцентра России (https://meteoinfo.ru/).

3.2. Вычисление показателей оценки воздействий опасных явлений

Для некоторых явлений вместо превышений пороговых значений в наблюденных данных лучше использовать комплексные показатели — индекс суровости и комфортность погоды, показатель пожарной опасности, характеристики устойчивости предприятий к внешним воздействиям ОЯ (влаго-, морозо-, засухо-, ветро-, волноустойчивость). Эти показатели вычисляются по наблюденным и прогностическим значениям параметров на основе физических или эмпирических моделей. Результаты оценки уровня опасности отдельных показателей и предоставления наблюденных, прогностических и климатических данных в одном интерфейсе для выделенной точки на карте представлены на рис. 2. Исследования по расчету показателя комфортности климата и погоды представлены в серии докладов на конференции [15, 16].

3.3. Прогноз интенсивности воздействий

Модели оценки воздействий условий среды на предприятия индивидуальны и разрабатываются по заказу, например при строительстве гидротехнических сооружений, которые являются защитными сооружениями от наводнений, селей и т.п. Такие модели используются также при проектировании крупных предприятий и позволяют оценить возможные воздействия с учетом характеристик этих объектов.

Эмпирические модели основаны на связи между количественными значениями показателей изменения климата и результатами конкретных экономических процессов. При расчете высоты волнолома используется правило: чем выше поднимется уровень мирового океана, тем выше должна быть высота дамбы. Надо найти компромисс между стоимостью строительства дамбы и суммарной ежегодной стоимостью ремонта. Много моделей связано с предсказанием урожайности зерновых и доставкой товаров в торговую сеть при учете изменений потребностей населения в товарах в зависимости от погоды [17, 18]. Другими примерами моделей являются:

— оптимизация расположения предприятий и транспортных путей, чтобы не заметало снегом, с учетом розы ветров, характеристик рельефа местности;

— выбор параметров ледозащитных сооружений для объектов на берегу;

— оптимизация процессов складирования грузов в порту в зависимости от условий хранения грузов, подверженных воздействию влажности, низких или высоких температур;

— повышение эффективности работы коммунальных служб: оптимизация отопления с учетом температуры воздуха на улице, розы ветров, других характеристик.

3.4. Оценка возможного ущерба

Авария танкера приводит к экологической катастрофе. Список ущербов, вызванных разливом нефти, и затрат на ликвидацию последствий включает [19]: стоимость операций по очистке воды и побережья от нефти; выплаты страховки за ущерб. Возникает ущерб от уменьшения уловов рыбы, утраты груза с нефтью у собственника груза, загрязнения пляжей, гибели биоты.

В МЧС России используются модели расчета [20] возможных разрушений, числа погибших и раненных при землетрясениях; подъема воды по цифровой модели рельефа для вычисления продолжительности эвакуации; зон распространения низового пожара с использованием направления и скорости ветра, влажности; времени остывания помещений до минусовых значений внутри помещения в зависимости от температуры воздуха.

Программа «Расчет вероятного количества погибших и спасенных пострадавших с комбинированными повреждениями в морских катастрофах» позволяет «на основе информации о количестве пассажиров на судне, времени начала спасательной операции, температуры морской воды, удаленности от берега рассчитать количество погибших в катастрофе и структуру пострадавших». [21].

Материальный ущерб, связанный с ОЯ, включает потери от простоя предприятия; стоимость неполученной продукции, ремонта испорченного оборудования, механизмов; ремонта зданий. Ущерб от прохождения судами сложных ледовых условий рассматривается в следующих вариантах: ледокольная проводка — ущерб связан с опозданием доставки груза, стоимостью проводки судна ледоколом; ожидание благоприятных ледовых условий — ущерб определяется простоем судов; прохождение судна в сложных условиях без ледоко-

ла, возможные вариант — гибель или авария судна. При гибели судна ущерб включает стоимость судна, груза и судового имущества; ущерб здоровью людей, включая их возможную гибель; выплаты за моральный ущерб. При аварии ущерб включает потери прибыли за время аварии и ремонта; стоимость ремонта, буксировки аварийного судна, вознаграждение за спасение; штраф, неустойку, неуплату фрахта в связи с опозданием доставки груза.

3.5. Расчет стоимости проведения превентивных мероприятий

«Для принятия решения, кроме возможного ущерба, необходимо знать стоимость превентивных мероприятий» [22]. До явления укрываются грузы, боящиеся влаги, эвакуируются ценные грузы в безопасное место и т.д. Большинство расчетов стоимости превентивных мероприятий складывается из [23] заработной платы, участвующих в превентивных мероприятиях, стоимости арендованной техники, дополнительного оборудования для усиления мер безопасности, заблаговременного строительства защитных сооружений, расходных материалов; затрат на эвакуацию людей и т.п. [7]. Единовременные затраты производятся для конкретного предприятия один раз, например строительство дамбы, предотвращающей наводнение, или волнолома, препятствующего прохождению волн в бухту, где находится порт. Постоянные затраты относятся к единице времени год — ежегодная подготовка жилищно-коммунального хозяйства к зимнему сезону; эксплуатация защитных сооружений.

3.6. Оптимизация проведения превентивных мероприятий

Модели оптимизации применяются в случае выдачи рекомендаций по проектам превентивных мероприятий, требующих больших материальных затрат на выполнение и расчета времени на эвакуацию людей и имущества. Предложенные системой поддержки решений рекомендации уточняются с помощью математических моделей [24]. Демонстрационный вариант модели расчета стоимости превентивных мероприятий и оценки ущерба [7] реализован в виде мобильного приложения, рис. 3. Приложение позволяет руководителю решить, проводить или не проводить превентивные мероприятия.

«Оптимальное решение — это хозяйственное решение потребителя, принимаемое на основании информации о состоянии окружающей среды и обеспечивающее получение максимального экономического эффекта или минимального ущерба, или обеспечение безопасности работ, людей, предприятий. В качестве критериев оптимальности выступают средние потери, минимальная вероятность потерь, превышающая некоторый заданный уровень максимально возможного ущерба, средний выигрыш, минимакс и др.» [24].

Для некоторых ОЯ (сели, цунами, смерчи) «методы прогноза не всегда дают точные результаты, и руководитель оказывается перед дилеммой: применять или не применять защитные мероприятия при прогнозе возникновения ОЯ. У него имеется три стратегии: никогда не применять защитные ме-

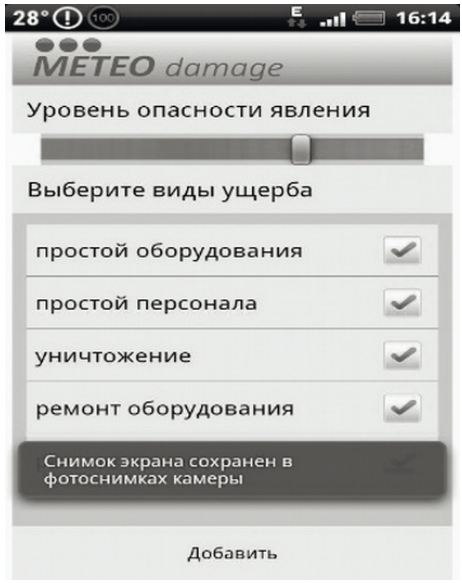


Рис. 3. Схема использования ЦД при моделировании.

роприятия; всегда применять защитные мероприятия; применять защитные мероприятия выборочно, ориентируясь на интуицию или дополнительную информацию» [7]. Использование экономических моделей позволяет иметь аргументированное решение по проведению тех или иных мероприятий.

Оценка ущерба носит вероятностный характер. При выработке решения руководитель не знает, какие значения будут принимать влияющие на ущерб показатели ОЯ в каждый момент времени, и рассматривает только распределение вероятностей этих значений. Поэтому потери потребителя тоже носят вероятностный характер. Чтобы выбрать то или иное решение в качестве экономически оптимального, определяется сначала закон распределения потерь, который используется в качестве критерия оптимального решения. С этой целью применяются средние в статистическом смысле потери. Этот метод минимизирует потери, ожидаемые в среднем за бесконечно долгое время, и никак не ограничивает вероятность ущерба.

Целесообразно найти оптимальную хозяйственную стратегию, придерживаясь которой, потребитель сводил бы риск к минимуму. Для отыскания такой стратегии применительно к каждой конкретной задаче строится модель, отражающая реакцию объекта на условия среды и позволяющая построить функцию полезности — зависимость затрат, ущерба, прибыли при ГМУ в виде матрицы с расчетом максимальных, минимальных и средних затрат. Критериями принятия решений при неопределенности являются критерии Лапласа, Сэвиджа, Гурвица [24]. Эти критерии используются для определения оптимального значения затрат для минимизации затрат на строительство дамбы и проведения превентивных мероприятий при нескольких значениях вероятностей наводнения [1].

Примеры других оптимизационных задач:

— оптимизация выделения складских помещений для грузов, боящихся воздействий окружающей среды (скоропортящиеся грузы, или грузы, боящиеся осадков, и др.);

— буксировка опасных, ценных или крупногабаритных грузов на большие расстояния;

— расчет прибыли порта при учете сроков вскрытия припая в арктических портах;

— рекомендованные курсы следования судов в зависимости от скорости ветра, высоты волн и ледовых условий.

4. Требования к моделям

4.1. Требования к математическим, экономическим и другим типам моделей, используемых для оценки воздействий окружающей среды на предприятия

Трудностями при использовании моделей являются [1]:

— дефицит открытых данных;

— незрелость и отсутствие междисциплинарного диалога при аналитике данных, полученных от поставщиков данных;

— необходимость расширения междисциплинарных исследований, посвященных моделированию воздействий окружающей среды на предприятия;

— большой объем данных, представляемых руководителям предприятий в виде карт, диаграмм, графиков, таблиц, что требует автоматизации аналитических функций;

— необходимость расширения интеграции разнородных и распределенных данных.

С учетом этих трудностей выработаны следующие требования к моделям. Модели готовы к повторному выполнению в любое время ($24 \times 7 \times 365$) и для них всегда готовы необходимые актуальные данные в виде ЦД. Модели работают в распределенной среде. Доставка данных до моделей происходит в форматах обмена JSON или XML с помощью REST-, веб- или API-сервисов. Модели прогноза автоматически запускаются в установленные регламентом сроки, связанные с поступлением новых порций данных. Модели оценки воздействий запускаются по событию в случае выявления ОЯ и возникновения опасности предприятию.

4.2. Требования к модели данных цифрового двойника

Использование ЦД в моделях требует использования универсальной модели данных [25]. Объектами такой модели являются реестры, справочники и перечисления. Реестры — представляет записи со значениями атрибутов и связи между записями, относящимися к одному объекту. Справочники — это классификаторы, используемые в реестрах. Перечисления — это список

множественных значений атрибутов. За счет перечислений в одном атрибуте хранится несколько значений одного свойства объекта. Для реализации модели данных применяются следующие принципы:

— ЦД состоит из двух типов данных (временные ряды, данные в узлах сетки);

— каждый тип данных включает несколько объектов, относящихся к различным предметным областям, данные которых объединяются в ЦД;

— объекты одного типа хранятся в одной модели данных;

— каждый экземпляр объекта ЦД имеет уникальный идентификатор;

— цифровые объекты имеют одинаковые поисковые атрибуты;

— множественные значения для каждого свойства объектов записываются в одном атрибуте в виде списка значений через разделитель;

— каждый цифровой объект имеет экземпляр метаданных, что облегчает их поиск;

— метаданные обновляются автоматически после поступления новой порции данных;

— данные для ЦД непрерывно загружаются, обрабатываются и доставляются потребителям на основе конвейера [10].

Временные ряды хранятся в виде двух таблиц — метаданные и данные. Метаданные отражают сведения о временных рядах — временное разрешение, координаты точки, название пункта, имя параметра, другие. Данные временного ряда представляются как множественные значения в виде списка для каждого параметра.

Данные в узлах регулярной сетки тоже состоят из двух сущностей — метаданные и данные. Метаданные включают сведения о свойствах сетки и каждого поля — координаты района, дата–время, шаг сетки, имя параметра и другие атрибуты метаданных. Данные в узлах сетки представляются в виде списка множественных значений атрибутов в узлах сетки — широта, долгота, время и значения параметров. Связи с другими объектами и классификаторами организуются в виде ссылок.

5. Заключение

Представлены подходы по использованию данных ЦД для оценки влияния окружающей среды на деятельность предприятия с помощью математических, экономических моделей для расчета показателей устойчивости предприятий к воздействиям среды, идентификации ОЯ, прогноза воздействий, оценки ущерба, расчета стоимости превентивных мероприятий и оптимизации решений. Предложена унифицированная модель данных ЦД для объекта, представленного в виде одной таблицы. Это позволяет иметь связи как между объектами, так и отдельными экземплярами свойств среды и предприятия.

ЦД отражает состояние окружающей среды и деятельность предприятий в четырехмерном пространстве — широта, долгота, высота, время в виде вре-

менных рядов и данных в узлах регулярной сетки. Получаемый цифровой аналог состояния окружающей среды в виде значений параметров среды и экономических показателей предприятий позволяет отслеживать показатели опасности окружающей среды для предприятий, моделировать и прогнозировать воздействия на деятельность предприятий. Модели, использующие данные ЦД, помогут руководителям получать сведения об ОЯ. Это позволит оптимизировать бизнес-процессы, сократить ущерб от воздействия ОЯ и изменений климата, повысить эффективность работы предприятия.

СПИСОК ЛИТЕРАТУРЫ

1. *Вязилов Е.Д.* Цифровая трансформация гидрометеорологического обеспечения потребителей. Обнинск: ВНИИГМИ-МЦД. Т. 2. Направления использования. 2022. 356 с.
2. Нагрузки и воздействия на гидротехнические сооружения (волновые, ледовые и от судов): СНИП 2.06.04-82. Госстрой СССР. М.: Стройиздат, 1986. 40 с.
3. Инженерно-гидрометеорологические изыскания на континентальном шельфе. М.: Гидрометеоиздат, 1993. 377 с.
4. European Union's project ECONADAPT Toolbox provides easily accessible information on the economic assessment of adaptation. https://econadapt-toolbox.eu/easy-access-guide. Accessed: 8 February 2019.
5. *Вязилов Е.Д.* Цифровой двойник для окружающей среды // Сборник трудов Международной конференции "ENVIROMIS 2022" и школа молодых ученых по измерениям, моделированию и информационным системам для изучения окружающей среды. Томск: ИМКЭС СО РАН. 12–17 сентября 2022. С. 323–326.
6. *Viazilov E.D.* About Creating a Digital Twins in Field of Earth Sciences // Int. J. Appl. Sci. Development. 2022. V. 1. Art. 6. https://doi.org/10.37394/232029.2022.1.6. https://wseas.com/journals/asd/2022/a12asd-006(2022).pdf. Published: December 31, 2022. P. 42–51.
7. *Вязилов Е.Д.* Новые подходы по доведению информации об опасных гидрометеорологических явлениях и повышению информированности лиц, принимающих решения // Конференция «Проблемы прогнозирования чрезвычайных ситуаций». XVI Всероссийская научная конференция. Москва, 27–28 сентября 2017 г. Сборник материалов. М.: МЧС России, ФКУ «Антистихия». 2017. С. 40–44.
8. *Viazilov E.D.* From Informing Users about Disasters to Issuing a Forecast of Possible Impacts and Recommendations // J. Engineering World. 2022. No. 4. P. 34–43. https://wseas.com/journals/ew/2022/a12engw-5115-806.pdf
9. ЕСИМО. Единая государственная система информации об обстановке в Мировом океане. 2013. URL: http://esimo.ru. Доступ: 04.01.2023.
10. *Viazilov E.D., Melnikov D.A., Mikheev A.S.* On the development of a pipeline for processing hydrometeorological data // Supplementary Proceedings of the XXIII International Conference on Data Analytics and Management in Data Intensive Domains DAMDID/RCDL. 2021. V. 3036. http://ceur-ws.org/Vol-3036/paper08.pdf.

11. *Olchev A.V., Rozinkina I.A., Kuzmina E.V., Nikitin M.A., Rivin G.S.* Influence of forest cover changes on regional weather conditions: estimations using the mesoscale model COSMO // IOP Publishing Ltd. 2018. IOP Conf. Ser.: Earth Environ. Sci. V. 107, 012105. 7 p. https://doi.org/10.1088/1755-1315/107/1/012105.

12. Динамика поля геофизического параметра атмосферы над акваториями Мирового океана: интегральное влагосодержание атмосферы (TPW), водозапас облаков (CLW) и скорость приповерхностного ветра (WND). Шаг по времени – 3 часа. Шаг сетки – 0,25°. https://fireras.su/tpw/. Доступ: 09.01.2024.

13. *Сорокин А.А., Королев С.П., Гирина О.А. и др.* Интегрированная программная платформа для комплексного анализа распространения пепловых шлейфов при эксплозивных извержениях вулканов Камчатки // Современные проблемы дистанционного зондирования Земли из космоса. 2016. Т. 13. В. 4. С. 9–19.

14. *Ермаков Д.М., Чернушич А.П., Шарков Е.А.* Геопортал спутникового радиотепловидения: данные, сервисы, перспективы развития // Современные проблемы дистанционного зондирования Земли из космоса. 2016. Т. 13. В. 3. С. 46–57.

15. *Коспанов А.А., Константинов П.И.* Сравнение влияния зеленых и белых крыш на городской остров тепла на примере трех волн жары в Москве // Международная молодежная школа и конференция по вычислительно-информационным технологиям для наук об окружающей среде. (CITES 2023). 13–23 июня 2023. М.: С. 68–69.

16. *Левищева Т.П., Константинов П.И.* Применение локальных метеорологических моделей для воспроизведения городского микроклимата на примере Москвы // Международная молодежная школа и конференция по вычислительно-информационным технологиям для наук об окружающей среде. (CITES 2023). 13–23 июня 2023. М.: С. 83.

17. Методика определения размера, вреда, который может быть причинен жизни, здоровью физических лиц, имуществу физических и юридических лиц в результате аварии судоходных гидротехнических сооружений. Утв. Приказом МЧС России и Минтранса России от 02.10.2007.

18. *Хандожко Л.А.* Экономическая метеорология. СПб.: Гидрометеоиздат, 2005. 490 с.

19. *Ивченко А.А., Зацепа С.Н., Солбаков В.В. и др.* Модельный комплекс SPILLMOD-RA для расчета статистических характеристик распространения разливов нефти в море на основе тематического набора данных реанализа метеорологических полей // Свидетельство о государственной регистрации программы для ЭВМ. Номер свидетельства: RU 2020665648. 2020. Номер заявки: 2020664664.

20. *Алабян А.М., Зеленцов В.А., Крыленко И.Н. и др.* Масштабируемая региональная система мониторинга и оперативного прогнозирования речных наводнений: результаты разработки и тестирования. М.: МЧС России, 2018. 11 с.

21. *Закревский Ю.Н.* Обоснование системы оказания медицинской помощи и лечения пострадавших в морских катастрофах // Автореф. дисс. . . . д-ра мед. наук по специальности 05.26.02 – «Безопасность в чрезвычайных ситуациях». Архангельск: ГБОУ «Северный государственный медицинский университет», 2013. 40 с.

22. *Вязилов Е.Д., Чуняев Н.В.* О смене парадигмы гидрометеорологического обслуживания сведениями об опасных явлениях / Труды Гидрометцентра России,

2016. Вып. 362. Гидрометеорологические прогнозы. Под редакцией д-ра геогр. наук Е.С. Нестерова. С. 224–235.

23. *Чуняев Н.В.* Информационная поддержка управления морской деятельностью в случае опасных природных явлений. В сб.: Труды Главной геофизической обсерватории им. А.И. Воейкова. 2015. В. 578. С. 156–173.

24. *Мадера А.Г.* Математические модели и принятие решений в управлении: Руководство для топ-менеджеров. М.: УРСС, 2021. 684 с.

25. *Viazilov E.D., Puzova N.V., Mikheev A.S., Melnikov D.A.* Choosing a Data Model for the Environmental Digital Twin. Supplementary Proceedings of the XXIV International Conference on Data Analytics and Management in Data Intensive Domains (DAMDID/RCDL 2022) // Pleiades Publishing, Ltd. Special issue of the Lobachevskii Journal of Mathematics. 2023. V. 44. Is. 1. P. 237–248. https://doi.org/10.1134/S1995080223010444

Статья представлена к публикации членом редколлегии А.А. Галяевым.

Поступила в редакцию 08.07.2023

После доработки 23.09.2023

Принята к публикации 20.01.2024

© 2024 г. К.А. НАЙДЕНОВА, канд. техн. наук (ksennaid@gmail.com)
(Военно-медицинская академия имени С.М. Кирова, Санкт-Петербург),
В.А. ПАРХОМЕНКО (vladimir.parkhomenko@spbstu.ru)
(Санкт-Петербургский политехнический университет Петра Великого),
Т.А. МАРТИРОВА (martta462@yandex.ru)
(Военно-медицинская академия имени С.М. Кирова, Санкт-Петербург),
А.В. ЩУКИН, канд. техн. наук (alexander.schukin@spbstu.ru)
(Санкт-Петербургский политехнический университет Петра Великого)

ПРАВДОПОДОБНЫЕ РАССУЖДЕНИЯ В АЛГОРИТМЕ ГЕНЕРАЦИИ ХОРОШИХ КЛАССИФИКАЦИОННЫХ ТЕСТОВ

Статья посвящена применению принципов (правил) правдоподобных рассуждений к символьному машинному обучению (МО). Эти применения существенны и необходимы для увеличения эффективности алгоритмов МО. Множество таких алгоритмов порождают и используют правила в форме импликаций. Обсуждается генерация этих правил по отношению к классам объектов. Эти классификационные правила специфичны. Их посылки, называемые хорошими замкнутыми тестами (ХЗТ), покрывают максимально возможное множество объектов. Представлен один из алгоритмов генерации ХЗТ, называемый NIAGARA. Алгоритм пересмотрен и предложены новые процедуры на основе правдоподобных рассуждений. Их корректность доказывается. Используются следующие правила: импликации, запреты, индуктивные правила расширения текущих множеств целевых объектов, правила сокращения области поиска решений. Они позволяют увеличить эффективность алгоритма.

Ключевые слова: правдоподобные рассуждения, замкнутые множества, хорошие диагностические тесты, анализ хороших тестов, символьное машинное обучение.

DOI: 10.31857/S0005231024030066, **EDN:** TRSWBL

1. Введение

В статье предлагается новая, более эффективная версия алгоритма NIAGARA, представленная в [1] для генерации максимально избыточных хороших тестов (ХМИТ, ЗХТ), введенных в [2]. Увеличение эффективности алгоритма основано на введении нескольких новых импликативных правил правдоподобного вывода, которые делают возможным использовать непосредственно свойства целевых объектов, генерируемых в алгоритме. Правила правдоподобных рассуждений играют большую роль в конструировании алгоритмов извлечения знаний из данных. В свою очередь, эти правила порождаются при использовании методов машинного обучения.

Правила правдоподобного вывода продуктивны в решении следующих проблем: формирование контекстов для решаемых проблем, сокращение области поиска решений, формирование описаний объектов, выделение существенных элементов в обработке данных, выделение отношений между эле-

ментами в области поиска решений, интерпретация полученных результатов и многие другие. С этой точки зрения разумно рассматривать правила правдоподобных рассуждений как системный элемент в задачах конструирования алгоритмов обработки данных.

Статья организована следующим образом. Раздел 2 посвящен короткому введению в правдоподобный вывод. Рассматриваются только такие правила, которые применяются в новой версии NIAGARA. Раздел 3 дает определения ХМИТ. В разделе 4 обсуждается применение правил правдоподобного рассуждения в алгоритме NIAGARA-2 для генерации ХМИТ и дается пример работы алгоритма. В конце статьи обсуждаются некоторые работы, относящиеся к рассматриваемым проблемам.

2. Правила правдоподобных рассуждений

В разделе рассматриваются правила правдоподобного вывода как «если..., то...» или логические утверждения. Эти правила подразделяются на следующие категории:

— ПРИМЕРЫ, или отношения между реально наблюдаемыми объектами и фактами;

— ПРАВИЛА ПЕРВОГО ТИПА, или логические утверждения, основанные на регулярных отношениях между объектами и (или) их свойствами;

— ПРАВИЛА ВТОРОГО ТИПА, или правила правдоподобного вывода, с помощью которых правила первого типа применяются, модифицируются и извлекаются из данных.

Описание этих правил и информация об их применении дается в [1].

2.1. Правила первого типа в алгоритме NIAGARA-2

Импликации. Правило импликации есть классическое логическое правило. Оно имеет следующую форму: $x, y, z \to w$. Левая и правая части этого выражения называются антецедентом (посылкой) и следствием (заключением) соответственно. Если все значения посылки истинны, то заключение также истинно.

Запрет, или запрещающее правило. Запрет есть импликация специального вида. Это правило может быть рассмотрено как следующее выражение: $x, y, z \to$ false (никогда). Можно представить запрет как несколько импликаций. Например, $x, y \to$ не z; $x, z \to$ не y; $y, z \to$ не x.

2.2. Правила второго типа в алгоритме NIAGARA-2

Предположим, что y есть множество значений признаков некоторого объекта, наблюдаемых одновременно. Пусть `p`, `antecedent(p)` и `consequent(p)` обозначают импликацию, ее антецедент и заключение соответственно. Тогда применения импликации и запрета имеют следующий вид.

Применение импликации. Если `antecedent(p)`$\subseteq y$, тогда y можно расширить за счет `consequent(p)`: $y \leftarrow y \cup$ `consequent(p)`. Это применение использует modus ponens: если X, то Y; X; следовательно Y.

Применение запрета. Предположим, что p есть $z \to$ не k. Если antecedent(p)$\subseteq y$, тогда k есть запрещенное значение для всех расширений y. Это применение использует modus ponendo tollens: X или Y; X; следовательно не Y; X или Y; Y; следовательно не X. X и Y называются альтернативами.

В дальнейшем используем эти правила для расширения элементов начального множества ХМИТ.

3. Анализ хороших тестов

Напомним основные определения анализа хороших тестов [1–3].

Предоположим, что R и S соответственно многозначная таблица описаний объектов [5] и множество индексов объектов. Тогда $S(k)$ и $R(k)$ называют соответственно множеством индексов и описаний k-объектов, где $k \in K$ есть класс объектов, например «положительных» (+) или «отрицательных» (–), в некотором смысле.

Пусть FM, определяемое как $R \backslash R(k)$, есть множество описаний объектов, отличающихся от k класса. Обозначим через U и T соответственно множества атрибутов и значений атрибутов («значений», для краткости). Каждое значение появляется по крайней мере в описании одного объекта (в «объекте», для краткости) из R. Обозначим через n и $dom(Atr)$ соответственно общее число объектов (индексов объектов) и область значений $Atr \in U$.

Соответствие Галуа [4] от значений атрибутов к индексам объектов задается функцией $s(\cdot)$, которая принимает t, подмножество T попарно различных значений атрибутов, и возвращает подмножество индексов объектов, в описание которых входит t. Полагаем, что для значений атрибутов используются номинальные шкалы [5].

Назовем $t \subseteq T$, $s(t) \neq \emptyset$, *диагностическим тестом* для $R(k)$, если и только если $t \not\subset d$, $\forall d \in FM$. Быть диагностическим тестом t означает, что условие $s(t) \subseteq S(k)$ и импликативное правило p «если t, то k» выполняются. Очевидно, t есть antecedent(p), где p есть правило первого типа.

Обозначим через $DT(k)$ множество всех диагностических тестов для $R(k)$. Для любой пары $t, d \in DT(k)$, одно и только одно из следующих условий выполняется: $s(t) \subset s(d)$, $s(d) \supset s(t)$ и $s(t) \sim s(d)$, где sim означает отношение несравнимости.

Тогда не пустой тест t называется *хорошим тестом* для $R(k)$, если и только если $s(t) \subseteq S(k)$ и одновременно $\forall g$, $g \in S(k) \backslash s(t)$, расширение $\{s(t) \cup g\}$ не является тестом для $R(k)$.

Множество $t \subseteq T$ называется *максимально избыточным* (*замкнутым*), если для любого имликативного правила $Y \to z$ в R имеем $(Y \subseteq t) \to (z \in t)$.

Соответствие Галуа $T \to S$ задается как $s(B) = \{i \mid i \in S, B \subseteq t_i$, где t_i есть описание объекта с индексом $i\}$. Другое соответствие Галуа $S \to T$ задается как $t(s) = \{$пересечение всех $t_i \mid t_i \subseteq T, i \in s\}$.

Существует два оператора замыкания [5] $generalization\_of(t) = t'' =$

$= t(s(t))$ и $generalization\_of(s) = s'' = s(t(s))$. Множество t *замкнуто*, если $t(s(t)) = t$, и s *замкнуто*, если $s(t(s)) = s$.

Взаимосвязь между анализом хороших тестов и анализом формальных понятий была освещена в [3]. Кроме того, в статье показано, что ХМИТ является популярным классификатором. Приведены взаимосвязи с другими символьными классификаторами, например JSM-гипотезами [6, 7].

4. Алгоритм NIAGARA-2 для генерации хороших тестов с использованием правил правдоподобного вывода

4.1. Идея алгоритма

NIAGARA-2 является пакетным алгоритмом для вывода всех ХМИТ для положительных или отрицательных примеров объектов. Это новый вариант NIAGARA алгоритма, описанного в [1]. Для этой цели порождается последовательность $S_0 \subseteq \ldots \subseteq S_q \subseteq S_{q+1} \subseteq \ldots \subseteq S_{q+m}$, где S_q есть множество всех подмножеств $S(+)$ мощностью q. Операция генерализации применяется к каждому элементу $(s_q, t(s_q))$ начиная с двух начальных множеств $R(+) = \{t_1, \ldots, t_i, \ldots, t_{nt}\}$ и $S(+) = \{1, \ldots, i, \ldots, nt\}$, где nt есть число положительных объектов.

Псевдокод обсуждаемых далее процедур размещен в следующем подразделе.

Процедура DEBUT выполняет расширения элементов множества $S(+) = \{1, \ldots, i, j, \ldots, nt\}$ и конструирует множество $\{s_{12}, s_{13}, \ldots, s_{ij}, \ldots\}$, где $s_{ij} = \{i, j\}$, $1 < i < j < nt$.

Каждый элемент $s_{ij} = i, j$ такой, что $(s_{ij}, t(s_{ij}))$ не является тестом для $R(+)$, хранится в множестве Q запрещенных пар объектов. А каждый набор $s_{ij} = i, j$ такой, что $(s_{ij}, t(s_{ij}))$ есть тест для $R(+)$, обобщается, а результат $s = generalization\_of(s_{ij})$ помещается в $S(test)$.

Когда DEBUT заканчивает работу, проверяется, не соответствует ли набор s из $S(test)$ набору ХМИТ для положительных объектов. Для этой цели служит правило: если некоторый объект j, где $j = 1, \ldots, nt$, принадлежит одному и только одному s из $S(test)$, то s не может быть расширен, следовательно, s является ХМИТ, который переносится из $S(test)$ в STGOOD.

$S(test)$ частично упорядочено и содержит все $s = \{i_1, \ldots, i_q\}$, $q = 1, \ldots, nt$, удовлетворяющие условию, что $(s, t(s))$ является максимально избыточным тестом для $R(+)$, но не хорошим. STGOOD есть частично упорядоченное множество, содержащее все $s = \{i_1, \ldots, i_q\}$, $q = 1, \ldots, nt$ и удовлетворяющее условию, что $(s, t(s))$ есть ХМИТ для положительных объектов. Для каждого s в $S(test)$, $ext(s)$ есть множество всех возможных его расширений, которые соответствуют тестам для положительных примеров. Алгоритм реализует направленный выбор объектов для расширения s с использованием правила генерализации.

Процедура $SELECT(s)$ возвращает множество $select(s)$ допустимых объектов для расширения s.

$Context(s)$ определяется как множество индексов объектов, которые в текущий момент могут быть использованы для расширения $s \subseteq S(+)$: $context(s) = \{\{\cup s*\} \mid Prefix(s*) = Prefix(s), s \prec s*\}$, где $Prefix(s)$ есть первый индекс s и $\prec$ есть символ лексикографического порядка.

Введение функции $context(s)$ позволяет использовать декомпозицию алгоритма на независимые подпроцессы. Множество $V(s)$ определяется как множество индексов объектов, которые должны быть удалены из $context(s)$, чтобы избежать повторной генерации тестов. $CAND(s) = context(s) \setminus V(s)$, где $V(s) = \{\cup s*, s \subseteq s*, s* \in \{\{S(test) \setminus s\} \cup STGOOD\}\}$.

Множество $V(s)$ есть объединение всех множеств в $\{\{S(test) \setminus s\} \cup \cup STGOOD\}$, содержащих s, следовательно, s в пересечении этих множеств. Если хотим, чтобы расширение s не было включено ни в один из элементов множества $\{\{S(test) \setminus s\} \cup STGOOD\}$, надо использовать для расширения s индексы объектов, не появляющиеся одновременно с s во множестве $V(s)$.

$Select(s)$ определяется как множество индексов объектов, разрешенных для расширения s (для этой цели используется множество запрещенных пар индексов Q): $select(s) = \{i, i \in CAND(s) : (\forall j)(j \in s), i, j \notin \{STGOOD \vee Q\}\}$.

Отметим следующие правила правдоподобных рассуждений, применяющиеся в NIAGARA-2:

— правила запрета;

— правило расширения множеств с элиминированием поисковых пространств, не содержащих решений;

— импликативные правила, основанные на известных свойствах хороших тестов как формальных понятий.

4.2. Псевдокод алгоритма NIAGARA-2

В этом разделе представлен псевдокод основных процедур алгоритма NIAGARA-2. При формировании STGOOD множество s сохраняется в STGOOD, если и только если оно не вложено ни в одно множество из STGOOD. В псевдокоде используется обозначение $gnrOf$ вместо $generalization\_of$ из соображений краткости. Представим основную процедуру алгоритма.

Главный алгоритм NIAGARA-2

Вход: $R(+), R(-), nt, S(+) = \{1, \ldots, nt\}$.
Выход: TGOOD
1. DEBUT;
2. **до тех пор, пока** *S(test)*$\neq \emptyset$ **выполнять**
3. SELECT(s);
4. EXTENSION(s);
5. ANALYSIS_OF_EXTENSION(s);
6. **до тех пор, пока** STGOOD **выполнять**
7. TGOOD $\leftarrow \{t(s) \mid s \in$ STGOOD$\}$;
8. STGOOD$\setminus s$;

По сравнению с предыдущей версией алгоритма [1], осуществлены следующие изменения:

— в процедуре SELECT новая функция «determining context(s)» замещает предыдущую версию этой функции; $context(s)$ определено выше;

— в процедуре Analysis of Extension(s) выделение всех множеств, содержащих нерасширяемое множество s, и передачи их в STGOOD (строки 7–12).

Основные улучшения представлены в следствии 3 в следующем подразделе. Положительное влияние улучшений оценивается в подразделе с обсуждаемым примером. Рассмотрим подробнее основные процедуры NIAGARA-2.

Процедура `DEBUT()`

Вход: $R(+), R(-), nt, S(+) = \{1, \ldots, nt\}$.
Выход: $S(test)$, Q, STGOOD.
1. STGOOD, $Q, S(test) \leftarrow \emptyset$;
2. **цикл** $i \in \{1, \ldots, nt\}$ **выполнять**
3. $\quad sum(i) \leftarrow 0$;
4. **цикл** $i \in \{S[1], \ldots, S[nt]\}$ **выполнять**
5. $\quad$ **цикл** $j \in \{S[i+1], \ldots, S[nt]\}$ **выполнять**
6. $\quad\quad$ **если** $to\_be\_test(t\{i,j\}) = false$ **тогда**
7. $\quad\quad\quad Q \leftarrow Q \cup \{i,j\}$
8. $\quad\quad$ **иначе**
9. $\quad\quad\quad s'' \leftarrow gnrOf(\{i,j\})$;
10. $\quad\quad\quad$ **для каждого** i *в* s'' **выполнять**
11. $\quad\quad\quad\quad sum(i) \leftarrow sum(i) + 1$;
12. $\quad\quad\quad$ вставить s'' в $S(test)$ в лексикогр. порядке;
13. **для каждого** i *в* $\overline{1, nt}$ **выполнять**
14. $\quad$ **если** $sum(i) = 1$ **тогда**
15. $\quad\quad$ найти $s : i \in s, s \in S(test)$;
16. $\quad\quad$ вставить s в STGOOD в лексикогр. порядке;
17. $\quad\quad$ удалить s из $S(test)$;
18. $nts \leftarrow \{\cup s \mid s \in S(test)\}$;

Algorithm `SELECT(`s`)`

Вход: $s, nts, Q, S(test)$, STGOOD.
Выход: множество объектов $select(s)$ для возможного расширения s.
1. Сформировать $context(s)$;
2. **если** $context(s) = \emptyset$ **тогда**
3. $\quad select(s) \leftarrow \emptyset$
4. **иначе**
5. $\quad V(s) = \{\cup s', s \subseteq s', s' \in \{\{S(test) \setminus s\} \cup \text{STGOOD}\}\}$;
6. $\quad$ **если** $V(s) = \emptyset$ **тогда**
7. $\quad\quad CAND(s) \leftarrow context(s)$;
8. $\quad$ **иначе**
9. $\quad\quad CAND(s) \leftarrow context(s) \setminus V(s)$;
10. $\quad\quad$ **если** $CAND(s) = \emptyset$ **тогда**
11. $\quad\quad\quad select(s) \leftarrow \emptyset$;
12. $\quad\quad$ **иначе**
13. $\quad\quad\quad select(s) = \{i \in CAND(s) \mid (\forall j)(j \in s), \{i,j\} \notin \{\text{STGOOD} \vee Q\}\}$

Процедура EXTENSION(s)

Вход: $s, select(s), S(test)$, STGOOD.

Выход: $ext(s)$ — множество всех расширений s' из s таких, что $t(s')$ есть тест.

1. $ext(s) = \emptyset$;
2. **до тех пор, пока** $select(s) \neq \emptyset$ **выполнять**
3. **для каждого** j *из* $select(s)$ **выполнять**
4. $s_{new} \leftarrow s \cup j$
5. **если** $to\_be\_test(t(s_{new}))$=*false* **тогда**
6. удалить s_{new};
7. **иначе**
8. $s_{new} \leftarrow gnrOf(s_{new})$;
9. вставить s_{new} в $ext(s)$ в лексикогр. порядке;

Процедура ANALYSIS_OF_EXTENSION()

Вход: $ext(s), S(test)$, STGOOD.

Выход: модифицированные $S(test)$ и STGOOD.

1. **если** $ext(s) = \emptyset$ **тогда**
2. **если** $s \subset s*$, $s* \in S(test)$ **тогда**
3. перенести $s*$ из $S(test)$ в STGOOD в лексикогр. порядке;
4. удалить s;
5. **иначе**
6. перенести s из $S(test)$ в STGOOD в лексикогр. порядке;
7. **если** $||ext(s)|| = 1$ **тогда**
8. $s = s_{new}, s_{new} \in ext(s)$;
9. **если** $s \not\subset s*$, $s* \in S(test)$ **тогда**
10. перенести s из $S(test)$ в STGOOD в лексикогр. порядке;
11. **иначе**
12. перенести $s*$ из $S(test)$ в STGOOD в лексикогр. порядке;
13. удалить s;
14. **иначе**
15. **для каждого** s_{new} *в* $ext(s)$ **выполнять**
16. вставить s_{new} в $S(test)$ в лексикогр. порядке;
17. delete s;

4.3. Главные характеристики алгоритмов NIAGARA и NIAGARA-2

Предлагаемый алгоритм основан на генерации замкнутых множеств. Пусть X незамкнутое множество и $c(X) = s(t(X))$ его замыкание. Рассмотрим две возможности: $c(X) = X$ и $X \subset c(X)$. В первом случае X замкнуто и должно быть расширено. Во втором случае X не замкнуто.

Утверждение 1. Если $X \subset c(X)$, *тогда* $t(c(X)) \subset t(X)$, *но* $t(X) \subset \subset t(c(X)) \rightarrow t(X) = t(c(X))$.

Следствие 1. Если $t(X) = t(c(X))$, *тогда* $c(X)$ *может заменить* $X \in \in S(test)$ *и* X *может быть удалено из* $S(test)$ *без потери какого-либо решения.*

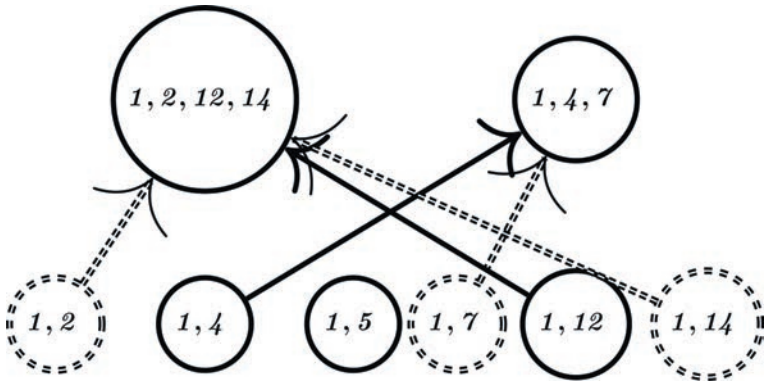


Иллюстрация использования утверждения 1 и следствия 1.

Утверждение 2. Если индекс объекта включен в одно и только одно множество в S(test), тогда это множество не может быть расширено.

Следствие 2. Если индекс объекта включен в одно и только одно множество в S(test), тогда это множество является хорошим тестом.

Следующее утверждение лежит в основании нового правила правдоподобных рассуждений, что увеличивает эффективность алгоритма NIAGARA-2 по сравнению с предыдущей версией.

Утверждение 3. Если замкнутое множество X индексов объектов содержит нерасширяемое подмножество Y, тогда X также нерасширяемое множество.

Следствие 3. Если множество X индексов объектов замкнуто, является тестом и содержит некоторое нерасширяемое подмножество, тогда X является хорошим тестом.

Назовем следствия 1, 2 и 3 правилами рассуждения 1, 2 и 3 соответственно. Хотя первое и второе утверждения (и следствия) впервые сформулированы, соответствующие правила уже использовались в NIAGARA.

Третье правило делает возможным избежать расширения множеств, содержащих нерасширяемые подмножества. Это правило с логической точки зрения является запрещающим правилом. С точки зрения решетки паттернов это правило значит, что если паттерн нерасширяемый, то все элементы его принципиального фильтра в решетке паттернов также нерасширяемы.

Рисунок дает пример работы следствия 1. Круги и сплошные стрелки, выполненные полужирным, означают замкнутые множества и связи между ними (см. диаграмму в [8]). Круги и стрелки, выполненные пунктиром, означают незамкнутые множества и связи с их замыканиями. Эти множества должны быть удалены и заменены на их замыкания.

Можно удалить из рассмотрения множества $\{1,2\}$, $\{1,7\}$, $\{1,14\}$ потому, что $t(\{1,2\}) = t(\{1,2,12,14\}$, $t(\{1,7\}) = t(\{1,4,7\})$, $t(\{1,14\}) = t(\{1,2,12,14\})$ и, тем самым, избежать расширения незамкнутых множеств, потому что по-

Таблица 1. Пример булевого представления множеств

	1	2	4	5	7	12	14	Замкнутый?	Удалить?
1	1	1				1	1	1	
2	1	1						0	1
3	1		1		1			1	
4	1			1				1	
6	1				1			0	1
7	1					1		1	
8	1						1	0	1

лучим тот же результат при расширении их замкнутых супер множеств. Таблица 1 иллюстрирует преимущество использования булевого представления множеств индексов объектов. Данные в табл. 1 согласованы с рисунком.

5. Оценка работы алгоритма

5.1. Иллюстративный пример

Данные для работы алгоритма приведены в [1]. В этом разделе приводятся результаты применения алгоритма NIAGARA-2 на множестве следующих на-

Таблица 2. Расширения элементов S(test)

S	$Context(s)$	$CAND(s)$	$Select(s)$	$Ext(s)$	Delete $s \in S(test)$	STGOOD
1, 4	5, 7, 12	5, 12	12	$\emptyset$	1	1, 4, 7
1, 4, 7						
1, 5	12	$\emptyset$	$\emptyset$	$\emptyset$	1	1, 5, 12
1, 5, 12						
1, 12	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	1	
2, 3, 4	7, 8.10	7, 8, 10	7, 8	2, 3, 4, 7	1	2, 3, 4, 7
2, 7	8, 10	8	8	2, 7, 8	1	2, 7, 8
2, 8	10	$\emptyset$	$\emptyset$	$\emptyset$	1	
2, 10	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	1	2, 10
3, 7	8, 10, 11, 12	8, 10, 11	8, 11	$\emptyset$	1	3, 7, 12
3, 7, 12						
3, 8	10, 11	10, 11	10, 11	$\emptyset$	1	3, 8
3, 10	11	11	11	$\emptyset$	1	3, 10
3, 11	$\emptyset$	$\emptyset$		$\emptyset$	1	3, 11
4, 6, 8, 11	12	$\emptyset$	$\emptyset$	$\emptyset$	1	4, 6, 8, 11
4, 6, 11	12	$\emptyset$	$\emptyset$	$\emptyset$	1	
4, 7	8, 11, 12	8, 11, 12	8, 11, 12	4, 7, 12	1	4, 7, 12
4, 8	11, 12	11, 12	11	4, 8, 11	1	
4, 11	12	12	$\emptyset$	$\emptyset$	1	
4, 12	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	1	
7, 8	11, 12	11, 12	11	7, 8, 11	1	7, 8, 11
7, 11	12	12	$\emptyset$	$\emptyset$	1	
7, 12	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	1	
8, 10	11	11	$\emptyset$	$\emptyset$	1	8, 10
8, 11	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	1	

чальных данных. *Вход*: $S = \{1, 2, \ldots, 14\}$; $T = \{A_1, \ldots, A_{26}\}$; STGOOD $= \emptyset$; $S(test) = \emptyset$; $Q = \emptyset$. *Выход*: после применения процедуры DEBUT имеем те же множества $S(test)$, Q и STGOOD, как в таблицах 21, 22 и 23 из [1] соответственно. Таблица 2 представляет результат расширения элементов множества $S(test)$. Различие между этой таблицей и соответствующей таблицей в [1] обсуждается в следующем разделе. Результат работы NIAGARA и NIAGARA-2 одинаковый (табл. 26 в [1]).

5.2. Сравнение работы алгоритмов NIAGARA и NIAGARA-2

Иллюстративный пример был обработан с помощью NIAGARA и NIAGARA-2, чтобы найти новые ХМИТ. Преимущество NIAGARA-2 определяется двумя основными оптимизациями: функцией контекста и новым правилом 3 на основе утверждения 3. Для сравнения алгоритмов вычисляются следующие меры:

1) общее число объектов, вовлеченных в контексты (Conts);

2) общее число объектов, включенных в множества $CAND(s)$ для всех расширяемых наборов (SCand);

3) общее число объектов, включенных в множества $Select(s)$ для всех расширяемых наборов (SSelect);

4) число наборов объектов (индексов объектов), которые необходимо было расширить (SExt).

Результат сравнения представлен в табл. 3.

Таблица 3. Сравнение NIAGARA и NIAGARA-2

Алгоритм	Conts	SCand	SSelect	SExt
NIAGARA	59	55	16	25
NIAGARA-2	29	22	14	20

Меры Const и SCand уменьшились почти на 50% и 40% соответственно. Мера SSelect для новой версии алгоритма изменилась незначительно. Мера SExt уменьшилась из-за ХМИТ, которые переводились из $S(test)$ в STGOOD по новому правилу (соответствующие расширения не производились). Эти ХМИТ с номерами 8, 10, 12 в табл. 26 из [1].

6. Работы с близкой тематикой

В первую очередь авторов интересуют методы, с помощью которых решается основная проблема алгоритмов: как избежать повторяющейся генерации одного и того же объекта или как проверить единственность сгенерированного объекта. Применительно к этой проблеме некоторые методы обобщены в [9, 10]. Некоторые другие работы приведены в обзоре [11].

Важность использования контекстов аргументированно показана в [12]. Также можно отметить, что идея использования деревьев решений (класси-

фикационной структуры) для организационного принципа извлечения онтологии из текстов предложена в [13].

Правила рассуждений широко используются во всех алгоритмах, имеющих дело с генерацией частых замкнутых наборов элементов. Например, правило, аналогичное правилу 1 в NIAGARA-2, поддерживается леммой 3 в [14]. Тем не менее правила рассуждений 2 и 3 являются оригинальными и используются только для генерации GMRT.

Во многих алгоритмах применяется индуктивное правило, основанное на поуровневом расширении множеств атрибутов, наборов значений атрибутов или объектов (индексов объектов) таким образом, что множества размера q строятся из множеств размера $q-1$ предыдущего уровня. Каждое множество может быть построено тогда и только тогда, когда на предыдущем уровне есть все его собственные подмножества. В алгоритмах проверяются различные свойства множеств. Если множество не обладает требуемым свойством, то его исключают из рассмотрения. Это сильно уменьшает количество множеств всех последующих уровней, которые нужно построить.

Поуровневая генерация множеств элементов используется для извлечения ассоциативных правил из данных. Алгоритм извлечения ассоциативных правил AIS впервые был введен в [15]. Новые алгоритмы Apriori, AprioriTid и AprioriHybrid являются улучшенными версиями первого алгоритма.

Теоретико-решеточное обоснование задачи извлечения ассоциативных правил из данных на основе анализа формальных понятий было сделано в [16]. Было показано, что множество часто встречающихся понятий однозначно определяет все часто встречающиеся паттерны. Решетка частых понятий может также быть использована, чтобы получить правило генерации множеств, из которых можно вывести все ассоциативные правила.

Существуют две стратегии генерации понятий с помощью пакетных алгоритмов: нисходящая (или сверху вниз) и восходящая. Однако важно, состоит ли ведущий процесс непосредственно в генерации всех подмножеств объектов (объемов понятий) или всех подмножеств атрибутов (содержаний понятий). В [17] используется стратегия «сверху вниз» для вывода понятий в «breadth first» (сначала в ширину) стиле. Ведущим процессом этого алгоритма является генерация подмножеств объектов заданного контекста с уменьшением их мощности. В алгоритме NextClosure [18] ведущим процессом является построение лексически упорядоченных подмножеств атрибутов.

Можно заключить, что каждый алгоритм, генерирующий паттерны (наборы элементов) и подразумевающий формирование логических правил (импликация, ассоциативное правило, функциональные зависимости, формальные понятия и многие другие), использует так или иначе правила правдоподобного рассуждения, и этот факт дает право утверждать, что искомые алгоритмы можно рассматривать как модели мыслительных процессов человека.

7. Заключение

В статье проанализировано применение правдоподобных рассуждений в NIAGARA и NIAGARA-2 для генерации ХМИТ. Более того, некоторые новые процедуры, основанные на правдоподобных рассуждениях, помогают сделать алгоритм NIAGARA-2 более эффективным. Производительность алгоритма по времени улучшена за счет следующих новых процедур: расширение текущих наборов целевых объектов (с использованием импликации, основанной на свойствах замкнутых хороших тестов, запретов, правил расширения) и сокращение пространства поиска. Доказана их корректность. Будущие работы включают в себя проведение экспериментов, показывающих эффективность предложенных оптимизаций.

СПИСОК ЛИТЕРАТУРЫ

1. *Naidenova X.* An incremental learning algorithm for inferring logical rules from examples in the framework of the common reasoning process / Data Mining and Knowledge Discovery Approaches Based on Rule Induction Techniques, Massive Comp., Springer. 2006. V. 6. P. 89–147.
2. *Найденова К.А., Полежаева Ю.Г.* Алгоритм нахождения наилучших диагностических тестов // Сб. научн. тр. 4 Всесоюзн. конф. “Применение методов математической логики”. Институт кибернетики АН ССР. 1986. С. 87–92.
3. *Naidenova X., Buzmakov A., Parkhomenko V., Schukin A.* Notes on relation between symbolic classifiers / CEUR-WS. 2017. V. 1921. P. 88–103.
4. *Ore O.* Galois connections // Trans. Amer. Math. Soc. 1944. V. 55. P. 494–513.
5. *Ganter B., Wille R.* Formal concept analysis: mathematical foundations. Berlin/ Heidelberg: Springer, 1999.
6. *Финн В.* О машинно-ориентированной формализации правдоподобных рассуждений в стиле Ф. Бекона-Д. С. Милля // Семиотика и информатика. 1983. Т. 20. С. 35–101.
7. *Kuznetsov S.* Mathematical aspects of concept analysis // J. Math. Sci. 1996. V. 80. No. 2. P. 1654–1698.
8. *Ganter B., Kuznetsov S.* Hypotheses and version spaces // LNCS. 2003. V. 2746. P. 83–95.
9. *Carpineto C., Romano G.* Concept Data Analysis: Theory and Applications. Chichester, UK: John Wiley & Sons, Ltd, 2004.
10. *Kuznetsov S.O., Obiedkov S.A.* Comparing performance of algorithms for generating concept lattices // J. Experiment. Theoret. Artific. Intelligence. 2002. V. 14. No. 2-3. P. 189–216.
11. *Poelmans J., Ignatov D.I., Kuznetsov S.O., Dedene G.* Formal concept analysis in knowledge processing: A survey on applications // Expert Syst. Appl. 2013. V. 40. No. 16. P. 6538–6560.
12. *Galitsky B., Ilvovsky D.I., Goncharova E.* Organizing contexts as a lattice of decision trees for machine reading comprehension // Proc. of the 10th Int. Worksh. “What can FCA do for Artificial Intelligence?”. 2022. P. 75–87.

13. *Goncharova E., Ilvovsky D.I., Galitsky B.* Concept-based chatbot for interactive query refinement in product search // Proc. of the 9th Int. Worksh. “What can FCA do for Artificial Intelligence?”. 2021. P. 51–58.
14. *Wang J., Han J., Pei J.* Closet+: Searching for the best strategies for mining frequent closed itemsets // Proc. ACM SIGMOD Int. Conf. Knowl. Discov. Data Mining. 2003. P. 236–245.
15. *Agrawal R., Imieliński T., Swami A.* Mining association rules between sets of items in large databases // Proc. ACM SIGMOD Int. Conf. Management Data. 1993. P. 207–216.
16. *Zaki M.J., Ogihara M.* Theoretical foundations of association rules / 3rd ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery ACM. 1998. P. 71–78.
17. *Bordat J.P.* Calcul pratique du treillis de galois d’une correspondence // Math. Sci. Humaines. 1986. V. 96. P. 31–47.
18. *Ganter B.* Two basic algorithms in concept analysis / Formal Concept Analysis. Berlin/Heidelberg: Springer. 2010. P. 312–340.

Статья представлена к публикации членом редколлегии А.А. Галяевым.

Поступила в редакцию 08.07.2023

После доработки 13.10.2023

Принята к публикации 20.01.2024

© 2024 г. Д.А. ЛЮТКИН (dalyutkin@gmail.com),
Д.В. ПОЗДНЯКОВ (dvpozdnyakov@hse.ru)
(Национальный исследовательский университет
«Высшая школа экономики», Москва),
А.А. СОЛОВЬЕВ (andrey.a.soloviev@gmail.com),
Д.В. ЖУКОВ (dimas.zhukov@gmail.com)
(ООО «Бэбиблог», Москва),
М.Ш.И. МАЛИК, д-р философии (Ph. D.) (mumalik@hse.ru),
Д.И. ИГНАТОВ, канд. техн. наук (dignatov@hse.ru)
(Национальный исследовательский университет
«Высшая школа экономики», Москва)

ПРИМЕНЕНИЕ ТРАНСФОРМЕРОВ ДЛЯ ОПРЕДЕЛЕНИЯ ПРОФИЛЬНОГО ВРАЧА НА ОСНОВЕ ЗАПРОСОВ ПОЛЬЗОВАТЕЛЕЙ[1]

Представлен новый подход, использующий модель RuBERT для классификации пользовательских запросов в области медицинских консультаций с учетом специализации эксперта. В ходе исследования был собран обширный набор данных, который использовался для дообучения модели RuBERT. Метрика качества полученной модели F1-score составила более 91,8% как при использовании блоковой кросс-валидации, так и при разделении набора данных на обучающую и тестовую выборки. Подход демонстрирует высокую обобщающую способность для различных медицинских подобластей, таких как кардиология, неврология и дерматология. Предложенный подход позволяет сократить время на определение наиболее подходящего специалиста и тем самым повышает качество консультации и медицинской помощи.

Ключевые слова: трансформер, медицинский текст, многоклассовая классификация.

DOI: 10.31857/S0005231024030076, **EDN:** TQAELK

1. Введение

Спрос на квалифицированную медицинскую помощь растет, особенно вместе с ростом доступности телемедицины в цифровую эпоху. Поскольку онлайн-платформы выступают в качестве источников медицинской информации [1], все большую важность приобретает обеспечение точности и достоверности консультаций. Одной из таких платформ является Babyblog.ru [2],

[1] Исследование выполнено с использованием суперкомпьютерного комплекса НИУ ВШЭ. Разработано при финансовой поддержке Фонда содействия развитию малых форм предприятий в научно-технической сфере fasie.ru. Два первых соавтора внесли равнозначный вклад в исследование и подготовку статьи к публикации. Исследование второго автора осуществлено в рамках Программы фундаментальных исследований НИУ ВШЭ. Онлайн-демо доступно по ссылке: https://www.babyblog.ru/classifier

где пользователи могут задавать вопросы и получать ответы, в том числе и от медицинских специалистов.

Однако большое количество пользовательского контента[2] может ставить под сомнение научную достоверность распространяемой информации [3]. Следовательно, существует необходимость в создании механизмов, обеспечивающих проверку и обогащение пользовательского контента за счет участия экспертов. Такой совместный подход позволяет специалистам проверять сообщения, комментарии и обсуждения пользователей и предоставлять экспертные мнения, корректируя ответы непрофессионалов и обеспечивая предоставление точной и достоверной медицинской информации.

С учетом значительного объема пользовательского контента на различных платформах, охватывающего широкий спектр тем, включая медицинские, псевдомедицинские и немедицинские области, задача идентификации контента, требующего медицинской или профессиональной проверки, становится все более актуальной. Кроме того, классификация на основе тематики и специализации позволит направить пользователя к соответствующему специалисту для консультации.

Для решения этой задачи был разработан автоматический классификатор медицинских текстов, который определяет вероятность принадлежности текста к материалам по той или иной медицинской специальности. Реализация включает в себя интеграцию классификатора на платформе, в которой он идентифицирует медицинский контент и присваивает ему соответствующие медицинские специальности. Впоследствии специалисты каждой из соответствующих специальностей получают уведомления для проверки контента и предоставления своего ответа.

Система классификации позволяет оптимизировать процесс проверки за счет сокращения количества нерелевантной информации, получаемой медицинским специалистом, уменьшения нагрузки, связанной с проверкой контента, и ускорения получения профессиональных ответов пользователем.

Цель исследования — разработать и изучить эффективность системы на основе трансформеров для классификации пользовательского медицинского контента в контексте сайта Babyblog.ru. Используя различные методы обработки естественного языка (Natural Language Processing, NLP), данное исследование улучшает доступность медицинской информации для пользователей с сохранением при этом ее научной строгости и надежности.

2. Обзор релевантной литературы

В сфере классификации медицинских текстов стоит отметить статью [4], в которой предлагается использовать гибридный подход (Hybrid Model, HyM), сочетающий сразу несколько технологий глубокого обучения: LSTM, TEXT-CNN, BERT, TF-IDF, а также механизм внимания (attention mechanism).

[2] Различное информационно-значимое содержимое цифровых носителей, которое создается пользователями.

Предложенный подход позволяет определить, к какому специалисту направить пациента на основе описания симптомов.

В [5] описываются важные аспекты обработки текста, в частности обработка больших текстовых данных, объем которых растет ежедневно. Авторы отмечают необходимость автоматизации обработки текста и приходят к выводу о том, что современные подходы, такие как трансформеры и механизм внимания, могут быть крайне эффективными в этой задаче.

В своем исследовании авторы представили модель двунаправленного трансформера (bidirectional transformer, BiTransformer), построенную на основе двух блоков двунаправленного позиционного кодирования. Такой подход позволяет поместить в один контекст информацию, находящуюся как перед, так и после каждого токена, что улучшает способность модели находить связи в тексте и повышает возможности модели в обработке сложных текстовых данных.

Чтобы оценить эффективность механизмов внимания в процессе классификации, авторы сравнивают четыре модели: с использованием долгой кратковременной памяти (Long Short Term Memory, LSTM), с механизмом внимания, трансформер и предложенный BiTransformer. Эксперименты проводятся на большом наборе текстов на турецком языке, включающем 30 классов.

В ходе экспериментов было показано, что модели классификации, использующие трансформер и механизм внимания, превосходят классические методы глубокого обучения. Это демонстрирует способность трансформеров выявлять полезные закономерности и учитывать контекст в текстовых данных.

Также авторы изучили влияние использования предобученных векторных представлений (“эмбеддингов”, от англ. embedding) на качество модели. Эмбеддинги хранят семантические представления слов и предобучаются на большом корпусе текстов. Они являются популярным способом улучшить качество модели в задачах обработки текстов на естественном языке (ОТЕЯ). Авторы показывают, как предобученные эмбеддинги могут еще больше увеличить эффективность и точность моделей классификации текстов.

В ходе экспериментов было показано, что среди всех рассмотренных подходов к классификации текстов наилучшие результаты показал предложенный авторами BiTransformer.

Работа [5] дает представление о потенциале механизма внимания и трансформеров в обработке текстов. Появление BiTransformer и его показатели в задаче классификации текстов открывают новые возможности для будущих исследований и применения моделей на основе трансформеров в задачах ОТЕЯ. Результаты исследования имеют важное значение для автоматизации обработки текстовых данных, анализа тональности текста (sentiment analysis), информационного поиска и других сфер, связанных с использованием текстов. Поскольку спрос на эффективные и точные методы обработки текстов продолжает расти, данное исследование вносит значительный вклад в развитие этой области и служит ценным пособием для исследователей и практиков в области обработки естественного языка.

3. Сбор данных: составление исчерпывающего набора данных для классификации медицинских текстов

В этом разделе описывается процесс сбора данных, включающий разработку парсера и применение нормализаторов для дальнейшего использования в эксперименте.

3.1. Парсинг данных

Для получения данных были рассмотрены русскоязычные сайты, которые предоставляют доступ к вопросам пользователей и ответам экспертов на эти вопросы. При выборе источников учитывались следующие критерии:

1) открытость данных по вопросам медицинской тематики,

2) наличие аннотации о медицинской специализации вопроса,

3) наличие верифицированного ответа от эксперта, обладающего соответствующей специализацией.

Таким образом были выбраны следующие источники: sprosivracha.com [6], doctu.ru [7], 03online.com [8] и health.mail.ru [9]. Был разработан парсер, позволяющий параллельно и асинхронно собирать информацию из открытых источников. В ходе сбора с помощью парсера каждый вопрос сохраняется в виде HTML (от англ. HyperText Markup Language — «язык гипертекстовой разметки») документа для дальнейшей обработки. В табл. 1 представлено количество данных, полученных из каждого источника данных.

Таблица 1. Сравнение платформ с медицинскими вопросами

Сайт	Количество вопросов	Процент от общего числа
sprosivracha.com	550 000	23,2
doctu.ru	83 000	3,5
03online.com	1 148 000	48,4
health.mail.ru	590 000	24,9

На следующем шаге алгоритм асинхронно обрабатывает каждый элемент полученных данных и извлекает части, содержащие текст вопроса к врачу и специальность врача. Полученные данные (текст вопроса и специальность врача) заносятся в таблицу вместе с URL (от англ. Uniform Resource Locator — «единообразный указатель местонахождения ресурса») исходного документа, который используется как идентификатор источника. После завершения процесса все данные экспортируются в CSV (от англ. Comma-Separated Values — значения, разделенные запятыми) файл для дальнейшего использования.

3.2. Аугментация данных

Анализ полученных данных показал, что распределение вопросов по специальностям имеет вид, похожий на распределение Парето. Это можно объяснить тем, что некоторые медицинские специальности гораздо более востребованы, чем другие, что приводит к дисбалансу классов [10].

Для минимизации дисбаланса и улучшения способности модели к обобщению были применены некоторые методы аугментации данных, предоставляе-

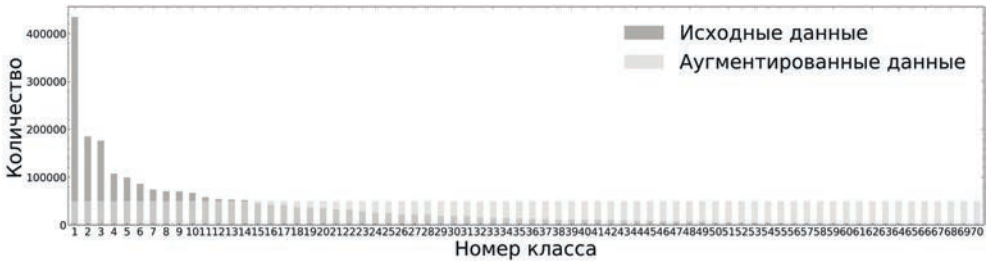


Рис. 1. Распределение классов после аугментации (первые 70 классов).

мые библиотекой nlpaug [11]. С ее помощью были созданы дополнительные синтетические записи путем перестановки слов в предложениях с сохранением прежнего контекста и смысла предложения, а также перестановки самих предложений.

Благодаря процессу аугментации были созданы дополнительные данные для непопулярных классов, что уменьшило дисбаланс и привело вид распределения к более равномерному по всем медицинским специальностям (см. рис. 1). Кроме того, это позволило улучшить способность модели работать с новыми данными в процессе тестирования.

В результате был получен набор данных с близким к равномерному распределением классов, размер которого составил 5 миллионов вопросов, в соотношении приблизительно 50 000 вопросов на класс для 97 классов.

Несмотря на то, что уже существуют аналогичные наборы данных, полученный набор данных превосходит их по количеству учитываемых заболеваний и медицинских специальностей. Кроме того, большинство существующих наборов данных составлены с использованием профессиональной лексики, которая отличается от описания состояния здоровья обычным человеком. Полученный набор данных устраняет это упущение, что будет способствовать более целостному пониманию опыта людей в области здоровья.

4. Предлагаемый метод

В этом разделе описывается предлагаемый метод, исследуются различные трансформеры и подходы к их обучению. Блок-схема метода представлена на рис. 2 (Start – Начало; Define tokenizer – Определить токенизатор; Define model for Sequence Classification – Определить модель для классификации последовательностей; Data – Данные; Separation into training and validation samples – Разделение на обучающие и проверочные выборки; length train and valid – длина обучающих и проверочных; Creating dataset – Создание набора данных; batch size – размер пакета; Train_dataloader – Загрузчик данных для обучения; Valid_dataloader – Загрузчик данных для проверки; Calculate loss and metrics – Рассчитать потери и метрики; Sending metrics to wandb – Отправка метрик в wandb; Metrics are better than in the previous iteration? – Метрики лучше, чем в предыдущей итерации?; Saved optimizer parameters, weights on file of pth.tar – Сохранены параметры оптимизатора,

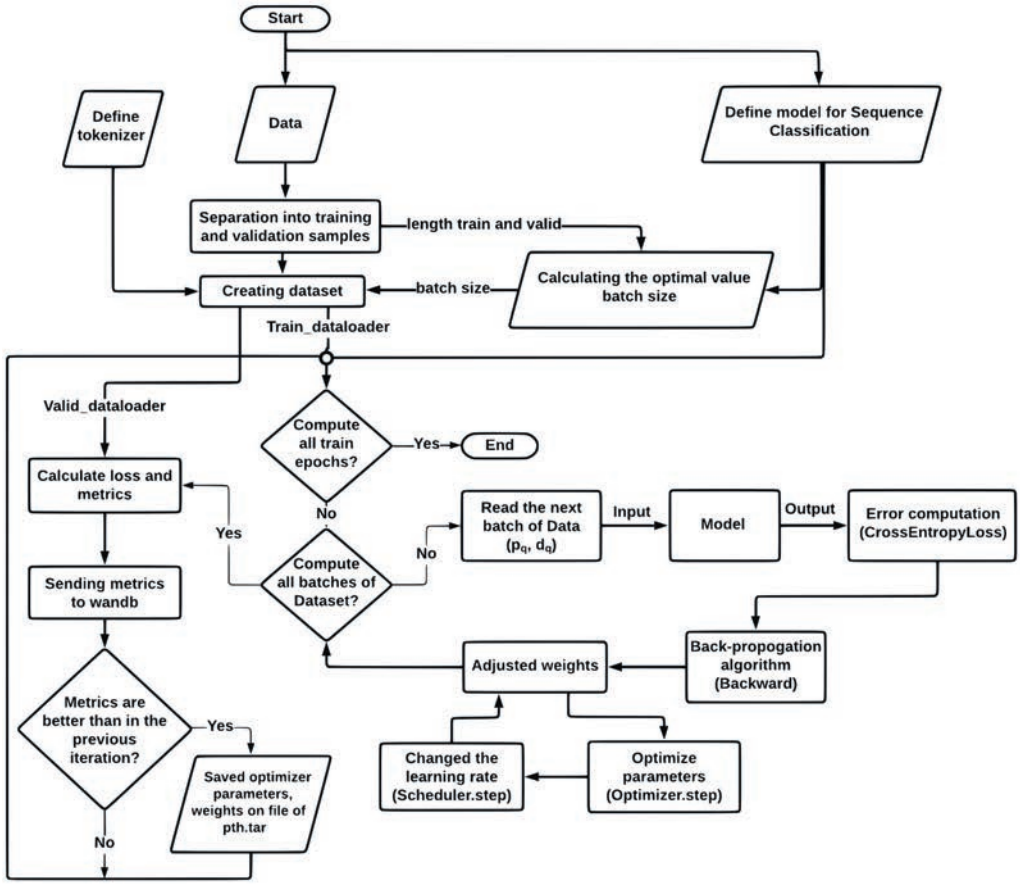


Рис. 2. Блок-схема предлагаемого метода.

веса в файле pth.tar; Compute all train epochs? – Вычислены все эпохи обучения?; Read the next batch of Data (pq, dq) – Считать следующий пакет данных (pq, dq); Adjusted weights – Настроенные веса; Changed the learning rate (Scheduler.step) – Изменена скорость обучения (Scheduler.step); Input – Вход; Model – Модель; Output – Выход; Error computation (CrossEntropyLoss) – Вычисление ошибки (Перекрестная энтропия); Back-propagation algorithm (Backward) – Алгоритм обратного распространения (Backward); Optimize parameters (Optimizer.step) – Оптимизировать параметры (Optimizer.step); End – Конец.).

4.1. Модели-трансформеры

Как правило, нейронные сети обучаются с использованием алгоритма обратного распространения ошибки [12], который оптимизирует параметры модели исходя из минимизации ошибки и улучшения целевой меры качества на тестовой выборке. Однако этот метод сильно зависит от выбора самого алгоритма оптимизации [13], поскольку алгоритм может сойтись к локальному минимуму в процессе вычисления градиентов, что делает модель неспособной

обобщать данные и ухудшает качество предсказания (затухание градиентов). Для решения данной задачи был задействован алгоритм AdamW [14], один из передовых методов оптимизации, который использует информацию об изменении коэффициента скорости обучения для аппроксимации направления антиградиента, а также имеет внутренний "импульс", что улучшает сходимость функции. Этот алгоритм значительно повышает качество обучения, однако он чувствителен к выбору изначального коэффициента скорости обучения. Для регулировки коэффициента скорости обучения был применен косинусный планировщик (cosine scheduler) [15]. Этот планировщик регулирует скорость обучения для каждого пакета (batch) данных. В качестве функции потерь была выбрана кросс-энтропия, поскольку ее величина показывает, насколько хорошо модель справляется с задачей классификации. Применение алгоритма AdamW и планировщика для обучения классификатора на основе BERT было выбрано по следующим причинам.

AdamW Optimizer: AdamW – это версия алгоритма Adam, которая показала свою эффективность в дообучении моделей трансформеров [16]. Данный алгоритм лучше реализует затухание весов, тем самым предотвращает переобучение [17].

Cosine Scheduler: Косинусный планировщик изменяет скорость обучения, начиная с низкой скорости обучения и постепенно увеличивая ее в течение обучения. Период разогрева с низкой скоростью позволяет модели сходиться быстрее, уменьшая нестабильность или колебания функции потерь во время обучения [18].

Трансформеры являются предпочтительным вариантом для классификации медицинских текстов по сравнению с классическими методами, поскольку верно следующее [19].

Предобучение. Трансформеры предобучены на больших корпусах данных, что с самого начала позволяет им получать лучшее представление о языке и закономерностях в нем. Это, в свою очередь, позволяет применять трансформеры в различных задачах после совсем небольшого дообучения.

Контекстное представление. Трансформеры используют двунаправленный механизм внимания, что позволяет им создавать контекстное представление слова в рамках целого предложения. Это важный аспект в задаче классификации текста, которая зависит от качества понимания смысла и контекста предложения.

Перенос обучения (Transfer Learning). Возможно предобучение модели на больших неразмеченных данных из смежной сферы с последующим дообучением на малых размеченных данных из основной сферы. Это позволяет упростить процесс обучения в условиях, когда количество размеченных данных невелико.

Высокая эффективность. Трансформеры показали, что превосходят традиционные методы машинного обучения в различных задачах ОТЕЯ, включая классификацию текстов. Это можно объяснить их способностью улавли-

вать контекстуальные представления и связи между словами, которые имеют большое значение для понимания семантики предложения.

Предобучение на русскоязычных текстах. Модели, предварительно обученные на больших русскоязычных корпусах, показывают более высокое качество в задачах обработки русских текстов по сравнению с обучением с нуля. Предварительное обучение позволяет модели выстроить представление об устройстве языка и упрощает построение контекстных связей.

Важно отметить, что традиционные методы машинного обучения попрежнему широко используются и могут давать хорошие результаты при решении конкретных задач NLP. Однако такие возможности трансформеров, как предобучение, контекстное представление и др., делают это семейство моделей мощным инструментом для классификации медицинских текстов.

4.2. Обучение модели

Используются архитектуры и предобученные веса моделей, полученные с помощью пакета transformers [20]. В самом начале модель инициализируется, а также подготавливается токенизатор с помощью модуля AutoTokenizer пакета transformers. Выходной слой модели изменен вручную для задачи классификации.

Затем подбирается оптимальный размер пакета данных (batch size). Для этого генерируется небольшой синтетический набор данных и затем на основе этого набора данных выполняется поиск по сетке значений (grid search), чтобы определить такой размер пакета, при котором скорость обучения и количество утилизируемых вычислительных ресурсов максимальны. Этот шаг важен для обеспечения максимальной эффективности модели при запуске на удаленном сервере.

В процессе обучения важным моментом является агрегация энергий после применения функции активации Softmax [21]. Итоговая энергия, представленная в виде вероятностных оценок, служит индикатором уверенности модели в отнесении входных данных к определенным классам. Такая мера уверенности играет важную роль в окончательных предсказаниях модели. Важно отметить, что здесь целевые метки — это числовые идентификаторы или номера классов, предварительно закодированные с помощью метода LabelEncoder, а входные данные представляют собой вопросы на естественном языке с описанием жалоб пользователя.

5. Постановка эксперимента

В данном разделе описывается порядок проведения эксперимента, длительность обучения каждой модели и используемое оборудование.

5.1. Оборудование

Для обучения моделей было задействовано два графических ускорителя NVIDIA V100 с 32GB памяти в каждом. Также системе было предоставлено

250GB оперативной памяти для более эффективного хранения и обработки большого набора данных. Обучение моделей выполнялось на суперкомпьютере cHARISMa [22].

5.2. Длительность обучения

Длительность обучения каждой модели зависит от ее архитектуры. Ниже представлены результаты измерений для каждой модели.

- **SBERT [23]**: Модель SBERT потребовала приблизительно 54 ч для обучения. Большая длительность обучения может быть обусловлена глубокой архитектурой и сложным механизмом внимания.
- **LaBSE [24]**: Модель LaBSE потребовала приблизительно 12 ч на обучение. Можно предположить, что на это оказала влияние эффективная архитектура модели и достаточный уровень предобучения.
- **RuBERT [25]**: Обучение модели RuBERT заняло приблизительно 13 ч. Архитектура модели, разработанная специально для применения с текстами на русском языке, потребовала дополнительное время для завершения дообучения.
- **BERT [26]**: Как и LaBSE, модель BERT потребовала приблизительно 12 ч на обучение.
- **BART [27]**: Модель BART потребовала больше времени на обучение – приблизительно 55 ч. Это может быть обусловлено сложностью модели и необходимостью дополнительного обучения структуры кодировщиков–декодировщиков.

Полный цикл оптимизации гиперпараметров, обучения и тестирования, включая кросс-валидацию по k-блокам со значением $k = 3$, занимает от 3 до 12 сут в зависимости от модели.

Для обучения некоторых моделей нужны значительные временные и вычислительные ресурсы, однако степень улучшения показателей моделей и получение сравнительных характеристик нескольких моделей оправдывает усилия, затраченные на дообучение моделей.

6. Результаты экспериментов

В этом разделе представлены результаты экспериментов и их анализ.

На рис. 3 изображены графики кривых обучения моделей LaBSE, SBERT, BERT и BART. Целевая метрика представлена F1-мерой, больше — лучше. Очевидно, что LaBSE показывает лучшую точность по сравнению со многими другими моделями. Кривая обучения LaBSE демонстрирует быструю сходимость и высокое качество. Однако именно для русскоязычного текста модель RuBERT достигает наивысшего качества благодаря предварительному обучению на русскоязычном корпусе текстов.

Модели SBERT, BART и BERT показывают более низкую точность и менее быструю сходимость в рассматриваемой задаче. Модели LaBSE и RuBERT лучше всего подходят для рассматриваемой задачи классификации текстов.

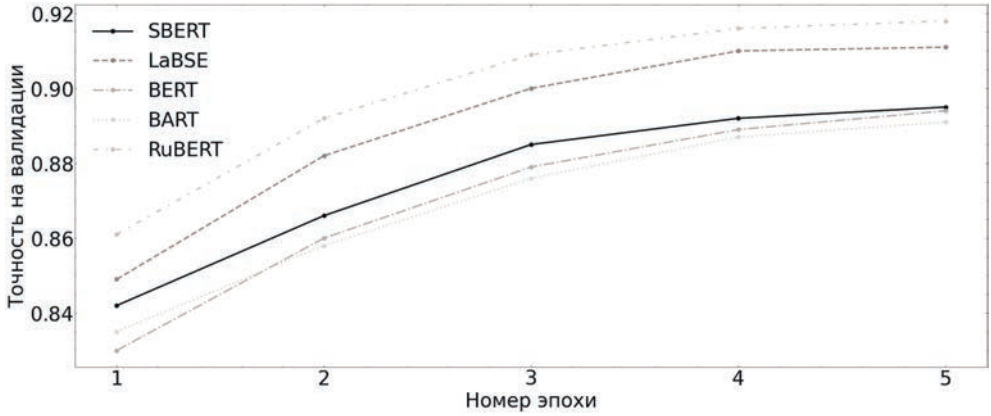


Рис. 3. Кривая обучения различных моделей.

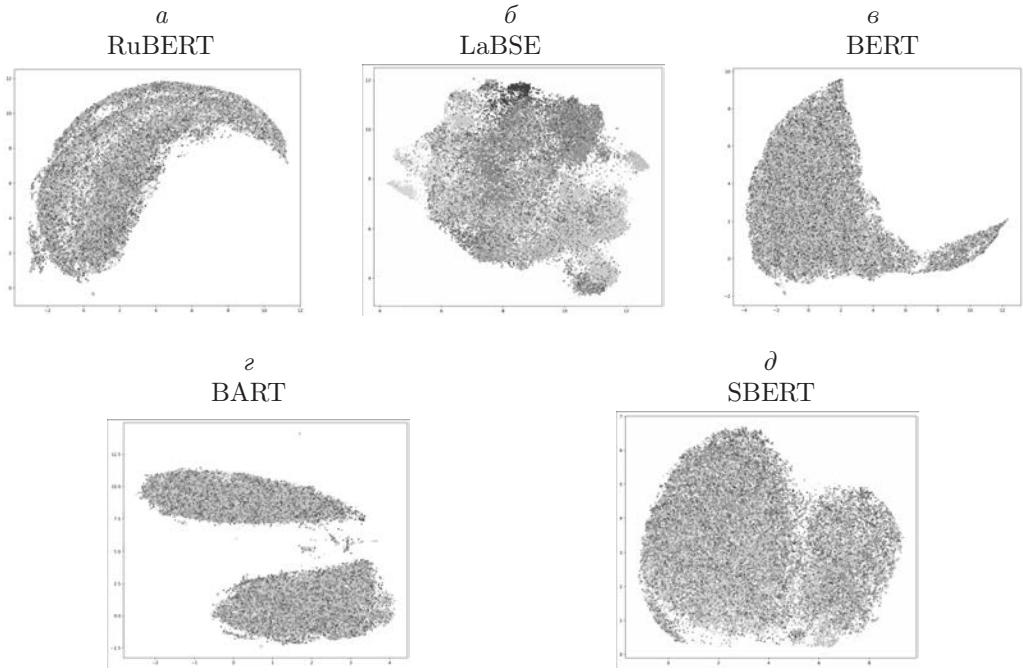


Рис. 4. Визуализация результатов UMAP для различных моделей с использованием собранного набора данных.

Это можно объяснить тем, что LaBSE хорошо различает сущности в тексте, что видно на изображении, полученном с помощью метода UMAP (Uniform Manifold Approximation and Projection[3]), который преобразует многомерные эмбеддинги в двумерное представление. Также видно, что изображение, полученное с помощью UMAP для RuBERT, похоже на другие для методов, которые работают хуже. Однако несмотря на это, благодаря адап-

[3] https://umap-learn.readthedocs.io/

Таблица 2. Сравнение моделей

Модель	K-fold (F1-score, $k = 3$)		Split (F1-score, train — 90%)	
	–	аугментация	–	аугментация
BART	0,798	0,891	0,794	0,896
BERT	0,743	0,894	0,760	0,903
LaBSE	0,824	0,911	0,833	0,913
LogRegression	0,457	0,552	0,531	0,564
Random Forest	0,521	0,579	0,596	0,603
RuBERT	**0,839**	**0,918**	**0,852**	**0,918**
SBERT	0,782	0,905	0,761	0,895
SVM	0,525	0,565	0,534	0,598

Таблица 3. Метрики модели RuBERT для классификации медицинской специальности

Специальность	Точность	Полнота	F1-Оценка	Поддержка
Венеролог	0,7763	0,8112	0,7934	15 140
Гастроэнтеролог	0,7574	0,7339	0,7455	14 839
Гинеколог	0,7834	0,7459	0,7642	14 844
Дерматолог	0,7111	0,6569	0,6829	14 941
Детский хирург	0,8405	0,8782	0,8589	14 847
Инфекционист	0,8409	0,7986	0,8192	14 924
Кардиолог	0,8646	0,8567	0,8606	14 836
ЛОР	0,7555	0,7432	0,7493	15 276
Невропатолог	0,6633	0,5834	0,6206	15 058
Нейрохирург	0,8797	0,9025	0,8910	14 898
Онколог	0,8796	0,8742	0,8769	14 957
Офтальмолог	0,9403	0,9210	0,9305	14 936
Педиатр	0,6482	0,5712	0,6073	15 087
Психолог	0,7759	0,7215	0,7477	15 020
Сексолог-андролог	0,7904	0,6955	0,7399	15 148
Стоматолог	0,8815	0,8893	0,8854	14 861
Терапевт	0,5066	0,3738	0,4302	15 080
Травматолог-ортопед	0,7981	0,7683	0,7829	15 081
Уролог	0,6445	0,6240	0,6341	15 110
Хирург	0,6705	0,5818	0,6230	14 929
Эндокринолог	0,8478	0,8072	0,8270	15 011
точность	0,9111	0,9111	0,9111	0,9031
среднее значение	0,9177	0,9205	0,9189	1 470 000
взвешенное среднее	0,9178	0,9201	0,9189	1 470 000

тации модели для русского языка, RuBERT после дообучения начинает показывать высокую точность, см. рис. 4.

В ходе обучения моделей были измерены результаты для каждой модели в различных режимах обучения.

Таблица 2 дает представление о качестве каждой модели в условиях эксперимента при использовании полученного набора данных. Высокий показа-

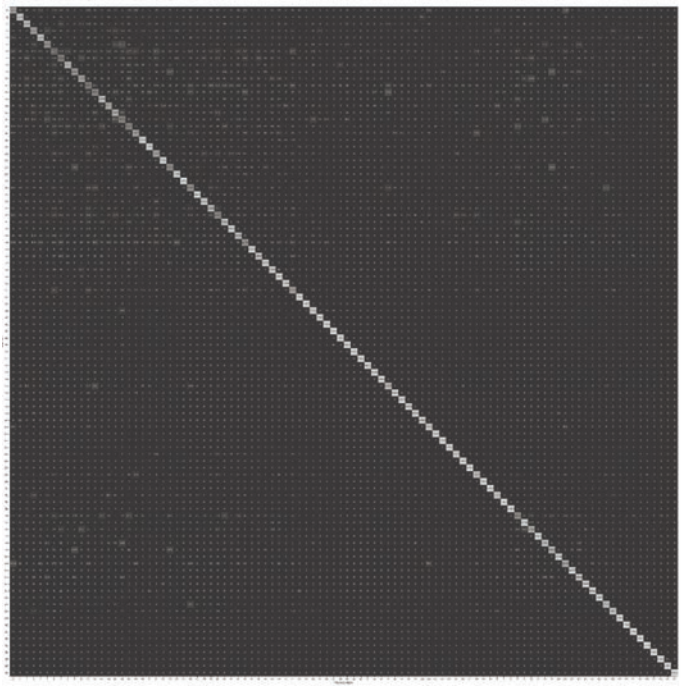


Рис. 5. Матрица ошибок классификатора RuBERT на тестовом наборе данных.

тель метрики у моделей RuBERT и LaBSE говорит о том, что эти модели эффективнее других отражают смысл и контекст текстовых данных в условиях эксперимента. SBERT и BERT также продемонстрировали достаточную производительность, хотя и несколько ниже, чем RuBERT и LaBSE. BART продемонстрировал еще более низкую F1-меру, что может быть объяснено влиянием задачи и конкретного набора данных.

В табл. 3 представлено качество предсказаний модели RuBERT в рамках отдельных классов (21 класс был выбран случайным образом и отсортирован по названию специальности). Представлены точность (precision), полнота (recall), F1-мера (F1-score) и количество текстов в конкретном классе (support). Вместе эти показатели позволяют судить о способности модели корректно различать и предсказывать медицинские специальности по входному тексту.

Матрица ошибок (см. рис. 5) позволила детально изучить результаты классификации, выделив истинно положительные, истинно отрицательные, ложно положительные и ложно отрицательные результаты. Этот анализ выявил заметную тенденцию: большинство ошибок связано со сложностью структуры и семантики настоящих медицинских текстов, где нюансы языка и контекста могут привести к проблемам классификации. Например, обращения, рассматриваемые терапевтом, имеют невысокую полноту (0,3738) в силу возможного общего характера практики (поскольку в реальных больницах посе-

щение терапевта обычно необходимо для обращения к специалистам других профилей).

Стоит отметить, что в ходе оценки модели с использованием только синтетических данных матрица ошибок показывает, что такие данные классифицировались лучше. Это резкое различие показывает, что модель хуже работает с неоднородным, но настоящим текстом пользователя по сравнению с более предсказуемыми и однородными синтетическими данными.

Также были обнаружены некоторые ограничения модели.

Во-первых, длина входного текста ограничена 128 словами и в некоторых случаях ее недостаточно для понимания тонких нюансов, доступных в более длинных данных. При превышении этого порога данные представляются в виде разреженных векторов, что потенциально приводит к потере информации и снижению точности.

Во-вторых, возможное различие в стиле письма тестовых данных и обучающих данных может повлиять на точность предсказаний. Обучение модели только одному определенному стилю ограничивает ее способность адаптироваться к новым, ранее не встречавшимся стилям письма.

Кроме этого, еще одно ограничение создает наличие вопросов, затрагивающих темы, слабо представленные в обучающих данных. Модель хуже предсказывает, когда сталкиваются с вопросами, затрагивающими незнакомые ситуации, поскольку ей не хватает информации о контексте и значении предложения.

7. Заключение

В данной работе был собран исчерпывающий набор данных из открытых источников для классификации медицинских текстов жалоб пользователей среди 97 классов медицинских специальностей по 50 000 экземпляров на класс. Набор данных сбалансирован различными методами аугментации. С использованием полученного набора данных были обучены современные модели трансформеров: SBERT, BERT, LaBSE, BART и RuBERT. Модель RuBERT показала лучшее качество с F1-мерой 91,8%. Полученные результаты позволяют сделать вывод о том, что модели-трансформеры, в частности RuBERT, крайне эффективны в задаче классификации текстов. Способность трансформеров “захватывать” контекстное представление и обнаруживать сложные закономерности в естественном языке делает их значительно точнее по сравнению с классическими методами.

Дальнейшие исследования могут быть направлены на изучение применимости этих моделей трансформеров для дообучения на небольших наборах данных или на узкоспециализированных наборах данных. Кроме того, существует потенциал для разработки новой архитектуры трансформеров, специально предназначенных для задач классификации текстов. Эти архитектуры могут включать в себя знания, специфичные для конкретной области, что, в свою очередь, еще больше повысит точность классификации.

Также перспективным направлением будущих работ может стать исследование новых методов переноса обучения, стратегий дообучения и оптимизации гиперпараметров для моделей-трансформеров. Существуют широкие возможности для развития области классификации медицинских текстов с использованием моделей-трансформеров, и эти будущие работы могут способствовать разработке более точных и эффективных моделей.

СПИСОК ЛИТЕРАТУРЫ

1. Trusting Social Media as a Source of Health Information: Online Surveys Comparing the United States, Korea, and Hong Kong / H. Song // J. Medic. Internet Res. 2016. V. 18. No. 3. P. 25. URL: https://www.jmir.org/2016/3/e25. https://doi.org/10.2196/jmir.4193
2. БэбиБлог — Ответы на любые вопросы о беременности, детях и семейной жизни. Accessed: December 19 , 2022. https://www.babyblog.ru/
3. *Keshavarz H.* Evaluating credibility of social media information: current challenges, research directions and practical criteria // Inform. Discover. Deliver. 2021. V. 49. No. 4. P. 269–279. https://doi.org/10.1108/IDD-03-2020-0033
4. Automatic medical specialty classification based on patients' description of their symptoms / C. Mao / BMC Medical Informatics and Decision Making. 2023. V. 23. https://doi.org/10.1186/s12911-023-02105-7
5. *Tezgider M., Yildiz B., Aydin G.* Text classification using improved bidirectional transformer // Concurrency and Computation: Practice and Experience. 2022. V. 34. No. 9. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpe.6486. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.6486. https://doi.org/https://doi.org/10.1002/cpe.6486
6. СпросиВрача: Задай вопрос врачу онлайн и получи ответ мгновенно. Accessed: February 17, 2023. https://sprosivracha.com/
7. ДОКТУ — поиск лучших врачей и клиник в России. Accessed: February 17, 2023. https://doctu.ru/
8. 03 Онлайн — медицинские консультации в режиме онлайн. Accessed: February 17, 2023. https://03online.com/
9. health.mail.ru — Поиск по болезням, лекарствам и ответам врачей. Accessed: February 17, 2023. https://health.mail.ru/
10. *Johnson J.M., Khoshgoftaar T.M.* Survey on deep learning with class imbalance // Journal of Big Data. 2019. V. 6. No. 1. P. 27. https://doi.org/10.1186/s40537-019-0192-5
11. *Ma E.* NLP Augmentation. 2019. Accessed: February 17, 2023. https://github.com/makcedward/nlpaug
12. *Hecht-Nielsen R.* III.3 – Theory of the Backpropagation Neural Network (Based on "nonindent" by Robert Hecht-Nielsen, which appeared in Proceedings of the International Joint Conference on Neural Networks 1, 593–611, June 1989). © 1989 IEEE / Neural Networks for Perception / H. Wechsler (Ed.). Academic Press, 1992. P. 65–93. ISBN 978-0-12-741252-8. https://doi.org/10.1016/B978-0-12-741252-8.50010-8. URL: https://www.sciencedirect.com/science/article/pii/B9780127412528500108

13. *Shaheen Z., Wohlgenannt G., Filtz E.* Large Scale Legal Text Classification Using Transformer Models. 2020. arXiv: 2010.12871 [cs.CL]
14. Understanding AdamW through Proximal Methods and Scale-Freeness / Z. Zhuang. 2022. arXiv: 2202.00089 [cs.LG]
15. Automated Learning Rate Scheduler for Large-batch Training / C. Kim. 2021. arXiv: 2107.05855 [cs.LG]
16. Attention Is All You Need / A. Vaswani. 2017. arXiv: 1706.03762 [cs.CL]
17. Large Batch Optimization for Deep Learning: Training BERT in 76 minutes / Y. You. 2020. arXiv: 1904.00962 [cs.LG]
18. Are Transformers more robust than CNNs? / Y. Bai // Advances in Neural Information Processing Systems. 2021. P. 34. Curran Associates, Inc. P. 26831–26843. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/e19347e1c3ca0c0b97de5fb3b690855a
19. A Survey on Text Classification: From Shallow to Deep Learning / Q. Li. 2021. arXiv: 2008.00364 [cs.CL]
20. Transformers: State-of-the-Art Natural Language Processing / T. Wolf [et al.] // Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. Online : Association for Computational Linguistics. 2020. P. 38–45. URL: https://www.aclweb.org/anthology/2020.emnlp-demos.6
21. *Maida A.* Cognitive Computing and Neural Networks: Reverse Engineering the Brain / Handbook of Statistics. V. 35. Elsevier. 2016. P. 39–78. https://doi.org/10.1016/bs.host.2016.07.011 URL: https://doi.org/10.1016/bs.host.2016.07.011
22. *Kostenetskiy P.S., Chulkevich R.A., Kozyrev V.I.* HPC Resources of the Higher School of Economics / J. Physics: Conf. 2021. P. 1740. No. 1. P. 012050. https://doi.org/10.1088/1742-6596/1740/1/012050 URL: https://dx.doi.org/10.1088/1742-6596/1740/1/012050
23. *Reimers N., Gurevych I.* Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. 2019. arXiv: 1908.10084 [cs.CL]
24. Language-agnostic BERT Sentence Embedding / F. Feng. 2022. arXiv: 2007.01852 [cs.CL]
25. *Kuratov Y., Arkhipov M.* Adaptation of Deep Bidirectional Multilingual Transformers for Russian Language. 2019. arXiv: 1905.07213 [cs.CL]
26. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding / J. Devlin. 2019. arXiv: 1810.04805 [cs.CL]
27. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension / M. Lewis. 2019. arXiv: 1910.13461 [cs.CL]

Статья представлена к публикации членом редколлегии А.А. Галяевым.

Поступила в редакцию 08.07.2023

После доработки 07.11.2023

Принята к публикации 20.01.2024

© 2024 г. А.Г. СОРОКА (andrew.soroka@student.msu.ru),
Г.В. МИХЕЛЬСОН (mikhelson.g@gmail.com)
(Московский государственный университет им. М.В. Ломоносова),
А.В. МЕЩЕРЯКОВ, канд. физ.-мат. наук (mesch@cosmos.ru)
(Московский государственный университет им. М.В. Ломоносова;
Институт космических исследований РАН),
С.В. ГЕРАСИМОВ (sergun@gmail.com)
(Московский государственный университет им. М.В. Ломоносова)

SMART ROUTES: СИСТЕМА ДЛЯ РАЗРАБОТКИ И СРАВНЕНИЯ АЛГОРИТМОВ РЕШЕНИЯ ЗАДАЧИ ОПТИМИЗАЦИИ МАРШРУТОВ С РЕАЛИСТИЧНЫМИ ОГРАНИЧЕНИЯМИ

Задача оптимизации маршрутов с реалистичными ограничениями становится крайне актуальной в условиях глобального роста городского населения. Подходы к оптимизации маршрутов, включая точные методы, сталкиваются с проблемой экспоненциальной сложности при увеличении размера задачи оптимизации маршрутов. В работе сравнивается точный решатель SCIP с эвристическими методами (LKH, 2-OPT, 3-OPT, OR-Tools) и моделью глубокого обучения JAMPR+. Для задач размером 50 глубокое обучение и классические эвристики достигают точности, сравнимой с SCIP, но требуют меньше времени. Для задач размером 100, эвристики и нейронные сети значительно опережают SCIP как по времени, так и по качеству первого найденного решения. Для проведения экспериментов разработана платформа Smart Routes для решения задачи оптимизации маршрутов, которая включает в себя точные, эвристические и нейросетевые модели и облегчает удобное интегрирование собственных алгоритмов и наборов данных.

Ключевые слова: CVRPTW, система оптимизации маршрутов, эвристики, точные решения, глубокое обучение с подкреплением.

DOI: 10.31857/S0005231024030083, **EDN:** TOTOOH

1. Введение

Задача оптимизации маршрутов (***Vehicle Routing Problem, VRP***) — это класс задач логистики, направленных на минимизацию затрат на транспортные ресурсы, стоимость маршрута и времени доставки груза группе клиентов. В условиях реального мира оптимизация маршрутов является актуальной проблемой для большинства компаний. По мере увеличения числа городов и клиентов, становится необходимым разработать решение, которое позволит оптимально использовать выделенные ресурсы при сохранении качества услуг. Чем короче маршрут, тем быстрее клиент может получить свой товар и тем больше времени остается для доставки товаров другим клиентам.

В логистических задачах зачастую *VRP* рассматривают с различными ограничениями, которые необходимо учитывать при построении маршрута. Наиболее популярными из них являются ограничения на *время посещения и обслуживания клиентов* (TW) и *вместимость транспортных средств* (C). На сегодняшний день возникают большие трудности при разработке и модификации алгоритмов, способных решать задачи с увеличивающимися размерностями и при этом соблюдающих компромисс между качеством решений и временем их поиска в зависимости от количества наложенных ограничений на задачу. Помимо этого, не существует единой платформы, которая позволяла бы пользователям не только решать задачи с помощью встроенных алгоритмов и экспериментировать с ними, но и добавлять собственные алгоритмы, методы чтения и обработки данных и т.п. без изменения архитектуры системы.

Вклад данной статьи заключается в следующем: 1) платформа для оптимизации задач маршрутизации; 2) сравнительный анализ эффективности распространенных методов оптимизации маршрутов. Усилия авторов сосредоточены на разработке инновационной системы под названием Smart Routes и подробном сравнительном анализе различных методов в контексте описанных задач и ограничений. Изучены эвристики, глубокое обучение с подкреплением и точные методы на базе библиотеки (SCIP). Исследование завершается полезной информацией об эффективности и качестве этих решений, в первую очередь оцениваемых двумя основными метриками: временем оптимизации решения и окончательной стоимостью маршрута, определенной целевой функцией. В статье раскрывается архитектура системы Smart Routes. Система без проблем интегрирует точные и эвристические подходы, а также подходы глубокого обучения с подкреплением, разработанные для решения проблемы маршрутизации транспортных средств (VRP) в различных формах и модификациях. Она полезна как экспертам в области логистики, так и тем, кто менее опытен, предоставляя единое место для тестирования новых идей. Следовательно, данное исследование не только углубляется в поиск наиболее эффективного алгоритмического подхода к решению проблемы маршрутизации транспортных средств с временными окнами (CVRPTW) — подкатегории VRP с популярными и практическими ограничениями, но также представляет инновационный инструмент, готовый решать эту проблему всесторонне и эффективно.

Статья устроена следующим образом: раздел 2 предоставляет обзор последних публикаций, содержащих как классический, так и глубокий подход к оптимизации маршрутов с использованием обучения с подкреплением; разделы 3 и 4 содержат описания моделей и разработанной системы Smart Routes соответственно; раздел 5 описывает данные, на основе которых проводились эксперименты; в разделе 6 представлены полученные результаты, а в разделе 7 содержатся выводы, сделанные в ходе проведенного исследования.

2. Обзор литературы

В качестве обзора рассмотрены три группы алгоритмов, способных решать рассматриваемую задачу. Отметим, что, несмотря на богатую историю данной проблемы, существует много исследований, сравнивающих различные подходы к решению задачи оптимизации маршрута. Большинство статей часто сталкиваются с отсутствием точных методов в сравнении, а также ограниченным размером рассматриваемых задач и, иногда, недостатком ограничений. Здесь постараемся охватить все требования, чтобы предоставить полную картину применимости классических подходов.

2.1. Эвристические алгоритмы

Эвристические алгоритмы включают *конструктивные эвристики* и *метаэвристики*, предоставляя субоптимальные решения для VRP. **Конструктивные эвристики** пошагово создают решения, например *эвристика ближайшего соседа* и *эвристика вставки*. Они обеспечивают быстрое формирование качественных решений. **Метаэвристические алгоритмы**, такие как *имитация отжига* и *генетические алгоритмы*, исследуют пространство решений с использованием рандомизации. Они могут находить более качественные решения, но требуют больше времени на вычисления.

Использование этих алгоритмов зависит от задачи. Конструктивные эвристики применяются благодаря их скорости и эффективности в получении быстрых качественных решений. Метаэвристики более гибки, часто используются в широком спектре задач, но могут требовать больше времени на вычисления.

2.2. Точные алгоритмы

Алгоритмы, которые предоставляют гарантированно оптимальное решение, — это **точные алгоритмы**. К ним относятся алгоритмы линейного программирования. Они решают задачи, в которых функции цели и ограничений являются линейными. Задачи линейного программирования хорошо изучены, и известны их свойства с точки зрения существования решения.

Задача линейного программирования (***LP***) — это задача оптимизации в стандартной форме, выражаемая соотношением

$$\max\left\{c^{\mathrm{T}}x \mid Ax \leqslant b, x \geqslant 0\right\}, \tag{1}$$

где $A \in R^{m,n}$ – технологическая матрица, $b \in R^m$ – вектор ресурсов, $c \in R^n$ – вектор цен, $x \in R^n$ – неизвестный вектор. Если некоторые или все переменные в векторе $\boldsymbol{x}$ являются целыми числами, проблема называется *проблемой смешанного целочисленного линейного программирования* (***Mixed Integer Linear Programming, MILP***). Существует несколько алгоритмов, которые могут решать задачи *MILP*:

1. *Метод ветвей и границ* (*Branch&Bound* [1]) разделяет задачу на подзадачи (узлы) и решает их с использованием линейного программирования. Если решение не целочисленное, алгоритм разветвляет узел и решает подзадачи, добавляя ограничения. Процесс повторяется до нахождения целочисленного решения или доказательства его отсутствия.
2. *Метод секущей плоскости* (*Cutting Plane* [2]) добавляет к задаче линейные неравенства, называемые разрезами, чтобы исключить нецелые решения. Разрезы генерируются путем решения LP-релаксации задачи. Алгоритм продолжается до нахождения целочисленного решения или доказательства неосуществимости.
3. *Метод ветвей и отсечений* (*Branch&Cut* [3]) комбинирует *Branch & Bound* и *Cutting Plane*. Генерируются разрезы для устранения нецелых решений, и одновременно разветвляются узлы. Этот метод часто более эффективен, чем отдельно взятые *Branch & Bound* или *Cutting Plane*.

В целом выбор алгоритма зависит от конкретного экземпляра задачи, требований к качеству решения и времени его поиска. Некоторые решатели *MILP*, такие как *CPLEX* [4], *SCIP* [5] и *Gurobi* [6], реализуют несколько из этих алгоритмов и автоматически выбирают наиболее подходящий из них для данного экземпляра задачи.

Основная трудность с задачами линейного программирования заключается в их размерности, так как могут быть тысячи переменных и ограничений. Объем памяти и время решения могут увеличиваться экспоненциально по мере добавления целочисленных переменных. Поэтому специально для того, чтобы находить приближенное к оптимальному решение за меньшее время, были разработаны эвристики. Для сложных проблем эвристические подходы часто могут предложить лучший компромисс между качеством решения и вычислительным временем.

2.3. Алгоритмы глубокого обучения и обучения с подкреплением

Первая модель глубокого обучения для решения VRP была предложена в [7], где был адаптирован Pointer Network (PtrNet) из [8] для работы с CVRP. В [7] полностью отказались от исходной части модели кодировщика RNN и заменили ее линейным слоем с общими параметрами. Более новый алгоритм AM в [9] заменил эту архитектуру адаптированной моделью трансформера с использованием внимания [10]. Прямым улучшением этой модели является подход JAMPR, предложенный в [11], где авторы добавили дополнительные полносвязные сети для текущего пути и положения грузовиков. Это дополнение позволило алгоритму успешно решить проблему CVRPTW. В [12] предлагают подход к улучшению на основе RL, который итеративно выбирает область на графике, а затем выбирает и применяет установленные локальные эвристики. Этот подход был дополнительно улучшен оператором разрушения, представленным в [13]. Последняя попытка использовать глубокое

обучение для разделения набора точек на подзадачи и решения их с помощью решателя черного ящика была предложена в [14]. Авторы предложили два подхода: регрессионное прогнозирование возможного улучшения конечной стоимости и классификацию на лучшую подзадачу. За счет уменьшения размерности и использования классических метаэвристических подходов в каждой подзадаче авторам удалось показать хорошие результаты на задачах большой размерности (более 1000 точек). Была подробно рассмотрена и исследована применимость подхода на базе глубокого обучения с подкреплением для решения задачи оптимизации маршрута с реалистичными ограничениями и продемонстрировано, что быстрое субоптимальное решение может быть получено с использованием обученной нейронной эвристики, как показано в статье авторов [15].

3. Алгоритмы и модели

3.1. Классические подходы

При рассмотрении классических эвристических подходов было принято решение сосредоточиться на группе алгоритмов, известных как *локальный поиск*. Эти алгоритмы обеспечивают хороший баланс между качеством решения и эффективностью поиска в реальных задачах и также служат фундаментальными компонентами для метаэвристик и генетических алгоритмов [16].

Основным эвристическим алгоритмом был выбран один из лучших эвристических подходов — *эвристика Лина–Кернигана* (**Lin–Kernighan heuristic**) [17]. Данный алгоритм относится к классу так называемых алгоритмов локальной оптимизации. Алгоритм задается в терминах *opt* (обмен или ходы), которые могут преобразовать один маршрут в другой. При наличии осуществимого маршрута алгоритм неоднократно выполняет обмены, которые сокращают продолжительность текущего маршрута, пока не будет достигнут маршрут, для которого никакой обмен не приведет к улучшению. Этот процесс может повторяться много раз с начальных маршрутов, сгенерированных каким-либо рандомизированным способом. В алгоритме предполагается, что некоторое начальное разбиение маршрута уже существует, затем имеющееся приближение улучшается в течение некоторого количества итераций. Применяемый способ улучшения состоит в обмене вершинами в каждом подмаршруте в отдельности.

SCIP [5], программное обеспечение с открытым исходным кодом, представляет собой мощный инструмент оптимизации, специально разработанный для решения задач смешанного целочисленного программирования [18, 19]. Он предназначен для решения сложных задач оптимизации, возникающих, например, в области логистики, маршрутизации и планирования производства. SCIP использует широкий спектр методов и алгоритмов оптимизации, включая методы Branch&Bound, методы секущей плоскости, методы распространения ограничений, эвристики, методы декомпозиции и целочисленное про-

граммирование. В целом *SCIP* представляет собой передовое программное обеспечение для оптимизации, использующее широкий спектр алгоритмов и методов для эффективного и точного решения сложных задач оптимизации.

В дополнение к SCIP был выбран OR-Tools (Operations Research Tools) [20], популярный фреймворк, заслуживший признание своей эффективностью в решении задач VRP (маршрутизации транспортных средств). OR-Tools — это универсальное программное обеспечение, предназначенное для решения комбинаторных задач оптимизации, уравнений и неравенств, а также задач планирования и маршрутизации. Он предлагает разнообразный набор методов и алгоритмов оптимизации, включая *линейное программирование (LP)*, *целочисленное программирование (IP)*, *методы дискретной оптимизации* и *методы решения уравнений и неравенств*. OR-Tools представляет собой мощный инструмент для решения различных задач оптимизации. Благодаря своей гибкости и широкому набору методов, OR-Tools может быть использован для решения разнообразных задач в областях логистики, планирования, транспорта и др.

3.2. Модели глубокого обучения с подкреплением

Модифицировав модель JAMPR [11], используем ее в качестве подхода глубокого обучения с подкреплением [15]. Модель JAMPR является модификацией модели внимания (AM) [9], которая использует архитектуру энкодер–декодер на основе внимания [10]. Обе модели рассматривают проблему оптимизации маршрута как задачу последовательного принятия решений, моделируемую как марковский процесс принятия решений и решаемую с использованием обучения с подкреплением. Решение проблемы строится поэтапно, создавая маршруты по одному узлу за раз. Текущее решение, маршрут и непосещенные узлы рассматриваются как состояние, а индекс всех непосещенных узлов, доступных для добавления к текущему маршруту, рассматривается как действие.

К подходу JAMPR были добавлены обучаемые матрицы (блок M+ на рис. 1), отдельные для каждого ограничения. Они применяются к результату работы декодера, модифицируя тем самым политику (π_θ). Окончательным результатом работы сети является измененная политика размером ($k*$, $n*$). Процесс оптимизации заключается в минимизации стоимости маршрута с учетом пропущенных клиентов (мягкая постановка задачи):

$$Q = \sum_{j=1}^{m} \sum_{i=1}^{n} c_{ij} x_{ij} + \lambda \sum_{i=1}^{n} z_i, \tag{2}$$

где i, $i = 1, \ldots, N$ — клиенты, j, $j = 1, \ldots, M$ — транспортные средства, c_{ij} — стоимость маршрута от j транспортного средства к i клиенту, z_i — бинарный параметр, указывающий, пропущен ли клиент.

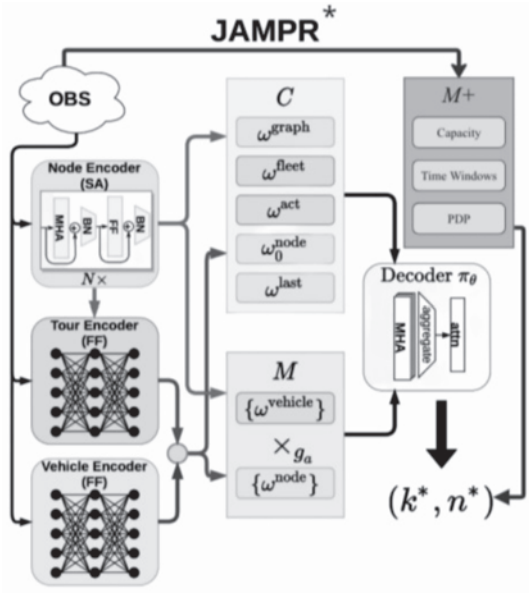


Рис. 1. Архитектура модели JAMPR, модифицированная для решения проблем CPDPTW.

Рабочий процесс модели: сначала кодировщик получает характеристики x_i каждого узла i (координаты, вес грузов, временные окна и т.д.) и кодирует их в скрытый линейный вектор $\tilde{x_i} \in R^{d_{emb}}$ размером d_{emb}. Затем модель декодера вычисляет запрос внимания для каждого $\tilde{x_i}$ по отношению к конкретному контексту $C^{(t)}$ на шаге декодирования t для получения оценок для всех узлов, которые могут быть добавлены к текущему маршруту. Здесь контекст включает неявное встраивание графа задачи и дополнительную информацию о проблеме, такую как индекс узла депо, последний узел, добавленный к текущему маршруту, и оставшуюся пропускную способность. Полученные оценки затем либо используются в жадной процедуре выбора, т.е. всегда выбирается узел с наивысшей оценкой, либо используется softmax для преобразования его в распределение, применяемое для выборки. В общем случае модель кодер–декодер представляет собой политику $\pi(i^{(t+1)}|C^{(t)}, x; \theta)$ с обучаемыми параметрами θ. Полная архитектура $JAMPR^*$ из [11] представлена на рис. 1. Здесь OBS — текущее состояние среды, Node Encoder — кодировщик состояния вершин графа, Tour Encoder — кодировщик текущего состояния маршрута, Vehicle Encoder — кодировщик состояния грузовика. Блок C — контекст пути, M — контекст Масок, каждый из весов w внутри ассоциируется с разными показателями текущего состояния среды: graph — граф, fleet — флот, act — последнее действие, node — вершины, last — последняя вершина, vehicle — текущий грузовик. Блок $M+$ обозначает обучаемые маски, каждый из Capacity, Time Windows и PDP относится к совему типу ограничений, Объем, Временные Окна и Вывоз и Доставка соответственно.

4. Система *Smart Routes*

4.1. Требования

Система должна соответствовать следующим функциональным **основным требованиям**.

— **Решение задачи VRP с популярными ограничениями.** Это включает в себя решение VRP с общими и важными ограничениями, такими как вместимость транспортных средств и временные окна для доставки. С расширением системы будут добавлены новые ограничения.

— **Решение задач большого масштаба (∼1000 точек).** Решение VRP большого масштаба представляет собой вызовы, поскольку увеличение количества клиентов для посещения затрудняет поиск решений в алгоритмах и приводит к увеличению времени поиска.

— **Использование данных реального мира.** Так как данная работа сравнивает алгоритмы с использованием искусственно сгенерированных данных, предлагаемая система также включает функцию обработки данных реального мира в заданном формате.

— **Проведение экспериментов с классическими эвристическими, точными и глубокими методами обучения с подкреплением.** Задача определения лучшего алгоритма для решения VRP остается актуальной. Поэтому предлагаемая система предоставляет возможность экспериментировать с алгоритмами из разных классов.

— **Интерактивное использование системы.** Для упрощения использования встроенных алгоритмов предусмотрен интуитивно понятный интерфейс для настройки параметров и взаимодействия с системой.

— **Визуальная интерпретация решений.** Реализация алгоритмов иногда может затруднить определение корректности и оптимальности решений. Визуализация решения упрощает этот процесс верификации, предоставляя визуализацию построенного маршрута.

— **Сравнение показателей качества.** При сравнении нескольких алгоритмов полезно анализировать различия в качестве полученных решений. Графики, демонстрирующие уменьшение или увеличение стоимости маршрута за определенный период времени, особенно полезны и удобны для проведения этих экспериментов.

4.2. Архитектура

Архитектура системы представлена на рис. 2. Архитектура системы Smart Routes включает три блока: 1) Модуль данных (Dataset) для обработки данных и загрузки в различных форматах; 2) Модуль алгоритма (Algorithm) занимается обучением и оценкой алгоритмов, позволяя добавлять пользовательские алгоритмы; 3) Модуль решения (Solution) формирует метрики, визуализирует результаты и предоставляет веб-интерфейс для взаимодействия. Значения на рис. 2: Input Dataset — Входной набор данных, Generation

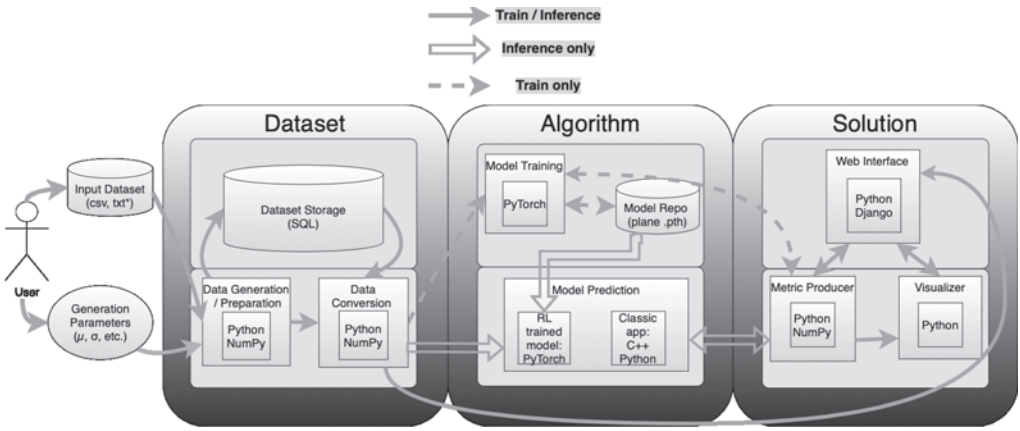


Рис. 2. Архитектура системы Smart Routes.

Parameters — Параметры генерации распределения, Dataset — Набор данных, Dataset Storage — Хранение набора данных, Data Generation — Генерация набора данных, Data Convertion — Преобразование набора данных, Algorithm — Алгоритм, Model Training — Обучение модели, Model Repo — Репозиторий модели, Model Prediction — Предсказание модели, RL Trained Model — Обученная модель обучения с подкреплением, Solution — Решение, Web Intefrace — Веб-интерфейс, Metric Producer — Предиктор метрик, Visualizer — Визуализатор.

Опишем основные шаги использования системы. Пользователь устанавливает систему, строго следуя документации. При запуске его перенаправляют в веб-браузер, где отображается веб-страница с предложением ввести необходимые параметры.

1. **Данные и параметры.**
   a. пользователь выбирает проблему, которую он собирается решить: $CVRP$, $VRPTW$, $CVRPTW$;
   b. выбирается имя алгоритма решения;
   c. устанавливается временное ограничение для решения одной задачи;
   d. выбирается вариант решения задач последовательно или параллельно на доступном количестве процессоров;
   e. загружается набор данных в определенном формате;
   f. указываются параметры для генерации данных, если это необходимо: 1) μ — математическое ожидание; 2) δ — дисперсия; 3) $capacity_{\min}$ — минимальное значение объема груза потребителя; 4) $capacity_{\max}$ — максимальное значение объема груза потребителя; 5) $n_{samples}$ — количество примеров задач; 6) $n_{customer}$ — количество клиентов в каждой задаче; 7) $service_{window}$ — правое временное окно для депо; 8) $service_{duration}$ — время обслуживания клиента.
2. **Набор данных.** Пользовательские параметры и загруженные данные передаются блоку *Генерации/Подготовки данных*, где данные либо ис-

кусственно генерируются на основе указанных параметров пользователя, либо сохраняются в переменных. Затем пользовательские наборы данных сохраняются в базе данных или немедленно передаются блоку *Преобразование данных*, где данные преобразуются и приводятся к необходимым типам.

3. **Алгоритм.** Этот блок получает предобработанные данные и параметры, указанные пользователем. Система определяет, к какому классу проблем принадлежит выбранный алгоритм: *classic_heuristics*, *exact_methods*, *neural_methods*.

   В результате для каждой представленной задачи пользователь получает конечный маршрут, т.е. список клиентов и его стоимость.

4. **Решение.** Результаты, полученные из предыдущего блока, передаются блоку *Генерации метрик*, который создает общий граф с метриками для выбранных пользователем алгоритмов, демонстрируя уменьшение стоимости маршрута в пределах указанного времени. Результаты также передаются блоку *Визуализатор*, который визуализирует полный графический маршрут на основе полученного списка клиентов.

По завершении работы система отображает сгенерированные графики, конечную стоимость маршрута и список клиентов в порядке их посещения на начальной веб-странице.

4.3. Преимущества

Предложенная платформа обладает ключевой особенностью, позволяющей пользователям добавлять собственные алгоритмы и модели, что придает ей значительное преимущество перед другими фреймворками, поскольку:

1. Понимание инструментария фреймворка может занять много времени. Например, реализация модели CVRPTW с использованием программного обеспечения SCIP требовала значительного времени для разбора того, как правильно реализовать ограничения, определить архитектуру модели и предобработать входные данные.
2. Пакеты программного обеспечения, такие как OR-Tools, SCIP, Gurobi, CPLEX, позволяют создавать модели только в качестве точных методов. По мере увеличения размеров экземпляров проблемы, время поиска решения также растет, что может стать критичным, требуется инструмент реализации эвристик, удобный для сравнения результатов.

В описанной системе пользователи могут интегрировать свои алгоритмы, используя базовые классы. Встроенные алгоритмы подходят для всех методов решения проблем VRP. Также доступны удобные функции: визуализация результатов для оценки эффективности разработанного подхода и API, упрощающий использование встроенных подходов без интеграции собственных алгоритмов.

Таким образом, функционал платформы упрощает разработку, тестирование и экспериментальное сравнение подходов решения VRP, а также демон-

стрирует значительное преимущество по сравнению с существующим программным обеспечением.

4.4. Сравнение Smart Routes и Классических фреймворков

В настоящее время существуют фреймворки, предоставляющие инструменты для решения проблемы маршрутизации транспортных средств (VRP) с различными ограничениями. Самые популярные из них: OR-Tools, Gurobi и SCIP.

В таблице представлены типы алгоритмов, поддерживаемые этими фреймворками (обозначено знаком +, в противном случае знаком —).

Тип алгоритма	Фреймворк			
	SCIP	Gurobi	OR-Tools	Smart Routes
Точные методы	+	+	+	+
Эвристические методы	+	+	+	+
Глубокое обучение с подкреплением	—	—	—	+
Визуализация результатов	—	—	—	+
Добавление собственных алгоритмов	—	—	—	+

Важно отметить, что OR-Tools и SCIP могут не предоставлять тот же уровень производительности и масштабируемости, что и Gurobi, при решении очень крупных задач оптимизации.

Предложенная система отличается от упомянутого программного обеспечения в нескольких аспектах. Помимо охвата всех типов алгоритмов для решения VRP, Smart Routes позволяет интегрировать собственные модели, демонстрируя гибкость, не подвергая риску общую архитектуру. В ней реализована визуализация для создания графиков производительности и представления конечных маршрутов. Кроме того, при установке системы OR-Tools и SCIP устанавливаются автоматически, что упрощает проведение экспериментов сравнения моделей пользователей с моделями, реализованными на наборах инструментов, предоставленных этими фреймворками. Нет необходимости разрабатывать алгоритмы для решения индивидуальных логистических задач, поскольку предложенная система уже дает встроенные реализации. Более того, система предоставляет интерфейс, который упрощает рабочий процесс и делает его удобным инструментом для пользователей.

5. Данные

Рассматриваемыми алгоритмами решалась задача *CVRPTW*, так как наибольший интерес представляет поведение рассмотренных подходов с несколькими ограничениями одновременно, и рассматриваемая задача решалась в *Soft-постановке*, т.е. могут быть точки, которые невозможно посетить. В таком случае они исключаются из итогового маршрута и выносятся в штраф, а

точнее, в формуле стоимости увеличивается переменная *missed_ nodes*. В работе для *CVRPTW* выбраны подходящие экземпляры задач из распределения на основе статистики R201, известного эталонного набора Соломона [21]. Объемы грузовиков заданы как $Q_{50} = 750$, $Q_{100} = 1000$ для задач размеров 50 и 100 соответственно. Общий временной горизонт равен $[a_0, b_0]$, где $a_0 = 0$ для всех примеров, а правая граница b_0 меняется в зависимости от размера задачи: 1000 для 50 и 100 точек. При этом продолжительность обслуживания h_i равномерно устанавливается равной 10.

6. Эксперименты

Для проведения экспериментов были сгенерированы искусственные наборы данных, состоящие из 100 примеров с размерами 50 и 100 точек. Каждой задаче был назначен определенный временной лимит для решения с использованием алгоритмов: 100 с для задач с 50 точками и 200 с для задач со 100 точками. Поскольку точные методы включают полный поиск возможных решений, им требовалось в 10 раз больше времени для каждой задачи, чтобы получить любое решение, даже если оно не оптимальное. Поэтому точному методу было выделено 1000 и 2000 с для задач с 50 и 100 точками соответственно.

Эксперименты проводились с использованием разработанной системы Smart Routes, которая упростила процесс настройки параметров при запуске различных алгоритмов и задач. Для экономии времени при экспериментах задачи были распараллелены на все доступные процессоры машины.

Все эксперименты выполнялись на сервере с графическим процессором NVIDIA Tesla A40 и процессором Intel(R) Xeon(R) Gold 6226R CPU@2.90GHz с 16 виртуальными ядрами.

6.1. Основные результаты

Основные результаты экспериментов отображены на рис. 3, где каждый алгоритм представлен в виде точки. Прямые линии различных цветов соединяют точки, показывая, что общая стоимость маршрутов увеличивается с увеличением размеров проблемы. Результаты, полученные с помощью SCIP, соединены наклонной сплошной линией, а пунктиры указывают на потенциальное геометрическое положение результатов SCIP. Ось x представляет среднее время решения для одной задачи, а ось y представляет среднюю стоимость маршрута для одной задачи. На рис. 3 горизонтальная линия указывает время, необходимое для алгоритма для достижения наилучшей стоимости пути, вертикальная линия указывает наилучшую достижимую алгоритмом стоимость. Наклонная сплошная линия показывает предел применимости точных методов: слева недостаточно времени для алгоритма, чтобы предоставить решение, зона справа представляет собой благоприятную область для выбора точных методов. Обозначения на рис. 3: Cost —

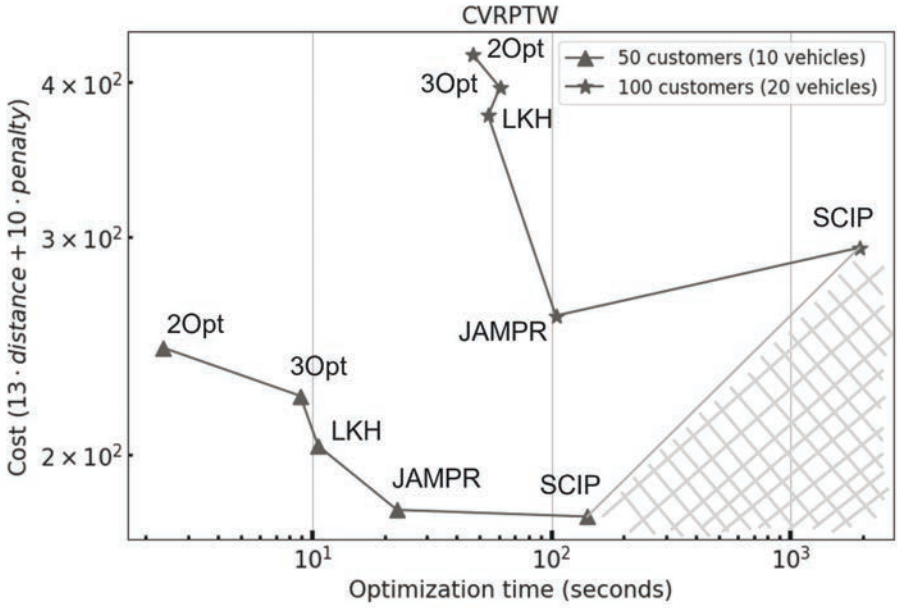


Рис. 3. Итоговый результат классических подходов.

стоимость пути, 50 customers (10 vehicles) — задача с 50 точками и 10 грузовиками, 100 customers (20 vehicles) — задача со 100 точками и 20 грузовиками, Optimization Time — время оптимизации.

На основе полученных результатов можно сделать следующие выводы.

1. Увеличение значения λ в алгоритме LKH (λ Opt) улучшает качество решений, но увеличивает время поиска.
2. По мере увеличения размерности задачи и количества транспортных средств время решения SCIP значительно увеличивается. Удвоение размерности задачи приводит к примерно 14-кратному увеличению времени решения SCIP. В отличие от этого нейронная сеть JAMPR проявляет себя лучше, чем точные методы, как по качеству решения, так и по времени поиска для экземпляров задач с размерностью 100 точек. Это подтверждает, что точные методы имеют ограниченную применимость для решения задач с размерностью 100 точек и более.

Две последующие главы предоставляют более подробное описание проведенных экспериментов на экземплярах задач с размерностью 50 и 100 точек.

6.2. Результаты для экземпляров задач с 50 точками

Для диаграммы с метриками GAP было рассчитано процентное отклонение алгоритмов в каждой точке по сравнению с окончательным лучшим результатом, достигнутым в течение 1000 секунд оптимизации (SCIP). Общая метрика представляет среднее значение среди всех решенных экземпляров задач.

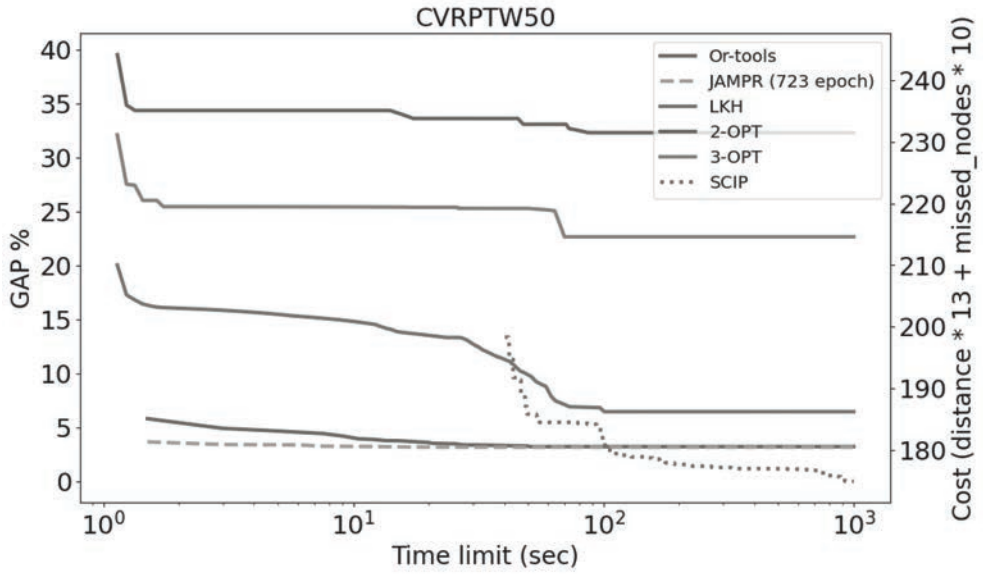


Рис. 4. Эффективность моделей на CVRPTW с 50 точками.

На рис. 4 отражено изменение качества окончательных стоимостей маршрутов по мере оптимизации времени. Сплошные линии — результаты эвристик. Пунктир — лучший результат SCIP за максимальное время. Пунктирная линия — JAMPR после 700 эпох — быстрое субоптимальное решение, сходящееся к эвристике. Две шкалы: общая стоимость (справа) и отклонение от лучшего решения (слева). Для задач с 50 точками как классические эвристики, так и глубокое обучение с подкреплением быстро предоставляют субоптимальные решения. Результаты методов 2-Opt, 3-Opt, LKH, OR-Tools и JAMPR достигают плато, но в начале оптимизации классические эвристики предоставляют начальные жадные решения, с LKH, постепенно приближающимся к результатам JAMPR и OR-Tools. Однако после примерно 10 с SCIP становится менее эффективным, чем LKH, OR-Tools и JAMPR, но после 100 с SCIP превосходит их. Несмотря на то, что точные методы обеспечивают лучшее решение, они требуют значительно больше времени, чтобы превзойти LKH, JAMPR и OR-Tools по качеству решения, при этом конечная разница между моделью глубокого обучения и SCIP составляет менее 5%. Обозначения на рис. 4: Cost — стоимость пути, GAP — метрика, процентное отставание от лучшего решения, Time limit (sec) — время оптимизации в секундах.

Таким образом, для задач с 50 точками, несмотря на то, что точные методы демонстрируют лучшее решение, их превосходят классические эвристики и методы глубокого обучения с подкреплением по времени на порядок. Важно отметить, что нейронные сети и классические эвристические подходы могут предоставить субоптимальное решение для CVRPTW уже в первые несколько с.

На рис. 5 показано, как меняется качество стоимости маршрута во времени для задач со 100 точками. Сплошные линии — результаты эвристик. Пунктир — результаты SCIP. Пунктирная линия — JAMPR после 23 эпох — быстрое субоптимальное решение, аппроксимирующее эвристику. Две шкалы: общая стоимость (справа) и отклонение от лучшего решения (слева). Как и в случае с 50 точками, была вычислена процентная стоимость для каждого алгоритма относительно лучшего результата, показанного SCIP, в пределах максимального времени оптимизации (200 с для классических эвристик и метода глубокого обучения, и 2000 с для SCIP). Обозначения на рис. 5: Cost — стоимость пути, GAP — метрика, процентное отставание от лучшего решения, Time limit (sec) — время оптимизации в секундах.

На графике видно, что классические эвристические методы предоставляют более быстрое субоптимальное решение, но уступают по качеству JAMPR и OR-Tools. Несмотря на то, что JAMPR достигает плато примерно на 100-й с, его начальное жадное решение превосходит LKH примерно на 10%, а конечный результат превосходит LKH GAP примерно на 50%. При увеличении размерности задачи увеличивается разница между начальными решениями в модели глубокого обучения и эвристиками.

Также на рис. 5 видно, что для SCIP потребовалось более 900 с для первого решения, что в 13 раз дольше, чем JAMPR и OR-Tools. В отведенное время SCIP не предложил оптимальное решение, и его первое решение было примерно на 50% дороже, чем решения, предоставленные эвристиками OR-Tools и JAMPR. Это указывает на то, что с увеличением размерности задачи точ-

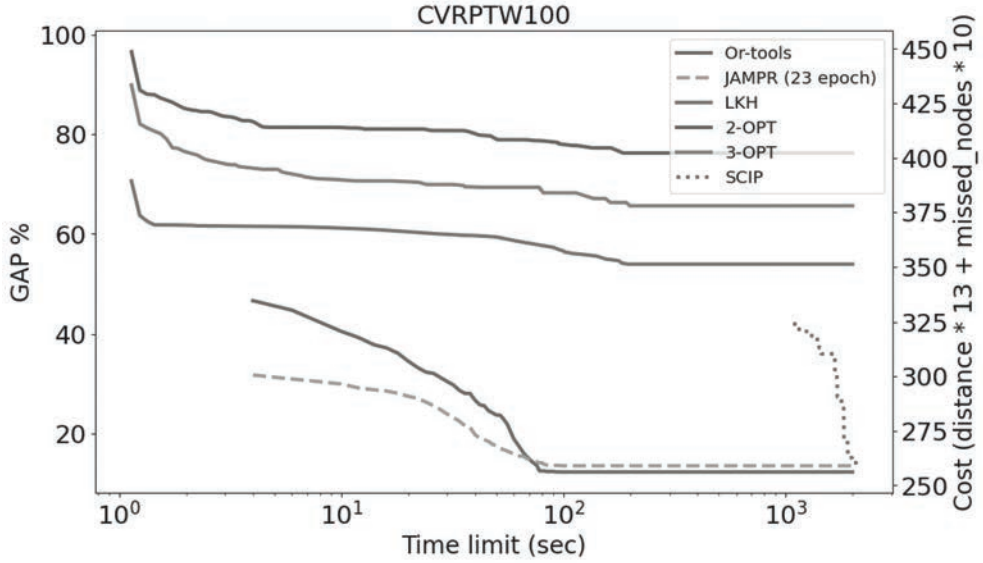


Рис. 5. Эффективность моделей на задаче CVRPTW с 100 точками.

ные методы становятся менее применимыми из-за экспоненциального роста времени.

Таким образом, в случае задач со 100 точками методы глубокого обучения с подкреплением могут предоставить быстрое субоптимальное решение, превосходящее классические эвристики по качеству из-за более длительного времени обучения, и превосходящее точные методы по времени, необходимому для нахождения решения.

7. Заключение

Проблема оптимизации маршрутов с учетом реалистичных ограничений становится чрезвычайно актуальной в свете глобального роста городского населения. Несмотря на наличие подходов, которые теоретически обеспечивают точное оптимальное решение, их применение становится трудным с увеличением размера задачи из-за экспоненциальной сложности. Здесь исследуется задача оптимизации маршрутов с ограничением по времени и вместимости транспортного средства (CVRPTW) и сравниваются решения, полученные с использованием точного решателя SCIP [5] с эвристическими алгоритмами, такими как LKH, 2-OPT, 3-OPT [17], фреймворком OR-Tools [20] и моделью глубокого обучения JAMPR [11].

Все метрики, представленные в статье, были получены с использованием платформы Smart Routes. Эта система обладает возможностью решения задачи оптимизации маршрута с использованием различных подходов, что облегчает исследования и сравнение результатов.

На основе экспериментальных результатов можно сделать следующие выводы:

— Для задач с 50 точками как классические эвристические методы, так и методы обучения с подкреплением демонстрируют свою эффективность, предоставляя быстрые субоптимальные решения. Общие результаты близки к результатам, полученным точным методом (SCIP), с разницей менее 5%. Это свидетельствует о том, что для задач с 50 точками классические эвристические методы и методы, основанные на нейронных сетях, могут предложить хороший баланс между качеством решения и временем поиска.

— Для задач со 100 точками точные методы требовали в 13 раз больше времени для нахождения начального решения. Они не смогли предоставить оптимальное решение в заданный срок, что привело к увеличению затрат на маршрут до 50%. Классические эвристические методы предоставляют быстрые субоптимальные решения, но начинают отставать от более продвинутых методов, таких как JAMPR и OR-Tools, по качеству решения. Эти результаты свидетельствуют о полной нецелесообразности точных методов для решения задачи оптимизации маршрута со 100 (и более) точками.

Разработанная платформа является важным компонентом для будущих исследований в области оптимизации маршрутов с различными ограничениями.

СПИСОК ЛИТЕРАТУРЫ

1. *Laporte G., Nobert Y.* A branch and bound algorithm for the capacitated vehicle routing problem // Operations-Research-Spektrum. 1983. V. 5. P. 77–85.
2. *Cook W., Rich J.L.* A parallel cutting-plane algorithm for the vehicle routing problem with time windows // Technical Report TR99-04, Computational and Applied Mathematics, Rice University, Housten. 1999.
3. *Augerat P., Naddef D., Belenguer J.M., et al.* Computational results with a branch and cut code for the capacitated vehicle routing problem / Research report — IMAG. 1995.
4. *IBM ILOG Cplex* V12. 1: User's Manual for CPLEX // Int. Busin. Machin. Corporat. 2009. V. 46. No. 53. P. 157.
5. *Bestuzheva K., Besancon M., Chen W.-K., et al.* The SCIP Optimization Suite 8.0 // Technical Report, Optimization Online. 2021. http://www.optimization-online.org/DB_HTML/2021/12/8728.html
6. *Gurobi Optimization, LLC* Gurobi Optimizer Reference Manual // 2023. https://www.gurobi.com
7. *Nazari M., Oroojlooy A., Snyder L., et al.* Reinforcement learning for solving the vehicle routing problem // Conf. Advances Neural Inform. Proc. Syst. 2018. V. 31.
8. *Vinyals O., Fortunato M., Jaitly N.* Pointer networks // Conf. Advances Neural Inform. Proc. Syst. 2015. V. 28.
9. *Kool W., Hoof H.V., Welling M.* Attention, learn to solve routing problems! // 2018. arXiv preprint arXiv:1803.08475.
10. *Vaswani A., Shazeer N., Parmar N., et al.* Attention is all you need // Conf. Advances Neural Inform. Proc. Syst. 2017. V. 30.
11. *Falkner J.K., Schmidt-Thieme L.* Learning to solve vehicle routing problems with time windows through joint attention // arXiv preprint arXiv:2006.09100. 2020.
12. *Chen X., Tian Y.* Learning to perform local rewriting for combinatorial optimization // Conf. Advances Neural Inform. Proc. Syst. 2019. V. 32.
13. *Lu H., Zhang X., Yang S.* A learning-based iterative method for solving vehicle routing problems // International conference on learning representations. 2019.
14. *Li S., Yan Z., Wu C.* Learning to delegate for large-scale vehicle routing // Conf. Advances Neural Inform. Proc. Syst. 2021. V. 34.
15. *Soroka A.G., Meshcheryakov A.V., Gerasimov S.V.* Deep Reinforcement Learning for the Capacitated Pickup and Delivery Problem with Time Windows // Patt. Recognit. Imag. Anal. 2023. V. 33. No. 2. P. 169–178.
16. *Groër C., Golden B., Wasil E.* A library of local search heuristics for the vehicle routing problem // Math. Program. Comput. 2010. V. 2. P. 79–101.
17. *Helsgaun K.* An effective implementation of the Lin–Kernighan traveling salesman heuristic // Eur. J. Oper. Res. 2000. V. 126. No. 1. P. 106–130.
18. *Çetinkaya C., Karaoglan I., Gökcen H.* Two-stage vehicle routing problem with arc time windows: A mixed integer programming formulation and a heuristic approach // Eur. J. Oper. Res. 2013. V. 230. No. 3. P. 539–550.
19. *Tahernejad S., Ralphs T.K., DeNegre S.T.* A branch-and-cut algorithm for mixed integer bilevel linear optimization problems and its implementation // Math. Program. Comput. 2020. V. 12. P. 529–568.

20. *Perron L.* Operations research and constraint programming at google // International Conference on Principles and Practice of Constraint Programming. 2011. P. 2–2.

21. *Solomon M.M.* Algorithms for the vehicle routing and scheduling problems with time window constraints // Oper. Res. Inf. 1987. V. 35. No. 2. P. 254–265.

Статья представлена к публикации членом редколлегии А.А. Галяевым.

Поступила в редакцию 08.07.2023

После доработки 10.10.2023

Принята к публикации 20.01.2024

СОДЕРЖАНИЕ

Тематический выпуск

CONTENTS

Surveys