

Федеральное государственное образовательное бюджетное учреждение
высшего образования
«Финансовый университет при Правительстве Российской Федерации»

На правах рукописи

Ехлаков Роман Сергеевич

МЕТОД МНОГОКРИТЕРИАЛЬНОЙ ОЦЕНКИ МОДЕЛЕЙ СЕТЕВЫХ СТРУКТУР НА ОСНОВЕ СХОДСТВА С ИДЕАЛЬНЫМ РЕШЕНИЕМ

1.2.2. Математическое моделирование, численные методы
и комплексы программ

ДИССЕРТАЦИЯ
на соискание ученой степени
кандидата технических наук

Научный руководитель

Судаков Владимир Анатольевич,
доктор технических наук, доцент

Москва – 2025

Оглавление

Введение.....	4
Глава 1 Анализ современных моделей транспортной сети	15
1.1 Транспортная сеть	15
1.2 Многокритериальные методы оценки.....	18
1.3 Модели загруженности и прогнозирования сети	29
1.4 Влияние и прогнозирование погодных условий	42
1.5 Поиск альтернативных маршрутов.....	44
Глава 2 Комплекс моделей сетевых структур дорожного движения и методика оценки эффективности маршрутизации транспортных средств по результатам моделирования.....	53
2.1 Модель многокритериальной оценки эффективности маршрута	54
2.2 Модель загруженности сетевой структуры с учетом погодных условий.....	61
2.3 Модель безопасности маршрута	77
2.4 Когнитивная модель расчета показателей оценки маршрута	85
Глава 3 Численный метод для маршрутизации транспортных средств	93
3.1 Методика обработки данных перемещения транспортных средств ...	94
3.2 Методика оценки загруженности УДС	109
3.3 Методика маршрутизации транспортных средств.....	112
Глава 4 Комплекс программ для маршрутизации транспортных средств	122
4.1 Архитектура комплекса программ.....	122
4.2 Вычислительные эксперименты с моделью загруженности с учетом погодных условий.....	127
4.3 Вычислительные эксперименты с моделью безопасности маршрутов.....	142
4.4 Вычислительные эксперименты с численным методом маршрутизации транспортных средств	144
4.5 Вычислительные эксперименты с моделью многокритериальной оценки эффективности маршрута	150

4.6	Вычислительные эксперименты с когнитивной моделью	157
4.7	Результаты сравнения комплекса программ со сторонними сервисами	158
	Заключение	165
	Список сокращений и условных обозначений	168
	Словарь терминов.....	169
	Список литературы	170
	Приложение А Применение MCDM методов в исследованиях.....	197
	Приложение Б Сравнение методов и моделей в транспортных исследованиях	209
	Приложение В Оценка загруженности дорожного сегмента на Golang.....	210
	Приложение Г Поиск альтернативных маршрутов двухсторонним A* на Golang.....	212
	Приложение Д Расчет веса параметров методом АНР на Golang.....	218
	Приложение Е Расчет альтернатив техникой TOPSIS на Golang	223
	Приложение Ж Расчет коэффициентов влияния когнитивной модели на Golang.....	226
	Приложение И Комбинационная матрица парных сравнений подкритериев «Безопасность»	230
	Приложение К Комбинационная матрица парных сравнений основных критериев	232
	Приложение Л Коэффициенты влияния критериев когнитивной модели	234
	Приложение М Свидетельство о государственной регистрации программы для ЭВМ	237

Введение

Актуальность темы исследования. Современные системы поддержки принятия решений в транспортных сетях – это информационные средства, функционирующие на принципах теории графов, машинного обучения и многокритериальной оценки, предназначенные для решения задач прогнозирования загруженности, обеспечения безопасности и выбора рациональных маршрутов. Проектирование алгоритмов маршрутизации строится как ряд итеративных процедур формирования промежуточных гипотетических вариантов моделей сетевых структур и требует оценки большого количества критериев эффективности. По причине длительности сроков создания прогнозных моделей при их проектировании необходимо учитывать: характеристики транспортных потоков с учетом погодных условий, возможность появления новой элементной базы и новых конструкторских разработок и технологий, которые позволяют повысить точность прогнозирования. Однако использование новых конструктивно-технических решений в процессе проектирования моделей транспортных сетей может привести к повышению рисков несвоевременного создания или невыполнения требований к точности и скорости обработки данных. В частности, при проектировании алгоритмов поиска маршрутов оценка выполнимости всех требований к безопасности и загруженности может стать неразрешимой задачей, так как для выполнения каждой характеристики требуется определенный ресурс вычислительных мощностей и данных, поэтому уже на первых стадиях жизненного цикла разработки программного комплекса становится важным решение вопроса повышения эффективности проектирования моделей и выбора совокупности наиболее важных критериев, выполнение которых с наименьшими рисками обеспечит требуемое качество поддержки принятия решений.

Указанные обстоятельства отражают наличие практического противоречия в необходимости учитывать – время на перемещение,

загруженность транспортной сети и безопасность маршрута. Существующие модели ориентированы на использование только загруженности, таким образом повышая риски несвоевременного создания маршрутов и выбора нерационального варианта. Это определяет важное практическое значение исследований, направленных на повышение эффективности поддержки принятия решений, и обосновывает актуальность исследования.

Степень разработанности темы исследования. Проблемам управления процессами проектирования и создания сложных технических систем, таких как транспортные сети, основанных на поэтапном контроле характеристик отдельных компонент и комплексной оценке эффективности процесса на протяжении всего жизненного цикла, посвящены работы М. Кастельса, М.А. Айзермана, А.В. Олескина, Б. Кернера, В.Л. Арлазарова, Н.Е. Емельянова, В.В. Подиновского, В.Д. Ногина, А.В. Лотова, И.И. Пospelовой, А.А. Морозова, Е.О. Черноусовой, В.А. Судакова и других авторов.

Вопросы математического моделирования транспортных сетей исследовали М. Лайтхилл, Дж. Уизем, П. Ричардс, А. Решель, Л. Пайп, А.В. Гасников, С.Л. Кленов, Е.А. Нурминский, Я.А. Холодов, Н.Б. Шамрай, В.И. Швецов, А.А. Петров, Ю.В. Шокин, А.Ю. Замятин, С.Е. Семенов, В.В. Буслаев, М.Ю. Шокин, М.В. Яшина, А.Г. Таташев и другие.

Критерии, как показатели качества и эффективности альтернатив (решений), исследовали такие ученые, как О.И. Ларичев, Э.А. Трахтенгерц и другие.

Проблемы влияния поведенческого фактора в сетевых структурах (водитель в потоке) исследовали А. Рейн, Р.В. Шкрабак, Н.У. Гюлев, В.Н. Басков, Д.О. Кожин, Д.Е. Алекминский, В.В. Евграшин, Ю.Н. Баранов, А.В. Олескин, М.М. Чучкевич, Н.Н. Марфенин и другие.

Существующие модели и методы управления транспортных сетей основаны на принципах всестороннего анализа и синтеза проектных решений. В работах классических алгоритмов маршрутизации в качестве

универсального показателя эффективности маршрутов рассматривается время поиска кратчайшего пути. Оценка рисков позволяет прогнозировать вероятность нахождения оптимального маршрута и минимизировать вычислительную сложность. Однако определение оптимального варианта состава критериев оценки рассматривается только с точки зрения готовности составных частей к применению в процессе разработки алгоритмов, не учитывая достигаемые показатели безопасности и загруженности, что не позволяет однозначно судить о выполнении требований к комплексной оценке маршрутов. Отмеченный недостаток проявляется в том, что в случае неверной оценки технического облика модели возможно увеличение сроков создания эффективного программного комплекса.

В работах многокритериального анализа отмечается, что при проектировании транспортных систем следует максимально возможно сокращать количество характеристик, влияющих на принятие решений о правильности проектирования. Результаты исследований, представленных в этих работах, показывают, что такие показатели качества проектирования как относительная близость к идеальному решению характеризуют поисковые возможности моделей и позволяют оценить выполнение основных требований заказчиков, что открывает возможности для совершенствования методов проектирования сетевых структур за счет включения предпочтений лица, принимающего решения (далее – ЛПР). Однако непосредственное использование методов и средств, представленных в вышеуказанных работах, применительно к решению задачи обоснования конструктивно-технических решений маршрутизации затруднительно из-за необходимости учета разнородных факторов (погодные условия, безопасность, загруженность) и вследствие ограниченности данных для машинного обучения.

Данное обстоятельство позволяет сформулировать противоречие в науке, которое заключается, с одной стороны, в необходимости учета множества критериев для повышения точности прогнозирования, и

отсутствии универсальных методов их комплексной оценки в условиях неполных данных.

Указанное противоречие позволяет сформулировать научную гипотезу, которая заключается в разработке ансамбля моделей, включающего когнитивные методы и нейронные сети для повышения эффективности многокритериальной оценки сетевых структур. Выполненный анализ позволил определить объект и предмет исследования.

Выявленное практическое противоречие определяет **цель работы** – повышение эффективности маршрутизации транспортных средств за счет повышения точности прогнозирования и комплексного учета факторов: нагрузки на сетевую структуру, погодных факторов, факторов безопасности движения, рассчитанных по результатам моделирования и объединенных в единый комплексный показатель эффективности маршрута.

На основании вышеизложенного и в соответствии с поставленной целью, решаемая научная задача заключается в разработке метода многокритериальной оценки моделей сетевых структур на основе сходства с идеальным решением. Декомпозиция поставленной научной задачи приводит к необходимости проанализировать современные подходы, методы моделирования и прогнозирования в сетевых структурах на примере транспортной сети и подготовить решения следующих частных **задач**:

- 1) разработать метод многокритериальной оценки сетевых структур на основе комплекса моделей дорожного движения, включающий: а) модель многокритериальной оценки комплексной эффективности маршрута; б) модель загруженности сетевой структуры с учетом погодных условий; в) модель безопасности маршрута, рассчитанную по разнородным показателям; г) когнитивную модель расчета показателей оценки маршрута на основе фреймово-факторного графа экспертных оценок;
- 2) разработать численный метод поиска альтернативных маршрутов для увеличения скорости нахождения рационального маршрута;

3) разработать комплекс программ для оценки и выбора маршрута, использующий большие данные от навигационных приемников.

Объектом исследования являются математические модели и численные методы сбора, расчета, анализа и прогнозирования значений критериев, влияющих на транспортную сетевую структуру. В работе рассматриваются объективные и экспертные критерии. Объективные могут быть рассчитаны на основе статистических данных, а экспертные – на основе мнения экспертов в области транспортного планирования.

Предметом исследования является процесс принятия решений с использованием многокритериальной оценки результатов моделирования на основе моделей для поиска рационального маршрута и когнитивных моделей расчета критериев оценки маршрута.

Область исследования. Диссертация соответствует пунктам 2. «Разработка, обоснование и тестирование эффективных вычислительных методов с применением современных компьютерных технологий», 8. «Комплексные исследования научных и технических проблем с применением современной технологии математического моделирования и вычислительного эксперимента», 9. «Постановка и проведение численных экспериментов, статистический анализ их результатов, в том числе с применением современных компьютерных технологий» Паспорта научной специальности 1.2.2. Математическое моделирование, численные методы и комплексы программ (технические науки).

Научная новизна результатов исследования. На основании проведенных исследований и экспериментов были получены новые научные результаты:

- разработан метод многокритериальной оценки сетевых структур на основе комплекса моделей дорожного движения, включающий: а) модель многокритериальной оценки комплексной эффективности маршрута, которая отличается от известных тем, что: учитывает предпочтения ЛПР, при этом предпочтения обеспечивают расчёт комплексного показателя эффективности

маршрута, что позволяет построить удовлетворяющий ЛПР маршрут; для расчета эффективности маршрутизации используется комплексный показатель, который показывает, насколько найденный методикой маршрут близок к идеальному маршруту (маршруту, у которого по всем критериям оценки маршрута указаны наилучшие значения). Близость к идеальному маршруту вычисляется с помощью евклидовой метрики в пространстве критериев оценки качества маршрута. В качестве критериев оценки качества маршрута используются следующие критерии: расстояние; время в пути; стоимость проезда; загруженность маршрута; погодные условия; безопасность; скоростные характеристики; геометрия маршрута; качество дорожного покрытия; б) модель загруженности сетевой структуры с учетом погодных условий, отличающаяся от известных тем, что за счет выбора архитектуры и гиперпараметров искусственных нейронных сетей – Long-Short Term Memory и Weighted Self-Attention (далее – LSTM + WSA), включая выбор количества слоев, размера пакета, размера скрытого слоя, количества эпох обучения, скорости обучения, удалось повысить точность прогнозирования загруженности маршрута на 5-7% в сравнении с классической LSTM; в) модель безопасности маршрута, отличающаяся от известных тем, что: показатель безопасности определяется по разнородным показателям, которые агрегируются с использованием методов теории принятия решений. Для из расчета используются открытые данные о состоянии дорожного покрытия, средствах регулирования движения (светофоры, переходные переходы), средствах фиксации нарушений (видеокамеры автоматической фиксации нарушений, радары), дорожно-транспортных происшествиях; используются экспертные оценки в форме парных сравнений приоритетов показателей безопасности, что отличается от подхода экспертного сравнения конкретных альтернативных маршрутов такое отличие позволяет произвести настройку модели один раз и использовать ее для оценки безопасности произвольного числа маршрутов; г) когнитивная модель расчета показателей оценки маршрута на основе многокритериальной оценки моделей сетевых структур

для увеличения вероятности нахождения оптимального маршрута, отличающаяся от известных моделей тем, что: использована фреймово-факторная модель экспертных оценок; данная модель позволяет приближенно оценить сетевую структуру в случае недостаточности эмпирических данных для машинного обучения нейросетевых моделей;

- разработан численный метод поиска альтернативных маршрутов, отличающийся от известных методов A^* тем, что: а) применена буферизация данных о структуре улично-дорожной сети для двух множеств вершин: первого множества вершин, просматриваемых на текущей итерации при движении от начальной вершины к конечной, второго множества вершин, просматриваемых на текущей итерации при движении от конечной вершины к начальной, таким образом минимизировано количество обращений к графу на диске; б) введен усовершенствованный критерий останова алгоритма для данного численного метода: алгоритм продолжает перебирать вершины из множеств после первого их пересечения, что повышает вероятность нахождения оптимального маршрута. В результате численный метод обеспечил увеличение скорости маршрутизации на 30% по сравнению с базовым методом A^* ;

- разработан комплекс программ для оценки и выбора маршрута, использующий большие данные от навигационных приемников отличающийся от наиболее распространённых навигаторов компаний Яндекс, 2GIS, Google Maps тем, что: а) маршрутизация происходит с учетом предпочтений лица принимающего решения по показателям загруженности и безопасности маршрута; б) опрос, проведенный по результатам тестирования, показал, что более 70% респондентов признали результаты работы комплекса программ приемлемыми, что на 10% выше уровня удовлетворенности другими техническими решениями, представленными на рынке коммерческих программных продуктов.

Полученные результаты увеличили: точность прогнозирования загруженности маршрута на 5-7%; скорость маршрутизации на 30%; рост

удовлетворенности ЛПР на 10%, а также обеспечили достижение цели диссертационного исследования по повышению эффективности маршрутизации транспортных средств.

Теоретическая значимость работы состоит в том, что, в отличие от результатов, полученных другими исследователями, разработанные модели позволяют учесть актуальные разнородные критерии, влияющие на результат ранжирования. Разработан метод многокритериальной оценки сетевых структур на основе сходства с идеальным решением. Создана когнитивная фреймово-факторная модель определения значений критериев в случае недостаточного объема статистических данных. Разработанные модели могут служить базой для постановки более широких, в том числе теоретико-игровых, задач анализа взаимодействия элементов сетевых структур.

Практическая значимость работы заключается в широкой применимости сформулированных положений ансамбля моделей сетевых структур в качестве инструмента помощи при принятии решений. На основе разработанной модели возможно моделировать кризисные ситуации, оценить рациональность маршрута при различных значениях и весах критериев в модели для многокритериальной оценки сетевых структур.

Методологию и методы исследования составили результаты трудов российских и зарубежных авторов, посвященные моделированию сетевых структур на основе транспортной сети, научные работы в области многокритериального анализа альтернатив, исследования математических моделей, описывающих влияние сложных факторов на сеть и ее участников, и влияние поведенческого фактора в сетевых структурах. В работе применялись методы научной абстракции, математического моделирования, машинного и глубокого обучения, теория множеств, теория графов, теория принятия решений, объектно-ориентированное и функциональное программирование.

Информационную базу составили анонимные GPS-треки пользователей различных устройств, аналитические отчеты и статистические данные о чрезвычайных ситуациях на дорогах РФ, научные исследования

российских и зарубежных ученых, а также материалы конференций, опубликованные в периодической печати и в сети Интернет.

Положения, выносимые на защиту. В соответствии с поставленными задачами, положения, выносимые на защиту, включают в себя следующее:

1) метод многокритериальной оценки сетевых структур на основе следующего комплекса моделей: а) модель многокритериальной оценки комплексной эффективности маршрута (С. 54-60); б) модель загруженности сетевой структуры с учетом погодных условий повысившую точность прогнозирования на 5-7% (С. 60-76); в) модель безопасности маршрута (С. 76-85); г) когнитивная модель расчета показателей оценки маршрута на основе многокритериальной оценки моделей сетевых структур и фреймово-факторного графа экспертных оценок (С. 85-89);

2) численный метод поиска альтернативных маршрутов, ускоряющий поиск маршрута на 30% (С. 111-119);

3) комплекс программ для оценки и выбора маршрута, использующий большие данные от навигационных приемников и одобренный более 70% респондентов после тестирования (С. 121-126).

Степень достоверности, апробация и внедрение результатов исследования. Достоверность положений исследования подтверждается их апробацией в установленном порядке, применением качественных и количественных методов оценки методов оценки рассматриваемых проблем, изучением научных публикаций отечественных и зарубежных авторов по тематике исследования.

Основные положения и результаты научного исследования докладывались и получили одобрение на международных и всероссийских научных мероприятиях: на 19-й Международной конференции «Авиация и космонавтика» (Москва, Московский авиационный институт, 23-27 ноября 2020 г.); на X Международной научно-практической конференции «Абалкинские чтения» (Москва, РЭУ имени Г.В. Плеханова, 26-27 апреля 2021 г.); на XII Всероссийской научно-практической

конференции «Высокие технологии, наука и образование: актуальные вопросы, достижения и инновации» (г. Пенза, МЦНС «Наука и Просвещение», 7 сентября 2021 г.); на IV Всероссийской научно-практической конференции «Цифровая трансформация управления: проблемы и решения» (Москва, Государственный университет управления, 12 мая 2022 г.); на XII Международной научно-практической конференции «Абалкинские чтения» (Москва, РЭУ имени Г.В. Плеханова, 26-27 апреля 2023 г.); на 15th International KES Conference On Intelligent Decision Technologies (г. Рим, Италия, Springer Singapore, 14-16 июня 2023 г.).

Программа для ЭВМ «Программа поддержки принятия финансовых решений на основе сетевых структур», подготовленная по результатам исследования, внесена в Реестр программ для ЭВМ Роспатента (свидетельство о регистрации Роспатента № 2024615278 от 05.03.2024; автор и правообладатель: Ехлаков Р.С.).

Материалы исследования используются в практической деятельности ООО «ВК», в частности внедрены разработанные модели оценки и прогнозирования загруженности транспортной сети, позволяющие существенно улучшить качество ответов API сервисов. Реализованы отдельные модули поиска маршрута при помощи алгоритма двустороннего A*. Выводы и основные положения диссертации реализованы в виде модулей веб-приложения для решения логистических задач любого уровня сложности с использованием актуальной информации о пробках и находятся на этапе бета-тестирования.

Материалы исследования используются Кафедрой анализа данных и машинного обучения Факультета информационных технологий и анализа больших данных ФГОБУ ВО «Финансовый университет при Правительстве Российской Федерации» в преподавании учебной дисциплины «Практикум по программированию». Результаты используются при проведении научно-исследовательских семинаров и выполнении самостоятельной работы.

Апробация и внедрение результатов исследования подтверждены соответствующими документами.

Публикации. Основные результаты и положения диссертации изложены в 9 научных работах общим объемом 7,52 п.л. (авторский объем – 6,35 п.л.), в том числе 3 работы общим объемом 2,23 п.л. (авторский объем – 2,05 п.л.) опубликованы в рецензируемых научных изданиях, определенных ВАК при Минобрнауки России, а также 4 работы общим объемом 3,04 п.л. (авторский объем – 2,5 п.л.) опубликованы в изданиях, включенных в международную цитатно-аналитическую базу Scopus. Получено одно свидетельство на программу для ЭВМ.

Личный вклад автора. В работах [44-49; 107-110] автору принадлежит постановка задачи, создание математических моделей, разработка комплекса программ, проведение вычислительных экспериментов.

Структура и объем диссертации. Общая структура работы включает: введение, четыре главы, заключение, список сокращений и условных обозначений, словарь терминов, список литературы из 214 наименований и 11 приложений. Текст диссертации изложен на 237 страницах, включает 29 таблиц, 46 рисунков, 65 формул.

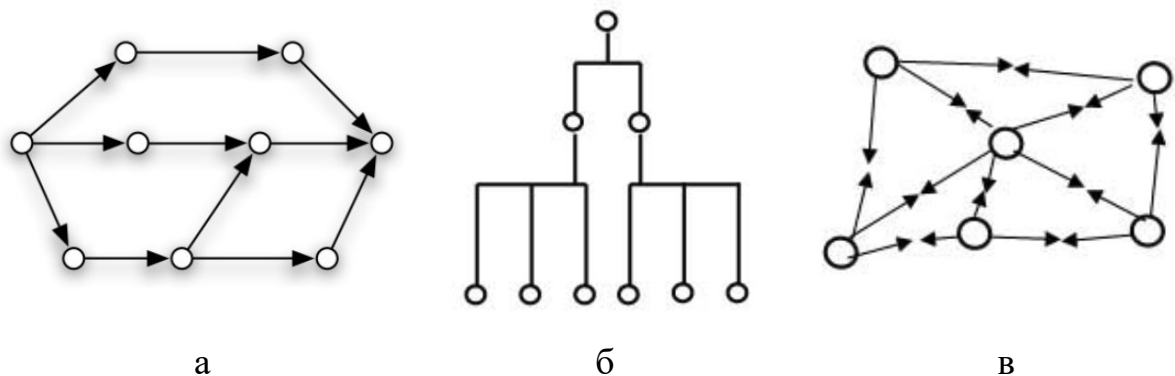
Глава 1

Анализ современных моделей транспортной сети

1.1 Транспортная сеть

Проблема моделирования процессов сетевых структур исследована не в полной мере, попыткой продвижения на пути к ее решению следует считать и данную работу. Сформулированная проблема относится к области графодинамики – направлению в теории управления, в котором значениями переменных являются графы. Фундаментальная работа в области аналитической графодинамики [38], в которой и введен термин «графодинамика», была выполнена под руководством профессора М.А. Айзермана и опубликована более 30 лет назад. Опираясь на целый ряд классических работ и публикаций последних лет, сетевые структуры представлены на рисунке 1.1 и делятся на:

- 1) классические (децентрализованные) структуры, не имеющие выраженного лидера и способные к горизонтальному расширению сети;
- 2) иерархические (вертикальными) структуры с единым управляющий центр;
- 3) квазирыночные структуры, в которых конкуренция между элементами преобладает над кооперацией [38].



а – классическая; б – иерархическая; в – квазирыночная.

Источник: [38].

Рисунок 1.1 – Сетевая структура

Одна из самых известных теорий была разработана в 1990-х годах американским ученым М. Кастельсом и изложена в книгах «Информационная эпоха: экономика, общество и культура» [7], где используется понятие «сетевое общество», чтобы, с одной стороны, показать определяющую роль компьютерных сетей в развитии современного социума, с другой стороны, чтобы обосновать, что развитие современных информационно-коммуникационных технологий ведет к изменению общественных отношений. Сети бывают разные: простые и сложные. Наиболее распространены и удобны для анализа однонаправленные сети, но могут быть и сети с обратными связями, циклами. Примерами сетевых структур являются современные информационные сети, в особенности созданные на базе Интернета. Сетевые структуры интернета переходят в структуры организационных типов – иерархические (во многом в силу воздействия государственного аппарата, полицейских инстанций, образовательных учреждений) и рыночные (в результате коммерциализации) [37; 39].

«Техника, технологии и экономика транспорта сегодня наполнены идеями сетевых структур и цифровых компонент с позиции информатизации управления процессами перевозок и финансового сопровождения договорно-правовых отношений» [40]. Аналогии, привнесенные в теорию экономических учений, классическую транспортную науку и другие научные направления, по сути, определили сегодняшнее состояние и перспективы развития перечисленных направлений исследований. Именно они послужили основой для развития сетевой экономики и управления цепями поставок. Создание интеллектуальных транспортных систем сегодня одна из ключевых тенденций в управлении процессами перевозок, динамикой потоков транспортных средств, спросом на услуги транспорта и в целом всем комплексом операций по доставке в цепях поставок. Подобные технологии требуют разработки современных подходов к организации и планированию при условии смещения акцентов от автоматизации выполняемых операций к автоматизации управления, включая все этапы жизненного цикла.

«В настоящее время много литературы посвящено изучению и моделированию транспортных потоков» [11]. Одной из важных современных теорий, по которым зачастую оцениваются модели транспортных потоков, является эмпирическая теория трёх фаз, разработанная Б. Кернером. Она рассматривает переход от свободного потока к плотному и возникающих в результате структурах в интенсивном потоке на скоростных магистралях. Менее требовательная к качеству предоставленных данных модель Нестерова-Пальмы, предложенная в 1998 году, также известная под названием модель стационарной динамики.

Р. Динг и другие [95] применили теорию сложных сетей к распространению риска фактора перегруженности городского транспорта. Они предложили концепцию важности сетевых вершин, основанную на сложных сетевых метриках, для классификации сетевых вершин и осуществления контроля над основными вершинами. Выявлено, что данный метод позволяет снизить степень риска фактора перегруженности дорожного движения.

Я. Янг и другие [190] применили теорию сложных сетей к идентификации критических вершин и интегрировали степень и коэффициенты кластеризации для идентификации критических вершин и расчета их веса с использованием методов энтропии. Результаты экспериментов показали, что интеллектуальный транспорт может уменьшить заторы на дорогах и повысить эффективность за счет управления прогнозирования транспортных потоков в режиме реального времени. С помощью различных датчиков возможно получить данные о трафике, которые затем могут использоваться в моделях прогнозирования.

Д. Танг и другие [172] использовали алгоритм Лемпеля-Зива для оценки сложности данных о транспортных потоках в различных временных масштабах. Сложные сети анализировались в соответствии с различными статистическими свойствами, такими как средняя длина пути и коэффициент кластеризации.

Результаты приведенных работ показывают, что сложные сети являются практическим инструментом для анализа характеристик динамического изменения временных рядов транспортной сети.

1.2 Многокритериальные методы оценки

В последние годы стали возможными новые методы оптимизации сетевых структур, позволяющие находить рациональные решения на основе большого количества данных. Существует несколько направлений оптимизации транспортных маршрутов, среди которых:

1) алгоритмы глубокого обучения – алгоритмы анализируют большие объемы данных и прогнозируют оптимальные маршруты, учитывая множество факторов, таких как дорожная обстановка, погодные условия и трафик;

2) генетические алгоритмы – алгоритмы имитируют процесс естественного отбора, чтобы найти наилучшие решения для определенной проблемы. Они используют операторы, такие как скрещивание и мутация, для поиска оптимальных маршрутов;

3) методы кластеризации – методы группируют данные в различные кластеры, что позволяет находить эффективные маршруты для каждого транспортного средства в каждом кластере. Это особенно полезно для распределения транспортных средств по различным зонам обслуживания;

4) эволюционные алгоритмы – подобно генетическим алгоритмам, эти методы используют принципы естественного отбора и мутации для поиска наилучшего решения, но могут включать дополнительные механизмы, такие как коэволюция и адаптивные стратегии;

5) методы динамического программирования – методы рассматривают все возможные комбинации маршрутов и выбирают рациональный маршрут для каждого транспортного средства. Динамическое программирование позволяет эффективно решать задачи оптимизации путем разбиения их на подзадачи;

б) методы мультиагентного моделирования – методы используют агентов, представляющих различные транспортные средства, для нахождения рациональных маршрутов в режиме реального времени. Агенты взаимодействуют друг с другом и окружающей средой, что позволяет адаптироваться к изменяющимся условиям на дорогах [13].

Известными примерами использования алгоритмов оптимизации являются следующие компании:

1) платформы такси Uber, Lyft, Gett, Яндекс.Такси и другие используют искусственный интеллект для поиска кратчайшего маршрута, динамического ценообразования и поиска ближайшего свободного водителя в каждой поездке;

2) логистические компании DHL, FedEx и другие используют методы математического моделирования, чтобы анализировать данные о грузе, транспорте, маршрутах и условиях доставки для оптимизации издержек на маршруте транспортного средства, что позволяет сократить время доставки и уменьшить стоимость транспортировки;

3) производитель электромобилей Tesla использует данные о расстоянии до ближайшей станции зарядки, скорости движения, времени суток и других параметрах, чтобы предложить рациональный маршрут зарядки для каждого электромобиля;

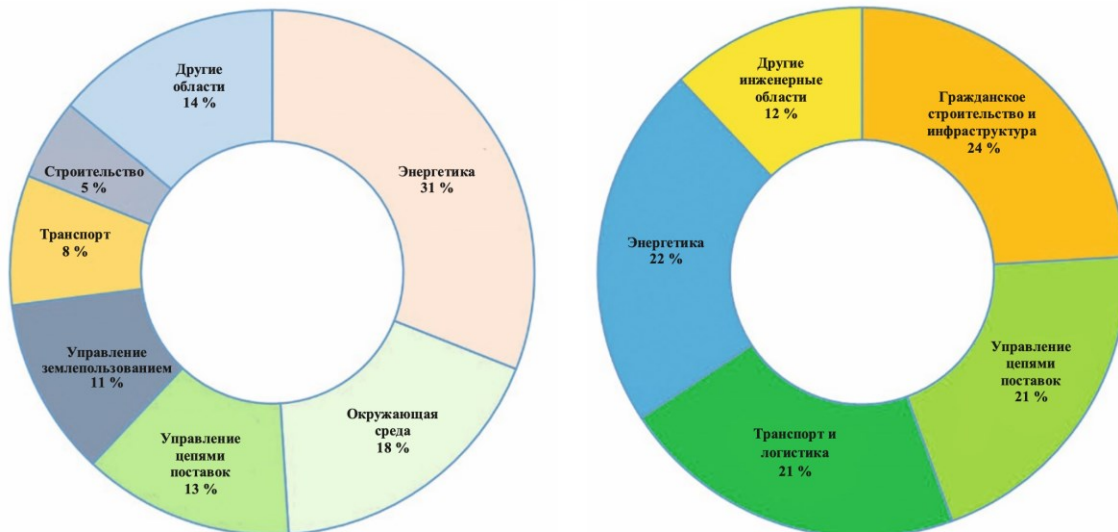
4) городские транспортные компании используют численные методы для оптимизации маршрутов наземного транспорта с учетом таких факторов, как: пробки, пассажиропоток и график движения.

Сложность принятия решений заключается в учете большого числа критериев оценки; в участии стейкхолдеров, преследующих различные цели; в наличии широкого разнообразия управленческих решений разного уровня и сложности расчетов. Необходимость учета перечисленных факторов привела к развитию научного направления в области многокритериального принятия решений – Multi-Criteria Decision Making (далее – MCDM). Методы MCDM

являются важной частью теории принятия решений и предназначены для решения четырех типов задач:

- 1) выбор наилучшего решения из множества альтернатив – определение оптимального варианта из доступных опций на основе множества критериев;
- 2) ранжирование и сортировка альтернатив – упорядочивание альтернатив по степени предпочтительности или эффективности;
- 3) описание и систематизация решений – анализ возможных решений и их систематизация для более глубокого понимания и структурирования информации;
- 4) оценка и управление последствиями реализации решений – анализ последствий принятых решений для их эффективного управления и корректировки в будущем.

На рисунке 1.2 проиллюстрирована доля применения методов MCDM в различных сферах.



Источник: составлено автором.
Рисунок 1.2 – Использование MCDM методов

Методы MCDM делятся на две категории: многоцелевое принятие решений (далее – MODM) и принятие решений по множеству атрибутов (далее – MADM). Модели MODM включают большое количество альтернативных решений, а цель рассмотрения проблемы состоит в

определении оптимальной альтернативы с учетом набора четко определенных ограничений путем решения математической модели. Модели MADM являются дискретными и применяются для ранжирования, когда конечное число предложенных альтернатив оценивается по отношению к различным атрибутам для получения рейтинга предпочтений, который описывает эффективность каждой альтернативы для достижения цели в отношении атрибутов. Среди недостатков применения MCDM моделей выделяют сложность сбора исходной информации и повышение трудоемкости процесса принятия решения. В работах различных авторов отмечается, что отсутствуют универсальные MCDM методы, подходящие для всех ситуаций принятия решений, что приводит к проблеме выбора. Наиболее часто используемыми MCDM методами являются:

- 1) АНР – аналитический иерархический процесс;
- 2) АНР – аналитический сетевой процесс;
- 3) TOPSIS – метод упорядоченного предпочтения по сходству с идеальным решением;
- 4) DEMATEL – метод испытаний и оценки принятия решений;
- 5) ELECTRE – исключение и выбор в условиях реальности;
- 6) PROMETHEE – метод организации сортировки предпочтений для оценок альтернатив;
- 7) VIKOR – многокритериальная оптимизация и компромиссное решение.

Процесс иерархического анализа (далее – АНР) – метод решения проблем многокритериального анализа, в котором каждая проблема проходит процесс деструктуризации и иерархического повторного расположения основных компонентов. Используется для создания иерархии вложенных альтернатив, в которой каждая задача раскладывается на ряд подчиненных. Иерархическая структура должна отвечать определенным целям. Альтернативы представляют собой абстракцию структуры анализируемой системы, которая позволяет лучше понять функциональные отношения между

компонентами и влияние на систему каждой из них. Процесс анализа не требует числовых значений для связывания критериев в альтернативу, представляющую какой-либо из аспектов проблемы, и позволяет использовать субъективные суждения, неподдающиеся измерениям для установления относительной важности критериев в соответствии с общей целью. Такие субъективные суждения используются для количественного определения относительной важности критериев при принятии решений и выборе вариантов. Одним из важных аспектов является определение веса, отражающих относительную важность каждой альтернативы в задаче решения с несколькими атрибутами. Т. Саати ввел использование матрицы парного сравнения (далее – МПС), которая позволяет лицу, принимающему решения, назначать сравнительное значение каждой из пар критериев, в результате чего матрица, содержащая сравнительные значения каждой пары критериев, преобразуется в вербально-числовую со значениями от 1 до 9. Характерная особенность АНР заключается в том, что он позволяет сопоставлять выбранные критерии и строить матрицу сравнений. Тем не менее, метод редко применяется при наличии большого набора критериев. Так, при 24 критериях матрица будет включать по 24 столбца и строки, тогда как обычно их число не превышает 10. Последние расширения метода АНР направлены на получение более надежных результатов на основе знаний или предпочтений лица, принимающего решения, при одновременной оптимизации распределения веса парных сравнений.

Техника порядка предпочтения по сходству с идеальным решением (далее – TOPSIS) – компенсационная методика сравнения набора альтернатив путем определения веса и нормализации оценок каждого критерия и вычисления геометрического расстояния между каждой альтернативой и идеальной альтернативой, которая является лучшей оценкой по каждому критерию, описанная следующими шагами:

- 1) определение набора альтернатив;
- 2) нормализация;

- 3) взвешивание;
- 4) определение положительного идеального решения;
- 5) определение отрицательного идеального решения;
- 6) расчет расстояния до идеальной точки;
- 7) расчет расстояния до анти-идеальной точки;
- 8) определение коэффициента подобия.

В результате будут получены значения в интервале между 0 и 1, из которых ближайшие к 1, будут приняты в качестве репрезентативных при выборе альтернативы.

Аналитический сетевой процесс (далее – ANP) разрешает конфликты, помогает учитывать циклические взаимосвязи между критериями.

Метод ELECTRE (устранение и выбор, выражающий реальность), основанный на концепциях согласия и несогласия и демократическом принципе большинства без сильного меньшинства.

Основная идея метода организации ранжирования предпочтений для обогащения оценок (далее – PROMETHEE) заключается в создании частичной или полной классификации альтернатив в соответствии с положительным потоком выигрыша, отрицательным потоком выигрыша и чистым потоком выигрыша.

По результатам обзора исследований в области многокритериальной оценки параметров дорожно-транспортной сети были проанализированы 58 работ. Результаты распределения по области применения приведены в таблице 1.1. Подробный анализ приведен в приложении А.

Таблица 1.1 – Распределение статей по областям применений

Область применения	Количество публикаций	Доля, в процентах
1	2	3
Автомобильная транспортная сеть	12	20,69
Общественная транспортная сеть	11	18,97
Логистика	10	17,24

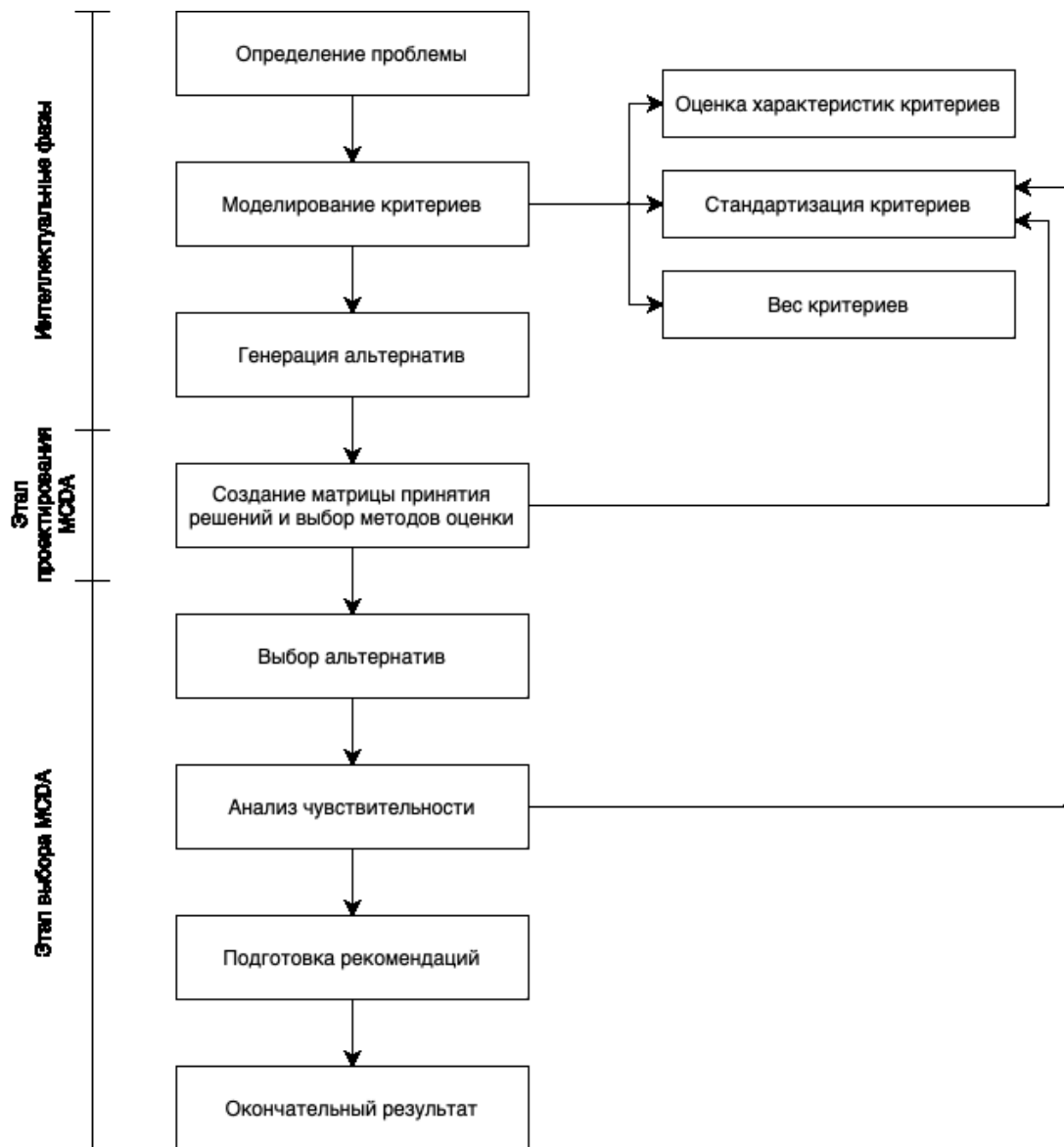
Продолжение таблицы 1.1

1	2	3
Воздушная транспортная сеть	8	13,79
Морская транспортная сеть	6	10,34
Железнодорожная транспортная сеть	4	6,90
Планирование транспортной сети	4	6,90
Транспортные операции	2	3,45
Пешеходная транспортная сеть	1	1,72
Итого	58	100

Источник: составлено автором по материалам [65-66; 68-71; 75-76; 78-82; 91; 93; 96-99; 101; 113; 119; 121; 123, 125-128; 131-132; 137; 140-141; 143; 145; 147-148; 150-151; 154-156; 162-163; 165-166; 178-179; 182; 183; 185-188; 191-193].

По этой причине метод многокритериального анализа выбран за основу диссертационного исследования и используется с учетом параметров системы, влияющих на результат. В научной литературе существуют различные методы классификации многокритериальных решений. Среди наиболее известных авторов выделяют Вилкас Э.Й. [4], Майминас Е.З. [5], Ларичев О.И. [8], Саати Т [10]. «Одной из критических проблем в использовании многокритериальных методов является необходимость присвоения веса для каждого критерия, что в большинстве случаев субъективно» [31]. Информация, полученная с помощью технических устройств, считается более надежной, чем оцененная и интерполированная экспертами информация. Таким образом, важную роль в решении играет автоматизированный способ обработки информации, чтобы исключить субъективный и человеческий фактор при поиске альтернатив. Для минимизации субъективности в современных исследованиях активно применяются алгоритмы машинного обучения, способные анализировать большие массивы данных и выявлять скрытые зависимости между критериями. Кроме того, использование гибридных методов, сочетающих MCDA с нейронными сетями или генетическими алгоритмами, позволяет повысить точность ранжирования альтернатив за счет адаптивной настройки весовых коэффициентов, что

особенно актуально в условиях неопределенности, когда традиционные экспертные оценки могут приводить к противоречивым результатам. Процесс интеграции MCDA метода изображен на рисунке 1.3.



Источник: составлено автором.

Рисунок 1.3 – Процесс интеграции MCDA

Для определения критериев, влияющих на характеристики поездки, поиск альтернатив и выбор рационального маршрута были подробно изучены как отечественные, так и зарубежные исследования транспортной сети во всем мире. Критерии и частота их использования в исследованиях описаны в таблице 1.2. Наиболее часто встречающимися критериями являются: расстояние, загруженность, геометрические характеристики транспортной сети и время в пути.

Таблица 1.2 – Наиболее часто встречающиеся в исследованиях параметры, влияющие на выбор маршрута

Критерий	Источник								
	[150]	[77]	[153]	[165]	[106]	[86]	[116]	[82]	[108]
Расстояние	нет	да	нет	да	нет	нет	да	нет	нет
Геометрические характеристики	да	нет	нет	да	нет	нет	нет	да	нет
Потребляемая энергия	нет	нет	да	нет	нет	да	нет	нет	нет
Качество дорожного покрытия	да	нет	нет	нет	да	нет	нет	нет	нет
Количество пассажиров	нет	нет	нет	нет	да	нет	нет	нет	нет
Время в пути	нет	да	нет	да	нет	нет	да	нет	да
Стоимость проезда	нет	нет	нет	нет	нет	нет	нет	да	нет
Загруженность	нет	да	нет	да	нет	нет	нет	да	нет
Наличие актуальных данных	да	нет	нет	нет	да	нет	да	нет	нет
Тип дороги	да	нет	нет	нет	да	нет	нет	нет	нет
Погодные условия	нет	нет	да	нет	нет	нет	нет	нет	да
Экологичность	нет	нет	нет	нет	нет	да	нет	нет	нет
Количество остановок	нет	нет	нет	нет	нет	нет	нет	да	нет
Безопасность	нет	нет	да	нет	нет	нет	нет	нет	нет

Источник: составлено автором по материалам [77; 82; 86; 106; 108; 116; 150; 153; 165].

Согласно совместному исследованию ВШЭ и ЦОДД по влиянию параметров транспортных потоков на дорогах Москвы: «Были рассчитаны средние значения скоростей движения в целом по сети за определенные интервалы времени в сутках, при этом усреднение параметров проводилось за месяц, полугодие и год. В отличие от множества параметров, влияющих на условия движения в городе, таких как интенсивность движения, пропускная способность дороги, потерянное время на перекрестках, длины очередей в заторах, средняя скорость транспортного потока является универсальным инструментом оценки работы дорожной сети» [20]. Значения скоростных параметров объединяют в себе множество факторов, которые влияют на движение автомобилей по городу, включая организацию дорожного движения, загрузку участков улично-дорожной сети, погодные условия, ремонтные работы, изменение транспортной инфраструктуры. Вследствие чего скоростные параметры были выбраны как индикатор качества движения автомобилей по сети и динамики изменения транспортной ситуации на

длительный период, например, сравнение среднемесячных значений параметров с шагом в один год. В настоящее время на рынке контента о загрузке улично-дорожной сети коммерческие компании (например, Яндекс, 2ГИС) используют бальную систему оценки. Как правило применяется 10 бальная шкала, которая дает качественную оценку состояния транспортной системы для конкретного часа суток, начиная от свободного движения (0 баллов) до значительных задержек в пути из-за множественных заторов на участках сети (10 баллов). При этом назначение «стоимости» каждого балла для каждой компании является субъективной процедурой, которая зависит от внутренних алгоритмов расчета скорости, выбора эталонных маршрутов, фильтрации данных и оценки границ значений параметров. Таким образом, назначенные 10 баллов могут в реальности означать как практически полную остановку движения (менее 5 км/час), так и движение (20-40 км/час) по магистралям, причем оценка одной и той же транспортной ситуации в каждой компании может различаться. Важно отметить, что бальный подход является хорошим и востребованным инструментом предоставления информационных услуг об общем состоянии транспортной сети, который оказывает помощь в принятии решений перед началом поездки. Помимо прочего, на скорость движения в городе также влияет фактор погоды. Для участников движения характерно снижение скорости во время сильных дождей и/или снегопадов, данный факт закономерен и соответствует практике всех без исключения городов мира: помимо физически обусловленных причин, таких как сцепление автомобиля с дорожным полотном, водитель также снижает скорость, так как сокращается расстояние видимости и увеличивается время реакции, что неминуемо приводит к увеличению дистанции между двумя транспортными средствами, находящимися в движении.

Среди известных научных работ таких авторов, как В.Л. Арлазаров, Н.Е. Емельянов «Информационно-аналитические аспекты в задачах управления» [1], А.А. Морозов, А.В. Гасников «Введение в математическое моделирование транспортных потоков» [6], В.И. Швецов «Математическое

моделирование транспортных потоков» [63] А.Ю. Замятин, С.Е. Семенов «Моделирование транспортных потоков: основные понятия и обзор методов», В.В. Буслаев, М.Ю. Шокин «Влияние параметров транспортного потока на пропускную способность автомобильных дорог» выделяют следующие критерии, влияющие на качество поездки по маршруту:

- 1) расстояние – чем короче маршрут, тем меньше времени и топлива потребуется для его преодоления;
- 2) загруженность – наличие пробок или ремонтных работ увеличивают время в пути;
- 3) погодные условия – плохая погода может ухудшить видимость и увеличить риск аварий, что в свою очередь замедляет движение транспорта;
- 4) безопасность – большое количество ТС, небольшое количество освещения и высокая аварийность на маршруте негативно влияют на качество поездки;
- 5) тип дороги – маршруты по скоростным автомагистралям обычно занимают меньше время, чем маршруты по городским улицам с ограничением скорости или проселочным дорогам с плохим качеством дорожного покрытия;
- 6) сезонность – загруженность и перекрытие маршрута может варьироваться в зависимости от времени суток, праздничных дней или мероприятий;
- 7) особенности маршрута – количество сложных поворотов, нерегулируемые перекрестки и развязки, большое количество светофоров и других технических средств также могут оказывать негативное влияние на качество поездки;
- 8) плотность потока – чем больше количество ТС на маршруте, тем больше загруженность и вероятность ДТП.

1.3 Модели загрузки и прогнозирования сети

Транспортное моделирование позволяет рассчитать среднюю скорость на сложном участке и предложить решения для повышения интенсивности движения. Модели, применяемые для анализа транспортных сетей, весьма разнообразны и классифицируются по уровню детализации транспортного потока, где выделяют макроскопические и микроскопические модели. **«Макроскопические модели** транспортных потоков рассматривают поток как единое целое – одномерная сжимаемая жидкость. В модели Лайтхилла-Уизема-Ричардса транспортный поток подобен потоку сжимаемой жидкости и описывается законом сохранения массы и основана на предположении о существовании функциональной зависимости между интенсивностью и плотностью. Один из самых серьезных недостатков модели – это то, что в любой момент времени средняя скорость соответствует равновесному значению при данной плотности потока, что не позволяет корректно исследовать часто возникающие случаи неравновесных потоков» [13].

В модели Танака поток АТС состоит из одной полосы. «Предположим, что скорость АТС не превышает V_{max} , что соответствует ограничениям на скорость движения машин внутри города или за ними» [14]. Тогда плотность можно посчитать по формуле (1.1)

$$\rho(v) = \frac{1}{d(v)}, \quad (1.1)$$

где $d(v) = L + c_1 v + c_2 v^2$ – среднее безопасное расстояние между АТС для определенной скорости потока v ;

L – среднестатистическая длина кузова;

c_1 – время реакции водителя;

c_2 – коэффициент, зависящий от тормозного пути.

В целом, коэффициент c_2 зависит от состояния дорожного покрытия и погодных условий. Из зависимости $d(v)$ можно получить зависимость $V(\rho)$,

удовлетворяющую условиям модели LWR. В связи с важностью учета данных условий модель получила широкое практическое применение. «Модель Уизема, в отличие от предыдущих моделей, дополнительно учитывает поведение водителей, которые уменьшают скорость движения, если видят впереди скопление транспортных средств и увеличивают скорость вплоть до максимально разрешенной на участке, если дорога свободна» [13]. Для реализации данного условия вводится специальный параметр $D(\rho) > 0$. Таким образом уравнение записывается по формуле (1.2)

$$v(t, x) = V(\rho(t, x)) - \frac{D(\rho(t, x))}{\rho(t, x)} \times \frac{\partial \rho(t, x)}{\partial x}, \quad (1.2)$$

где $v(t, x)$ – локальная скорость, зависящая от плотности потока и её пространственного изменения;

$V(\rho(t, x))$ – оптимальная скорость в зависимости от плотности ρ ;

$D(\rho(t, x))$ – коэффициент диффузии, связанный с поведением водителей;

$\rho(t, x)$ – плотность транспортного потока (количество транспортных средств на единицу длины) в момент времени t в точке x .

Описывается уравнением типа Бюргера по формуле (1.3)

$$\frac{\partial \rho}{\partial t} + \frac{\partial Q(\rho)}{\partial x} = \frac{\partial}{\partial x} \left(D(\rho) \frac{\partial \rho}{\partial x} \right), \quad (1.3)$$

где $\frac{\partial \rho}{\partial t}$ – изменение плотности потока во времени;

$\frac{\partial Q(\rho)}{\partial x}$ – пространственное изменение потока.;

$\frac{\partial}{\partial x} \left(D(\rho) \frac{\partial \rho}{\partial x} \right)$ – диффузионный член, описывающий «распространение» плотности.

«Модель Пейна отличается от остальных тем, что скорость, с которой движется автотранспортное средство регулируется некоторой желаемой

скоростью, к которой стремится водитель» [11]. Уравнение модели показано в формуле (1.4)

$$\frac{\partial}{\partial t}v = \frac{\partial v}{\partial t} + v \frac{\partial v}{\partial x} = -\frac{1}{\tau} \left(v - \left(V(\rho) - \frac{D(\rho)}{\rho} \frac{\partial \rho}{\partial x} \right) \right), \quad (1.4)$$

где τ – скорость стремления к желаемой скорости движения.

В связи с тем, что транспортный поток в данном случае обладает «мотивацией», основное различие между его гидродинамическим аналогом заключается лишь в правой части уравнения, а значит можно использовать уже известные алгоритмы расчетов и вычислений для соответствующих жидкостей. Тем не менее, модель подверглась серьезной критике – были найдены существенные недостатки, такие как появление отрицательной скорости и слишком близкое движение автомобилей друг к другу, приводящее к неизбежной аварийной ситуации.

В **микроскопических моделях** транспортный поток рассматривается с точки зрения взаимодействия между несколькими отдельными машинами, а не целой группы. Основным интересом моделей является описание поведения и движения одной конкретной машины в зависимости от остальных транспортных средств в потоке. Таким образом, модель движения может быть описана формулой (1.5)

$$\frac{dv_a(t)}{dt} = \frac{v_a^0 + d_a(t) - v_a(t)}{t_a} + f_{a,(a-1)}(t), \quad (1.5)$$

где $\frac{dv_a(t)}{dt}$ – скорость изменения скорости автомобиля a во времени;

v_a^0 – скорость водителя в отсутствие препятствий;

$d_a(t)$ – расстояние от автомобиля a до лидера;

t_a – характеризует, насколько быстро автомобиль a реагирует на изменения условий движения;

$f_{a,(a-1)}(t) \leq 0$ – описывает воздействие лидера $(a - 1)$.

Воздействие лидера – есть общая функция от:

- 1) относительной скорости: $\Delta v_\alpha(t) = [v_\alpha(t) - v_{\alpha-1}(t)]$;
- 2) скорости $v_\alpha(t)$ АТС α , влияющей на безопасную дистанцию до впереди идущего АТС;
- 3) расстояния между головными частями АТС (расстояние брутто) $d_\alpha(t) = [x_{\alpha-1}(t) - x_\alpha(t)]$ или просвет (расстояние нетто) $s_\alpha(t) = [d_\alpha(t) - l_{\alpha-1}]$, где l_α – длина АТС α .

«Заслуживают упоминания и модели следования за лидером: модель Трайбера, модель Джиппса, модель CARSIM, модель Кастильо, модель Мэзона-Вудса» [12]. Одним из недостатков моделей является отсутствие описания поведения водителя вне окружения. Без движущегося впереди АТС при $d_\alpha \rightarrow \infty$, АТС α не будет ускоряться совсем, но желательная скорость v_α^0 должна достигаться именно на трассе, в свободном трафике. Таким образом, модели следования не предполагают адаптацию к скорости лидирующего АТС, а предлагают адаптацию к скорости $v'_e(d_\alpha)$, которая должна отражать требования безопасности и иногда называется оптимальной скоростью. Ньюэл выразил адаптацию в формуле (1.6)

$$v_\alpha(t + \Delta t) = v'_e(d_\alpha(t)) = v_e(s_\alpha(t)). \quad (1.6)$$

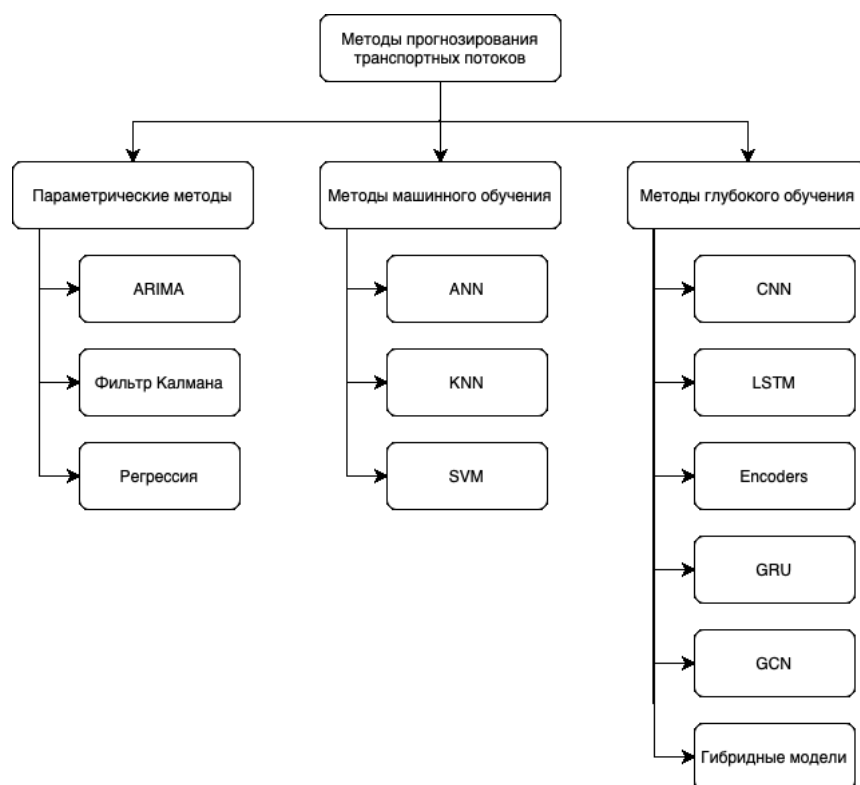
Модель оптимальной скорости не содержит отклика водителя на относительную скорость δv_α со впереди движущимся АТС, она весьма чувствительна к конкретному выбору функции $v'_e(d_\alpha)$ и приводит к авариям, когда быстро движущееся АТС приближается к стоящему. Чтобы избежать этого, требуется ввести отношения, зависящие от скорость/дистанция и выбрать очень маленькие значения τ во избежание нереально больших ускорений. В действительности, однако, время ускорения от пяти до десяти раз больше времени торможения. Более того, водители сохраняют достаточно большую дистанцию безопасности и раньше снижают скорость, когда относительная скорость, $\Delta v_\alpha(t)$ высока.

Гибридные модели пытаются скомбинировать сильные стороны моделей для каждого подходящего случая. «Многие модели предусматривают

множественность моделирования и подходят как для микроскопических, так и для макроскопических расчетов. Часто используемый подход – сочетание возможности макроскопических моделей и моделей клеточных автоматов, что позволяет моделировать происходящие процессы численным образом. Из наиболее известных можно выделить, например, модели, полученные дискретизацией модели Ава-Раскла, основанные на расширении модели Даганзо» [11]. На сегодняшний день не существует общепринятого определения загруженности транспортной сети. К. Даганзо в своей теории выдвигает гипотезу о том, причина пробок кроется в узких местах – сужениях и расширениях дороги, как постоянных, так и временных, что является причиной образования, роста и распространения очередей на дорогах.

Согласно отчетам Департамента транспорта и развития дорожно-транспортной инфраструктуры Москвы: «Движение потока на городских магистралях происходит на стадиях синхронного движения с частыми внезапными переходами к старт-стоп режиму, причем, пробки растекаются по сети за счет того, что подъезжающие к затору водители разворачиваются на объездные дороги, на которых в свою очередь возникают новые заторы» [19]. Различные авторы исследовали ряд методов прогнозирования транспортного потока с высокой точностью, принимая во внимание различные аспекты и связанные с ними характеристики. Среди параметрических методов популярны модели временных рядов, такие как ARIMA, которые учитывают сезонные колебания и исторические паттерны движения. Методы машинного обучения, включая градиентный бустинг и случайные леса, позволяют анализировать многомерные данные, такие как погодные условия и события в городе, улучшая точность прогнозов. Глубокое обучение, в свою очередь, использует рекуррентные нейронные сети (например, LSTM) для обработки последовательностей данных в реальном времени, что особенно эффективно для моделирования нелинейных зависимостей в динамике транспортных потоков. В целом, можно выделить три категории моделей прогнозирования

транспортных потоков, которые приведены на рисунке 1.4, параметрические методы, методы машинного обучения и методы глубокого обучения.



Источник: составлено автором.

Рисунок 1.4 – Категории моделей прогнозирования транспортных потоков

Статистические методы по сравнению с другими современными методами являются менее точными и хоть такие методы могут извлекать особенности из временных рядов, они представляют трудности в представлении пространственных отношений в задачах прогнозирования транспортных потоков. Фильтры Калмана и интегрированная модель авторегрессии – скользящего среднего (далее – ARIMA) являются наиболее изученными статистическими методами прогнозирования временных рядов. Модели должны быть обучены на большом наборе данных для достижения более точных результатов. А. Емами и другие [109] предлагают метод, основанный на адаптивных фильтрах Калмана, диагностики потока трафика, объединяя данные, предоставляемые подключенными транспортными средствами и устройствами Bluetooth. В процессе эксперимента использовались данные более 100 подключенных транспортных средств в городской среде. Результаты показывают, что по мере увеличения объема

транспортного средства генерируется большой поток данных, следовательно, фильтр Калмана более эффективно прогнозирует поток.

Н. Темпельмайер и другие [175] предложили контролируемую модель пространственно-временных зависимостей на данных транспортного потока с учетом запланированных городских мероприятий. Для обеспечения точных прогнозов использовались различные функции событий, поведение мобильности и городской инфраструктуры. Регрессионная модель имеет возможность предсказывать пространственную размерность эффекта событий, используя алгоритмы k-ближайших соседей (далее – kNN), регрессию Риджа и опорных векторов. Применив методологию к историческим данным событий в Ганновере, Германии в течение четырех месяцев для 150 городских событий, результаты исследования показали среднюю абсолютную погрешность менее 27%.

Х. Братсас и другие [78] сравнивают четыре метода: регрессия опорных векторов (SVR), множественная линейная регрессия (далее – MLR), нейронные сети (далее – NN) и метод случайного леса (далее – RF) с целью прогнозирования скорости автомобильного движения в городе Салоники, Греция с интервалом 15 минут. Результаты показывают, что при оценке MAE модель на основе SVR представляет погрешность 6,25% и более эффективна, чем другие примерно на 0,2 км/ч. Модели SVR и NN показали лучшие результаты при оценке показателя RMSE.

М. Пейман и другие [156] указывают, что автомобильная самоорганизующаяся сеть (далее – VANET) может внести большой вклад в оперативное управление транспортным потоком. Однако большой объем трафика и транспортных средств в городской сети вызовет нестабильность в ее топологии из-за оперативной обработки информации. Основываясь на более ранние исследования [41], было обнаружено, что задержка в пути является серьезной проблемой, которая влияет на результаты прогнозирования. Авторы предлагают гибридный авторегрессионный алгоритм, основанный на задержках движения транспортных средств для

улучшения результатов прогноза, который не учитывался в предыдущих пространственно-временных моделях. Для составления набора данных исследования было выбрано семь различных маршрутов с разным объемом потока. Предложенная модель показала низкие показатели MSE и MAPE 484,51 и 4,92% по сравнению с 6325,88 и 12,35% метода ARIMA. Однако выборка собранных данных неадекватно описывает представление о транспортном потоке за неделю или месяц.

В исследовании [144] были описаны несколько моделей, подходящих для сценариев беспроводных децентрализованных самоорганизующихся сетей MANET и VANET. Ключевым преимуществом моделей является способность точно имитировать пространственные и временные измерения мобильности и подходят для сложных крупномасштабных сценариев. Однако, их эффективность зависит от наличия больших массивов GPS-данных, которые редко находятся в открытом доступе и доступны научному сообществу. Было инициировано несколько проектов по сбору траекторий движения транспортных средств, таких как OpenStreetMap [32] и HighD. После чего наборы данных должны быть обогащены большим количеством трассировок, чтобы обеспечить более точные результаты в будущем.

Модели мобильности на основе симуляторов дорожного движения (далее – TSBMM), описанные Д. Барсело в [72] относятся к моделям, основанным на инструментах моделирования мобильности транспортных средств. Инструменты являются либо коммерческими продуктами, либо бесплатным программным обеспечением с открытым исходным кодом. PARAMICS, CORSIM, AIMSUN и PTV Vissim – одни из первых инструментов моделирования, разработанные для разных транспортных сценариев, позволяющие детально моделировать движение. Однако создание сложной конфигурации и очень длительное время моделирования крупномасштабных сценариев представляют основной недостаток моделей на основе симуляторов.

Модели машинного обучения позволяют прогнозировать транспортные потоки в случае нелинейного и стохастического поведения. Исследователи используют модели машинного обучения для решения проблем, связанных с неоднородностью данных, количественной оценкой пространственно-временных зависимостей, сильными нелинейными характеристиками транспортного потока и различными внешними динамическими и статическими факторами.

П. Кай и другие [79], принимая во внимание существующую пространственно-временную корреляцию в транспортной сети, предлагают улучшенную модель kNN, использующую динамическую меру расстояния, основанную на физических данных функции для замены расстояния и описывают матрицы состояний для моделирования дорожной ситуации. Предложенная модель оценивается на данных дорожного движения в Пекине и сравнивается с параметрическими методами. Предлагаемая модель демонстрирует большую точность со значениями MAPE на 2,96% выше, чем исходная модель kNN.

Лиу и другие [132] признают сильные нелинейные характеристики городского транспортного потока, множественные динамические и статические факторы воздействия, а также растущую проблему прогнозирования транспортного потока в городской среде. В исследовании авторы собрали различные типы данных из нескольких источников: данные о транспортном потоке, погодных условиях и достопримечательностях в городе Гуйчжоу, Китай. Алгоритм машины структурированных опорных векторов (далее – SSVM) имеет достаточно плохие показатели при изменяющемся трафике высокой интенсивности.

Лин и другие [133] упоминают, что векторные машины релевантности обладают превосходным обобщением и высокой скоростью тестирования. Авторы исследуют подход к прогнозированию условий дорожного движения, основанный на векторных машинах релевантности и предлагают оптимизировать параметры модели, используя генетический алгоритм

и оптимизацию роя частиц. Модель оценивается на основе данных на дорогах Канады. Результаты получаются более точными, чем другие модели прогнозирования, а технология распараллеливания значительно сокращает время, затрачиваемое алгоритмом оптимизации параметров. Предложенная модель превосходит другие модели, однако для оптимизации параметров скорость сходимости очень высока в первых циклах обучения и имеет тенденцию к снижению в последующих итерациях.

Модели глубокого обучения являются расширением машинного обучения и состоят из большего количества слоев обработки, способных изучать характеристики с высоким уровнем абстракции. В последние годы такие модели используются для фиксации параметров мобильности в городах с помощью датчиков движения, предоставляющих большое количество данных разных типов и характеристик. Среди методов глубокого обучения в рамках алгоритмов прогнозирования транспортных потоков, можно выделить: сверточные нейронные сети (далее – CNN), сети долгой краткосрочной памяти (далее – LSTM), управляемый рекуррентный блок (далее – GRU), графовые нейронные сети (далее – GCN) и гибридные модели, которые объединяют несколько методов.

Ю. Ву и другие [185] предложили модель, основанную на моделях CNN и GRU для извлечения пространственных и временных характеристик о транспортном потоке в наборе данных системы измерения производительности (далее – PeMS). В модели были использованы исторические данные за неделю для отображения периодичности транспортного потока. Аналогичным образом, данные о скорости использовались для обучения модели и определения насколько сильна корреляция исторических и будущих данных потока. Используя такие метрики как MAE, RMSE и MSE, предложенная модель показала улучшенную среднюю производительность по сравнению с другими методами. Однако частота ошибок увеличивается в других точках прогнозирования.

Д. Чжан и другие [202], основываясь на исторических данных о траектории, погоде и событиях, используют алгоритм CNN со сверхглубокой структурой, до 1000 слоев. В исследовании используются наборы данных TaxiBJ (Beijing City Taxi GPS) и BikeNYC (New York Bicycle System Path Data). Сначала входной и выходной поток преобразуется в двухканальное изображение, временная зависимость представляет три временные характеристики: расстояние, период и тенденция. Модель извлекает пространственные характеристики между соседними и удаленными вершинами. Кроме того, они вручную извлекают характеристики из внешних наборов данных, таких как информация о событиях и осадках, чтобы включить их в связанную нейронную сеть. Результаты превосходят ARIMA и DNN. Несмотря на то, что методы используют для прогнозирования изображения трафика с метками времени за прошлые периоды, зависимость временной последовательности явно не моделируется.

Ю. Женг и другие [208] рассматривают сеть, которая включает в себя модуль нейронной поисковой сети для пространственно-временной архитектуры (далее – ST-NASNet). Авторы отмечают, что предлагаемый метод не учитывает факторы, которые могут повлиять на эффективность прогнозирования, такие как количество осадков, классификация улиц и пространственное соответствие между вершинами.

Д. Ван и другие [183] предлагают двунаправленную схему LSTM для захвата глубинных характеристик транспортного потока с использованием климатических данных. Для проверки точности предложенной методики используются несколько наборов данных, а результаты сравниваются с моделями ARIMA и SVR. По результатам эксперимента с параметрами входного окна – 30, 40 и 50 минут предложенная модель превосходит другие алгоритмы. Авторы утверждают, что дополнительная информация о данных погодных условий улучшает производительность модели. Критическим и очевидным недостатком является то, что они сжимают всю информацию

в вектор фиксированной длины, что приводит к невозможности запоминания длинных параметров.

Ш. Маджумдар и другие [143] используют архитектуру LSTM для прогнозирования распространения заторов дорожной сети. Модель собирает данные о скорости с сенсорных устройств и прогнозирует распространение заторов каждые пять минут. Экспериментальные результаты показывают, что предложенный подход обеспечивает приемлемые прецизионные характеристики распространения заторов за короткие промежутки времени. Точность предлагаемого метода составляет от 84 до 95% во временном окне в 5 минут и представляет выходы 95-97% для 2-х минутных временных окон. Несмотря на то, что LSTM упрощает структуру сети с удовлетворительными результатами, ее точность зависит главным образом от конструкции дорожной сети. При применении в долгосрочном прогнозировании транспортных потоков средняя абсолютная погрешность увеличивается, что означает, что общая эффективность RNN и LSTM будет незначительно отставать.

Т. Сан и другие [171] предложили модель CNN, основанную на двунаправленной пространственно-временной сети, которая использует скорость движения и метеорологическую информацию. Сверточная сеть фиксирует пространственную зависимость топологии трафика для составления набора входных данных временных рядов, а алгоритм LSTM выполняет обучение набора данных. Результаты эксперимента превышают сценарий без метеорологической информации, а средняя погрешность составляет менее 5%. Предлагаемая модель подчеркивает использование данных из внешних источников с более точными результатами в качестве основы для будущих исследований.

Ш. Ду и другие [96] рассмотрели многослойную гибридную модель для улучшения прогноза автомобильного потока. Модель CNN используется для характеристики пространственных зависимостей, соответствующих локальной проекции, а сеть LSTM моделирует характеристики времени и регулярности данных о трафике. Экспериментальный набор данных

содержал около 70 тысяч точек, из которых 80% использовались для обучения, а 20% – для тестирования. Авторы отмечают, что модель сравнивалась с алгоритмами: ARIMA, RBF, Linear Kernel SVR, RNN, GRU, LSTM и показала более высокие результаты точности.

Авторы сходятся во мнении, что ключевым пунктом при прогнозировании транспортных потоков является учет пространственно-временных характеристик, поскольку последовательность исторических данных определяет поведение данных временных рядов и влияет на развитие следующего тренда. Помимо очевидных особенностей транспортного потока существуют внешние факторы, влияющие на динамику движения и пробки, такие как дорожно-транспортные происшествия, погодные условия, события, достопримечательности и другие. По мере того, как источники данных становятся более неоднородными, алгоритмы глубокого обучения проецируются в качестве соответствующих альтернатив из-за способности представлять многоуровневую структуру восприятия.

В приложении Б обобщены результаты сравнительного анализа моделей описанных выше исследований разных авторов. Обобщив результаты, можно сделать вывод, что наибольшая точность достигается при краткосрочном прогнозировании с окном горизонта до 20 мин. Однако, результаты зависят от данных, размерности и метрик оценки. Модели глубокого обучения и гибридные модели достигают лучших результатов прогнозирования потока трафика, превосходя традиционные параметрические модели машинного обучения.

Модели глубокого обучения широко используется для решения множества задач, включая прогнозирование транспортных потоков и пробок. Однако существующие исследования в основном основаны на готовых данных о транспортных потоках, полученных со стационарных датчиков обнаружения. Для сбора таких данных требуется большое покрытие транспортных сетей устройствами обнаружения. Поэтому исследования

применимы только к тем дорогам, где установлены датчики обнаружения. Таким образом область применения исследований ограничена.

1.4 Влияние и прогнозирование погодных условий

С начала 1950-х годов было установлено, что погодные условия оказывают значительное влияние на такие параметры, как средняя скорость потока, пропускная способность, поведение и взаимодействие участников транспортной сети. Исследования [92; 138] подтверждают, что неблагоприятные погодные условия приводят к снижению пропускной способности и безопасности и увеличению риска несчастных случаев. По оценкам исследователей, примерно 28% всех дорожно-транспортных происшествий и 19% смертельных случаев связаны с погодными условиями на дорогах в качестве сопутствующего фактора [107], а во время дождя происходит снижение средней скорости на 14%. Однако авторы не исследовали зависимость скорости от интенсивности дождя. В условиях небольшого снега средняя скорость потока снижается в среднем на 20-30%. Темное время суток приводит к снижению средней скорости на 5-10%. Влияние погодных условий на соотношение и параметры транспортных потоков варьируется в зависимости от типов дорог.

Исследователи применяли модель Маркова, модель авторегрессионного интегрированного скользящего среднего (далее – ARIMA), модель k ближайших соседей, метод опорных векторов, модель искусственной нейронной сети и модели глубокого обучения для прогнозирования транспортного потока в нормальных условиях. Чтобы повысить точность прогнозирования исследователи использовали погодные условия в качестве вспомогательных данных прогнозирования транспортных потоков и получили хорошие результаты. Однако исследователи не тестировали предложенные модели в неблагоприятных погодных условиях. Неблагоприятные погодные условия могут уменьшить дальность видимости, восприятие окружающей среды и радиус обзора водителя и влияют на

поведение участников транспортной сети. Пан и другие [153] утверждают, что дорожно-транспортные происшествия на шоссе в китайской провинции Чжэцзян, вызванные дождливыми, туманными и снежными днями, занимают 33,33%, 26,31%, 14,10% соответственно. При дальности видимости менее 200 м средняя скорость движения снизится на 15-25%. Основываясь на данных о транспортных потоках в китайском городе Пекин, Чжан [205] обнаружил, что пропускная способность дороги снижается на 7,26%, 10,87% и 17,09%, а скорость свободного потока снижается на 3,07%, 5,29% и 6,64% при слабом, умеренном и сильном дожде соответственно. Аналогичные закономерности наблюдались при слабом, умеренном и сильном снегопаде. Кроме того, исследователи отмечают, что сильный ветер оказывает воздействие на условия вождения, что означает необходимость в полной мере учитывать влияние неблагоприятных погодных условий на транспортный поток.

Некоторые исследователи классифицировали дождь, снег, туман как плохую погоду для создания модели прогнозирования транспортных потоков в условиях плохой погоды с использованием метода случайного леса (далее – Random Forest), а также использовали гибридную модель глубокого обучения. В таблице 1.3 собраны современные исследования влияния погодных условий на загруженность транспортного потока и используемые модели.

Таблица 1.3 – Исследования прогнозирования транспортных потоков в условиях неблагоприятной погоды

Ссылка	Модель	Исследуемые погодные условия
[150; 164]	Нечеткая нейронная сеть	Дождь
[149]	ARIMA и SVM	Дождь
[95]	Теория мягких множеств	Дождь
[135]	Функция множественной регрессии	Снег
[96]	Скрытая марковская модель	Снег
[122]	Глубокая нейронная сеть	Метель
[67]	Метод опорных векторов	Плохие погодные условия
[152]	CNN, LSTM, GRU	Дождь и ветер

Источник: составлено автором по материалам [67; 95; 96; 122; 137; 149; 150; 152; 164].

Исследования сосредоточены на прогнозировании транспортных потоков в дождливую погоду, а некоторые исследователи рассматривали снежную погоду. Несмотря на достигнутый прогресс другие типы неблагоприятных погодных условий, включая туман и ветреную погоду, также оказывают влияние на движение. Существующие исследования грубо классифицировали прогнозирование транспортных потоков при каждом типе неблагоприятных погодных условий в одну ситуацию. Не обсуждались воздействия различных типов неблагоприятных погодных условий на транспортный поток, а предложенные модели не тестировались при множественных неблагоприятных погодных условиях и наличии исторических данных. Таким образом, необходимо разработать более общую модель прогнозирования транспортных потоков при неблагоприятных погодных условиях, основанную на исторических данных о погоде на маршруте. Простые модели не способны точно отразить характеристики транспортного потока в неблагоприятных погодных условиях, так как они используют ограниченное количество данных о погоде. Большинство моделей прогнозирования транспортных потоков в вышеупомянутых исследованиях были созданы с применением статистических моделей и моделей машинного обучения, небольшое число исследователей выбирали модели глубокого обучения для прогнозирования транспортных потоков в условиях неблагоприятной погоды.

1.5 Поиск альтернативных маршрутов

Задача поиска альтернативных маршрутов – это поиск пути с наименьшей стоимостью между двумя вершинами. При этом стоимость может быть измерена в расстоянии между точками, времени в пути, коэффициентом безопасности, стоимостью проезда и так далее. Для маршрутизации в сетевых структурах созданы многочисленные алгоритмы, фактически формирующие инструменты поиска кратчайшего пути по расстоянию. Наиболее известными и широко применяющимися являются

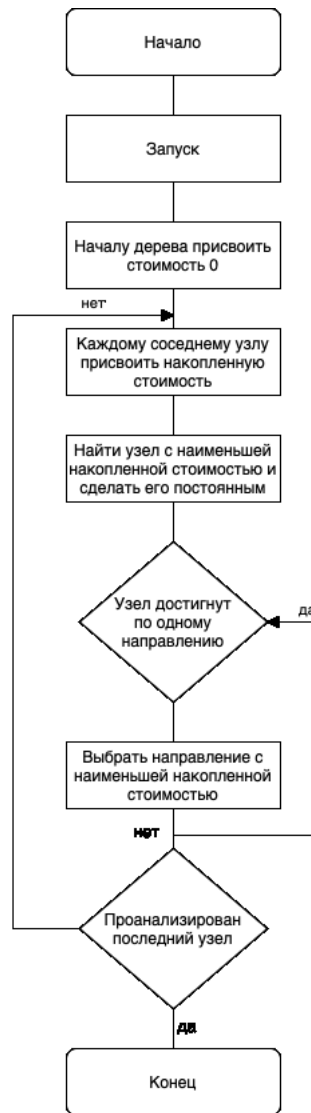
алгоритмы поиска в ширину, поиска в глубину, Дейкстры, Беллмана-Форда, A Star (далее – A^*), Флойда-Уоршелла и другие. Среди них есть алгоритмы, которые работают только с графом дорог и не требуют предварительной подготовки индекса, например, алгоритмы Дейкстры и A^* , а есть алгоритмы, которые требуют генерации индексов на основе графа перед началом расчета маршрутов. Наиболее радикальный вариант – посчитать и запомнить кратчайшие пути, соединяющие все дуги графа со всеми дугами. Такой подход потребует огромного индекса, но зато маршруты в среднем будут вычисляться за константное время – время поиска по ключу, состоящему из старта и финиша в хеш-таблице.

Алгоритм поиска в ширину (далее – BFS) является одним из фундаментальных алгоритмов. Популярное использование этого алгоритма – поиск кратчайшего пути в сетке или лабиринте. Некоторые из реальных применений этого алгоритма включают GPS-отслеживание, анализ сетей и графиков, а также поисковые системы. Алгоритм посещает каждую вершину. Сначала посещаются, которые находятся всего в одном шаге от начальной, затем в двух шагах, и это продолжается до тех пор, пока не будут посещены все вершины. Основное преимущество алгоритма заключается в том, что решение всегда будет найдено. Если существует несколько решений, все решения будут найдены и из всех алгоритм выберет решение с минимальной стоимостью. Недостатком является его отношение к использованию памяти. С помощью этого алгоритма все узлы сохраняются в дереве, и каждый узел будет проверен на уровне n , чтобы получить решение для уровня $n + 1$. BFS подходит только для невзвешенных графов.

Алгоритм поиска в глубину (далее – DFS) сначала исследует самую глубокую вершину, а затем возвращается для изучения других в дереве поиска. Поиск продолжается до самого глубокого уровня дерева. Одним из немногих преимуществ алгоритма является то, что он требует гораздо меньше памяти по сравнению с BFS. Одним из недостатков является то, что не все решения будут найдены в случае нескольких решений, в отличие от BFS.

Кроме того, поскольку алгоритм движется вглубь, существует возможность бесконечного цикла. DFS не ищет оптимальные маршруты.

Алгоритм Дейкстры используется для нахождения пути в графе из одной вершины до всех остальных. Считается одним из самых простых и хорошо выполняется в графах с небольшим количеством вершин. В случае с сетью, количество вершин в графе может достигать до нескольких тысяч. Алгоритм Дейкстры работает по принципу жадного алгоритма. «На каждом шаге стремится минимизировать текущую общую стоимость пути» [14]. Сначала инициализируются два множества, содержащие уже обработанные вершины (изначально пустое) и содержащие все остальные вершины графа. Каждой вершине графа присваивается вес, который представляет минимальную известную стоимость пути от начальной вершины до заданной. Для начальной вершины вес равен 0, для всех остальных – бесконечность. На каждом шаге алгоритм выбирает вершину из непосещенного множества с наименьшим весом, перемещает эту вершину в множество посещенных вершин и обновляет веса всех соседей выбранной вершины. Вес соседа обновляется, если через выбранную вершину можно добраться до соседа с меньшей стоимостью. Для оптимизации работы в крупных графах часто применяют структуры данных, такие как приоритетная очередь, что снижает вычислительную сложность до $O(m + n \times \log n)$, где m – количество рёбер. Кроме того, алгоритм активно используется в транспортных навигационных системах для поиска кратчайших маршрутов, но требует модификаций при наличии динамически изменяющихся условий, например, пробок. Процесс продолжается, пока не будут посещены все вершины или пока не будет найден путь до конечной вершины. Недостатками алгоритма является поиск кратчайшего пути от одной вершины графа до всех остальных и некорректная работа с отрицательными дугами. Вычислительная сложность алгоритма – $O(n^2)$. Блок-схема алгоритма изображена на рисунке 1.5.



Источник: составлено автором.

Рисунок 1.5 – Алгоритм Дейкстры

Алгоритм Беллмана-Форда вычисляет кратчайшие пути от исходной вершины ко всем остальным во взвешенном ориентированном графе. Он медленнее, чем алгоритм Дейкстры, но более универсален, так как способен обрабатывать ребра с отрицательным весом. Если граф содержит «отрицательный цикл» (цикл, сумма ребер которого равна отрицательному значению), достижимый из источника, тогда не существует самого дешевого пути: любой путь, имеющий точку на отрицательном значении. Цикл можно удешевить еще одним обходом отрицательного цикла. В таком случае алгоритм Беллмана-Форда может обнаружить отрицательный цикл и сообщить о нем. Алгоритм Беллмана-Форда выполняется релаксацией, в котором приближения к правильному расстоянию заменяются лучшими до тех

пор, пока они в конечном итоге не достигнут решения. Приблизительное расстояние до каждой вершины всегда является завышенным по сравнению с истинным расстоянием и заменяется минимумом своего старого значения и длиной вновь найденного пути. Однако алгоритм Дейкстры использует приоритетную очередь для жадного выбора ближайшей вершины, которая еще не была обработана, и выполняет этот процесс релаксации на всех ее исходящих дугах. Алгоритм Беллмана-Форда, напротив, расслабляет все дуги и делает это $N - 1$ раз, где N – количество вершин в графе. При каждом повторении количество вершин, с рассчитанными расстояниями растет, в конечном итоге все вершины будут иметь правильные расстояния, что позволяет применять алгоритм Беллмана-Форда к более широкому классу данных. Основными недостатками алгоритма являются: плохая масштабируемость; изменения в топологии сети не отражаются быстро, поскольку обновления распространяются от вершины к вершине; при сбое канала расчет происходит до бесконечности, так как вершина становится недоступным для набора других вершин, и такие вершины постоянно увеличивают свои оценки расстояния до него, создавая петли маршрутизации. Вычислительная сложность алгоритма – $O(VE)$, где V – число вершин в графе, E – число ребер в графе. Йен описал [189] усовершенствование алгоритма Беллман-Форда для графа без циклов отрицательного веса, делая алгоритм быстрее на практике, он не улучшает наихудший сценарий. Улучшение сокращает количество шагов релаксации, которые необходимо выполнять на каждой итерации алгоритма – если вершина имеет значение расстояния, которое не изменилось с момента последнего ослабления ребер, тогда нет необходимости ослаблять дуги во второй раз. Таким образом, по мере роста числа вершин с правильными значениями расстояний число исходящих ребер, которые необходимо ослаблять на каждой итерации, уменьшается, что приводит к экономии времени с постоянным коэффициентом для плотных графов.

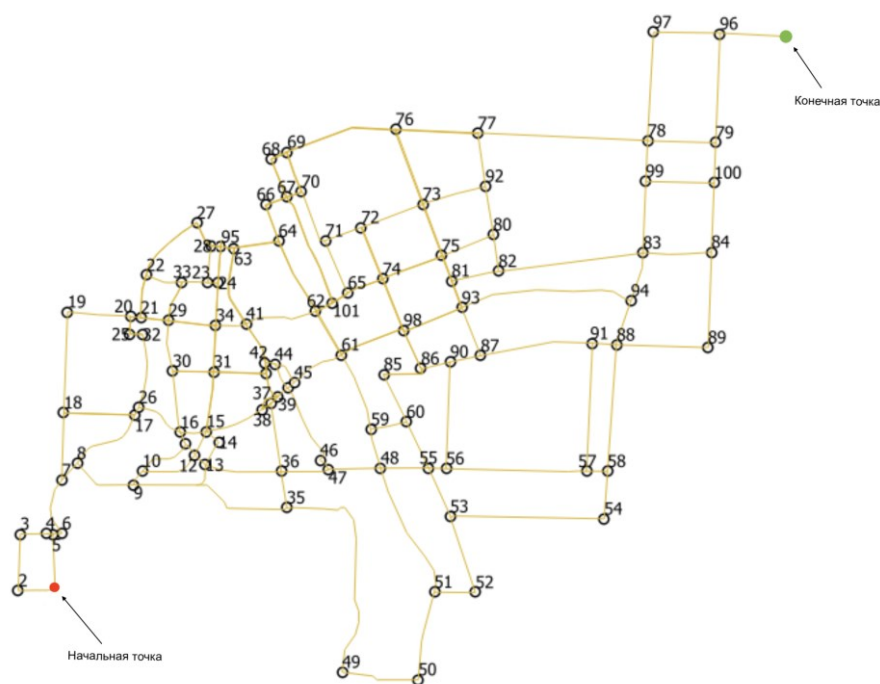
«Алгоритм Флойда-Уоршелла является динамическим алгоритмом поиска кратчайших расстояний между всеми вершинами взвешенного ориентированного графа. Если в графе есть циклы отрицательного веса, то формально алгоритм Флойда-Уоршелла неприменим» [14]. Но на самом деле алгоритм корректно работает для всех пар, пути между которыми никогда не проходят через цикл негативной стоимости, а для остальных будут получены какие-нибудь числа, возможно сильно отрицательные. После отработки такого графа на диагонали матрицы кратчайших путей возникнут отрицательные числа – кратчайшее расстояние от вершины в этом цикле до неё самой будет меньше нуля, что соответствует проходу по этому циклу, так что алгоритм можно использовать для определения наличия отрицательных циклов в графе. Является эффективным алгоритмом расчета кратчайших маршрутов в плотных графах, когда имеет место большое количество пар ребер между парами вершин. Вычислительная сложность алгоритма $O(n^3)$. Алгоритм A* фактически является расширением алгоритма Дейкстры. «Алгоритм A*, используя эвристическую функцию, обходит только те вершины, которые, как предполагается, приведут к конечной, что может значительно уменьшить количество обрабатываемых вершин и увеличить производительность, особенно в больших графах. A* учитывает, где находится финиш и рассматривает вершины графа преимущественно в его сторону, что позволяет раскрыть в среднем меньше вершин для нахождения кратчайшего пути. Вычислительная сложность алгоритма – $O(n \times \log n)$, где n – количество ребер в графе. Для нахождения оптимального маршрута в графах с весами лучше использовать: алгоритм Дейкстры – находит кратчайший путь во взвешенном графе без отрицательных ребер или Алгоритм A* – улучшенная версия Дейкстры с эвристикой, быстрее в реальных задачах или Беллмана-Форда – если граф содержит отрицательные веса. Сравнение алгоритмов отображено в таблице 1.4.

Таблица 1.4 – Сравнение сложности алгоритмов

Алгоритм	Сложность
Дейкстры	$O(n^2)$
Беллмана-Форда	$O(n^2)$
Флойда-Уоршелла	$O(n^3)$
A*	$O(n \times \log n)$

Источник: составлено автором.

В рамках эксперимента была сгенерирована случайная транспортная сеть, состоящая из множества точек (узлов). Между начальной и конечной точками необходимо построить кратчайший маршрут. Для каждого ребра случайным образом был назначен вес (расстояние), по которым производилось сравнение алгоритмов. Пример случайно сгенерированной сети и данных представлен на рисунке 1.6 и в таблице 1.5.



Источник: составлено автором.

Рисунок 1.6 – Случайно сгенерированная транспортная сеть

Таблица 1.5 – Расстояние между точками

Точка старта	Точка финиша	Относительная стоимость
Начальная точка	2	0,23
2	3	0,16
...	...	...
96	Конечная точка	0,37

Источник: составлено автором.

Эксперименты проводились на сервере с 2xCPU Intel Xeon Gold 6238R, RAM 12x32Gb, SSD 2x7680Gb, LAN 2x10Gb/s, OS CentOS 8. Результаты экспериментов приведены в таблицах 1.6 и 1.7.

Таблица 1.6 – Время движения от начальной до конечной точки

Тест	Время обработки, мс					
	Поиск в ширину	Поиск в глубину	Дейкстры	Беллмана-Форда	Флойда-Уоршелла	A*
1	1,4163	1,4169	1,2011	1,2405	1,1444	1,1205
2	1,3951	1,4255	1,1899	1,1953	1,1196	1,1150
3	1,3992	1,4110	1,1950	1,2090	1,1132	1,1099

Источник: составлено автором.

Таблица 1.7 – Количество итераций цикла поиска маршрута

Тест	Количество итераций, шт					
	Поиск в ширину	Поиск в глубину	Дейкстры	Беллмана-Форда	Флойда-Уоршелла	A*
1	147	151	122	116	104	98
2	150	149	122	116	104	98
3	150	154	122	116	104	98

Источник: составлено автором.

По результатам экспериментов можно сделать следующие выводы. Алгоритмы Дейкстры, Беллмана-Форда, Флойда-Уоршелла и A* показали наилучшие результаты при поиске кратчайшего маршрута для набора случайных точек. Тем не менее, алгоритм A* предоставит решение быстрее из-за меньшего количества итераций при поиске кратчайшего маршрута. Чем

больше точек (узлов), тем больше разница между количеством итераций цикла и временем.

Выводы по первой главе

В первой главе проанализированы современные российские и зарубежные исследования: моделирования транспортной сети; поиска критериев, влияющих на транспортную сеть; методов многокритериального анализа в транспортной сети.

Изучены актуальные модели, описывающие поведение участников, загруженность и прогнозирование в транспортной сети.

Изучены работы влияния погодных условий на загруженность транспортной сети и прогнозирования загруженности с учетом негативных погодных условий.

Проведено сравнение сложности и эксперимент поиска и расчета маршрутов случайно сгенерированной транспортной сети на основе существующих алгоритмов.

По результатам существующих исследований можно сделать вывод, что исследователям удалось достаточно продвинуться в моделировании транспортной сети, загруженности и прогнозировании. Тем не менее, недостаточно изучен вопрос комплексной оценки результатов моделирования на основе множества разнородных критериев. Современные модели учитывают не более трех критериев – расстояние, время в пути и погодные условия, что является недостаточным для более точного анализа транспортной сети и поиска рациональных маршрутов на основе большего числа критериев. Большинство исследователей не имеют актуальных данных о текущем состоянии транспортной сети и не анализируют ситуацию в режиме, приближенному к реальному времени.

Глава 2

Комплекс моделей сетевых структур дорожного движения и методика оценки эффективности маршрутизации транспортных средств по результатам моделирования

Необходимые для разработки эффективных алгоритмов маршрутизации транспортных средств с учетом особенностей улично-дорожной сети, в том числе наиболее значимых деструктивных факторов, модели сетевых структур должны обеспечивать:

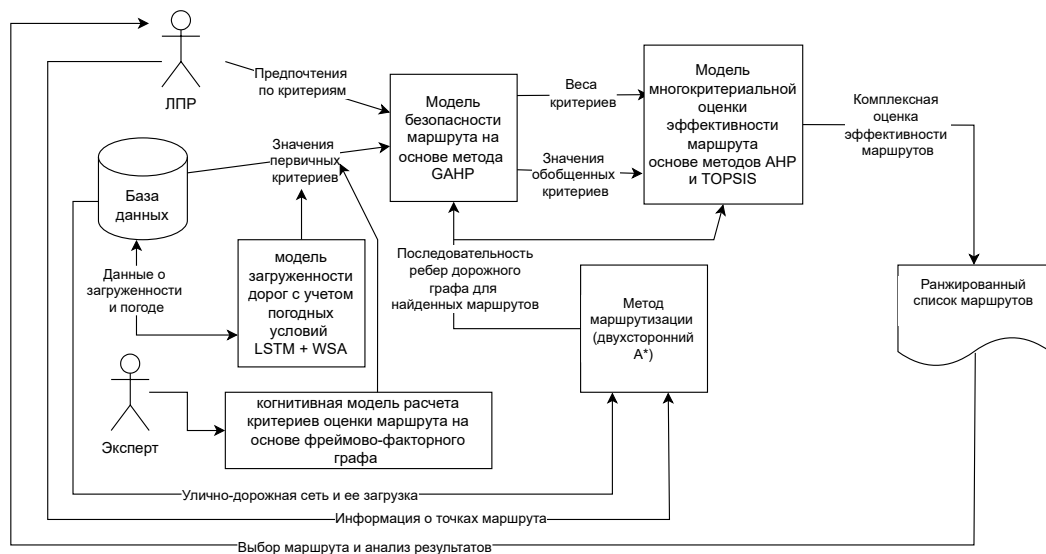
- 1) расчет нового комплексного критерия эффективности, включающего показатели: загруженность улично-дорожной сети, безопасность движения для выбора наилучшего маршрута для транспортных средств с точки зрения;
- 2) формирование аналитического представления модели машинного обучения для прогноза загруженности улично-дорожной сети с учетом погодных факторов;
- 3) расчет безопасности движения на конкретных сегментах улично-дорожной сети;
- 4) расчет комплексного критерия эффективности маршрута в случае недостаточных данных для определения загруженности методами машинного обучения.

Для реализации заданных требований в диссертации разработаны следующие взаимосвязанные модели:

- 1) модель загруженности сетевой структуры с учетом погодных условий для прогнозирования на основе сопоставления метрик точности архитектур искусственных нейронных сетей и последующего выбора архитектуры LSTM + WSA и оптимизации гиперпараметров (включая выбор количества слоев, размера пакета, размера скрытого слоя, количества эпох обучения, скорости обучения);

- 2) модель безопасности маршрута, использующую открытые данные о состоянии дорожного покрытия, средствах регулирования движения (светофоры, переходные переходы), средствах фиксации нарушений (видеокамеры автоматической фиксации нарушений, радары), дорожно-транспортных происшествиях, радиоактивном загрязнении;
- 3) когнитивная модель расчета показателей оценки маршрута на основе фреймово-факторного графа экспертных оценок, позволяющая оценить сетевую структуру в случае недостаточности эмпирических данных;
- 4) модель многокритериальной оценки эффективности маршрута на основе сходства с идеальным решением, позволяющая рассчитать комплексный показатель эффективности маршрута.

На рисунке 2.1 представлена схема информационного взаимодействия моделей (прямоугольники – это модели, а стрелки показывают какие данные между элементами схемы передаются).



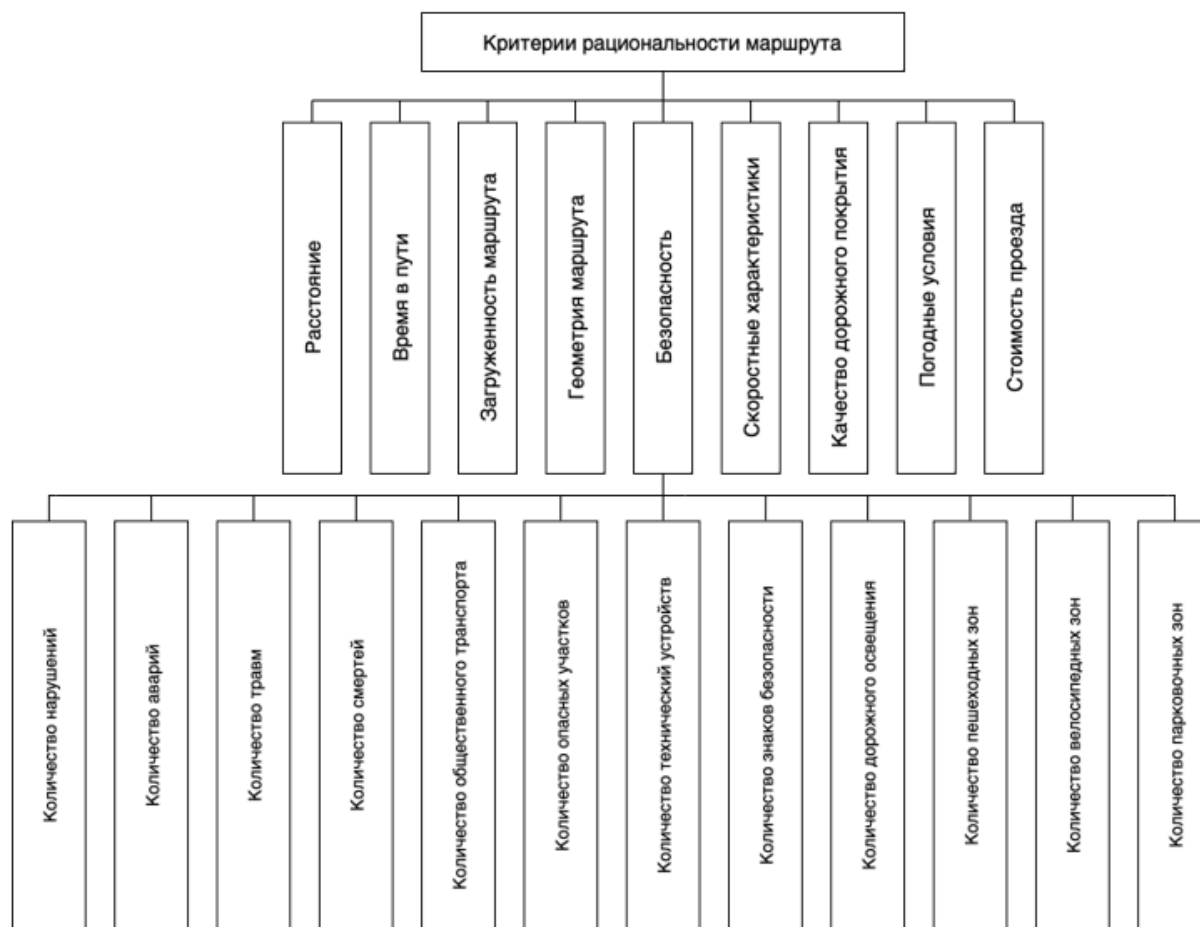
Источник: составлено автором.

Рисунок 2.1 – Структурно-логическая схема информационного взаимодействия комплекса моделей и методов маршрутизации транспортных средств

2.1 Модель многокритериальной оценки эффективности маршрута

Определим критерии оценки эффективности маршрута. Согласно [45] критерии для оценки эффективности маршрута должны учитывать

особенности движения, загруженность, безопасность, геометрию транспортной сети и погодные условия. Структура критериев и показателей представлена на рисунке 2.2.



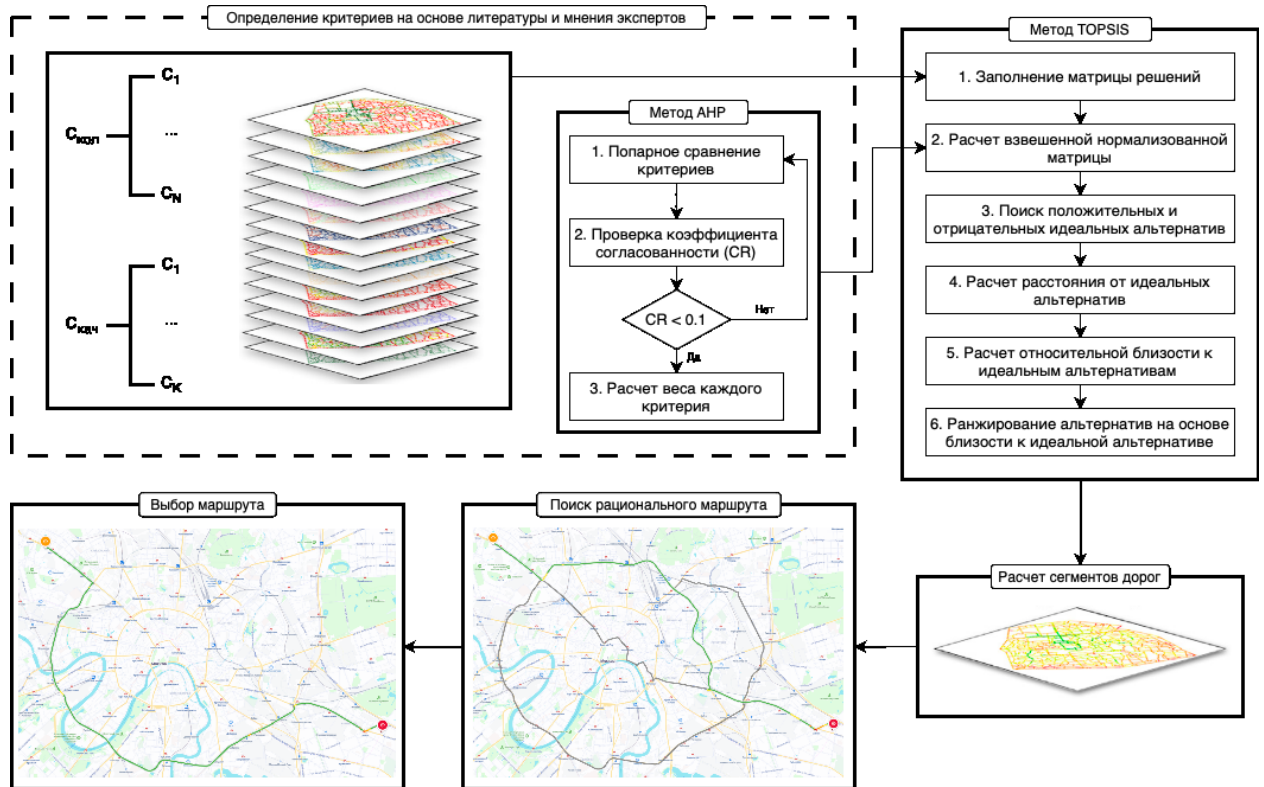
Источник: составлено автором.

Рисунок 2.2 – Критерии и показатели, влияющие на выбор рационального маршрута

В данной задаче математически строгий подход к ранжированию маршрутов на основе оптимальности по Парето не является возможным в связи с тем, что используется 21 показатель и практически все альтернативы становятся недоминируемыми. Поэтому в диссертации при оценке и выборе маршрута, применялось комплексирование нескольких методов многокритериального анализа, в том числе:

- 1) метод анализа иерархий для расчета веса критериев и показателей;
- 2) метод определения порядка предпочтений по сходству с идеальным решением для ранжирования маршрутов и выбора решения на основе мнений ЛПР и экспертов [72].

На рисунке 2.3 представлен процесс принятия решений о выборе маршрута с использованием разрабатываемых моделей.



Источник: составлено автором.

Рисунок 2.3 – Формализация процесса принятия решений для поиска альтернативных маршрутов

В таблице 2.1 критерии разделены на группы, что обеспечивает возможность попарного сравнения небольшого числа критериев в группе. Небольшие группы необходимы, так как ЛПР в состоянии одновременно анализировать не более 7 ± 2 критерия [14]. Количественные критерии можно описать при помощи математического аппарата на основе исторических данных и актуальной информации. К ним относятся:

- 1) расстояние – дальность маршрута и потраченное топливо;
- 2) время в пути – длительность поездки;
- 3) загруженность – плотность потока движения (пробки);
- 4) погодные условия – метеорологические данные, влияющие на среднюю скорость и загруженность маршрута;
- 5) безопасность – данные о ДТП на маршруте;
- 6) стоимость проезда – наличие платных дорог на маршруте.

Качественными критериями в данной задаче являются критерии, которые относятся к особенностям или объектам дороги, их состояние определяется экспертами. К ним относятся:

- 1) геометрия маршрута – класс дороги и наличие резких поворотов;
- 2) качество дорожного покрытия – тип дороги (новый асфальт, выбоины на дороге, проселочная дорога и тому подобное);
- 3) скоростные характеристики – максимальная скорость, ограничения на маршруте и другие характеристики.

Таблица 2.1 – Объективные и экспертные критерии

Количественные критерии	Качественные критерии
Расстояние, км.	Геометрия маршрута
Время в пути, мин.	Качество дорожного покрытия
Загруженность маршрута, в процентах	Скоростные характеристики
Погодные условия, в процентах	—
Безопасность, в процентах	—
Стоимость проезда, руб.	—

Источник: составлено автором.

В задаче выбора маршрута для перемещения транспортных средств ЛПР обычно является физическое или юридическое лицо. ЛПР несет ответственность за принятие окончательного решения о выборе маршрута.

Основная роль ЛПР – определить критерии и структуру предпочтений и, возможно, с привлечением экспертов выбрать окончательный вариант маршрута. Однако, при планировании маршрутов требуется не только учитывать процессы перемещений, которые определяются конкретным водителем.

В сложных системах, при принятии решений об изменении транспортной инфраструктуры необходимо комплексно оценить, как изменяться возможные маршруты в транспортной сети. Другое назначение комплексной оценки маршрутов – это решение задач оптимизационных задач логистики – составление планов грузоперевозок, в процессе которого необходимо учитывать факт конфликта значений критериев при выборе

маршрута (например, более безопасный маршрут может быть более долгим), что обуславливает необходимость поиска компромиссов между критериями.

Будем полагать, что выбранный маршрут должен иметь наименьшее расстояние от положительного идеального решения (маршрута) и наиболее удаленное от отрицательного идеального решения (маршрута). Тогда математически модель можно выразить в виде формул, применяемых в следующей последовательности:

1) расчет нормализованной матрицы решений (N), где нормализованное значение N_{ij} рассчитывается по формуле (2.1)

$$N_{ij} = \frac{X_{ij}}{\sqrt{\sum_{j=1}^n X_{ij}^2}}, \quad (2.1)$$

где $i = \overline{1, m}$;

$j = \overline{1, n}$;

X_{ij} – значения показателей;

2) расчет взвешенной нормализованной матрицы решений (V), где взвешенное нормализованное значение V_{ij} описывается формулой (2.2)

$$V_{ij} = W_i N_{ij}, \quad (2.2)$$

где $i = \overline{1, m}$;

$j = \overline{1, n}$;

W_i – вес i -го показателя и $\sum_{i=1}^m W_i = 1$;

3) определение положительного идеального и отрицательного идеального решения описывается формулой (2.3). Положительное идеальное решение (A^+) для максимизируемых показателей включает наивысшие значения показателя, а для минимизируемых показателей включает наименьшие значения показателя. Кроме того, идеальное отрицательное решение (A^-) для максимизируемого и минимизируемого показателей включает наименьшее и наибольшее значения показателя соответственно (2.3)

$$\begin{aligned}
 A^+ &= \{V_1^+, \dots, V_n^+\}, \text{ где } V_j^+ = \begin{cases} \text{Max}_i V_{ij}, & \text{если } j \in K \\ \text{Min}_i V_{ij}, & \text{если } j \in K' \end{cases} \\
 A^- &= \{V_1^-, \dots, V_n^-\}, \text{ где } V_j^- = \begin{cases} \text{Min}_i V_{ij}, & \text{если } j \in K \\ \text{Max}_i V_{ij}, & \text{если } j \in K' \end{cases}
 \end{aligned} \tag{2.3}$$

где $i \in \{1, 2, \dots, m\};$

$j \in \{1, 2, \dots, n\};$

K соответствуют положительным показателям;

K' соответствуют отрицательным показателям;

4) расчет мер разделения, используя n -мерное евклидово расстояние.

Расстояние каждой альтернативы от положительного идеального и отрицательного идеального решения описываются формулами (2.4)

$$\begin{aligned}
 d_j^+ &= \left\{ \sum_{i=1}^m (V_{ij} - V_i^+)^2 \right\}^{\frac{1}{2}}, \\
 d_j^- &= \left\{ \sum_{i=1}^m (V_{ij} - V_i^-)^2 \right\}^{\frac{1}{2}},
 \end{aligned} \tag{2.4}$$

где $j = \overline{1, n};$

5) расчет относительной близости к идеальному решению.

Относительная близость альтернативы A_j по отношению к A^+ определяется по формуле (2.5)

$$R_j = \frac{d_j^-}{(d_j^+ + d_j^-)}, j = \overline{1, n}, \tag{2.5}$$

где $d_j^+ \geq 0;$

$d_j^- \geq 0;$

$R_j \in [0, 1];$

б) ранжирование порядка альтернатив производится на основе значений R_j . Каждая альтернатива с большим R_j лучше, чем другие альтернативы.

Приведенные выше операции в отличие от традиционных подходов к маршрутизации транспорта (основанных на минимизации по одному критерию времени проезда с переводом других критериев в ограничения), позволяют искать компромиссные решения по нескольким критериям одновременно. Кроме того, разработанная модель многокритериальной оценки эффективности маршрута на основе сходства с идеальным решением (2.3) отличается от известных решений [3] тем, что:

1) использует гибкое ранжирование показателей: при котором ЛПР может изменять приоритеты показателей в зависимости от конкретной ситуации. В отличие от существующих моделей, фиксированных в своих приоритетах, данная модель позволяет адаптировать вес каждого показателя под текущие потребности пользователя, обеспечивая повышение эффективности маршрутизации транспортных средств за счет комплексного учета факторов: нагрузки на сетевую структуру, погодных факторов, факторов безопасности движения;

2) с использованием парных сравнений значений критериев, модель интегрирует в единую комплексную оценку как количественные, так и качественные показатели, в то числе такие как геометрия маршрута и качество дорожного покрытия, что комплексно учесть показатели выбора маршрута по сравнению с моделями, учитывающими только на количественные показатели (например, расстояния и времени в пути), что расширяет возможности использования модели в сложных условиях транспортного планирования;

3) адаптирована для работы с множеством источников данных, включая как открытые, так и коммерческие данные, что обеспечивает полное и актуальное представление о дорожной обстановке. Большинство существующих моделей ограничены лишь одним или двумя источниками

данных, что снижает их точность в условиях изменяющихся транспортных потоков и погодных условий.

Проведенный анализ показал, что ключевыми критериями при выборе маршрута являются критерии основанные на оценке его загруженности. Значения этих критериев динамичны, то есть подвержены существенным изменениям в процессе маршрутизации транспорта, поэтому важно обеспечить удовлетворительную точность их прогнозирования.

2.2 Модель загруженности сетевой структуры с учетом погодных условий

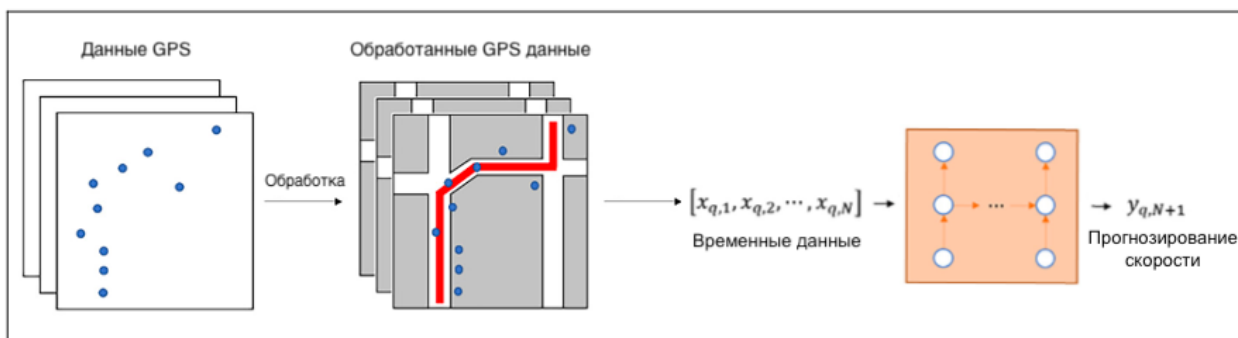
Модели загруженности сетевой структуры с учетом погодных условий предназначены для прогнозирования средней скорости на каждой дуге улично-дорожной сети. В решаемой задаче прогнозирования как правило имеются достаточно большие объемы данных о скоростях движения в прошлом. Однако при этом модели поведения участников движения с необходимой точностью построить достаточно проблематично ввиду сложности получения данных о конкретных участниках движения, в этой связи в диссертации были применены регрессионные модели.

Для обеспечения прозрачности расчетов процесс выбора архитектуры и параметров модели был разделен на этапы:

- сбор данных: а) исторические данные о транспортных потоках; б) погодные данные – температура, осадки;
- подготовка данных: а) форматирование временных рядов, представленных как последовательности средней скорости движения по участкам сети;
- обучение модели: а) настройка гиперпараметров – количество слоев, размер скрытого слоя, размер временного окна; б) оптимизация механизма внимания для выделения значимых временных периодов.

Для прогнозирования временных рядов хорошие результаты показывают модели на основе глубокого обучения RNN

и ее производные [103]. Задача прогнозирования загруженности транспортной сети заключается в определении средней скорости (временных рядов) на выбранном участке дороги. При этом входными данными являются данные о средней скорости движения на участке дороги, а результатом работы модели является оценка прогнозируемой средней скорости на дороге. На рисунке 2.4 представлена структурно-логическая схема модели прогнозирования загруженности дорожной сети.



Источник: составлено автором.

Рисунок 2.4 – Структурно-логическая схема модели прогнозирования загруженности дорожной сети

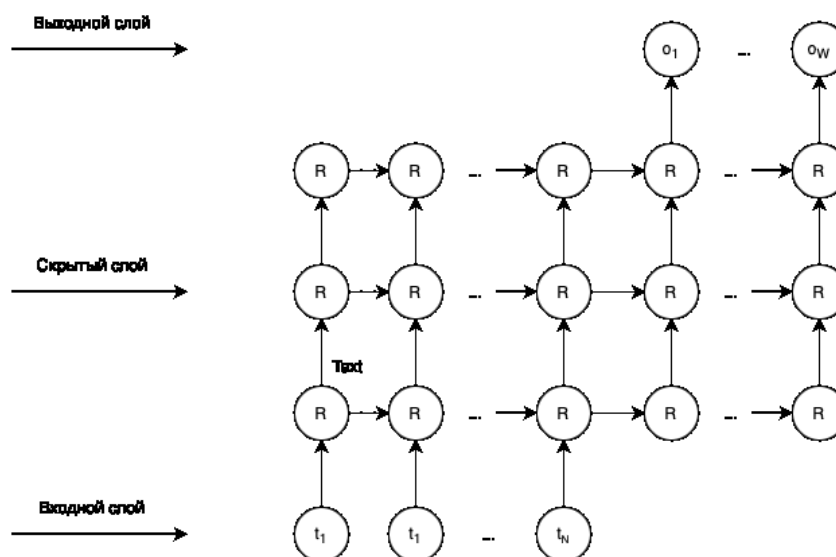
Большие объемы данных, возможность появления сезонных факторов (периодического возрастания и снижения нагрузки на сетевую структуру), появление дополнительных признаков, связанных с погодными условиями, требует применения методов машинного обучения, которые способны работать с пространством признаков высокой размерности и самообучаться для повышения точности прогнозирования. Способностями к самообучению обладают искусственные нейронные сети, поэтому целесообразно их применение в данной задаче. Длина входных единиц временного окна, используемых при прогнозировании, определяется как средняя скорость движения на дороге в течение 5 минут и описывается формулой (2.6)

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1N} \\ x_{21} & x_{22} & \dots & x_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ x_{Q1} & x_{Q2} & \dots & x_{QN} \end{bmatrix}, \quad (2.6)$$

где N – длина входных единиц временного окна;

Q – количество выбранных дорог.

Для прогноза использована модель RNN, так как она обладает способностями к самообучению в условиях меняющейся обстановки, способна работать с временными рядами. Архитектура модели RNN, используемая в работе представлена на рисунке 2.5 и включает один входной слой, три скрытых слоя и один выходной слой. Входной и выходной слой по определению RNN должен быть один [103], число промежуточных слоев может быть и другим в данной работе оно выбиралось в процессе настройки гиперпараметров в следующем разделе. Целесообразно использовать минимальное число промежуточных слоев (так как это делает модель проще и снижает вычислительную сложность) при обеспечении не обходимой точности оценки.



Источник: составлено автором.
Рисунок 2.5 – Архитектура модели RNN

Входные данные временного ряда имеют вид (2.7)

$$x_i = [x_{i,t}^k, x_{i,t+1}^k, \dots, x_{i,t+N}^k], \quad (2.7)$$

где i – индекс выборки;

$x_{i,t}^k$ – средняя скорость движения по k -й дороге в момент времени t .

Единица временного окна во входных данных определяется как средняя скорость движения дорожной сети в течение 5 минут. Расчет в скрытом слое можно описать по формуле (2.8)

$$s_t = f(Vx_i + Uu_i + b_u), \quad (2.8)$$

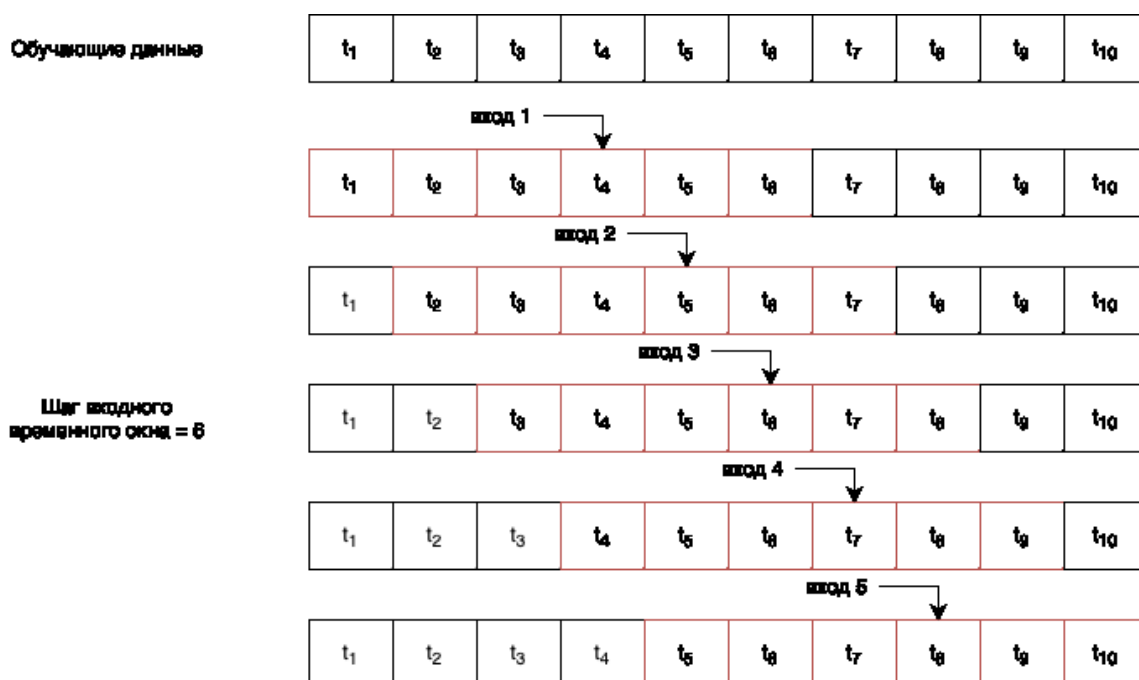
где f – функция активации скрытого слоя;
 V – весовая матрица выходного слоя;
 x_i – входной вектор i ;
 U – весовая матрица распределительного слоя;
 u_i – состояние скрытого слоя;
 b_u – вектор сдвигов скрытого слоя.

Расчет выходного слоя можно описать по формуле (2.9)

$$o_t = g(Wu_i + b_o), \quad (2.9)$$

где g – функция активации выходного слоя;
 W – весовая квадратная матрица обратных связей скрытого слоя;
 b_o – вектор сдвигов выходного слоя.

Входные данные временных рядов изображены на рисунке 2.6.



Источник: составлено автором.

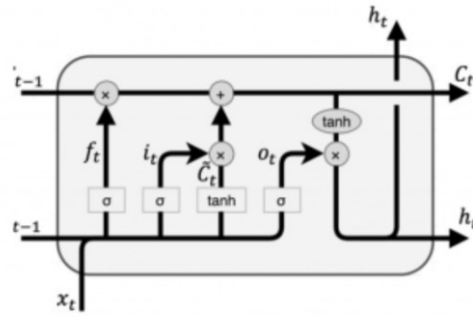
Рисунок 2.6 – Организация входных данных их временных рядов

Таким образом модель RNN позволяет работать с историческими данными произвольной длины, что и необходимо для решаемой задачи, так как продолжительность временного ряда с данными о средней скорости не фиксирована.

Приведенные формулы 2.8-2.9 описывают модель RNN в достаточно обобщенном виде. Так для применения RNN на практике необходимо определить конкретный вид функций в них входящих. Для прогнозирования загруженности сети воспользуемся из разновидностей RNN моделью нейронной сети с долгой краткосрочной памятью [95].

В отличие от других вариантов RNN, LSTM сеть хорошо приспособлена к обучению на задачах прогнозирования временных рядов в случаях, когда важные события разделены временными лагами с неопределённой продолжительностью и границами, которые характерны для дорожного движения. Во входных данных о загруженности транспортной сети возможны периоды отсутствия данных, они связаны как со сбоями оборудования по сбору данных, так и с периодами отсутствия какого-либо движения на участке улично-дорожной сети. Невосприимчивость к длительности временных разрывов даёт LSTM преимущество по отношению к альтернативным RNN, скрытым марковским моделям и другим методам обучения для последовательностей в сфере моделирования движения на сетевой структуре [37].

На рисунке 2.7 представлена внутренняя структура скрытого слоя достаточно эффективная для решения задач самообучения и содержащая три вентиля: входной, забывания и выходной. Входные данные скрытого слоя включают выходные данные предыдущего скрытого слоя. Функция шлюза забывания заключается в сокрытии некоторой информации о состоянии в ячейке. Такое сокрытие обеспечивает большей приоритет недавним наблюдениям и меньший приоритет данным, которые наблюдались давно, что необходимо при нестационарных процессах, а временной ряд загруженности дорог не стационарен [91].



Источник: составлено автором.

Рисунок 2.7 – Структура скрытого слоя LSTM

Расчет функции забывания данных о загруженности графа улично-дорожной сети, f_t выполняется по формуле (2.10)

$$f_t = \sigma(W_f \times [o_{t-1}, x_t] + b_f), \quad (2.10)$$

где σ – нелинейная функция активации;

W – весовая матрица;

b – смещение;

o_{t-1} – выход скрытого слоя в момент времени $t - 1$;

x_t – вход в момент t .

Входной вентиль можно описать по формуле (2.11)

$$i_t = \sigma(W_i \times [o_{t-1}, x_t] + b_i). \quad (2.11)$$

Информация о ячейке-кандидате $\hat{C}_t$ в момент времени t может быть описана по формуле (2.12)

$$\hat{C}_t = \tanh(W_c \times [o_{t-1}, x_t] + b_c), \quad (2.12)$$

где $\tanh$ – функция активации.

Информация о ячейке C_t в момент времени t описывается по формуле (2.13)

$$C_t = f_t \times C_{t-1} + i_t \times \hat{C}_t. \quad (2.13)$$

Выходной вентиль h_t можно описать по формуле (2.14). Здесь описывается работа вентиля забывания, который удаляет данные по прошлым

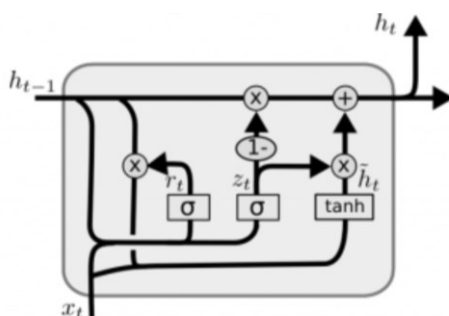
периодам из текущего состояния модели для учета нестационарности временного ряда по формуле (2.14)

$$h_t = \sigma(W_o \times [o_{t-1}, x_t] + b_o). \quad (2.14)$$

Для расчета выхода промежуточного слоя используется формула (2.15)

$$o_t = h_t \times \tanh(C_t). \quad (2.15)$$

Модель управляемого рекуррентного блока (GRU) является развитием модели LSTM. По сравнению с LSTM у данного механизма меньше параметров, т.к. отсутствует выходной вентиль [73]. За счет этого модель становится проще и потенциально быстрее обучается. Модель GRU объединяет два вентиля: входной и забывания в один вентиль. Внутренняя структура скрытого блока GRU показана на рисунке 2.11. Скрытый блок GRU содержит два вентиля: сброса (он определяет, как объединить новый прогноз загруженности с предыдущей памятью о состоянии модели GRU и шлюз обновления [38]. Учет такого нового фактора как забывание предыдущих наблюдений загруженности дорожного графа обеспечивает повышение точности его прогнозирования, за счет того, что при изменении динамики движения по улично-дорожной сети влияние наблюдений из прошлых периодов уменьшается чтобы отразить нестационарность временного ряда, который мы анализируем. Структура скрытого блока GRU представлена на рисунке 2.8.



Источник: составлено автором.

Рисунок 2.8 – Структура скрытого блока GRU для прогноза загруженности сегмента улично-дорожной сети

Функцию обновления прогноза загруженности сегмента улично-дорожной сети z_t можно описать по формуле (2.16)

$$z_t = \sigma(W_z \times [h_{t-1}, x_t]), \quad (2.16)$$

где h_{t-1} – выходные данные скрытого слоя в момент времени $t - 1$;

x_t – входные данные в момент времени t ;

W_z – весовая матрица вентиля обновления.

Функцию сброса r_t можно описать по формуле (2.17)

$$r_t = \sigma(W_r \times [h_{t-1}, x_t]), \quad (2.17)$$

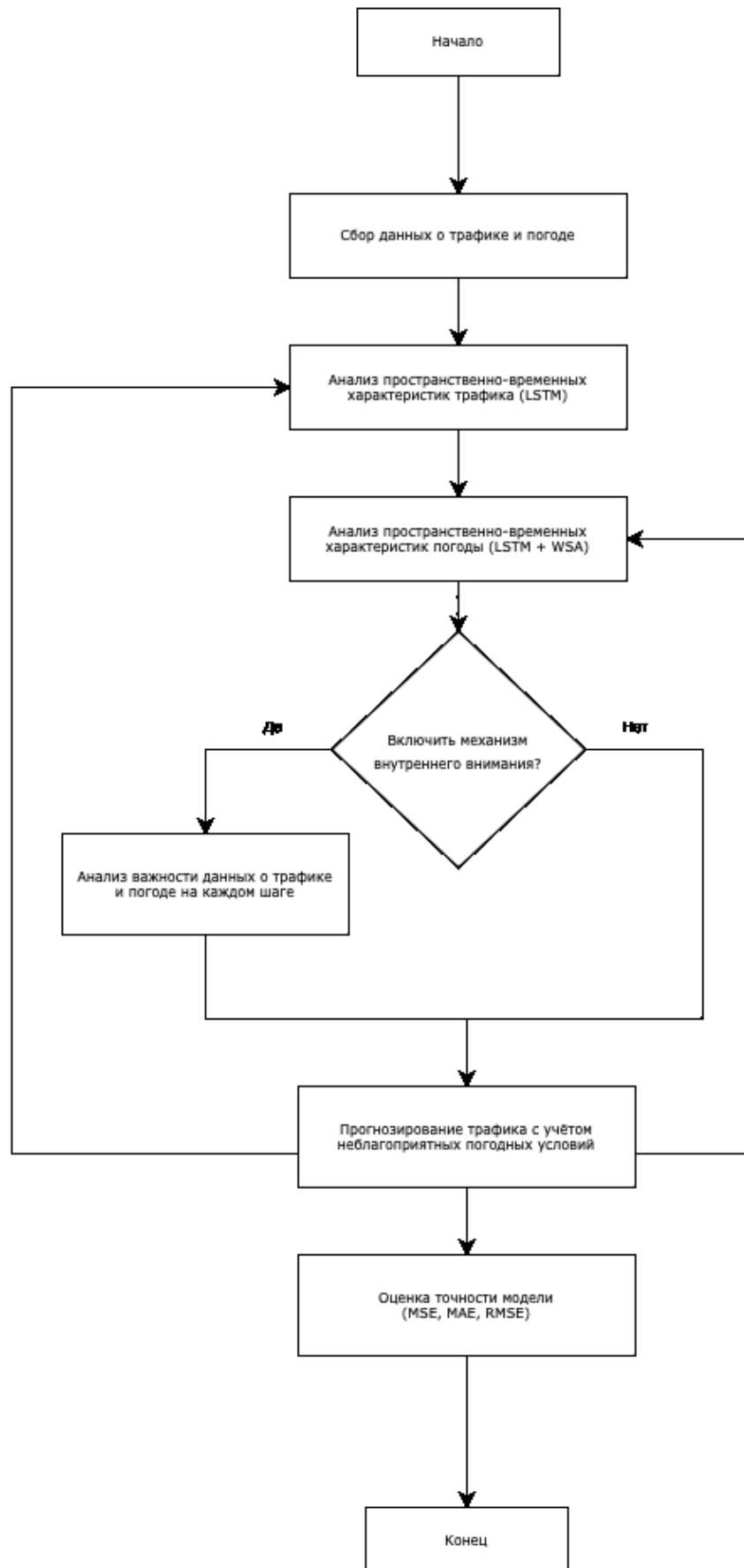
где W_r – весовая матрица функции сброса.

Выход скрытого слоя h_t можно описать по формуле (2.18)

$$\begin{aligned} \tilde{h}_t &= \tanh(W \times [r_t \times h_{t-1}, x_t]), \\ h_t &= (1 - z_t) \times h_{t-1} + z_t \times \tilde{h}_t, \end{aligned} \quad (2.18)$$

где $\tilde{h}_t$ – информация о ячейке-кандидате в момент времени t .

По сравнению с моделью LSTM модель GRU имеет более простую внутреннюю структуру, поэтому скорость ее обучения выше при работе с большими наборами обучающих данных. Однако в представленной модели GRU не учитываются погодные условия, влияющие на загруженность улично-дорожной сети. Поэтому, для учета влияния погодных условий на загруженность транспортной сети создана модель на основе глубокого обучения – LSTM + WSA, учитывающая такие погодные факторы, как дождь, туман и снег [69; 166]. Она показала наилучшие значения точности. LSTM + WSA представляет собой рекуррентную нейронную сеть для пространственно-временного прогнозирования с механизмом внутреннего внимания для увеличения производительности при работе с большими наборами данных. Модель объединяет данные прогнозирования загруженности с блоком погодных условий. Структурно-логическая схема модели загруженности с учетом погодных условий представлена на рисунке 2.9.



Источник: составлено автором.

Рисунок 2.9 – Структурно-логическая схема модели загруженности с учетом погодных условий с выбранными по результатам экспериментов слоями

Согласно результатам, приведенным в работах [98; 106; 161], для повышения точности прогноза загруженности дорожного графа в разработанном научно-методическом аппарате учтен фактор состояния погоды. Данные о транспортных потоках описываются по формуле (2.19)

$$X_s = (x_{1,s}, x_{2,s}, \dots, x_{T,s}), \quad (2.19)$$

где $x_{t,s}$ – данные о потоке в интервале времени t для участка s ;
 T – количество интервалов времени.

Для учета факторов погоды в модели загруженности введем модель внимания, которая позволяет учесть зависимость средней скорости движения от факторов дождя, снега, ветра. Модель внимания, используется в рекуррентных нейронных сетях и сверточных нейронных сетях для поиска взаимосвязей между различными частями входных и выходных данных. Модель внимания связывает ближайшие значения временного ряда по формуле (2.20)

$$\begin{aligned} Q_i &= W_q \times X_i, \\ K_i &= W_k \times X_i, \\ V_i &= W_v \times X_i, \end{aligned} \quad (2.20)$$

где Q_i – запрос;
 K_i – ключ;
 V_i – значение;
 X_i – входной слой;
 W_q, W_k, W_v – веса параметров.

Взаимосвязь между Q_i и K_i можно рассчитать по формуле (2.21)

$$Sim(i) = Q_i \times K_i, \quad (2.21)$$

где K_i – ключ i -го значения для X_i , $i \in \{1, 2, \dots, N\}$;
 N – количество значений X .

Тогда в выходном слое значение загруженности улично-дорожной сети вычисляется по формуле (2.22)

$$a(i) = \text{Softmax}(\text{Sim}(i)) = \frac{e^{\text{Sim}(i)}}{\sum_{i=1}^N e^{\text{Sim}(i)}}, \quad (2.22)$$

где Softmax – многомерная логистическая функция, применяемая для задач классификации, когда количество возможных классов больше двух.

Функция преобразует вектор z размерности K в вектор σ той же размерности, где каждая координата σ_i полученного вектора представлена вещественным числом в интервале $[0, 1]$ и сумма координат равна 1. Координаты σ_i вычисляются по формуле (2.23)

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{i=1}^N e^{z_N}}. \quad (2.23)$$

Координаты σ_i полученного вектора при этом трактуются как вероятности того, что объект принадлежит к классу i . Вектор-столбец z рассчитывается по формуле (2.24)

$$z = \omega^T x - \theta, \quad (2.24)$$

где ω^T – транспонированная матрица весовых коэффициентов признаков, имеющая размерность $K \times M$;

x – вектор-столбец признаков объекта размерности $M \times 1$;

θ – вектор-столбец с пороговыми значениями размерности $K \times 1$;

K – количество классов объектов;

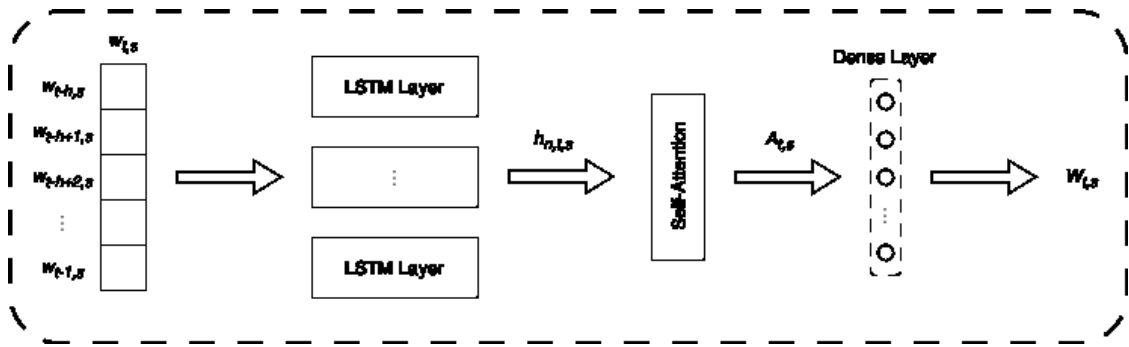
M – количество признаков объектов.

Таким образом, значение переменной внимания вычисляется по формуле (2.25)

$$A(Q_i, K_i, V_i) = \sum_{i=1}^N [a(i) \times V(i)], \quad (2.25)$$

где $V(i)$ – значение i -й переменной входного слоя X .

Структурно-логическая схема расчета факторов погоды представлена на рисунке 2.10. На ней анализируется взаимосвязь между данными о погодных условиях и транспортных потоках. Кроме того, веса блоков трафика и погоды определяются автоматически во время обучения модели. Если данные о погоде оказывают существенное влияние на прогноз транспортного потока в течение одного периода, то вес факторов погоды увеличивается. Новизна предложенной модели заключается в оригинальной формализации погодных условий, обоснованном выборе слоев рекуррентной нейронной сети для расчета загруженности сегментов дорожного графа.



Источник: составлено автором.

Рисунок 2.10 – Структурно-логическая схема модели прогнозирования загруженности с учетом погодных условий

Исследование показало, что наилучшим образом, пространственно-временные характеристики погодных данных вычисляются по формуле (2.26)

$$\begin{aligned}
 i_{n,t,s} &= \sigma_{n,s}^i(W_{n,s}^i \otimes [\bar{h}_{n,t-1,s}, \bar{w}_{n,t,s}] + b_{n,s}^i), \\
 f_{n,t,s} &= \sigma_{n,s}^f(W_{n,s}^f \otimes [\bar{h}_{n,t-1,s}, \bar{w}_{n,t,s}] + b_{n,s}^f), \\
 o_{n,t,s} &= \sigma_{n,s}^o(W_{n,s}^o \otimes [\bar{h}_{n,t-1,s}, \bar{w}_{n,t,s}] + b_{n,s}^o), \\
 c_{n,t,s} &= f_{n,t,s} \odot c_{n,t-1,s} + i_{n,t,s} \odot \tanh(W_{n,s}^c \otimes [\bar{h}_{n,t-1,s}, \bar{w}_{n,t,s}] + b_{n,s}^c), \\
 \bar{h}_{n,t,s} &= o_{n,t,s} \odot \tanh(c_{n,t,s}),
 \end{aligned} \tag{2.26}$$

где $i_{n,t,s}$, $f_{n,t,s}$ и $o_{n,t,s}$ – выходные данные входного, забывания и выходного вентилей n -го слоя для времени t и секции s ;

$$n \in \{1, 2, 3, \dots, N\};$$

N – количество слоев;

$\bar{w}_{n,t,s}$ – входной вектор n -го слоя для времени t и секции s ;

$c_{n,t,s}$ и $c_{n,t-1,s}$ – состояния ячеек n -го слоя для интервала времени $[t-1, t]$ и секции s ;

$\bar{h}_{n,t,s}$ и $\bar{h}_{n,t-1,s}$ – состояние ячеек памяти для n -го слоя интервала времени $[t-1, t]$ и секции s ;

$\sigma_{n,s}^i, W_{n,s}^i$ и $b_{n,s}^i$ – функция активации, вес и вектор смещения для элемента $i_{n,t,s}$;

$\sigma_{n,s}^f, W_{n,s}^f$ и $b_{n,s}^f$ – функция активации, вес и вектор смещения для элемента $f_{n,t,s}$;

$\sigma_{n,s}^o, W_{n,s}^o$ и $b_{n,s}^o$ – функция активации, вес и вектор смещения для элемента $o_{n,t,s}$;

$W_{n,s}^c$ и $b_{n,s}^c$ – вес и вектор смещения для элемента $c_{n,t,s}$.

Запоминание информации о погодных условиях обеспечивает решение задачи прогнозирования загруженности улично-дорожной сети, для этого на каждом шаге факторы внимания вычисляются по формуле (2.27)

$$\begin{aligned}\tilde{Q}_{t,s} &= \tilde{W}_q \times \bar{h}_{n,t,s}, \\ \tilde{K}_{t,s} &= \tilde{W}_k \times \bar{h}_{n,t,s}, \\ \tilde{V}_{t,s} &= \tilde{W}_v \times \bar{h}_{n,t,s},\end{aligned}\tag{2.27}$$

$$\tilde{A}_{t,s} = \text{Attention}(\tilde{Q}_{t,s}, \tilde{K}_{t,s}, \tilde{V}_{t,s}),$$

где $\tilde{Q}_{t,s}, \tilde{K}_{t,s}$ и $\tilde{V}_{t,s}$ – запрос, ключ и значение для $\bar{h}_{n,t,s}$;

$\tilde{W}_q, \tilde{W}_k$ и $\tilde{W}_v$ – веса для $\tilde{Q}_{t,s}, \tilde{K}_{t,s}$ и $\tilde{V}_{t,s}$ соответственно;

$\tilde{A}_{t,s}$ – вектор внимания для времени t и секции s .

Слой Dense изучает вес каждого значения в векторе внимания к погодным условиям согласно формуле (2.28)

$$\widehat{W}_{t,s} = \sigma_s^W(W_s^W \times \tilde{A}_{t,s} + b_s^W), \quad (2.28)$$

где $\widehat{W}_{t,s}$ – выходной вектор для времени t и секции s ;
 σ_s^W , W_s^W и b_s^W – функция активации, вес и вектор смещения.

Выходные данные блоков трафика и погоды объединяются. $\bar{A}_{t,s}$ и $\widehat{W}_{t,s}$ объединяются в вектор, прогнозируемое значение можно рассчитать после слоя Dense по формуле (2.29)

$$\hat{y}_{t,s} = \sigma_s(W_s \times [\bar{A}_{t,s}, \widehat{W}_{t,s}] + b_s), \quad (2.29)$$

где $\hat{y}_{t,s}$ – вектор прогнозируемых значений предложенной модели для времени t и секции s ;
 σ_s , W_s и b_s – функция активации, вес и вектор смещения.

Обучение модели происходит путем минимизации функции потерь L по формуле (2.30)

$$L = \sum_s [(x_s - \hat{y}_s)^2], \quad (2.30)$$

где x_s и $\hat{y}_s$ – векторы фактического и прогнозируемого значений загруженности сегментов улично-дорожной сети [71; 93; 96].

Таким образом минимизация функции потерь ошибки загруженности сегментов улично-дорожной сети L позволяет повысить точность прогнозирования, что ведет к повышению эффективности маршрутизации транспортных средств за счет улучшения точности и оперативности прогнозирования, в части учета факторов: нагрузки на сетевую структуру и погодных факторов.

На основе анализа точности прогнозирования загруженности транспортной сети, устойчивости модели к разреженным данным о движении в прошлом и способности учитывать временные характеристики транспортного потока из различных архитектур нейросетей была выбрана

LSTM + WSA. Среди рассмотренных альтернатив были такие архитектуры, как RNN, GRU, и их вариации с механизмами внимания.

Архитектура LSTM + WSA показала лучшие результаты за счет возможности более точного учета пространственно-временных связей и влияния погодных факторов.

Выбор архитектуры проводился из множества доступных моделей, включая RNN, GRU, LSTM, и их модификации. Основными показателями оценки были:

- 1) точность прогнозирования (увеличение точности на 5-7% в сравнении с другими архитектурами);
- 2) устойчивость модели к разреженным и нестабильным данным;
- 3) производительность: скорость обучения и предсказания;
- 4) адаптивность к изменяющимся условиям (например, влияние погодных факторов).

Методы оптимизации включали перебор гиперпараметров (количество слоев, размер скрытого слоя, скорость обучения) и тестирование на реальных данных сетевой структуры. Для выбора оптимального решения использовался набор метрик: MAPE, RMSE и MAE.

Выбор архитектуры модели был основан на задаче прогнозирования временных рядов для загруженности транспортной сети. Рассматривались следующие архитектуры:

- 1) RNN. Используется для работы с последовательностями данных. Ограничения: «эффект затухания градиента», что снижает точность при длинных временных зависимостях;
- 2) GRU. Модификация RNN с упрощенной структурой, содержащая меньше параметров. Преимущества: быстрее обучается, меньше памяти, но не столь эффективна при сложных временных зависимостях;
- 3) LSTM. Усовершенствованная RNN с механизмом долговременной памяти. Преимущества: эффективно обрабатывает длинные временные зависимости;

4) LSTM + WSA. Расширенная версия LSTM, включающая механизм внимания. Преимущества: выделяет наиболее важные временные периоды, что особенно полезно для учета сезонных и погодных факторов.

Модель нейросети LSTM обеспечивает учет наблюдений за дорожным движением для продолжительного периода времени, не фиксированной длины, а добавление механизма взвешенного внимания (WSA) позволило учесть погодные условия, которые привели к увеличению точности. Подробные описания классической LSTM приведены в работе [101], а механизм внимания был впервые подробно исследован в работе [161]. Механизм внимания позволяет модели LSTM + WSA выделять наиболее важные временные интервалы в данных, используя взвешенные параметры значимости, что повышает точность прогнозирования загруженности улично-дорожной сети и устойчивость к шуму в данных. Математическое описание этого механизма подробно рассмотрено в [161]. Модель LSTM + WSA была выбрана для прогнозирования загруженности дорожной сети за счет ее способности обрабатывать временные ряды с учетом влияния погодных факторов. Механизм внимания в этой модели выделяет наиболее важные временные интервалы, что обеспечивает потенциальную возможность повышения точности прогноза.

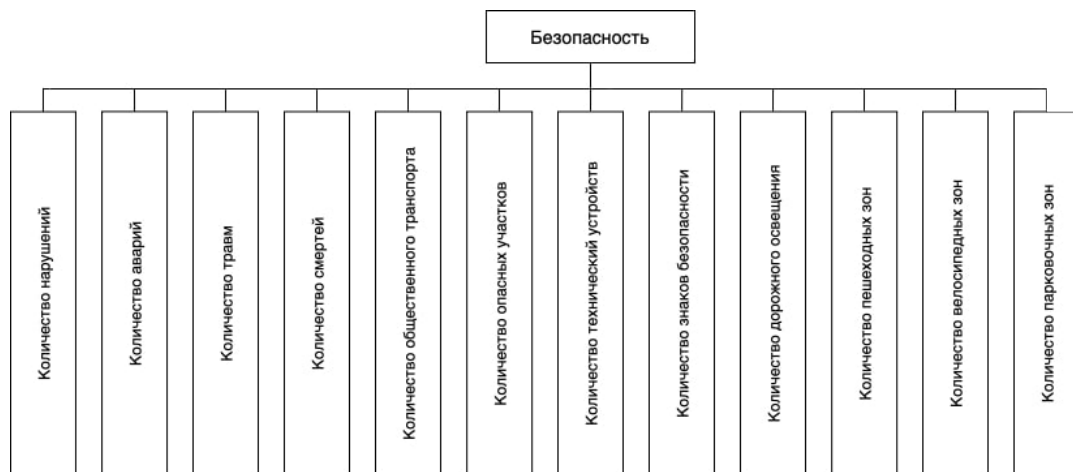
Таким образом, формулы (2.18)-(2.30) определяют последовательные шаги обработки данных, обучения модели и интеграции погодных факторов для повышения точности прогнозирования загруженности транспортной сети. Эти формулы отражают процесс построения и обучения модели LSTM + WSA, которая оптимально использует временные и погодные данные для прогнозирования загруженности транспортной сети, повышая точность маршрутизации транспортных средств. Результаты вычислительных экспериментов, подтверждающих требуемую точность на заданном горизонте прогнозирования приведены в главе 4.

Таким образом модель имеет в составе следующие компоненты рекуррентную искусственную нейронную сеть для прогнозирования на

временных рядах произвольной длины, функцию забывания для учета не стационарности временного ряда, механизм внимания для учета погодных условий. Однако повышение эффективности маршрутизации транспортных средств обеспечивается не только результатом прогнозирования загруженности с необходимой точностью, важным деструктивным фактором является безопасность маршрута, для расчета которых требуется соответствующая модель.

2.3 Модель безопасности маршрута

Безопасность дорожного движения является важным фактором, влияющим на поведение участников и движение потоков в транспортной сети. Постоянная оценка безопасности маршрутов на основе актуальных данных о ДТП и технических устройствах оказывает существенное влияние на качество поездки и безопасность для всех участников транспортной сети. «По мнению экспертов в области безопасности дорожного движения существует как минимум 12 показателей» [46-47], влияющих на критерий безопасности, которые изображены на рисунке 2.11.



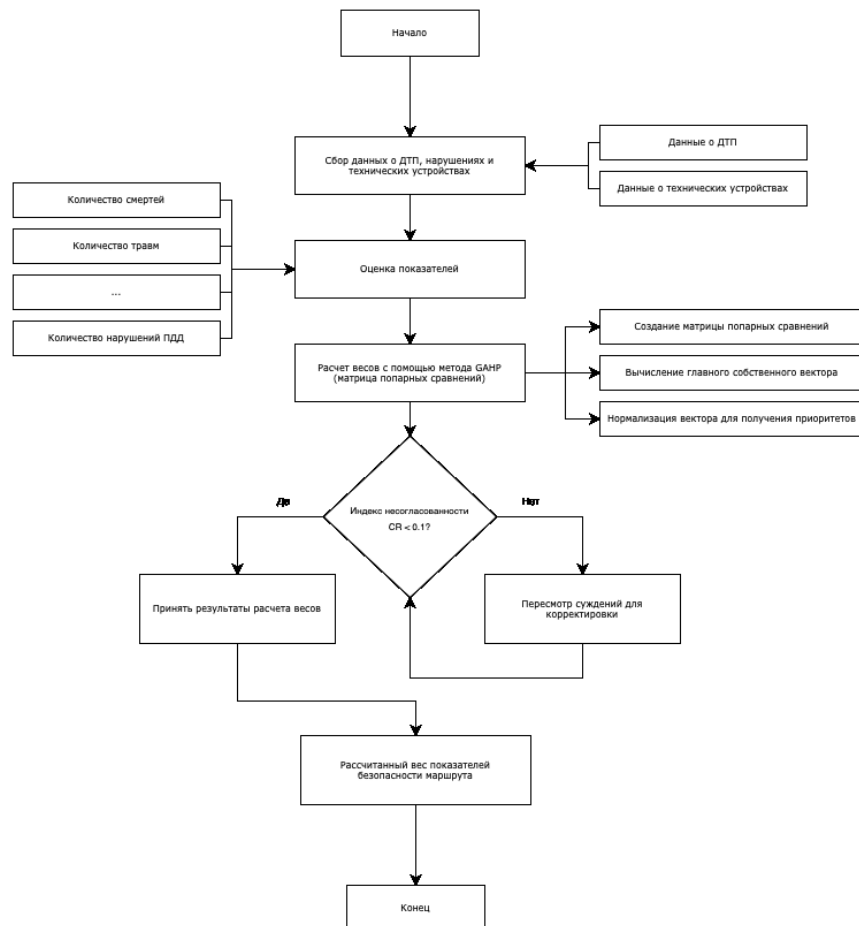
Источник: составлено автором.

Рисунок 2.11 – Показатели критерия безопасности

Среди них такие характеристики, как информация о количестве нарушений ПДД, количестве аварий, количестве травм, количестве смертельных случаев, количестве общественного транспорта, количестве

опасных участков, количестве технических устройств, количестве знаков безопасности, количестве дорожного освещения, количестве пешеходных зон, количестве велосипедных зон и количестве парковочных зон. Такие требования и определяют состав модели безопасности маршрута структурно-логическая схема которой представлена на рисунке 2.12.

Таким образом модель безопасности маршрута предназначена для расчета комплексного критерия безопасности на основе разнородных показателей, которые агрегируются с использованием методов теории принятия решений, что позволяет алгоритму построить приемлемый для ЛПР маршрут. ЛПР действительно на уровне требований/предпочтений. Маршрут, построенный в результате, должен удовлетворять ЛПР. Задачей модели является расчет комплексного критерия безопасности на основе первичных показателей, показанных на рисунке 2.11 и информации, полученной от экспертов в форме парных сравнений показателей по важности.



Источник: составлено автором.

Рисунок 2.12 – Структурно-логическая схема модели безопасности маршрута

Для создания модели безопасности маршрута, которая должна повысить эффективность маршрутизации транспортных средств, в данном подразделе должны быть решены следующие задачи:

- 1) определить показатели, которые определяют безопасность маршрута;
- 2) разработать методику для расчета весов показателей на основе мнений экспертов с проверкой согласованности их суждений;
- 3) выбрать метод свертки показателей в единую скалярную оценку безопасности маршрута.

Решение поставленных задач должно удовлетворять следующим требованиям:

- 1) учитывать показатели по количеству нарушений ПДД, количеству аварий, смертельных случаев и наличию инфраструктурных элементов (например, велосипедные и пешеходные зоны);
- 2) использовать статистику ДТП и данные о технических устройствах (в том числе, освещение, дорожные знаки), что позволяет учитывать динамику изменений в транспортной системе;
- 3) включать возможность анализа данных о ДТП и дорожных условиях на конкретных участках дороги в режиме реального времени, что существенно повышает её точность и актуальность по сравнению с моделями, не использующими актуализированные данные.

При обработке экспертных суждений есть масса других моделей, основанных на отношениях доминирования альтернатив, некоторые из них рассмотрены в работе [40]. Однако в данном случае требуется оценка достаточно большого числа альтернатив – сотни элементов улично-дорожной сети. При этом решение о выборе маршрута должно приниматься быстро – несколько секунд, поэтому предлагается метод, в котором эксперты сравнивают не конкретные альтернативы (альтернативные маршруты), а критерии, по которым они оцениваются. Таким образом при появлении новых альтернатив не требуется повторное привлечение экспертов.

Для решения поставленных задач и учетом данных требований расчет веса проводится с помощью существующего метода группового анализа иерархий (ГАНП) [12], для которого эксперты в области безопасности транспортного движения заранее заполнили матрицы попарного сравнения по шкале вербальных суждений.

Определим показатели безопасности маршрута $C_1, \dots, C_n$ некоторого уровня иерархии, для которых необходимо определить веса(приоритеты) $\omega_1, \dots, \omega_n$ влияния на некоторый элемент следующего уровня. Основным инструментом будет матрица чисел, представляющих суждения о парных сравнениях показателей безопасности маршрута. Для представления приоритетов выбран собственный вектор, соответствующий наибольшему собственному значению. Обозначим через a_{ij} число, соответствующее значимости элемента C_i по сравнению с C_j . Матрицу, состоящую из этих чисел, можно обозначить по формуле (2.31)

$$A = \|a_{ij}\|, \quad (2.31)$$

где $a_{ij} = \frac{1}{a_{ji}}$, т.е. матрица A – обратно-симметричная и согласованная.

Для согласованной матрицы является случай, когда сравнения основаны на точных измерениях, т.е. веса $\omega_1, \dots, \omega_n$ известны. Тогда (2.32)

$$a_{ij} = \frac{\omega_i}{\omega_j}, i, j = 1, 2, \dots, n. \quad (2.32)$$

Поэтому (2.33)

$$a_{ij}a_{jk} = \frac{\omega_i}{\omega_j} \times \frac{\omega_j}{\omega_k} = \frac{\omega_i}{\omega_k} = a_{ik}. \quad (2.33)$$

А также (2.34)

$$a_{ji} = \frac{\omega_j}{\omega_i} = \frac{1}{\omega_i/\omega_j} = \frac{1}{a_{ij}}. \quad (2.34)$$

Матричное уравнение модели безопасности маршрута (2.35):

$$A \times x = y, \quad (2.35)$$

где $x = (x_1, \dots, x_n)$ и $y = (y_1, \dots, y_n)$ соответствует краткой записи системы уравнений (2.36) и $i = 1, 2, \dots, n$

$$\sum_{j=1}^n a_{ij}x_j = y_i. \quad (2.36)$$

Тогда получаем (2.37) и $i, j = 1, 2, \dots, n$

$$a_{ij} \frac{\omega_j}{\omega_i} = 1, \quad (2.37)$$

$$\sum_{j=1}^n a_{ij}\omega_j = n\omega_i,$$

что эквивалентно выражению (2.38)

$$A\omega = n\omega. \quad (2.38)$$

Формула (2.37) отражает то, что ω – собственный вектор матрицы A с собственным значением n [30]. Уравнение модели безопасности маршрута, расписанное поэлементно, выглядит следующим образом (2.39)

$$A = \begin{matrix} & \begin{matrix} A_1 & A_2 & \dots & A_n \end{matrix} \\ \begin{matrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{matrix} & \begin{bmatrix} \frac{A_1}{\omega_1} & \frac{A_2}{\omega_1} & \dots & \frac{A_n}{\omega_1} \\ \frac{A_1}{\omega_2} & \frac{A_2}{\omega_2} & \dots & \frac{A_n}{\omega_2} \\ \frac{A_1}{\omega_n} & \frac{A_2}{\omega_n} & \dots & \frac{A_n}{\omega_n} \end{bmatrix} \end{matrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_n \end{bmatrix} = n \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_n \end{bmatrix}. \quad (2.39)$$

Если диагональ матрицы A состоит из единиц ($a_{ii} = 1$) и A – согласованная матрица, то при малых изменениях в a_{ij} наибольшее собственное значение λ_{max} остается близким к n , а остальные собственные значения – близкими к нулю. Таким образом, если A – матрица значений парных сравнений, то для нахождения вектора приоритетов нужно найти вектор ω [30]. Вектор должен удовлетворять условию (2.40)

$$A\omega = \lambda_{max}\omega, \quad (2.40)$$

где λ_{max} – наибольшее собственное значение матрицы.

Искомый собственный вектор или вектор взвешивания может быть определен по формуле (2.41)

$$(A - \lambda_{max}E) \times \omega = 0, \quad (2.41)$$

где E – единичная матрица размерности $n \times n$.

Точный расчет максимального собственного значения матрицы и вектора взвешивания часто требует объемных вычислений. Поэтому широко используются методы для приближенного расчета значения собственного вектора, или вектора взвешивания [56]. Например, из матрицы попарного сравнения показателей безопасности маршрута A можно вывести из приоритеты (2.42)

$$A^1 \times E; A^2 \times E; A^3 \times E; \dots A^q \times E, \quad (2.42)$$

где $A = n \times n$ – матрица сравнения пар;

$E = n \times 1$ – единичный вектор.

При достаточно большом значении показателя q , вектор $A^q \times E$ представляет собой приемлемое решение для задачи определения собственного вектора. Расчеты могут быть прекращены, если разница значений двух рассчитанных векторов, следующих друг за другом, не превышает заранее заданного значения. Данный численный метод определения весов в модели безопасности маршрута через последовательность, не применялся к показателям оценки безопасности маршрутов, которые позволят ЛПР выбрать более целесообразный с точки зрения комплексной безопасности маршрут, как и во многих других сходных многокритериальных задачах. Вместо этого зачастую использовали приближенное среднее арифметическое или среднее геометрическое для первичных показателей безопасности по строке матрицы A , что может привести к существенным погрешностям при плохой согласованности и нарушении транзитивности суждений ЛПР [41]. Определенные таким образом

приблизительные значения элементов собственного вектора (весов первичных показателей безопасности) необходимо затем нормировать [46].

После алгебраических вычислений матриц необходимо определить непротиворечивость суждений. В обратно-симметричной матрице наибольшее собственное значение больше или равно ее порядку. Обратно-симметричная матрица непротиворечива, если ее наибольшее собственное значение равно порядку матрицы. Индекс согласованности (CI) определяется по формуле (2.43)

$$CI = \frac{\lambda_{max} - n}{n - 1}. \quad (2.43)$$

После чего рассчитывается индекс несогласованности (CR) по формуле (2.44)

$$CR = \frac{CI}{RI}, \quad (2.44)$$

где RI – это случайный индекс со значениями шкалы Т. Саати, рассчитанный для квадратной n -мерной положительной обратно-симметричной матрицы, элементы которой сгенерированы датчиком случайных чисел, распределенных по равномерному закону для интервала значений: $\frac{1}{9}, \frac{1}{8}, \frac{1}{7}, \frac{1}{6}, \frac{1}{5}, \frac{1}{4}, \frac{1}{3}, \frac{1}{2}, 1, 2, 3, 4, 5, 6, 7, 8, 9$.

В таблице 2.2 представлены величины случайного индекса для матриц различного порядка.

Таблица 2.2 – Значения случайного индекса шкалы Т. Саати

Порядок матрицы, n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Случайный индекс, RI	0	0	0,58	0,90	1,12	1,24	1,32	1,41	1,45	1,49	1,51	1,48	1,56	1,57	1,59

Источник: [10].

Если индекс несогласованности $CR < 0,1$, то степень согласованности следует считать хорошей. Считается, что в некоторых задачах приемлемыми значениями можно считать и диапазон $0,1 - 0,3$. Однако при оценке

безопасности маршрута принимаемые по экспертным заключениям решения влекут за собой серьезные последствия, связанные с ДТП и причинением вреда здоровью, поэтому значения $CR \geq 0,1$ не желательны.

Если индекс несогласованности (CR) превышает допустимые значения, то экспертам, которые проводят парные сравнения показателей безопасности, рекомендуется пересмотреть свои суждения. Процедура экспертного опроса проводится перед внедрением данной модели в промышленную эксплуатацию. Согласование парных сравнений в модели безопасности маршрута обеспечит повышение комплексной эффективности предлагаемого маршрута. Для этого следует выполнить следующие шаги:

- 1) выявление проблемных позиций – определить элементы матрицы суждений, которые вносят наибольший вклад в величину отношения согласованности;

- 2) анализ причин несогласованности – понять, почему возникли несогласованности в оценках элементов матрицы парных сравнений показателей оценки безопасности маршрута, что может потребовать дополнительного сбора данных, пересмотра показателей или обсуждения с другими экспертами для получения более точных оценок;

- 3) изменение мер несогласованности – внести коррективы в оценки таким образом, чтобы уменьшить значение индекса несогласованности, рассчитываемого по формуле (2.43).

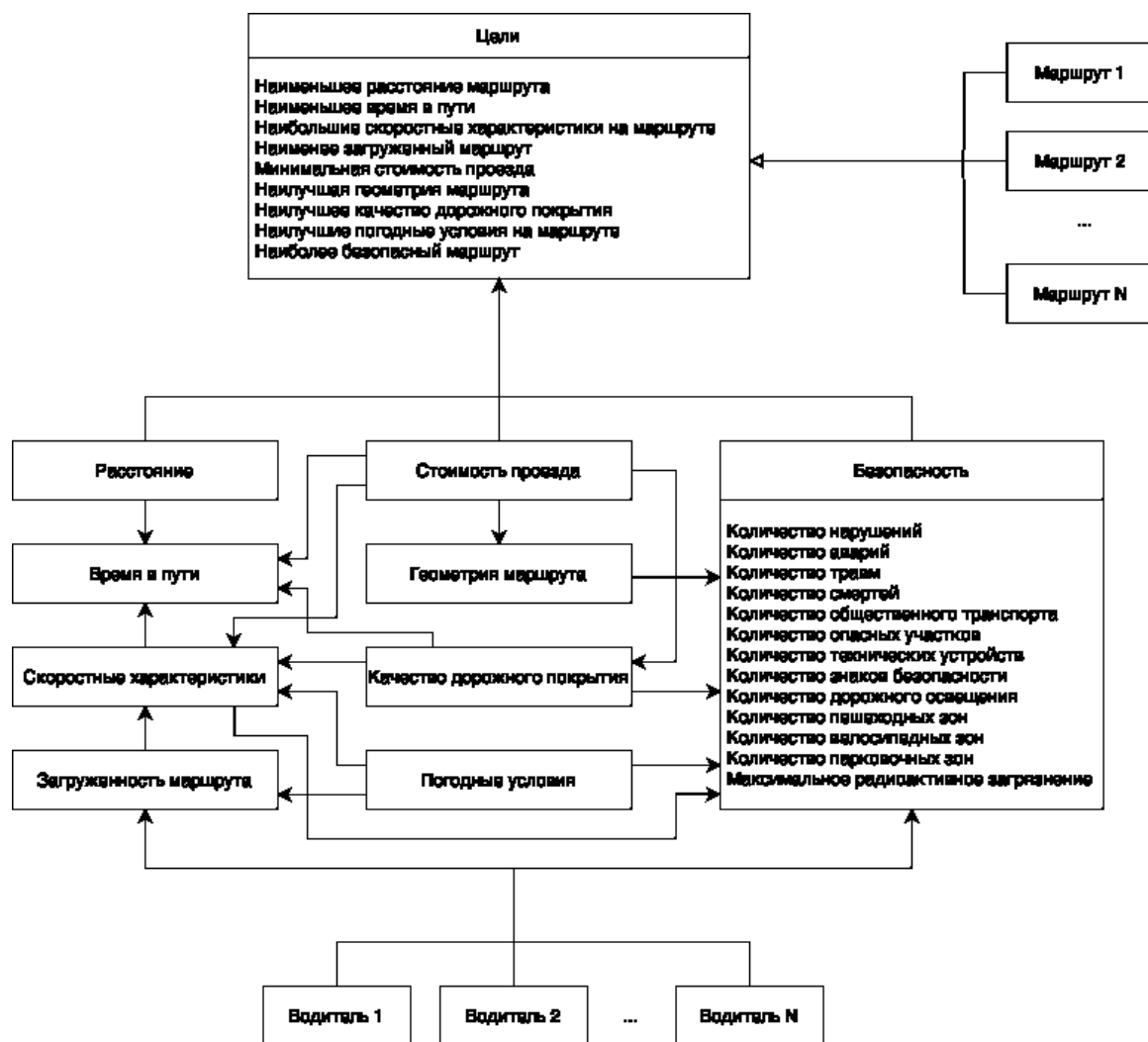
Таким образом модель позволяет проводить комплексную оценку безопасности. Для расчета весов по методу группового анализа иерархий (ГАНР) применяется итеративная процедура, что делает её гибкой и адаптируемой к различным условиям и регионам по сравнению с моделями, которые применяют непосредственное назначение весов показателей.

Модель использует проверку согласованности экспертных оценок с помощью индекса согласованности и случайного индекса Т. Саати, что позволяет выявить ситуации нарушения транзитивности суждений экспертов (например, когда показатель А лучше Б, Б лучше В, а В лучше А). В случае

нарушения транзитивности их необходимо устранить, иначе задача диссертации, по комплексной оценке, безопасности маршрута, не может быть решена. Модель безопасности маршрута позволяет рассчитать приоритеты его частных показателей, а модель из параграфа 2.1 позволяет искать компромисс между загруженностью и безопасностью.

2.4 Когнитивная модель расчета показателей оценки маршрута

Рассмотрим задачу расчета показателей оценки маршрута при недостаточном количестве данных (о загруженности дуг улично-дорожной сети, о погодных условиях и о показателях безопасности). В этом параграфе для расчета показателей средней скорости, загруженности, погодных условий, безопасности разработана когнитивная модель. «Когнитивная» означает основанная на знаниях эксперта в области транспорта. Таким образом когнитивная модель должна обеспечить достижение цели исследования в части оценки маршрута по ряду факторов: нагрузки на сетевую структуру, погодных факторов, факторов безопасности движения в случае отсутствия или недостаточного количества данных для машинного обучения искусственных нейросетей из параграфа 2.2. Её ключевое преимущество — способность учитывать контекстуальные особенности и экспертные эвристики, которые сложно формализовать в данных, например, локальные дорожные правила или сезонные аномалии. Кроме того, модель позволяет оперативно адаптировать критерии оценки при изменении внешних условий, таких как введение новых ограничений скорости или ремонтных работ. При наличии достаточных данных целесообразно использование моделей из параграфов 2.2-2.3, как обладающих большей точностью. И только если данных о каких-либо показателях недостаточно следует применять данную когнитивную модель. На рисунке 2.13 представлена структурно-логическая схема когнитивной модели оценки маршрута транспортной сети.



Источник: составлено автором.

Рисунок 2.13 – Структурно-логическая схема когнитивной модели оценки маршрута транспортной сети

Степень влияния одного фактора когнитивной модели на другой определяется через коэффициенты влияния, который назначается каждой дуге и отражает степень воздействия одного фактора дорожного движения на другой. Значения таких коэффициентов могут быть определены на основании экспертных мнений, накопленного опыта, вероятностных оценок, а в более сложных случаях коэффициенты могут демонстрировать нелинейные зависимости от других факторов дорожного движения. В данной работе используется статический метод оценки коэффициентов влияния, основанный на мнениях экспертов в области дорожного движения [39]. Когнитивная модель формализуется в матричный вид по формуле (2.45)

$$X = AX + F, \quad (2.45)$$

где X – вектор значений факторов дорожного движения размерности n ;
 A – матрица коэффициентов влияния между соответствующими факторами;
 F – вектор значений, характеризующий внешнее влияние на улично-дорожную сеть.

Значения факторов можно вычислить с помощью итерационной процедуры по формуле (2.46)

$$X^{k+1} = AX^k + F, \quad (2.46)$$

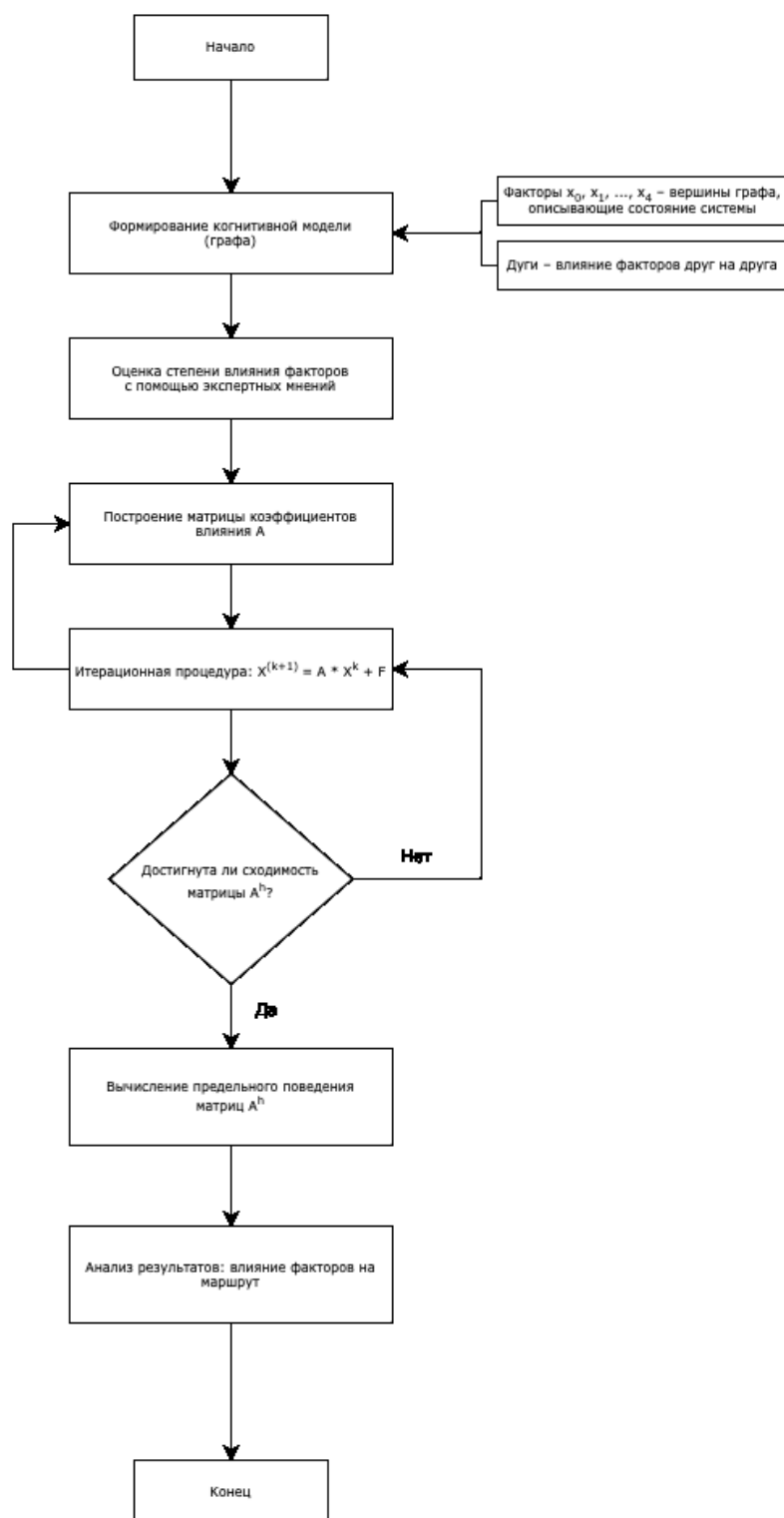
где k – номер итерации.

Скорость сходимости процедуры определяется собственными значениями матрицы A . При увеличении числа факторов дорожного движения возрастает количество вычислений и требования к вычислительным ресурсам, а визуализация графа становится трудной для восприятия. Поэтому далее реализуется моделирование факторов дорожного движения на основе концепции фреймов. Фрейм представляет собой структуру знаний о дорожном движении. Он состоит из слотов, которые содержат структурированные знания. На рисунке 2.13 показаны 2 фрейма с показателями безопасности маршрута и показателями с целями, которые желает достичь ЛПР. Слот – это значение фактора дорожного движения в определенный момент времени. Фреймовая структура помогает уменьшить сложность системы, не углубляясь в детали внутренней структуры фрейма. Это позволяет исследовать влияние слотов одного фрейма на другой. Например, влияние показателей загруженности на безопасность и на цели принятия решений по выбору маршрута.

В данном случае факторами являются показатели оценки маршрутов. Модель позволяет рассчитать распространение влияния одних показателей оценки маршрутов на другие показатели по всем возможным путям в графе влияний, который представлен матрицей A . Элементы матрицы показывают

только прямое влияние каждого показателя оценки маршрута на остальные. Слотами фрейма могут быть другие фреймы, что создает иерархическую структуру взаимосвязанных показателей. Показатели могут оказывать косвенное влияние друг на друга. Важно рассмотреть все возможные пути влияния показателей через эти транзитные показатели для полного понимания взаимодействий в моделируемой сетевой структуре. На данном этапе привлечение экспертов при моделировании уже не требуется. Их непосредственное участие заканчивается после формирования структуры фреймов и факторов и коэффициентов влияния, однако они могут привлекаться в дальнейшем для верификации результатов моделирования. Для этого построим последовательность матриц влияния: A^1, A^2, A^3, A^4 и так далее. Каждая следующая матрица получается путем умножения на предыдущую и нормировки так, чтобы сумма элементов в столбце была равна 1. Как только изменение матрицы на очередном шаге возведения в степень будет незначительным, или изменения начнут повторяться с заданным периодом, то итерационный процесс останавливается. Нормировка матриц на каждом шаге гарантирует сходимость модели и предотвращает неконтролируемый рост коэффициентов влияния, что особенно важно при работе с многоуровневыми зависимостями. Этот подход также позволяет выявлять ключевые показатели-«драйверы», оказывающие наибольшее воздействие на итоговую оценку маршрута, даже если их влияние изначально кажется косвенным. Полученная предельная матрица определит значения всех показателей оценки маршрутов включая показатели их безопасности для расчета комплексной эффективности маршрутов.

Предложенная модель позволяет проводить анализ вида «что-если». Например, если изменится значение показателя погодные условия то, как изменится загруженность и безопасность? Или как загруженность отразится на безопасности? Структурно-логическая схема расчета показателей оценки маршрута в рамках когнитивной модели представлена на рисунке 2.14.



Источник: составлено автором.

Рисунок 2.14 – Структурно-логическая схема расчета показателей оценки маршрута в когнитивной модели

Предложенная когнитивная модель оценки маршрута позволяет без достаточного количества исторических данных о показателях оценки маршрутов получить приближенные оценки этих показателей и таким образом выбрать маршрут. Она отличается тем, что:

1) впервые использует фреймовую структуру, предложенную академиком Б.Н. Четверушкиным и В.А. Судаковым [61] для уменьшения сложности оценки маршрута, что позволяет моделировать сложные взаимосвязи между различными факторами, влияющими на процесс выбора маршрута, в то время как существующие модели ориентируются на прямолинейные или иерархические подходы без учета транзитных факторов [39];

2) позволяет гибко изменять структуру при появлении новых показателей и факторов, что делает её адаптивной по сравнению с традиционными моделями, которые зачастую требуют полной переработки при изменении входных данных или показателей [38].

Модель безопасности маршрута за счет учета экспертных суждений о влиянии одного показателя на другой позволяет получить оценки маршрутов без использования данных о дорожном движении в предыдущие периоды времени. Таким образом она позволяет оценить как управленческие решения по изменению показателей оценки улично-дорожной сети (например, безопасности, загруженности) повлияют на комплексную оценку маршрута.

Таким образом когнитивную модель следует применять в таких ситуациях, когда расчета показателей загруженности дорожного графа нет достаточного количества исходных данных для построения модели машинного обучения. Когнитивная модель менее точная, но при достаточно квалифицированных экспертных оценках взаимного влияния факторов оценки дорожного движения позволяет получить приемлемые результаты.

Выводы по второй главе

Разработан комплекс моделей сетевых структур дорожного движения (улично-дорожной сети с участниками движения), включающий:

1) модель загруженности сетевой структуры с учетом погодных условий повысившую точность прогнозирования на 5-7% за счет сравнительного анализа современных архитектур искусственных нейронных

сетей и последующего выбора архитектуры LSTM + WSA и настройки гиперпараметров;

2) модель безопасности маршрута, использующая открытые данные о состоянии дорожного покрытия, средствах регулирования движения (светофоры, переходные переходы), средствах фиксации нарушений (видеокамеры автоматической фиксации нарушений, радары), дорожно-транспортных происшествиях, радиоактивном загрязнении;

3) когнитивная модель расчета критериев оценки маршрута на основе фреймово-факторного графа экспертных оценок, позволяющая оценить сетевую структуру в случае недостаточности эмпирических данных.

Сформирован набор критериев, влияющих на поиск рационального маршрута, на основе научных исследований и мнения экспертов. Критерии разделены на объективные и экспертные. Первые можно рассчитать на основе статистических данных и моделей машинного обучения, вторые были оценены при помощи экспертов в области транспорта.

Создана методика оценки эффективности маршрута на основе метода анализа иерархий и метода порядка предпочтений по сходству с идеальным решением. Предложенный комплекс моделей:

1) использует ансамбль методов многокритериальной оценки (MCDM), включающий АНР, TOPSIS и когнитивную модель, что позволяет более гибко адаптироваться к различным условиям выбора маршрута в зависимости от приоритетов ЛПП и расширяет возможности по сравнению с существующими моделями, которые обычно ограничиваются одним методом многокритериальной оптимизации;

2) учитывает количественные и качественные критерии, при этом последние оцениваются экспертами на основе опыта и условий маршрута, что позволяет интегрировать субъективные факторы, такие как качество дорожного покрытия и геометрия маршрута. В то время как другие модели часто ограничиваются только количественными показателями;

3) использование когнитивной модели в случаях, когда недостаточно данных для применения машинного обучения или методов MCDM, что дает преимущество при работе с ограниченными или неполными данными, чего не предусматривают существующие модели, полагающиеся только на большие данные;

4) позволяет пользователю через мобильное приложение настраивать вес критериев в реальном времени, обеспечивая персонализированный выбор маршрута в зависимости от целей поездки, что делает модель более гибкой по сравнению с традиционными системами, которые не предлагают возможности для динамической настройки предпочтений;

5) интегрирует прогнозируемые погодные условия и данные о безопасности, такие как аварии и нарушения ПДД, в процесс принятия решений, что позволяет более точно учитывать влияние внешних факторов на качество маршрута, тогда как многие существующие модели ограничиваются только статическими данными.

Глава 3

Численный метод для маршрутизации транспортных средств

В главе представлен численный метод поиска рациональных маршрутов. Суть метода заключается в том, что:

1) процесс поиска рационального маршрута формализован как поиск множества альтернативных вариантов маршрутов путем просмотра вершин графа при движении как от начальной вершины к конечной, так и от конечной к начальной;

2) комплексно решена задача оптимизации времени поиска маршрута за счёт минимизации обращений к элементам улично-дорожной сети.

Данный численный метод отличается от известного и распространённого алгоритма A^* , тем, что с целью повышения оперативности поиска маршрута с наилучшими параметрами, тем что:

1) использована буферизация (кэширование) данных о структуре улично-дорожной сети (далее – УДС) как для двух множеств вершин: первого множества вершин (M_1), просматриваемых на текущей итерации при движении от начальной вершины к конечной, второго множества вершин (M_2), просматриваемых на текущей итерации при движении от конечной вершины к начальной;

2) минимизировано количество обращений к графу на диске;

3) введен усовершенствованный критерий останова алгоритма для данного численного метода: алгоритм продолжает перебирать вершины из множеств M_1 и M_2 после первого их пересечения, что повышает вероятность нахождения оптимального маршрута.

Разработанный численный метод состоит из методики обработки данных о перемещениях транспортных средств, методики оценки загруженности УДС, методики маршрутизации транспортных средств и позволяет повысить оперативность поиска маршрута на 30%.

3.1 Методика обработки данных перемещения транспортных средств

Как было представлено в параграфе 2.2 для оценки дорожного движения используются следующие показатели:

- 1) интенсивность – количество транспортных средств, проходящих через определённую точку дороги за единицу времени;
- 2) плотность – количество транспортных средств на единицу длины дороги;
- 3) скорость – средняя скорость движения транспортных средств на определённом участке дороги.

Представленные показатели являются входными признаками для формирования модели загруженности сетевой структуры, по данным которой осуществляется оперативная оценка дорожного движения они необходимы как. Традиционно [43; 168] актуальная информация в дорожной сети собирается из различных источников данных, среди которых стационарные передатчики, камеры наблюдения и другие устройства. Однако такая методика имеет существенные ограничения, в связи с тем, что оснастить дорожную сеть таким оборудованием крайне затратно. Кроме того, из-за сбоев аппаратного или программного обеспечения, связи и технического обслуживания стационарные источники информации могут содержать существенные пропуски в данных. В то же время существуют достаточные массивы данных с приемников сигналов глобальных навигационных спутниковых систем (далее – ГНСС). Однако они не привязаны к УДС и могут содержать существенные погрешности в определении координат. Эта погрешность приводит к тому, что местоположение транспортного средства не попадает на улицу, а нам необходимо определить улицу или перекресток, на котором находится транспортное средство, по его координатам.

В диссертации сформулирована гипотеза о возможности улучшения комплексной эффективности маршрутизации транспортных средств за счет

снижения погрешностей в определении местоположения транспортного средства с привязкой к УДС. Привязки данных ГНСС к УДС, позволит восстановить маршрут движения на основе наблюдений и минимизировать погрешности определения траектории движения. Такая задача, как показали проведенные исследования, наиболее эффективно может быть решена через оценку правдоподобия маршрута с использованием комбинации расстояния Фреше [95] (которое позволяет найти элемент УДС наиболее близкий траектории по отклонению от координат полученных от приемника ГНСС) и инкрементального сопоставления ГНСС-данных с графом дорог, что повышает точность привязки и учитывает такие факторы, как направление движения.

В качестве приемников сигналов ГНСС чаще всего используются мобильные телефоны. Данные, полученные с мобильных телефонов, позволяют оценивать движение транспортных средств. Мобильные телефоны генерируют данные о местоположении, которые можно агрегировать для анализа дорожного движения. Чем чаще приемник сигналов ГНСС передает свои координаты в единую базу данных, тем точнее будут треки. Треки – это траектории движения, построенные на основе последовательности координат (широта, долгота, время) с устройств. Они используются для расчёта параметров дорожного движения, таких как: интенсивность, плотность, скорость движения. Важно отметить, что все данные обезличены и не содержат информации о конкретном пользователе или его автомобиле. Треки поступают в единую базу данных не только от частных водителей, но и от компаний-партнеров – организации с большим парком автомобилей, курсирующих по городу.

Таким образом задача определения местоположения транспортного средства на УДС фактически позволяет осуществлять косвенный мониторинг дорожного движения. Закрывающийся в том, что данные изначально собираются для предоставления услуг навигации пользователям мобильных устройств, а затем используются для анализа транспортных потоков в целом.

Мобильные устройства используют доступ к сети Интернет и генерируют данные о местоположении при передаче информации на ближайшую станцию сотового оператора. Характеристики пространственно-временного покрытия и низкая стоимость получения таких данных позволяют осуществить анализ дорожного движения. Таким образом предложенный подход позволяет решить задачу по повышению комплексной эффективности маршрутизации транспортных средств.

Для того чтобы найти маршрут близкий к оптимальному нам необходимо оценить загруженность УДС, а для оценки загруженности УДС необходимо понять какие транспортные средства и с какой скоростью по ним двигаются.

С этой целью необходимо по координатам полученным с приемников ГНСС определить улицу или перекресток УДС и соответственно представить положение транспортного средства на них. Дорожный граф, под которым в диссертации понимается модель УДС, включая вершины – перекрёстки, дуги – участки дороги между перекрёстками, геометрия дуг – координаты дорог, он является основой для обработки ГНСС-данных, так как точки трека должны быть привязаны к дугам графа для корректного построения маршрута.

Для решения задачи использования данных ГНСС для определения загруженности УДС, что позволяет улучшить оперативность маршрутизации, обосновано применение следующих алгоритмов обработки данных:

- 1) восстановление наиболее вероятного маршрута на основе зашумленных ГНСС-данных алгоритмом Витерби (суть которого заключается в определении условных вероятностей для последовательности ребер дорожного графа);

- 2) определение расстояния Фреше: для сопоставления траекторий с реальным дорожным графом, учитывая направление движения (расстояние Фреше позволяет найти элемент УДС наиболее близкий траектории по отклонению от координат полученных от приемника ГНСС);

3) фильтрация аномальных точек, вызванных погрешностями ГНСС. Это достигается путем сравнения скорости необходимой для перехода из предыдущей точки маршрута в новую точку маршрута и порога скорости, которую выше которого не может развивать никакое транспортное средство (500 км/ч).

Таким образом предлагаемая методика обработки данных ГНСС о перемещениях транспортных средств включает в себя вышеуказанные три алгоритма. На рисунке 3.1 представлена схема процесса получения и обработки данных. На нем под необработанными данными понимается информация, поступающая с ГНСС приемника сигнала мобильного устройства – это широта, долгота, высота, курс, ускорение по трем осям координат.



Источник: составлено автором.

Рисунок 3.1 – Схема процесса получения и обработки данных ГНСС методикой позволяющих восстановить траекторию перемещения транспортных средств на элементах УДС с целью определения загруженности элементов

Рассмотрим факторы, которые необходимо учитывать при привязке транспортного средства к дорожному графу (а привязка позволит определить загруженность элемента УДС):

1) расстояние от точки до геометрии дуги графа – оценивает как кратчайшее расстояние, так и вероятность того, что приёмник мог допустить ошибку;

2) совпадение направлений движения – оценивает угол между вектором движения транспортного средства и направлением участка геометрии ребра, к которому привязывается точка. Данная мера устойчива к систематической погрешности ГНСС-приёмника, но подвержена случайной;

3) изменение направления движения транспортного средства – вероятность того, что автомобиль свернет с главной дороги, в общем случае меньше, чем вероятность того, что он продолжит движение по ней. Так минимизируется количество маневров;

4) физическая возможность перехода с одной дуги на другую – адекватность скорости, с которой должен был двигаться автомобиль, чтобы совершить переход.

Для обработки ГНСС-данных (в том числе `id` – уникальный идентификатор устройства (`uuid`), `lat` – широта, `lon` – долгота, `course` – курс/азимут $0-359^\circ$, `hdop` – точность позиции по высоте, `height` – высота над уровнем моря, `mnc` – данные сотовой вышки (код оператора), `mcc` – данные сотовой вышки (код страны), `cellid` – данные сотовой вышки (идентификатор передатчика), `lac` – данные сотовой вышки (код зоны), `axel_x` – данные акселерометра (ось X, поперечная), `axel_y` – данные акселерометра (ось Y, продольная), `axel_z` – данные акселерометра (ось Z, вертикальная), `date` – время создания сообщения) и используется алгоритм Витерби, позволяющий определить наиболее вероятное состояние скрытой Марковской модели на основе последовательности наблюдений за вышеуказанными данными полученными с ГНСС приемника [201].

Последовательность наблюдений используется для восстановления скрытых состояний Марковской модели. Скрытое состояние Марковской модели – это данные о том на каком элементе УДС находится транспортное средство. Алгоритм Витерби обрабатывает последовательность наблюдаемых данных y_t , чтобы восстановить наиболее вероятную последовательность скрытых состояний x_t . y_t – это вышеуказанные данные полученные с приемника ГНСС (широта, долгота и прочее), x_t – это признак наличия соответствующего транспортного средства. При этом под скрытым состоянием понимается такое состояние, которое не наблюдаемо нами непосредственно [37].

При помощи алгоритма Витерби определяется, наиболее вероятная траектория конкретного участника дорожного движения, использование альтернативных подходов определения траектории не позволяет осуществить ее привязку к УДС, необходимой для поиска рационального маршрута. Таким образом снижается погрешность определения загруженности элемента УДС. Рассмотрим на суть методики обработки данных ГНСС для снижения погрешностей при соотнесении транспортного средства элементу УДС. Пусть существует скрытая марковская модель с пространством состояний $S = \{s_1, s_2, \dots, s_K\}$, где K – количество возможных различных состояний сети, характеризующих состояние участка (загруженность и скорость потока на дуге графа). При этом состояния, которые принимает сеть, невидимы для наблюдения (видимые данные с ГНСС приемников не содержат привязки у УДС, а невидимые – это как раз привязка к УДС). Обозначим через x_t состояние сети в момент t . На выходе сети в момент t появляется видимое для наблюдения значение $y_t \in O = \{o_1, o_2, \dots, o_N\}$, где N – число возможных различных наблюдаемых значений на выходе. Пусть π_i – начальная вероятность нахождения сети в состоянии i , а $a_{i,j}$ – вероятности перехода сети из состояния i в состояние j . Пусть на выходе сети наблюдается последовательность $y_1, \dots, y_T$. Тогда наиболее вероятная последовательность состояний сети $x_1, \dots, x_T$ для наблюдаемой последовательности может быть

определена с помощью следующих рекуррентных соотношений по формуле (3.1)

$$\begin{aligned} V_{1,k} &= P(y_1|k) \times \pi_k, \\ V_{t,k} &= \max_{x \in S} (P(y_t|k) \times a_{x,k} \times V_{t-1,x}), \end{aligned} \quad (3.1)$$

где $V_{t,k}$ – это вероятность наиболее правдоподобной последовательности состояний, соответствующей первым t наблюдаемым значениям, завершающейся в состоянии k .

Путь Витерби находится при помощи указателей, под которыми в диссертации понимается вспомогательная структура данных, которая запоминает предыдущие состояния при расчёте вероятностей $Ptr(k, t)$. Она используется для восстановления пути: для каждого состояния k в момент t указывается состояние x из момента $t-1$, которое привело к наблюдаемому результату. Пусть $Ptr(k, t)$ – функция, которая возвращает значение x , использованное для подсчета $V_{t,k}$, если $t > 1$, по формуле (3.2)

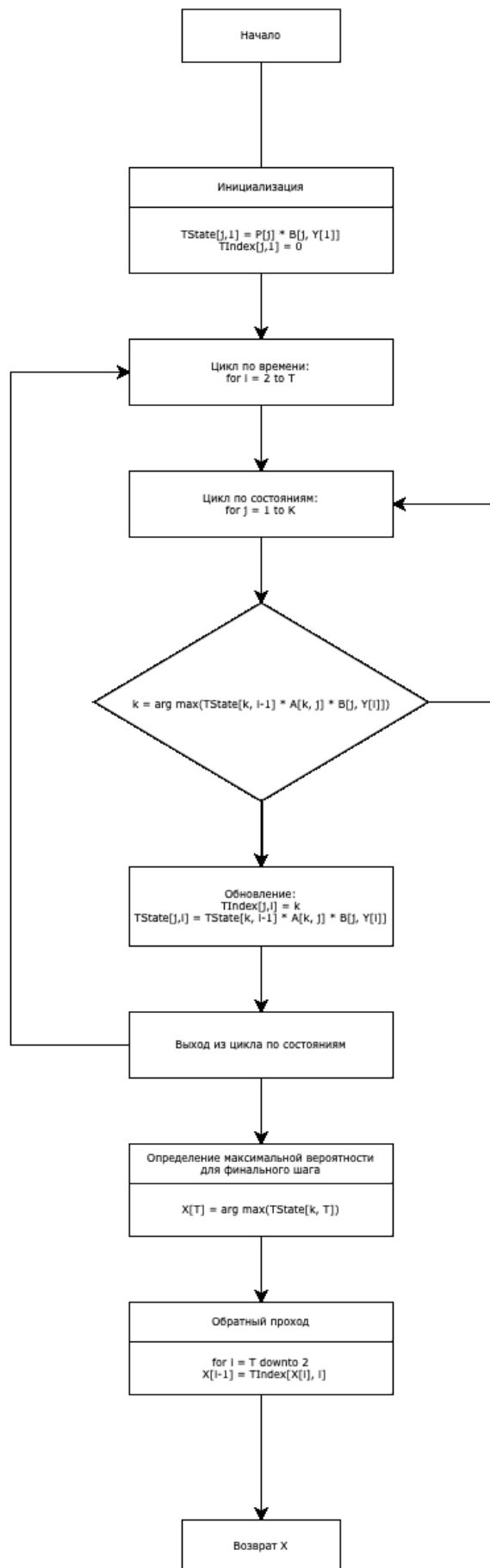
$$\begin{aligned} x_T &= \arg \max_{x \in S} (V_{T,x}), \\ x_{t-1} &= Ptr(x_t, t), \end{aligned} \quad (3.2)$$

где T – длина последовательности наблюдений.

Определенные выше ГНСС-данные, обработанные алгоритмом, используются для:

- 1) восстановления маршрутов движения;
- 2) оценки загруженности дорожной сети.

В данном параграфе речь о поиске рационального маршрута пока не идет. Прежде чем найти маршрут нужно оценить загруженность, а чтобы оценить загруженность нам нужно знать, как двигались другие. Как двигались другие – тут используется методика данного параграфа. Схема алгоритма привязки данных ГНСС к УДС представлена на рисунке 3.2.



Источник: составлено автором.

Рисунок 3.2 – Схема алгоритма привязки данных ГНСС к УДС

Данный алгоритм привязки данных ГНСС к УДС позволяет решить задачу привязки ГНСС-данных к дорожному графу. Оригинальный численный метод Витерби применяется для определения наиболее вероятной последовательности скрытых состояний, но в данном случае он используется с учётом специфики дорожной сети (направления движения, длины участков и другие). Таким образом данный алгоритм является составной частью методики обработки данных о перемещениях транспортных средств.

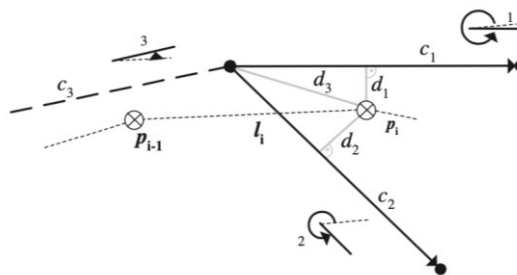
Алгоритм обрабатывает входящий трек, получая на выходе последовательность ребер УДС, которые близки к входному треку, что позволяет связать разрозненные ГНСС координаты в единый маршрут. ГНСС координаты имеют погрешности и могут не совпадать с геометрией дуг УДС, то есть находится не на элементе УДС, на котором находилось транспортное средство, а в произвольном месте. Поэтому данный алгоритм использует такие показатели как расстояние, направление, вероятность манёвра для выбора наиболее подходящей дуги. Таким образом строится связанный маршрут, минимизирующий ошибки.

Таким образом последовательность действий (в рамках представленной в настоящем разделе методики обработки данных о перемещениях транспортных средств) по обработке ГНСС данных для оценки загруженности в общем виде следующая:

- 1) получить данные с ГНСС приемников;
- 2) по полученными данным используя алгоритм (на основе метода Витерби) на рисунке 3.2 рассчитать вероятности нахождения на различных элементах УДС для каждого транспортного средства (это вероятности скрытых состояний);
- 3) выбрать для каждого транспортного средства элемент УДС с максимальной вероятностью нахождения на нем;
- 4) применить алгоритм (на основе расстояния Фреше) привязки трека к дорожному графу, описанный ниже и показанный на рисунке 3.3;

5) посчитать загруженность элемента УДС по всем транспортным средствам, которые находятся на нем, формулы расчета приведены в параграфе 3.2.

На основе данных, полученных с ГНСС приемника для пункта 4 вышеуказанной последовательности действий методики обработки данных о перемещениях транспортных средств, необходимо рассчитать оценку правдоподобия полученного маршрута транспортного средства. Для этого используется метрика – расстояние Фреше. В данной работе расстояние Фреше применяется к положению транспортного средства полученному после обработки ГНСС данных алгоритмом Витерби. Оценка ориентируется только на географическую удаленность прокладываемого трека (маршрута движения транспортного средства) [13]. Для привязки треков используется оценочная формула для инкрементального алгоритма привязки данных, согласно работе [49]. Учитывая серию данных о положении транспортного средства, алгоритм сопоставления реализует выборку по позиции и стратегию по краю. Чтобы сопоставить позицию p_i с УДС, учитывая, что предыдущая позиция p_{i-1} уже сопоставлена, алгоритм обработки данных о перемещениях транспортных средств действует следующим образом. Во-первых, ребра-кандидаты на включение в маршрут транспортного средства, которые должны быть сопоставлены с текущей позицией, идентифицируются как набор инцидентных данной позиции транспортного средства ребер, выходящих из последнего совпавшего ребра, включая также само совпадающее ребро [49]. На рисунке 3.3 представлены ребра c_1 , c_2 и c_3 , где c_3 – это ребро, соответствующее p_{i-1} .



Источник: составлено автором.

Рисунок 3.3 – Пример сопоставления ГНСС-данных

На рисунке 3.4 видно, что позиции p_{i-1} и p_i отличаются от двух возможных дорог c_1 и c_2 и нам нужно выбрать положение на одной из них исходя из метрики близости позиции p_i к ним. При оценке ребер необходимо использовать два показателя сходства: S_d и S_α . Показатель S_d описывает расстояние от позиции до различных ребер и рассчитывается на основе взвешенного расстояния d между точкой p_i и ребром-кандидатом c_j , используя коэффициенты масштабирования μ_d, n_d по формуле (3.3)

$$S_d(p_i, c_j) = \mu_d - \alpha \times d(p_i, c_j)^{n_d}, \quad (3.3)$$

где μ_d, n_d, α – это коэффициенты масштабирования;

$d(p_i, c_j)$ – это расстояние от точки p_i до геометрии дуги c_j .

Показатель S_α оценивает направление траектории по отношению к ребру-кандидату и рассчитывается на основе углового различия $\alpha_{i,j}$ между направлением ребра-кандидата c_j и направлением отрезка $l_i = (p_{i-1}, p_i)$ с использованием коэффициентов масштабирования μ_α, n_α по следующей формуле (3.4)

$$S_\alpha(p_i, c_j) = \mu_\alpha \times \cos(\alpha_{i,j})^{n_\alpha}, \quad (3.4)$$

где μ_α и n_α – это коэффициенты масштабирования;

$\cos(\alpha_{i,j})$ – косинус угла i -й дуги графа и направлением движения по дуге трека.

Для алгоритма необходимо корректное соотношение параметров μ_α и μ_d , так как оно определяет, какой из факторов будет иметь больший вес при сравнении. Выбор большего значения μ_d по сравнению с μ_α означает, что расстояние имеет большее значение, чем направление. Параметр n определяет скорость уменьшения соответствующего веса при увеличении расстояния отрезка линии или разницы углов. Использование $\cos$ подразумевает, что с увеличением угловой разности показатель S_α уменьшается, а при угловых

разностях $90 < \alpha < 270$ и выборе нечетной степени n и положительном значении коэффициента μ_α , S_α становится отрицательным.

Для конкретного набора данных, эмпирическим путем были установлены следующие значения параметров: $\mu_d = 10$, $\alpha = 0,17$, $n_d = 1,4$, $\mu_\alpha = 10$, $n_\alpha = 4$. Комбинированная мера сходства вычисляется как сумма отдельных оценок по формуле (3.5)

$$S(p_i, c_j) = S_d(p_i, c_j) + S_\alpha(p_i, c_j). \quad (3.5)$$

Чем выше оценка показателя сходства (3.5), тем лучше совпадение. То есть алгоритм обработки заключается в поиске наилучшего сходства траектории по данным ГНСС с траекторией по элементам УДС. В оценке сходства (3.5) сумма используется по нескольким причинам:

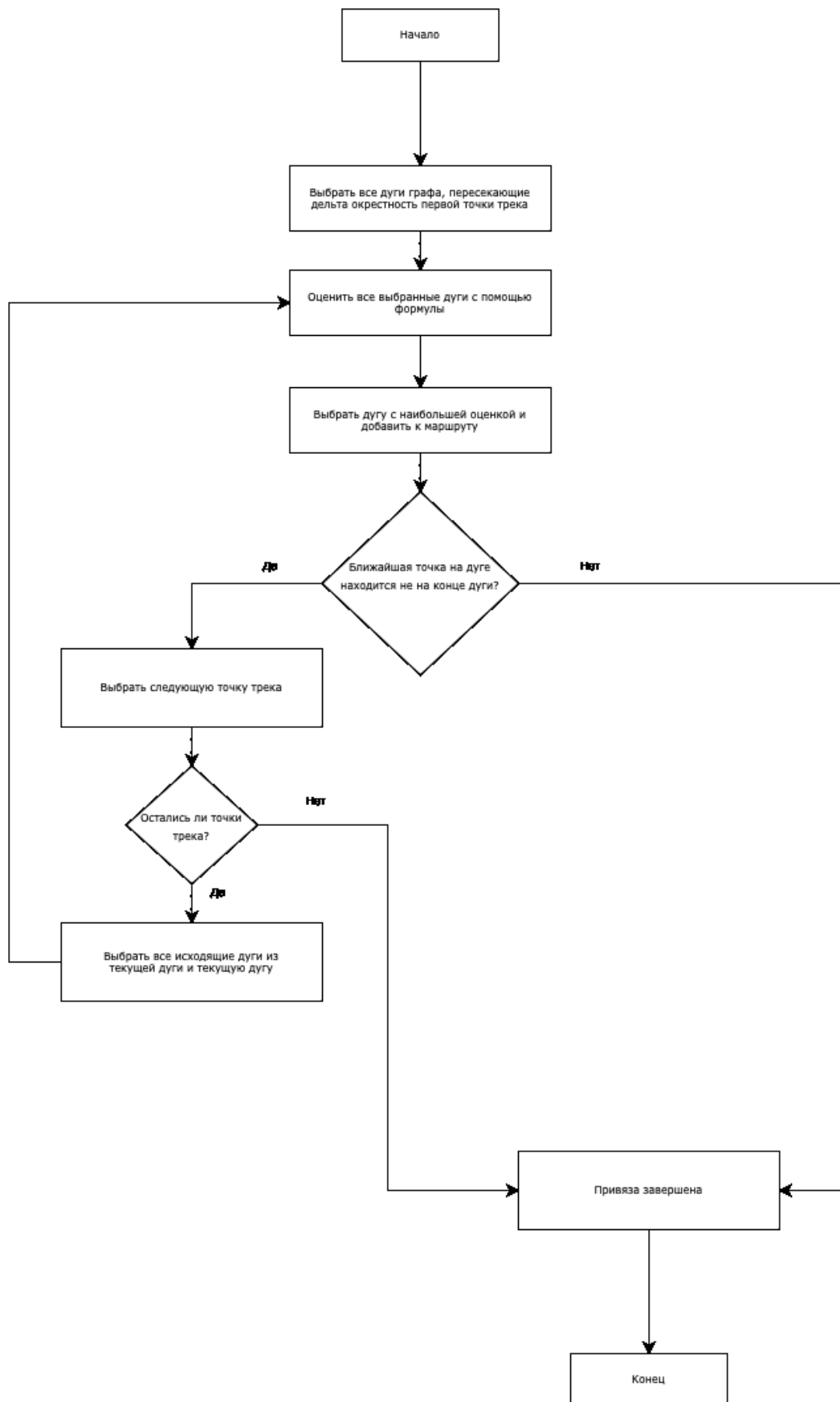
- 1) необходимо, чтобы увеличение S_d и S_α приводило к росту S ;
- 2) использование произведения не подходит, т.к. сделает формулу нечувствительной к изменению одного из аргументов, если второй аргумент равен нулю;
- 3) возможно использование других нелинейных функций, но анализировать придется большие данные – тысячи треков с десятками точек в секунду, поэтому необходимо высокой быстродействие, которое обеспечивает простое суммирование;
- 4) в работе [48] используется аналогичный подход, который хорошо себя зарекомендовал, как в указанной работе, так и в данном диссертационном исследовании при апробации работы.

Алгоритм привязки данных ГНСС к УДС, показанный на рисунке 3.2 позволяет только определить последовательность дуг дорожного графа в маршруте транспортного средства, но он не определяет положение транспортного средства на конкретном участке УДС. Для определения загруженности по методике, разработанной в параграфе 3.2, части элемента УДС, необходим следующий алгоритм привязки трека к дорожному графу:

- 1) выбрать все дуги графа с геометрией, пересекающей дельта окрестность первой точки трека;

- 2) оценить все выбранные дуги с помощью формулы (3.5);
- 3) выбрать дуги с наибольшей оценкой, сделать его текущим и добавить к готовому маршруту;
- 4) если ближайшая к точке трека точка на геометрии дуги находится не на конце дуги, то выбрать следующую точку трека. Если точек больше нет, то привязка закончена;
- 5) выбрать все исходящие из текущей дуги графа и текущую дугу;
- 6) перейти к пункту 2.

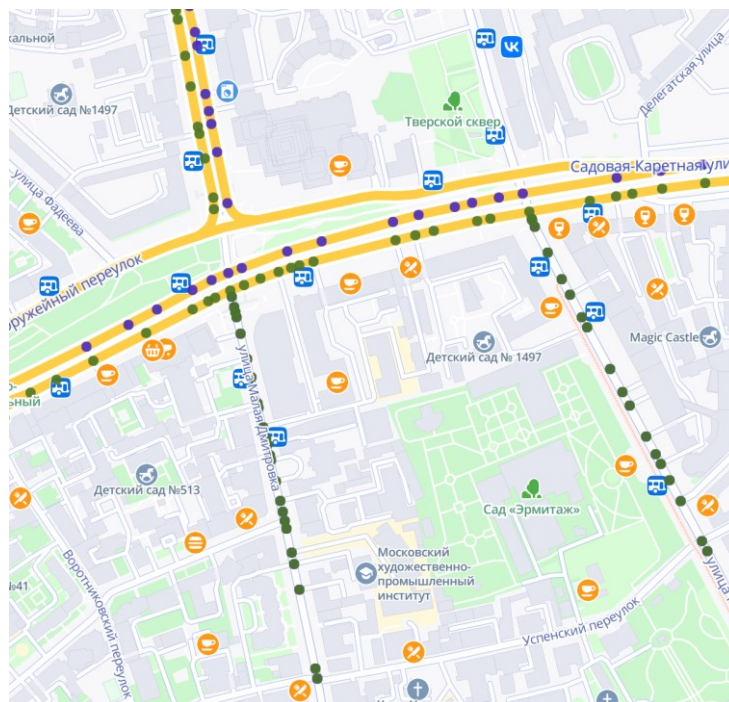
Для повышения устойчивости алгоритма к шумам в данных ГНСС рекомендуется использовать скользящее окно из нескольких точек трека, что снижает влияние случайных выбросов координат. Кроме того, размер дельта-окрестности может динамически адаптироваться в зависимости от плотности дорожной сети: уменьшаться в городских условиях и увеличиваться на магистралях с редкими развязками. В случае неоднозначности выбора дуг (например, на пересечениях) применяется дополнительная проверка по историческим данным о типичных маршрутах транспортных средств. Для ускорения обработки больших массивов данных алгоритм может быть оптимизирован через пространственное индексирование дуг графа, что сокращает время поиска кандидатов на каждом шаге. Схема алгоритма привязки координат к графу дорог представлена на рисунке 3.4. Таким образом параграф определяет методику привязки транспортного средства к УДС, то есть он про то на каком элементе УДС находится транспортное средство. Пока про оперативность речи не идет. Таки образом разработана единая методика обработки данных с ГНСС. То есть координаты с приемника ГНСС заменяются на координаты, которые попадают на элемент УДС. Именно привязка у УДС и позволяет в следующем параграфе посчитать загруженность.



Источник: составлено автором.

Рисунок 3.4 – Схема алгоритма привязки координат к УДС позволяющего определить на каком элементе УДС находится транспортное средство и по этим данным оценить загруженность элементов УДС

В результате алгоритм привязки данных ГНСС к УДС на основе метода Витерби позволил восстановить последовательность дуг графа, максимально соответствующую ГНСС-трекам, а алгоритм привязки координат к УДС на основе расстояния Фреше: использовался для оценки близости трека и дуг графа. Каждый частный алгоритм, рисунки 3.2 и 3.4, решал только часть задачи обработки данных ГНСС для последующего в параграфе 3.2 расчета загруженности, их комбинированное использование составляет единую методику обработки данных для оценки интенсивности, плотности и скорости дорожного движения, что необходимо для оценки загруженности маршрута. В результате полученные координаты с устройств накладываются на УДС, по которой строится единый маршрут движения с информацией о скорости его прохождения. Пример работы методики показан на рисунке 3.5.



Источник: составлено автором.

Рисунок 3.5 – Движение объектов после привязки к графу дорог

Если не использовать предложенную методику, то точки с транспортными средствами могут оказаться не на дороге, а скажем на крыше здания (за счет погрешностей в ГНСС), что маловероятно, после применения указанного алгоритма все транспортные средства будут находиться на элементах УДС, что и показывает рисунок 3.5. В результате применения

данной методики обработки данных о перемещениях транспортных средств получены координаты транспортных средств и их скорости на конкретных элементах УДС. На качественном уровне анализа методики можно говорить о том, что погрешность ГНСС данных должна уменьшиться. В результате данной методики получены точки на элементах УДС. Нельзя говорить, что совсем нет погрешности. Так как у нас нет истинных координат, посчитать уменьшение погрешности здесь не получается, основная цель в привязке к УДС, а не в уменьшении погрешности, и эта цель достигнута.

Таким образом, создана единая методика обработки данных о перемещениях транспортных средств. В следующем параграфе она будет применена для расчета загруженности.

3.2 Методика оценки загруженности УДС

Привязка треков и оценка скорости на дугах графа в параграфе 3.1 позволяет осуществить обработку для расчёта загруженности элементов УДС, включаемых в маршрут транспортного средства. Методика оценки загруженности (измеряемой в баллах) УДС на основе сравнения расчетной и фактической средней скорости для каждого сегмента дороги, позволит не только отслеживать текущие пробки, но и оценивать степень загруженности в разных условиях, включая время суток и климатические факторы. Для этого необходимо разработать шкалу оценки загруженности, которая позволила бы интерпретировать степень загруженности дороги. Перекресток является вершиной графа, а участки дороги между двумя перекрестками – дугами, которые имеют атрибуты длины и скорости. Длина известна заранее на основе открытых данных OpenStreetMap (далее – OSM). OSM – некоммерческий картографический проект по созданию географической карты мира. Среднюю скорость движения на дороге можно рассчитать по формуле (3.6)

$$x_{ab} = \frac{1}{M_r} \sum_{i=1}^{M_r} \left(\frac{1}{Tra_i - 1} \sum_{j=1}^{Tra_i - 1} \frac{distance(o_{i,j}, o_{i,j+1})}{interval(o_{i,j}, o_{i,j+1})} \right), \quad (3.6)$$

где x_{ab} – средняя скорость движения по дороге a за время b ;
 M_r – количество траекторий дороги a ;
 Tra_i – количество точек ГНСС на части траектории i , проходящей через дорогу a ;
 $o_{i,j}$ – ГНСС точка траектории i , проходящая через дорогу a ;
 $distance(o_{i,j}, o_{i,j+1})$ – большое дуговое расстояние совпадающих точек $o_{i,j}, o_{i,j+1}$;
 $interval(o_{i,j}, o_{i,j+1})$ – интервал времени выборки между $o_{i,j}$ и $o_{i,j+1}$.

После обработки данных значения средних скоростей для дорог сохраняется в БД каждые 5 мин. Если данные о скорости отсутствуют, то значение принимается максимально разрешенной скорости без нарушения ограничений ПДД. После расчета скорости и времени прохождения трека модель вычисляет насколько реальное время отличается от эталонного. Загруженность сети представляет шкалу от 0 до 10, где 0 баллов означает свободное движение, а 10 баллов – движение остановилось. Шкала настраивается по-разному для каждого города и имеет коэффициенты, учитывающие сезонность, климатические параметры и другие факторы. С помощью такой оценки ЛПР сможет понять, сколько времени необходимо на прохождение маршрута. Например, если оценка загруженности составляет 7 из 10, то поездка займет примерно в два раза больше времени, чем по свободным дорогам. Для расчета загруженности дорожного сегмента в модель загружаются следующие данные:

- 1) расчетная средняя скорость – среднестатистическая скорость для отрезка дороги (км/ч), источник данных – OSM [32];
- 2) фактическая средняя скорость – медианная скорость для отрезка дороги в конкретный момент времени (км/ч), источник данных – это методика обработки данных (которые включают треки движения транспортных средств

на элементах УДС, положение транспортных средств и их скорости) о перемещениях транспортных средств из параграфа 3.1.

Вышеприведенные данные позволяют оценить текущую загруженность дорожного сегмента, сравнивая фактическую скорость движения с расчетной. Различие между значениями может указывать на наличие пробок или других факторов, влияющих на поток. Использование данных из OSM для расчетной средней скорости и модели обработки данных для фактической скорости обеспечивает актуальность и точность анализа загруженности дорожного сегмента, что позволяет принимать обоснованные решения для оптимизации транспортных маршрутов. Программная реализация модели оценки треков представлена в приложении В. Зная расчетную и фактическую скорости, между ними можно рассчитать разницу в процентах от плановой по формуле (3.7) и сопоставить с данными таблицы 3.1.

$$\Delta = \frac{speed_p - speed_f}{speed_p} \times 100\%. \quad (3.7)$$

Таблица 3.1 – Оценка загруженности дорожного графа

Δ, в процентах	Балл	Описание	Степень загруженности
≥ 90	10	Быстрее пешком	Быстрее дойти пешком
< 90	9	Серьезные пробки	Пробки, быстрее на метро
< 80	8	Серьезные пробки	Пробки, возможно, быстрее на метро
< 60	7	Движение затруднено	Большая загруженность
< 50	6	Движение затруднено	Типичная загруженность в рабочий день
< 40	5	Местами затруднения	Движение с замедлениями
< 30	4	Местами затруднения	Движение с незначительными замедлениями
< 20	3	Дороги свободны	Типично для дорог ночью и в выходные
< 10	2	Дороги свободны	Типично для дорог ночью и в выходные
< 5	1	Пустые дороги	Свободные дороги

Источник: составлено автором.

Таким образом в данном параграфе разработана методика оценки загруженности УДС на основе сравнения расчетной и фактической средней скорости, для каждого сегмента дороги, отличием от существующих методик является использование для расчета загруженности алгоритм обработки данных о перемещениях транспортных средств из параграфа 3.1. Данная методика позволяет не только отслеживать текущие пробки, но и оценивать степень загруженности в разных условиях, включая время суток и климатические факторы. Методика оценки загруженности УДС, следующая:

- 1) зная координаты, обработанные в параграфе 3.1, посчитать путь;
- 2) посчитать скорость, разделив путь на время;
- 3) усреднить скорость по всем транспортным средствам на дуге;
- 4) посчитать отклонение средней скорости от плановой;
- 5) поделить отклонение на плановую скорость – получить долю, то есть отклонение в неких нормированных единицах измерениях – долях плановой скорости;
- б) переводим эти доли в качественную шкалу.

Создана шкала оценки загруженности, которая позволяет наглядно и точно интерпретировать степень загруженности дороги, предоставляя ЛПР возможность быстро оценить ситуацию и принять решение по изменению маршрута.

3.3 Методика маршрутизации транспортных средств

Таким образом в параграфе 3.1 обработаны данные с ГНСС с целью соотнести координаты транспортных средств элементам УДС, далее для этих элементов УДС рассчитана загруженность в параграфе 3.2, которая используется в данном разделе чтобы получить рациональный маршрут транспортных средств.

В контексте разработки моделей оценки маршрута, численный метод поиска A^* [78; 141; 198] представляет собой вычислительный компонент, необходимый для практического применения комплекса моделей сетевых

структур дорожного движения. Двухсторонний алгоритм A^* позволяют решать задачи поиска оптимального маршрута. A^* является полным в том смысле, что он всегда находит решение, если таковое существует. По результатам обзора исследований [12; 13; 70] и вычислительных экспериментов установлено, что A^* продемонстрировал хорошую производительность. Для уменьшения времени работы классического алгоритма A^* , с учетом использования на современных многоядерных мобильных процессорах, предложен двухсторонний A^* . В нем одновременно рассматриваются два множества вершин: M_1 – множество вершин, просматриваемых на текущей итерации при движении от начальной вершины к конечной, M_2 – множество вершин, просматриваемых на текущей итерации при движении от конечной вершины к начальной. При использовании необходимо учитывать следующие факторы:

- 1) для M_1 , поиск новых вершин для добавления в маршрут происходит по исходящим дугам. M_2 необходимо искать входящие в вершины дуги для добавления вершин;

- 2) двухсторонний A^* , запущенный в один поток поочередно добавляет вершины в M_1 и M_2 . Имеет смысл добавлять не по одной вершине в каждое из этих множеств, а по несколько десятков или сотен, а затем переключаться на вершины другого множества. Переключение между M_1 и M_2 требует времени и вычислительных ресурсов в основном при обращении к графу дорог, т.к. граф представляет собой большой массив данных и хранится на жестком диске устройства, к которому необходимо обращаться каждый раз за вершинами. Кроме графа в ряде случаев нужно обращаться к другим структурам данных, например, к информации о высоте дуги над уровнем моря. Существует ненулевая вероятность, что к данным каждой вершины будет несколько обращений что ухудшит скорость работы алгоритма, тогда необходимо применять технологии буферизации (кэширования);

- 3) критерий остановки алгоритма несколько сложнее, чем у одностороннего A^* . Первый общий элемент множеств M_1 и M_2 может

не являться элементом УДС, соединяющей два участка рационального маршрута и для подбора наилучшего маршрута требуется продолжить вычисления. Но в среднем, количество раскрытых вершин у двухстороннего A^* существенно меньше, чем у одностороннего A^* .

Наилучший маршрут должен соответствовать наилучшему значению комплексной эффективности маршрута, которая рассчитывается по формулам из параграфа 2.1. Один из ключевых показателей эффективности – это загруженность маршрута. Суть модификации заключается в параллельной обработке множеств M_1 и M_2 и переходе от целевой функции суммарной длины пути для исходного алгоритма A^* к целевой функции, по комплексной оценке, эффективности маршрута из параграфа 2.1.

Двухсторонний алгоритм A^* использует два параллельных процесса, которые запускаются навстречу друг другу, сокращая количество раскрытых вершин по сравнению с классическими моделями, такими как односторонний A^* и алгоритм Дейкстры, что особенно заметно на примерах, где двухсторонний A^* раскрывает в 2-3 раза меньше вершин, что значительно увеличивает скорость поиска. Двухсторонний алгоритм A^* позволяет гибко настраивать алгоритм в зависимости от требований к производительности и времени ответа. В сценариях, где важно максимизировать количество ответов в секунду, предпочтение может быть отдано однопоточному выполнению алгоритма, а при необходимости максимально быстрой прокладки маршрута – двухпоточному, чего не предусматривают многие существующие подходы.

Таким образом достигается цель по повышению оперативности поиска маршрута. Число потоков – это число процессов, одновременно существующих в вычислительной системе и использующих общие данные, в том числе множества M_1 и M_2 . Процесс – это программа для ЭВМ во время выполнения.

В общем случае, поиск наилучшего (по комплексному критерию эффективности) маршрута можно описать следующими шагами:

1) на каждой итерации алгоритма множества M_1 и M_2 дополняются новыми вершинами, пока у них не появится хотя бы один общий элемент (общая раскрытая вершина);

2) осуществляется поиск наилучшего маршрута, который начинается в одной из раскрытых вершин M_1 , после чего необходимо соединить его одной дугой с раскрытыми от финиша вершинами из M_2 .

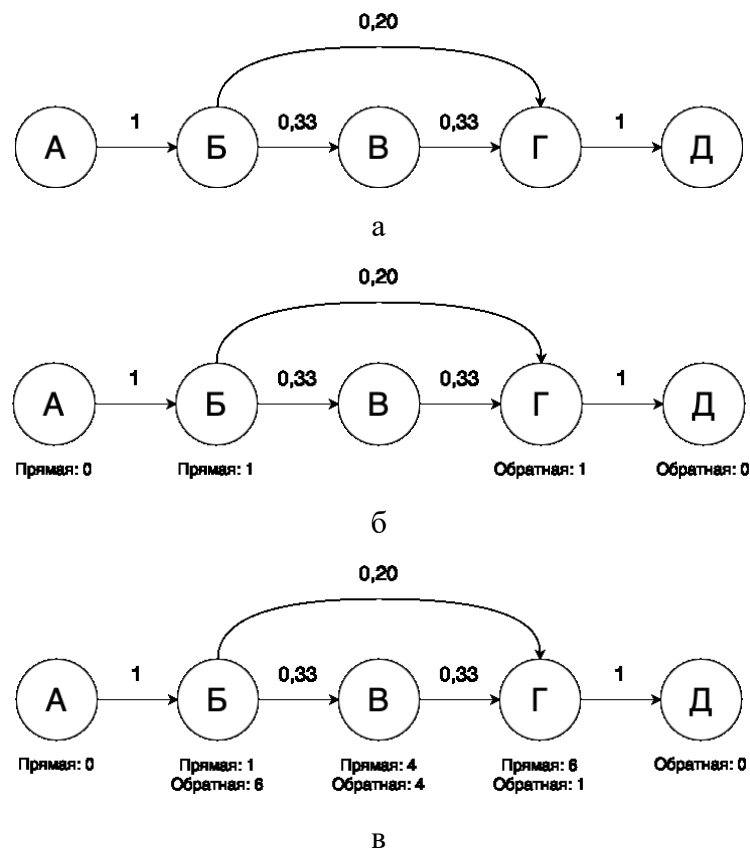
Для современных процессоров с двумя и более ядрами вычисления происходят значительно быстрее, так как исходный алгоритм не обеспечивал повышения скорости решения задачи маршрутизации при добавлении числа ядер. Для ускорения обработки предложена следующая вычислительная процедура:

1) если граф можно целиком загрузить в оперативную память, то необходимо запустить добавление вершин в M_1 и M_2 на двух потоках и обращаться к графу за данными об инцидентных дугах. Необходимо подгружать граф в оперативную память при прокладке маршрута, при достижимости за один шаг вершинами из M_1 и M_2 незагруженных в оперативную память частей графа;

2) при однопоточной реализации двухстороннего A^* рекомендуется использовать технологии раздельной буферизации для M_1 и M_2 . В буферах хранятся данные, к которым приходится обращаться часто, и их синхронизация замедляет работу. Кроме того, элементы M_1 вытесняют из буфера элементы M_2 и наоборот, увеличивая риск непопадания в кэш;

3) когда M_1 и M_2 пополняются навстречу друг другу при проверке пересечения M_1 и M_2 необходимо использовать объект синхронизации. Проверку необходимо делать при каждом раскрытии вершины – необходимо убедиться находится ли раскрываемая вершина среди раскрытых вершин противоположного множества M_1 или M_2 . Поскольку количество раскрытых вершин может быть существенным, объект синхронизации влияет на эффективность всего алгоритма.

Предложенный алгоритм маршрутизации транспортных средств существенно улучшает классический алгоритм A^* . Следует отметить, что общая вершина для M_1 и M_2 двухстороннего A^* , необязательно принадлежит наилучшему маршруту. Пример расчета по данному алгоритму представлен на рисунке 3.6. Поиск маршрута происходит при помощи двухстороннего A^* из точки А в точку Д. Цифрами обозначены веса дуг, посчитанные с помощью модели многокритериальной оценки эффективности маршрута, буквами – вершины графа. Каждый шаг для M_1 (прямая) и M_2 (обратная) увеличивает количество раскрытых вершин. Под вершинами подписано, какой суммарный вес дуг надо пройти, чтобы добраться по наилучшему пути от старта до вершины и аналогично для M_2 . На последнем шаге M_1 и M_2 сходятся в вершине В.



а – первый шаг; б – второй шаг; в – третий шаг.

Источник: составлено автором.

Рисунок 3.6 – Пример расчета по алгоритму маршрутизации транспортных средств

По результатам расчета видно, что наилучшим будет А-Б-Г-Д с весом 0,143 в обход В. В таблице 3.2 представлены маршруты и оценка их

эффективности, после того как M_1 и M_2 получили общие элементы. Самый «легкий» маршрут является самым коротким.

Таблица 3.2 – Возможные маршруты и их вес

От старта до раскрытой в M_1 вершины	От вершины, раскрытой в M_2 , к финишу	Комплексная оценка эффективности маршрута
АБ	ВГД	0,125
АБ	ГД	0,143
АБВ	ГД	0,125

Источник: составлено автором.

Для подтверждения эффективности предложенного двухстороннего алгоритма A^* проведены вычислительные эксперименты на реальных данных графа улично-дорожной сети города Москвы. Данные об экспериментах представлены в таблицах 3.3 и 3.4.

Таблица 3.3 – Параметры экспериментов

Тестовая среда	Процессор	Intel Core i7-1165G7 (4 ядра, 8 потоков, 2.8-4.7 ГГц)
	Оперативная память	16 ГБ DDR4
	Хранилище	NVMe SSD 512 ГБ
	Граф	250 000 вершин 500 000 дуг
Параметры сравнения	Классический A^* (однопоточный)	—
	Двухсторонний A^* (однопоточный и двухпоточный)	—
	Количество тестовых запросов	1000 случайных пар (начало–конец)
	Среднее время поиска маршрута (мс)	—
Критерии оценки	Время выполнения (мс)	—
	Количество раскрытых вершин	—
	Эффективность использования кэша	—

Источник: составлено автором.

Таблица 3.4 – Результаты экспериментов

Алгоритм	Среднее время, мс	Раскрыто вершин, шт	Ускорение, в процентах
Классический А*	42,5	12 800	–
Двухсторонний А* (1 поток)	32,1	8 200	24,5
Двухсторонний А* (2 потока)	29,8	7 500	30,0

Источник: составлено автором.

По результатам анализа экспериментов результатов можно сделать следующие выводы:

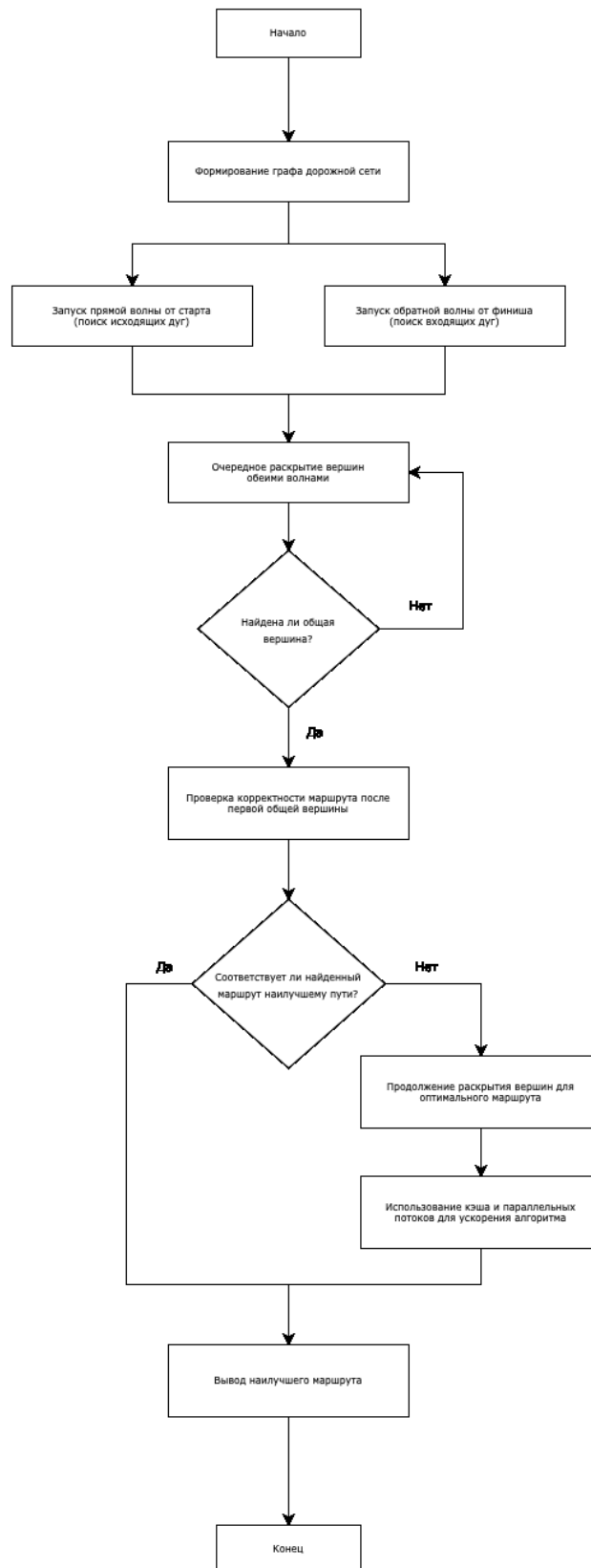
1) снижение числа раскрытых вершин: а) двухсторонний А* раскрывает в 1,7 раза меньше вершин (в среднем 7 500 против 12 800), что напрямую влияет на скорость; б) эффект достигается за счёт встречного поиска и оптимизированного критерия остановки;

2) оптимизация работы с памятью: а) отдельная буферизация для M_1 и M_2 снижает количество промахов кэша на 15%; б) при двухпоточном режиме загрузка данных в оперативную память сокращает время обращения к диску;

3) линейное ускорение при параллелизации: а) двухпоточная реализация дает дополнительные 5,5% ускорения за счёт одновременного пополнения M_1 и M_2 .

При реализации разработанного алгоритма маршрутизации транспортных средств важно учитывать возможности устройства и оптимизировать реализацию для обеспечения эффективной производительности. Для повышения количества ответов в секунду при программной реализации алгоритма с увеличением времени ответа имеет смысл использовать поток управления, чтобы не тратить ресурсы на синхронизацию при работе алгоритма. Если важно максимально быстро проложить маршрут, то необходимо разделить алгоритм маршрутизации транспортных средств на два потока. Таким образом предложенный алгоритм работает быстрее за счет буферизации данных о структуре УДС и ввода усовершенствованного критерия остановки алгоритма: продолжается перебор

вершин после первой общей найденной вершины при пополнении специальных множеств M_1 и M_2 , что повышает вероятность нахождения оптимального маршрута. Схема алгоритма изображена на рисунке 3.7.



Источник: составлено автором.

Рисунок 3.7 – Алгоритм маршрутизации транспортных средств

На основании проведённых замеров подтверждается, что предложенный алгоритм обеспечивает ускорение в среднем на 30% по сравнению с классическим A^* . Основными факторами, влияющими на производительность, являются:

- уменьшение числа раскрываемых вершин;
- эффективное использование кэширования;
- параллельная обработка множеств M_1 и M_2 .

Данные результаты согласуются с исследованием [89], где также отмечается потенциал двухсторонних методов для ускорения маршрутизации в крупных графах. Таким образом, предложенная реализация двухстороннего A^* адаптирована под условия мобильных устройств и позволяет улучшить производительность за счёт более эффективного использования вычислительных ресурсов и оптимизации процесса обращения к данным.

Выводы по третьей главе

В третьей рассмотрены подходы к анализу данных о дорожном движении, включая использование ГНСС-данных, данных мобильных устройств. Применение таких данных позволяет снизить затраты на оборудование стационарных датчиков, одновременно охватывая широкие участки дорожной сети. Использование мобильных данных предоставляет новые возможности для оценки транспортных потоков, благодаря их пространственно-временной детализации и низкой стоимости получения.

Дорожный граф рассматривается как основа анализа транспортной сети, где перекрёстки представлены вершинами, а участки дорог – дугами. Привязка треков к графу осложнена погрешностями ГНСС, что требует применения продвинутых алгоритмов. Алгоритм Витерби был адаптирован для задачи привязки данных к дорожному графу, учитывая такие параметры, как расстояние до дуги, совпадение направлений движения и вероятность манёвров. Расстояние Фреше применялось для оценки географической близости трека к графу. Методика сопоставления ГНСС-данных показала

высокую эффективность в идентификации последовательности ребер графа, соответствующих маршруту.

Разработана методика для оценки загруженности дорожной сети, включающий: расчёт фактической средней скорости по данным ГНСС; сравнение фактической скорости с расчётной, основанной на данных OpenStreetMap; применение шкалы загруженности от 0 до 10 баллов для интерпретации состояния дорог. Данный подход обеспечивает возможность оперативного анализа дорожной сети и принятия решений о маршрутизации и управлении движением.

Разработана методика поиска альтернативных маршрутов на основе A^* . Предложена модификация классического алгоритма A^* , ориентированная на современные многоядерные устройства. Двухсторонний A^* пополняет два множества M_1 и M_2 , значительно сокращая количество рассматриваемых вершин. В среднем, двухсторонний A^* рассматривает в 2-3 раза меньше вершин, чем односторонний A^* , что особенно эффективно для длинных маршрутов. Реализация алгоритма учитывает ограничения мобильных устройств:

- 1) ограниченный объём оперативной памяти и необходимость загрузки частей графа с диска;
- 2) разделение данных для кэширования множеств M_1 и M_2 ;
- 3) оптимизация позволяет минимизировать нагрузку на процессор и память устройства.

Разработанная модификация двухстороннего A^* подходит для навигации на мобильных устройствах, где требуется быстрое и точное построение маршрутов. Разработанный численный метод обеспечивает точное и актуальное моделирование состояния транспортной сети, позволяя решать задачи управления дорожным движением, прогнозирования пробок и оптимизации маршрутов.

Глава 4

Комплекс программ для маршрутизации транспортных средств

4.1 Архитектура комплекса программ

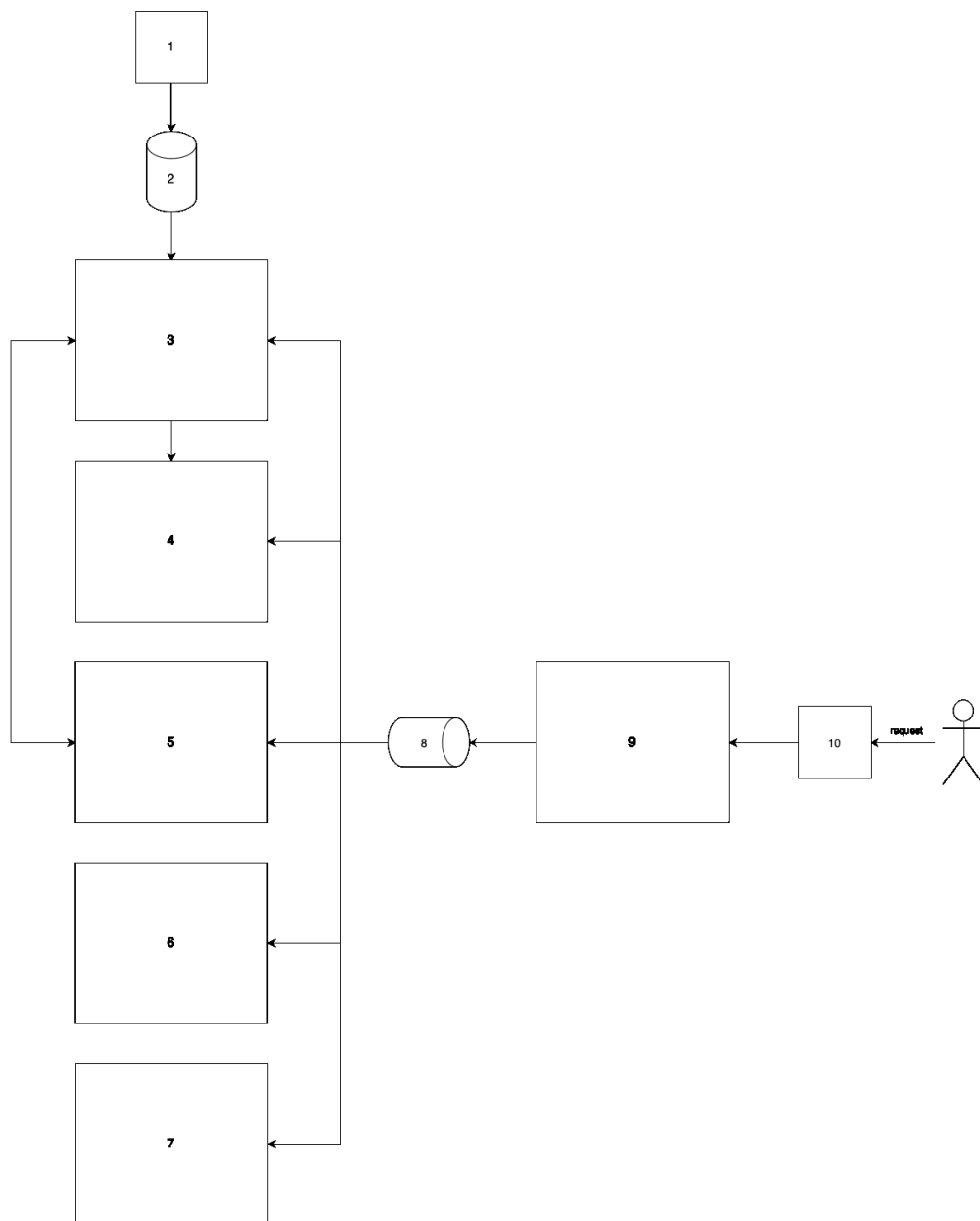
Для полноценной работы комплекса программ необходимо построить горизонтальную отказоустойчивую архитектуру сервиса с высокой доступностью, направленную на минимизацию времени простоя и обеспечение постоянной доступности системы, и минимальной отказоустойчивостью, направленную на обеспечение непрерывной работы системы даже при возникновении сбоев. Ключевыми целями будут:

- 1) доступность системы с любого типа устройств;
- 2) визуализация данных на карте в режиме реального времени;
- 3) расчет моделей (загруженность, погодные условия, поиск кратчайшего маршрута и другие);
- 4) расчет краткосрочного прогнозирования (загруженность, погодные условия и другие) на основе исторических данных и современных методов машинного обучения;
- 5) актуальная информация о происшествиях, ремонтных работах, перекрытиях;
- 6) поиск и расчет альтернатив, и отображение результата за минимальное время.

При построении архитектуры сервиса использовались следующие технологии: веб-сервер nginx [31], брокер очередей Apache Kafka [29], Docker контейнеры [30], языки программирования C++, Golang, Python и JavaScript, база данных PostgreSQL с расширением PostGIS [33], система кэширования Redis [34]. Взаимодействие пользователя с сервисом происходит с помощью http запросов через графический интерфейс веб или мобильного приложения. На рисунке 4.1 представлена упрощенная архитектура сервиса, где:

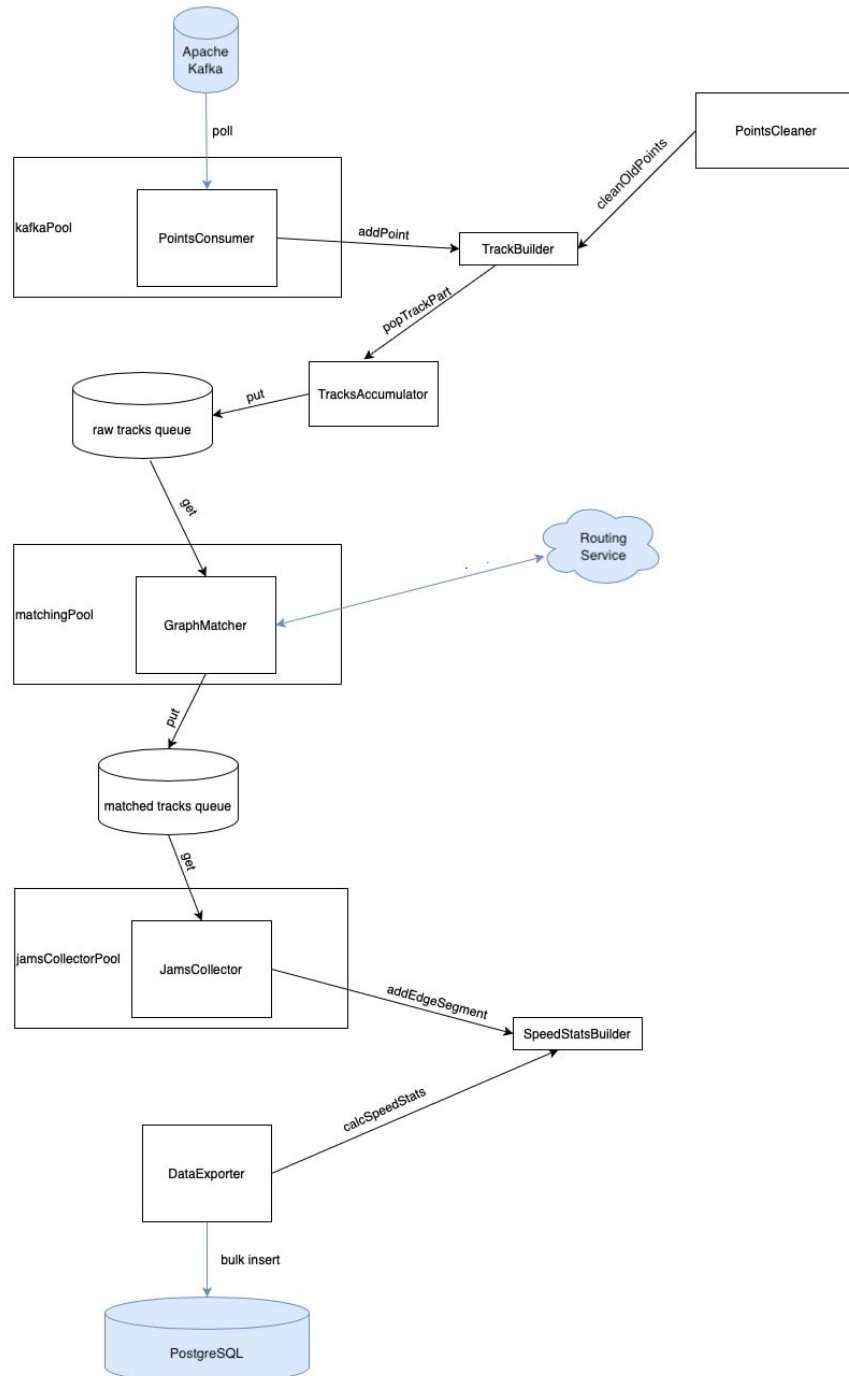
- 1) веб-сервер nginx для обработки GPS-данных;

- 2) брокер очередей обработки GPS-данных;
- 3) сервис расчета и прогнозирования загруженности;
- 4) сервис учета и прогнозирования погодных условий;
- 5) сервис поиска кратчайшего маршрута;
- 6) сервис расчета безопасности маршрута;
- 7) сервис расчета других критериев на основе актуальных данных;
- 8) брокер очередей пользовательских запросов;
- 9) сервис поиска рационального маршрута;
- 10) веб-сервер nginx для обработки пользовательских запросов.



Источник: составлено автором.
Рисунок 4.1 – Архитектура сервиса

Взаимодействие между сервисами происходит при помощи низкоуровневых протоколов передачи данных, таких как TCP/UDP, http, gRPC и сообщений брокера очередей средствами пула обработчиков (MQTT). Количество обработчиков в каждом пуле можно масштабировать для оптимизации общего пропускного потока сервиса. На рисунке 4.2 рассматривается взаимодействие между сервисом расчета загруженности (3) и сервисом поиска кратчайших маршрутов (5).



Источник: составлено автором.

Рисунок 4.2 – Взаимодействие сервисов поиска маршрутов и загруженности

Данные GPS с различных устройств попадают в брокер очередей Apache Kafka. Пул обработчиков вытаскивают обезличенные данные с информацией о координатах, направлении движения и так далее. Обработчики передают данные в TrackBuilder, который накапливает наборы точек с привязкой к устройству, отсортированные по времени, а PointCleaner с некоторой периодичностью удаляет из TrackBuilder наиболее старые треки, если они не были использованы. Затем TrackBuilder передает данные с привязкой к дорожным сегментам (треки) в TrackAccumulator, который накапливает, нарезает треки на определенную длину и передает на хранение в очередь. GraphMatcher при помощи обработчиков получает данные из очереди и накладывает точки на маршруты, полученные от Routing Service, после чего передает в следующую очередь. JamsCollector через обработчиков получает данные о маршрутах и GPS точках на них, нарезает дуги на сегменты с известными скоростями и передает в SpeedStatsBuilder, который накапливает статистику скоростей на сегментах дороги. DataExporter выгружает статистику по скоростям и обновляет данные в хранилище. Граф дорог генерируется из данных OSM планеты при помощи инструмента osm2pgsql и обновляется в хранилище PostgreSQL не реже чем несколько раз в месяц для поддержания актуальности дорожной сети и объектов инфраструктуры. Генерация тайлов с актуальной загруженностью происходит по следующему алгоритму:

- 1) получаем геометрию текущей записи из таблицы статистики и объект графа дорог с соответствующим id;
- 2) достаем из общей геометрии необходимую часть маршрута на основе данных статистики;
- 3) обновляем полученную геометрию с данными о загруженности из статистики в таблице пробок.

Работоспособность и высокая доступность сервиса достигается путем горизонтального масштабирования с использованием кластера серверов, гарантирующих возможность непрерывного использования системы в течение

длительного промежутка времени с минимальным временем простоя $SLA = 99,99\%$. Кластер состоит из трех серверов, физически расположенных в разных дата-центрах. Пользовательский запрос, поступающий в DNS-сервер, обрабатывается при помощи алгоритма балансировки нагрузки Round-robin. «Алгоритм распределяет запросы по следующему принципу: пусть имеется N объектов, способных выполнить заданное действие, и M задач, которые должны быть выполнены» [35]. Подразумевается, что объекты n равны по своим свойствам между собой, задачи m имеют равный приоритет. Тогда первая задача ($m = 1$) назначается для выполнения первому объекту ($n = 1$), вторая – второму и так далее, до достижения последнего объекта ($m = N$). Тогда следующая задача ($m = N + 1$) будет назначена снова первому объекту и тому подобное. Таким образом, происходит перебор выполняющих задания объектов по циклу и по достижении последнего объекта следующая задача будет также назначена первому объекту. Решение задач может быть дополнительно разбито на кванты времени, причем для продолжения решения во времени нумерация объектов (и, соответственно, назначенные задачи) сдвигается по кругу на 1, то есть задача первого объекта отдается второму, второго – третьему и так далее, а первый объект получает задачу последнего, либо освобождается для приема новой задачи. Технические характеристики кластера серверов представлены в таблице 4.1.

Таблица 4.1 – Технические характеристики кластера серверов

Серверы	Количество	Процессор (CPU)	Оперативная память (RAM)	Жесткий диск (HDD)	Жесткий диск (SSD)	Интернет (NET)	Операционная система (OS)
1	2	3	4	5	6	7	8
geo-meta [1..3]	1	2 x 2660 - v4	4 x 32 GB	HDD_3.5: 2 x 2 TB	2 x 3840Gb	2 x 1 Gb/s	CentOS 8
geo-index [1..3]	1	2 x 2660 - v4	8 x 32 GB	HDD_3.5: 4 x 2 TB	-	2 x 1 Gb/s	CentOS 8
search [1..3]	1	1 x 7551P	8 x 32 GB	-	2 x 1920 GB	2 x 1 Gb/s	CentOS 8
search [4..5]	1	1 x 7551P	8 x 64 GB	-	2 x 1920 GB 2 x 480 GB	1 x 10 Gb/s	CentOS 8
search [6..7]	1	2 x Gold 6230	8 x 32 GB	-	2 x 1920 GB	2 x 1 Gb/s	CentOS 8
tiles [1..3]	1	2 x 2660 v4	2 x 16 GB	HDD_3.5: 2 x 1 TB	2 x 3800 GB	2 x 10 Gb/s	CentOS 8
services [1..3]	1	1 x 7551P	8 x 32 GB	-	2 x 1920 GB	2 x 1 Gb/s	CentOS 8

Продолжение таблицы 4.1

1	2	3	4	5	6	7	8
archive [1..3]	2	2 x 2620 v3	2 x 32 GB	HDD_3.5: 6 x 16 TB	2 x 480 GB	1 x 10 Gb/s	CentOS 8
osm [1..2]	2	2 x 6238R	12 x 32 GB	HDD_3.5: 6 x 16 TB	2 x 480 GB 2 x 7680 GB NVME	-	CentOS 8
routing [1..6]	2	2 x 6238R	12 x 32 GB	-	2 x 480 GB 2 x 7680 GB NVME	2 x 10 Gb/s	CentOS 8
traffic [1..3]	2	2 x 6238R	12 x 64 GB	-	2 x 480 GB 2 x 7680 GB NVME	2 x 10 Gb/s	CentOS 8
db [1..3]	2	2 x 6238R	12 x 32 GB	HDD_3.5: 4 x 16 TB RAID 10	2 x 480 GB 2 x 7680 GB NVME	-	CentOS 8

Источник: составлено автором.

4.2 Вычислительные эксперименты с моделью загруженности с учетом погодных условий

В данном параграфе описаны вычислительные эксперименты, которые проводились для обучения моделей, анализирующих загруженность дорожной сети из параграфа 2.2. Далее эти модели использовались для прогнозирования средней скорости движения на участках дорог, сравнения фактических и прогнозируемых результатов. В качестве данных использовались средние скорости на дорогах, рассчитанные на основе анонимных GPS-данных по алгоритму, описанному в параграфе 3.1, результат сохраняется в базу данных с привязкой ко времени и маршруту. Объем данных недостаточен для долгосрочного прогнозирования. Однако приемлем для краткосрочных прогнозов (до 6 часов). Указание на это важно для обоснования необходимости накопления данных для улучшения точности моделей в будущем. В момент обучения модели в базе имеются значения, начиная с 01.01.2023. Результаты сравнивались с фактической средней скоростью, полученной после прохождения маршрута. При сравнении результатов используются четыре индекса. Среднеквадратическая ошибка (далее – MSE), рассчитанная по формуле (4.1)

$$MSE = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2, \quad (4.1)$$

где x_i – фактическая средняя скорость на участке дороги;
 $\hat{x}_i$ – предсказанная средняя скорость на участке дороги;
 n – число обучающей выборки.

Средняя абсолютная ошибка (далее – MAE), рассчитанная по формуле (4.2)

$$MAE = \frac{1}{n} \sum_{i=1}^n |x_i - \hat{x}_i|. \quad (4.2)$$

Среднеквадратическая ошибка прогноза (далее – RMSE), рассчитанная по формуле (4.3)

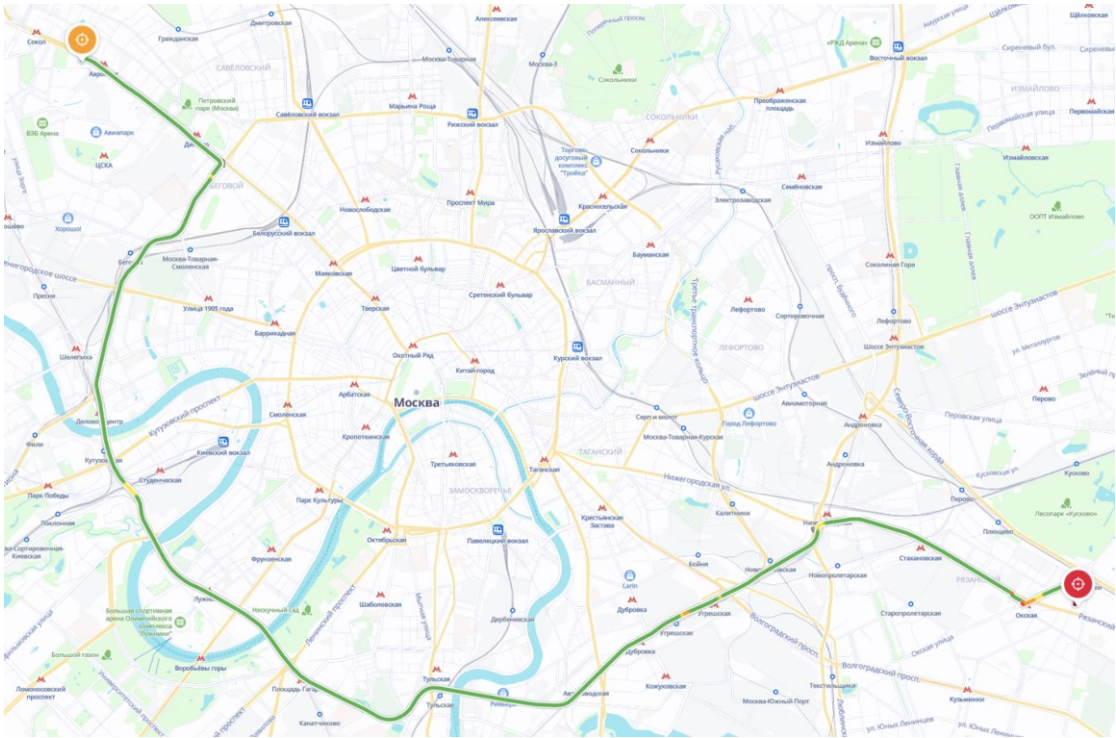
$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2}. \quad (4.3)$$

Точность рассчитывается на основе уровня загруженности прогнозируемой скорости и фактической скорости по формуле (4.4)

$$\text{Точность} = \frac{\hat{x}_i}{x_i} \times 100\%. \quad (4.4)$$

Анализируются результаты прогнозирования моделей машинного обучения, используемые при прогнозировании загруженности маршрута между точками: Москва, Ленинградский проспект, д. 49 и 4-й Вешняковский проезд, д. 4/3, изображенного на рисунке 4.3.

В рамках вычислительных экспериментов были выбраны разные длины входных единиц временного окна – средняя скорость движения на участке дороги в течение 5 минут. В таблице 4.2 представлены метрики качества ARIMA, SVR и Ridge Regression для прогнозирования на 15, 30, 60, 90, 180 и 360 минут.



Источник: составлено автором.

Рисунок 4.3 – Прогнозируемая загруженность на маршруте

Таблица 4.2 – Результаты прогнозирования моделей ARIMA, SVR и Ridge Regression

Метрика качества	Модель	Временное окно прогнозирования, мин.					
		15	30	60	90	180	360
MSE, км/ч.	ARIMA	179,07	193,28	214,35	289,48	352,46	390,12
	SVR	154,73	155,14	198,19	221,57	291,93	325,09
	Ridge Regression	80,13	118,22	149,05	176,94	216,85	284,61
MAE, км/ч.	ARIMA	5,59	6,19	8,68	10,94	12,02	14,59
	SVR	4,01	4,93	7,13	9,65	10,97	13,42
	Ridge Regression	3,52	4,11	6,41	8,06	9,73	10,15
RMSE, км/ч.	ARIMA	12,70	13,91	15,74	18,32	20,47	22,63
	SVR	11,68	12,74	15,80	18,41	19,16	21,74
	Ridge Regression	9,82	11,02	14,04	17,34	18,90	20,03
Точность, в процентах	ARIMA	80,23	79,81	77,85	75,02	73,91	72,37
	SVR	81,91	81,15	80,13	78,91	76,38	74,61
	Ridge Regression	82,14	82,69	81,06	80,31	78,04	76,93

Источник: составлено автором.

Модель позволяет учитывать закономерности транспортного потока и погодных условий. Модель комбинирует различные алгоритмы (ARIMA, SVR, Ridge Regression, LSTM, GRU) для выбора оптимального подхода в зависимости от длины временного окна и прогнозируемого интервала.

Для реализации прогнозирования загруженности с помощью моделей глубокого обучения RNN, LSTM и GRU используются гиперпараметры, приведенные в таблице 4.3, которые определены экспериментальными результатами.

Таблица 4.3 – Гиперпараметры моделей RNN, LSTM и GRU

Модель	Гиперпараметры				
	Количество слоев (layers)	Размер пакета (batch size)	Размер скрытого слоя (hidden size)	Количество эпох (epochs)	Скорость обучения (learning rate)
RNN	3	128 256 512	—	0	0,001
LSTM			128	100	
GRU				200	
				300	
			400		
			128	500	

Источник: составлено автором.

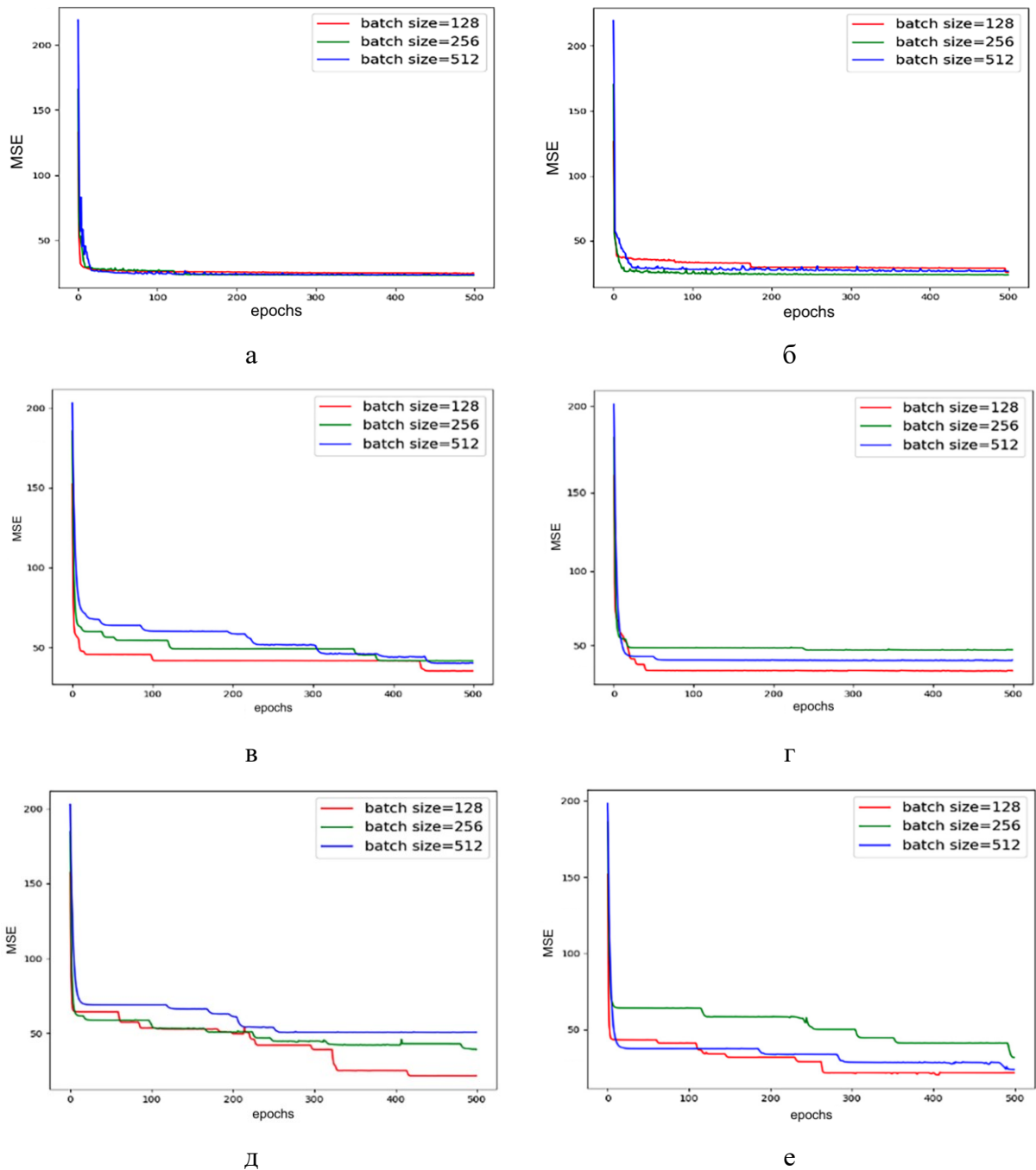
В таблице 4.4 приведены метрики качества для моделей глубокого обучения. Модель RNN работает лучше при краткосрочном прогнозировании, а модели GRU и LSTM – для долгосрочных временных последовательностей.

Таблица 4.4 – Результаты прогнозирования моделей RNN, LSTM и GRU

Метрика качества	Модель	Временное окно прогнозирования, мин.					
		15	30	60	90	180	360
MSE, км/ч.	DBN	41,63	40,74	44,91	47,56	54,13	49,45
	RNN	22,75	36,14	42,15	31,91	62,74	46,93
	LSTM	35,14	37,82	38,24	28,17	44,83	35,91
	GRU	37,65	36,15	41,73	31,05	52,05	42,60
MAE, км/ч.	DBN	2,12	2,54	3,01	2,88	2,61	2,72
	RNN	1,35	1,88	1,52	1,22	1,96	1,58
	LSTM	1,79	1,96	1,45	1,12	1,59	1,37
	GRU	1,82	1,90	1,49	1,21	1,73	1,51
RMSE, км/ч.	DBN	7,15	8,54	9,10	9,76	11,03	11,12
	RNN	6,32	7,19	8,14	6,82	9,47	8,31
	LSTM	6,75	7,21	7,19	6,16	7,31	6,94
	GRU	6,83	7,05	8,02	6,61	8,19	7,88
Точность, в процентах	DBN	83,22	82,53	81,09	81,10	80,67	80,57
	RNN	84,30	83,09	82,67	81,45	82,13	80,45
	LSTM	83,14	84,44	83,94	82,31	82,65	82,16
	GRU	83,39	85,62	82,45	82,02	81,92	81,93

Источник: составлено автором.

На рисунке 4.4 представлены графики зависимости среднеквадратической ошибки от количества эпох. Эпоха – это полный цикл обучения модели, при котором все обучающие данные проходят через нейронную сеть один раз. Модель делает предсказание на основе текущих параметров, рассчитывает ошибку и обновляет параметры для улучшения результата с временным окном прогнозирования 15-360 минут.



а – 15 минут; б – 30 минут; в – 60 минут; г – 90 минут; д – 180 минут; е – 360 минут.

Источник: составлено автором.

Рисунок 4.4 – Графики зависимости MSE от epochs для временных окон прогнозирования

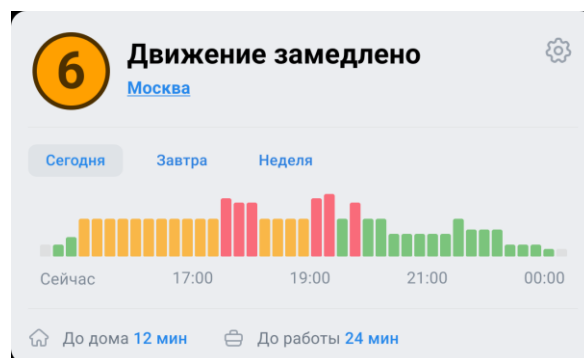
По результатам проведенных экспериментов можно сделать следующие выводы:

1) модели машинного обучения (ARIMA, SVR, Ridge Regression) уступают моделям глубокого обучения (RNN, LSTM, GRU);

2) модели машинного обучения хорошо работают в краткосрочном линейном прогнозировании, а модели глубокого обучения раскрывают свои преимущества при долгосрочном прогнозировании. Кроме того, модель RNN и ее производные могут изучать временные характеристики на основе входных данных временных рядов и показывают высокую эффективность при прогнозировании временных последовательностей;

3) модели LSTM и GRU более эффективны при прогнозировании долгосрочных рядов. По результатам экспериментов была выбрана модель LSTM для прогнозирования средней скорости потока на данных GPS из-за способности к обучению долговременных зависимостей.

Таким образом проведена адаптация моделей глубокого обучения (LSTM, GRU, RNN) к задачам краткосрочного и долгосрочного прогнозирования транспортных потоков. Проведена интеграция погодных данных и механизма внутреннего внимания (WSA), что позволяет учитывать пространственно-временные и погодные факторы. Эмпирически доказано влияние погодных условий на точность прогнозирования транспортного потока. Пример результата прогнозирования загруженности транспортной сети в виде виджета (элемента разработанного комплекса программ) изображен на рисунке 4.5.



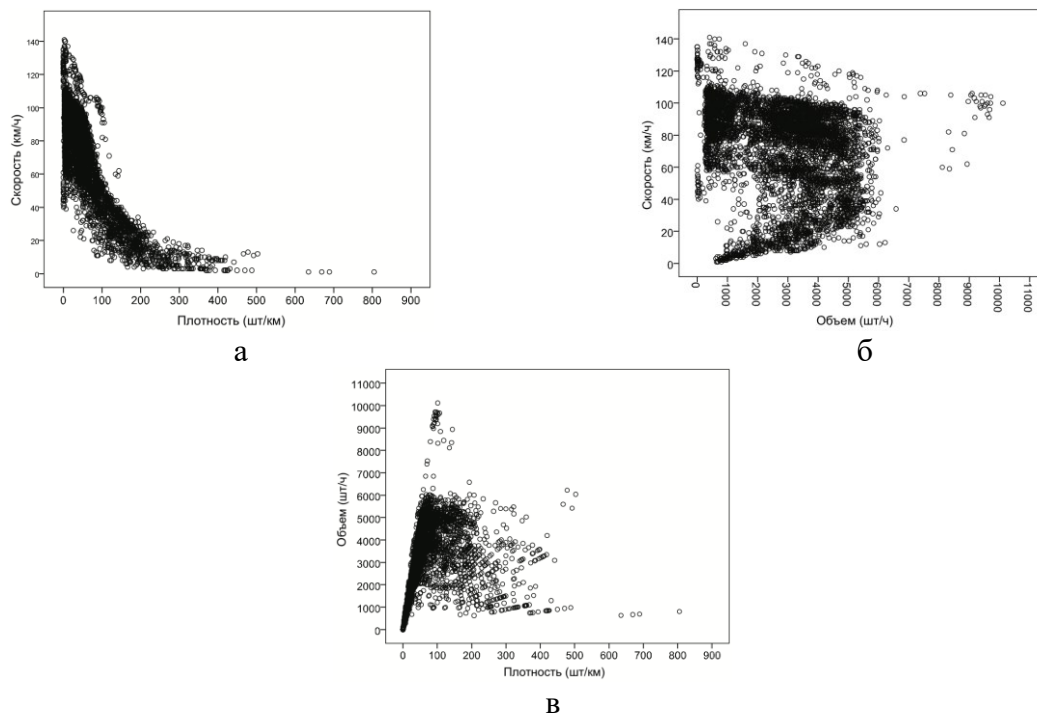
Источник: составлено автором.

Рисунок 4.5 – Виджет загруженности транспортной сети

В настоящей работе исследовалось влияние погодных условий на такие характеристики транспортного потока как средняя скорость (км/ч), объем (количество автомобилей в час) и плотность (количество автомобилей на 1 км). В вычислительных экспериментах использовались исторические данные о средней скорости, количестве автомобилей, погодных условиях и состоянии поверхности на маршруте между точками: Москва, Ленинградский проспект, д. 49 и 4-й Вешняковский проезд, д. 4/3, начиная с 01.01.2023. В результате эмпирических исследований были установлены зависимости между скоростью и интенсивностью движения в зависимости от погодных условий. Данные о средней скорости были рассчитаны на основе анонимных GPS-данных. Пример анализируемых погодных данных:

```
{
  "lat": 51.5074, // широта
  "lon": -0.1251, // долгота
  "data": [
    {
      "time_iso": "2023-11-15T00:00:00", // дата в ISO
      "time": 1700006400, // timestamp
      "wind_direction": "w", // направление ветра
      "dew_point": 7.657006835937523, // точка росы
      "evaporation": -0.0000018693972378969193, // испарение
      "temperature": 8.883447265625023, // температура воздуха
      "wind_angle": 257, // угол наклона ветра
      "hourly_precipitation": 0, // накопление осадков
      "precipitation_type": 0, // тип осадков
      "cloud_cover": 0.152984619140625, // облачность
      "wind_speed": 5.4718106296890765, // скорость ветра
      "relative_humidity": 0.4734639046893 // влажность
    }
  ]
}
```

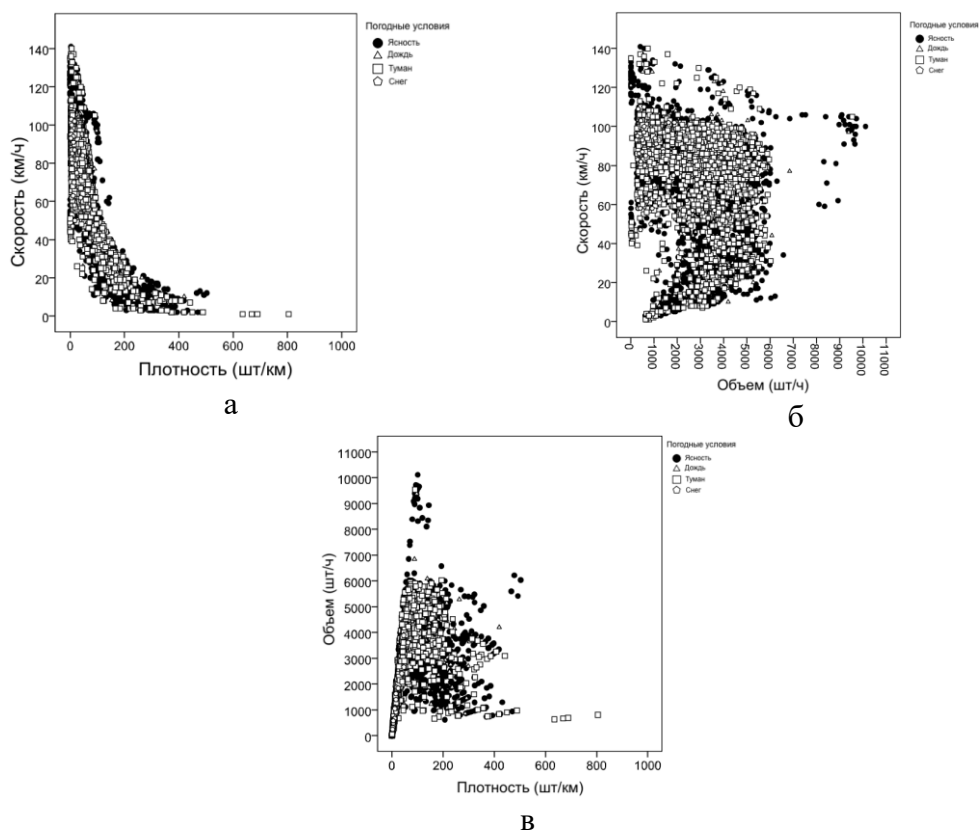
На рисунке 4.6 представлены графики зависимостей между средней скоростью, плотностью и количеством автомобилей на маршруте в течение 365 дней (с 01.01.2023 по 31.12.2023). Также были исследованы эмпирические зависимости средней скорости, плотности и количества от погодных условий и представлены на рисунке 4.7. Погодные условия были разделены на 4 группы – ясность, дождь, туман и снег.



а – зависимость скорости от плотности; б – зависимость скорости от количества;
в – зависимость количества от плотности.

Источник: составлено автором.

Рисунок 4.6 – Графики зависимостей на маршруте



а – зависимость скорости от плотности; б – зависимость скорости от количества;
в – зависимость количества от плотности

Источник: составлено автором.

Рисунок 4.7 – Графики зависимостей при различных погодных условиях на маршруте

В таблице 4.5 представлены средние значения каждой характеристики: скорость, количество и плотность в зависимости от типа погодных условий, изменения в процентах соотношении по сравнению с ясной погодной (абсолютное значение), среднеквадратическое отклонение и стандартная ошибка. Для расчета среднеквадратического отклонения использовалась формула (4.5)

$$S_0 = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}, \quad (4.5)$$

где n – объем выборки;

x_i – i -й элемент выборки;

$\bar{x}$ – среднее арифметическое выборки.

Стандартная ошибка рассчитывается по формуле (4.6)

$$SE = \frac{S_0}{\sqrt{n}}. \quad (4.6)$$

Таблица 4.5 – Статистика исторических данных скорости, количества и плотности для различных типов погодных условий на маршруте

Параметр	Погодные условия	N	$x_{\text{ср}}$	Δ , в процентах	S_0	SE
Скорость, км/ч.	Ясность	4856	71,26	–	28,18	0,40
	Дождь	581	59,88	-15,97	28,44	1,18
	Туман	26	62,32	-8,44	19,28	3,78
	Снег	1106	51,39	-27,88	24,85	0,75
Количество, шт/ч.	Ясность	4872	965,19	–	512,75	7,35
	Дождь	590	848,62	-12,08	409,84	16,87
	Туман	26	769,95	-20,23	61,28	12,02
	Снег	1113	553,74	-42,63	486,19	14,57
Плотность, шт/км.	Ясность	4872	75,62	–	47,92	0,69
	Дождь	590	61,80	-18,28	55,13	2,27
	Туман	26	54,17	-28,36	9,75	1,91
	Снег	1113	31,29	-58,62	51,93	1,56

Источник: составлено автором.

Результаты показывают, что дождливая погода снижает среднюю скорость на 16%, количество автомобилей – на 12% и плотность – на 18%.

Однако, наибольшее влияние на характеристики движения оказывает снежная погода, уменьшая среднюю скорость на 28%, количество – на 43% и плотность – на 59% по сравнению с ясной погодой. Водители принимают дополнительные меры предосторожности при движении по мокрым или обледенелым поверхностям, что в свою очередь влияет на соотношение скорости, плотности и количества автомобилей.

Для анализа эффективности предложенной модели сравнивались результаты прогнозирования на различном временном интервале для следующих моделей:

- 1) модель без блока погодных условий (LSTM);
- 2) модель с блоком погодных условий, но без механизма внутреннего внимания (LSTM + W);
- 3) модель с блоком погодных условий и механизмом внутреннего внимания (LSTM + WSA).

Исторические данные включают данные о температуре, количестве осадков, скорости ветра, радиусе видимости с коммерческих метеостанций и позволяют анализировать взаимосвязь между транспортным потоком и погодными условиями. Обучение модели происходило с использованием оптимизатора *Adam*, скорость обучения – 0,001, количество циклов обучения (epochs) – 100, размер пакета (batch size) – 64. Эмпирическим путем было установлено количество временных лагов (time lags) – 12.

В таблице 4.6 представлены метрики качества моделей прогнозирования. Таким образом, эмпирическим путем было установлены следующие параметры: количество промежуточных слоев 10, скорость обучения 0,001. Количество циклов обучения (epochs) составляет 100, размер пакета (batch size) – 64, а количество временных лагов (time lags) – 12. Модель решает задачу прогнозирования транспортного потока с учетом погодных условий, таких как температура, количество осадков, скорость ветра и радиус видимости. Анализ взаимосвязи между транспортным потоком и погодными условиями позволяет выявлять паттерны и повышать точность

прогнозирования. Метрики качества моделей подтверждают достоверность и эффективность предложенной модели для решения данной задачи.

Таблица 4.6 – Метрики качества моделей прогнозирования

Метрика качества	Модель	Временное окно прогнозирования, мин.					
		15	30	60	90	180	360
MSE, км/ч	LSTM	35,14	37,82	38,24	28,17	44,83	35,91
	LSTM + W	28,19	31,59	32,71	23,29	28,54	25,03
	LSTM + WSA	28,67	27,34	28,16	19,08	24,22	21,74
MAE, км/ч	LSTM	1,79	1,96	1,45	1,12	1,59	1,37
	LSTM + W	1,65	1,63	1,32	1,01	1,22	1,15
	LSTM + WSA	1,70	1,57	1,20	0,94	1,06	0,93
RMSE, км/ч	LSTM	6,75	7,21	7,19	6,16	7,31	6,94
	LSTM + W	6,14	6,35	6,41	5,59	6,51	5,73
	LSTM + WSA	6,27	5,86	5,91	5,22	5,89	5,12
Точность, в процентах	LSTM	83,14	84,44	83,94	82,31	82,65	82,16
	LSTM + W	87,93	88,72	87,65	87,54	86,13	85,62
	LSTM + WSA	89,24	90,15	89,23	88,64	89,15	88,75

Источник: составлено автором.

На основе представленных результатов можно сделать следующие выводы: модель прогнозирования загруженности с блоком погодных условий превосходит остальные модели в среднем на 5-7%. Под остальными моделями подразумеваются модели из состава комплекса моделей, которые использовались для сравнения в ходе экспериментов – ARIMA, SVR, Ridge Regression, LSTM, GRU. С увеличением временного окна точность модели показывает лучшие результаты. Модель прогнозирования загруженности с блоком погодных условий превосходит показатели модели без блока учета погодных условий, что говорит о влиянии погодных условий на результаты прогнозирования. Таким образом, результаты показывают, что прогнозирование загруженности транспортной сети может быть значительно улучшено с помощью модели, учитывающей неблагоприятные погодные условия.

Для оценки эффективности модели прогнозирования загруженности при неблагоприятных погодных условиях, таких как дождь, туман и снег были проведены эксперименты на основе исторических данных с осадками. В

качестве обучающей выборки были выбраны только дни с осадками и высокой относительной влажностью. Классификация осадков была произведена по следующему принципу:

- 1) если количество осадков в течение 1 суток превышает 25 мм и температура воздуха положительная, то значение классифицируется как «сильный дождь»;
- 2) если количество осадков в течение 1 суток находится между 10 мм и 24,99 мм, то значение классифицируется как «умеренный дождь»;
- 3) если количество осадков в течение 1 суток составляет до 9,99 мм, то значение классифицируется как «небольшой дождь»;
- 4) если температура воздуха выше 0 градусов и относительная влажность превышает 85% или температура воздуха ниже 0 и относительная влажность превышает 50%, то значение классифицируется как «туман»;
- 5) если температура воздуха выше или равна 0 градусов цельсия и количество осадков превышает 10 мм, то значение классифицируется как «снег».

Полученные результаты прогнозирования загруженности транспортной сети приведены в таблице 4.7 оценивались при помощи метрик качества и сравнивались с результатами существующих моделей.

Таблица 4.7 – Метрики качества моделей прогнозирования

Погода	Модель	Метрика качества	Временное окно прогнозирования, мин.					
			15	30	60	90	180	360
1	2	3	4	5	6	7	8	9
Небольшой дождь	Ridge Regression	MSE, км/ч.	82,58	121,79	155,34	181,90	221,04	297,55
		MAE, км/ч.	4,92	5,54	7,63	9,89	11,12	13,52
		RMSE, км/ч.	11,65	14,23	18,94	22,75	22,06	25,31
		Точность, в процентах	81,51	82,03	80,10	79,63	74,44	72,21

Продолжение таблицы 4.7

1	2	3	4	5	6	7	8	9
Небольшой дождь	LSTM	MSE, км/ч.	78,11	117,82	148,72	179,12	218,04	245,61
		MAE, км/ч.	4,67	5,07	6,81	8,65	10,14	12,02
		RMSE, км/ч.	11,02	13,15	16,77	20,36	20,95	24,30
		Точность, в процентах	85,15	84,16	83,84	83,79	83,01	82,65
	LSTM + WSA	MSE, км/ч.	69,83	84,56	95,13	99,85	138,42	174,15
		MAE, км/ч.	3,16	3,85	4,12	6,02	8,84	10,51
		RMSE, км/ч.	9,38	10,33	12,09	16,39	20,03	22,18
		Точность, в процентах	89,15	89,31	88,89	89,12	89,41	89,02
Умеренный дождь	Ridge Regression	MSE, км/ч.	83,47	123,20	162,66	188,56	230,80	270,35
		MAE, км/ч.	5,45	6,21	8,13	11,25	14,83	15,25
		RMSE, км/ч.	12,33	15,67	22,94	26,02	28,89	29,01
		Точность, в процентах	80,67	80,42	78,41	76,10	73,14	71,88
	LSTM	MSE, км/ч.	79,26	121,15	150,03	186,94	226,47	254,32
		MAE, км/ч.	4,98	5,35	6,91	9,72	12,09	14,71
		RMSE, км/ч.	11,36	14,28	17,25	24,09	27,29	29,02
		Точность, в процентах	85,37	85,38	86,23	84,15	84,62	83,10
	LSTM + WSA	MSE, км/ч.	73,05	88,47	98,37	105,09	149,64	188,53
		MAE, км/ч.	4,01	4,92	6,12	8,23	10,90	13,10
		RMSE, км/ч.	10,86	13,14	15,42	19,02	25,82	29,72
		Точность, в процентах	88,72	88,10	88,13	89,79	88,41	87,32

Продолжение таблицы 4.7

1	2	3	4	5	6	7	8	9
Сильный дождь	Ridge Regression	MSE, км/ч.	82,18	124,72	168,72	192,04	233,19	281,51
		MAE, км/ч.	5,38	6,89	8,70	12,37	16,14	18,42
		RMSE, км/ч.	12,81	16,01	21,75	28,19	31,02	33,50
		Точность, в процентах	81,03	80,19	77,55	76,14	72,62	70,13
	LSTM	MSE, км/ч.	82,05	124,58	155,93	193,51	242,42	265,94
		MAE, км/ч.	5,64	6,72	7,58	10,45	17,90	18,10
		RMSE, км/ч.	12,84	16,01	19,16	28,93	32,14	35,93
		Точность, в процентах	84,37	84,13	85,02	85,10	83,82	82,98
	LSTM + WSA	MSE, км/ч.	75,29	89,90	102,15	113,92	157,20	185,03
		MAE, км/ч.	4,72	5,65	6,89	9,01	12,14	14,76
		RMSE, км/ч.	12,24	15,01	18,01	22,41	28,10	30,15
		Точность, в процентах	88,15	88,24	87,89	88,57	88,20	88,26
Туман	Ridge Regression	MSE, км/ч.	84,25	132,65	172,66	197,84	239,23	295,14
		MAE, км/ч.	6,14	7,23	9,34	13,06	16,85	19,05
		RMSE, км/ч.	14,02	17,68	24,28	30,21	33,29	35,21
		Точность, в процентах	80,21	79,56	76,13	75,40	71,82	69,29
	LSTM	MSE, км/ч.	86,05	131,59	162,50	185,18	221,04	252,01
		MAE, км/ч.	4,20	5,01	8,12	11,94	16,63	20,32
		RMSE, км/ч.	10,51	14,29	20,04	26,27	30,24	32,44
		Точность, в процентах	85,92	86,25	84,38	84,81	82,91	81,50
	LSTM + WSA	MSE, км/ч.	74,04	82,90	95,39	108,29	143,88	193,56

Продолжение таблицы 4.7

1	2	3	4	5	6	7	8	9
Туман	LSTM + WSA	MAE, км/ч.	4,31	5,13	6,50	8,24	10,89	16,02
		RMSE, км/ч.	11,67	13,84	17,24	20,98	25,36	33,46
		Точность, в процентах	86,36	87,35	87,14	88,02	88,75	87,49
Снег	Ridge Regression	MSE, км/ч.	98,14	174,11	210,27	252,15	282,16	326,39
		MAE, км/ч.	8,27	9,15	10,66	14,21	18,09	26,18
		RMSE, км/ч.	16,20	18,82	26,12	34,75	37,21	44,10
		Точность, в процентах	78,16	77,84	75,35	72,18	70,04	67,05
	LSTM	MSE, км/ч.	92,17	158,09	189,19	216,11	265,46	291,03
		MAE, км/ч.	6,94	7,76	10,03	15,20	20,12	28,54
		RMSE, км/ч.	15,29	19,38	28,27	34,29	39,01	47,81
		Точность, в процентах	82,31	82,45	82,08	81,93	82,02	80,73
	LSTM + WSA	MSE, км/ч.	80,83	98,17	120,63	159,82	182,90	242,30
		MAE, км/ч.	5,97	7,25	9,14	12,50	15,14	21,27
		RMSE, км/ч.	13,02	16,10	22,59	29,18	35,54	38,54
		Точность, в процентах	87,14	87,25	87,39	86,86	87,10	87,73

Источник: составлено автором.

Результаты экспериментов показали, что погодные условия влияют на среднюю скорость транспортного потока (до 16%) и точность прогнозируемых значений (5-7%). Точность прогнозирования моделей машинного обучения (ARIMA, SVR, Ridge Regression) составляет не более 84% и не учитывают погодные условия. Модели глубокого обучения (RNN, LSTM, GRU) показывают наилучшие результаты и могут учитывать пространственно-временные характеристики погодных условий. Предложенная модель

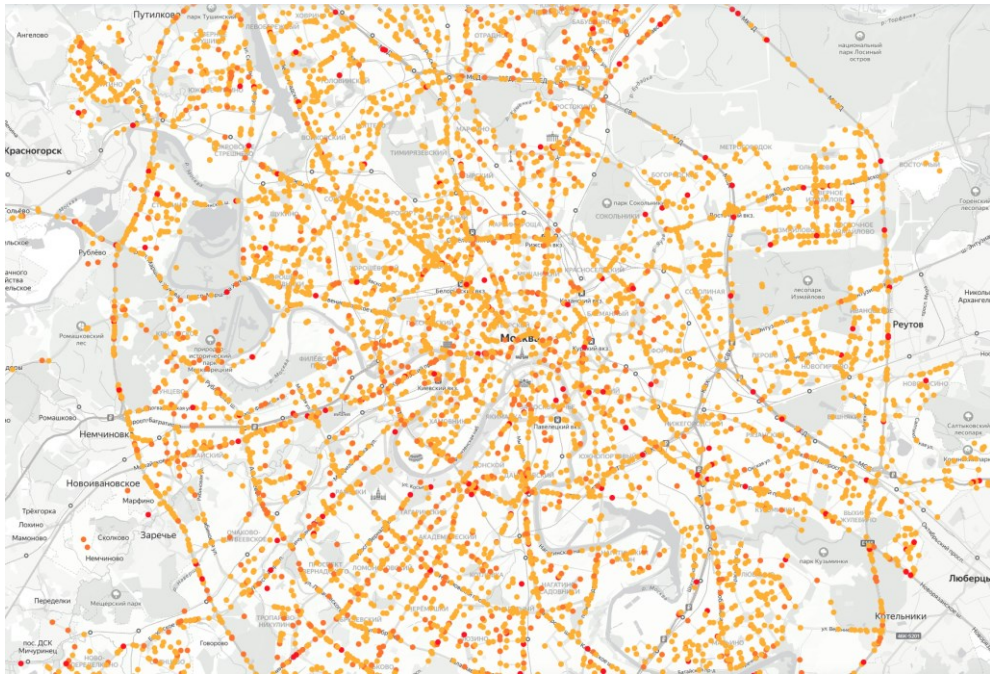
глубокого обучения на основе LSTM + WSA, включающая блок погодных условий и механизм внутреннего внимания, демонстрирует наилучшую точность прогнозирования при неблагоприятных погодных условиях. Модель использует пространственно-временные данные транспортного потока и погодные характеристики с учетом веса, рассчитанным с помощью механизма внутреннего внимания.

4.3 Вычислительные эксперименты с моделью безопасности маршрутов

В экспериментах использовались открытые данные со следующих ресурсов: статистические данные о ДТП и технических устройствах [24], некоммерческий проект «Карта ДТП» [21], онлайн-мониторинг радиоактивного излучения [25]. Данные о ДТП для Москвы изображены на рисунке 4.8 и содержат следующую информацию.

Для расчетов используются как открытые данные о ДТП, нарушениях, наличии дорожных знаков и ограничениях проезда, так и коммерческие данные о количестве участников дорожного движения, их местоположении и погодных условиях из следующих источников информации:

- 1) OSM [32];
- 2) статистика ГИБДД и данные о технических устройствах [24];
- 3) карта ДТП [21];
- 4) данные о текущих и прогнозируемых погодных условиях [27];
- 5) онлайн-мониторинг дозы радиоактивного загрязнения [25];
- 6) пользовательские данные;
- 7) другие данные коммерческих компаний.



```
{
  "id": 384094, # идентификатор
  "tags": ["Дорожно-транспортные происшествия"], # данные ГИБДД
  "point": {"lat": 50.6039, "long": 36.5578}, # координаты
  "region": "Белгород", # город/район
  "address": "г Белгород, ул Сумская, 30", # адрес
  "weather": ["Ясно"], # погода
  "category": "Столкновение", # тип ДТП
  "datetime": "2017-08-05 13:06:00", # дата и время
  "severity": "Легкий", # тяжесть ДТП/вред здоровью
  "vehicles": [ # участники - транспортные средства
    {
      "year": 2010, # год производства транспортного средства
      "brand": "ВАЗ", # марка транспортного средства
      "model": "Priora", # модель транспортного средства
      "category": "С", # категория ТС
      "participants": [ # участники внутри ТС
        {
          "role": "Водитель", # роль участника
          "gender": "Женский", # пол участника
          "violations": [], # нарушения правил участником
          "health_status": "Раненый, находящийся...", # травмы
          "years_of_driving_experience": 11 # стаж вождения
        }
      ]
    }
  ],
  "dead_count": 0, # кол-во погибших в ДТП
  "participants": [], # участники без транспортных средств
  "injured_count": 2, # кол-во раненых в ДТП
  "parent_region": "Белгородская область", # регион
  "road_conditions": ["Сухое"], # состояние дорожного покрытия
  "participants_count": 3, # кол-во участников ДТП
}
```

Источник: [21].

Рисунок 4.8 – Данные о ДТП для Москвы

Попарные сравнения показателей в матрицах проводились на основе 9-вариантной шкалы отдельно двумя экспертами в области безопасности транспортного движения. Объединив мнения экспертов, была получена комбинационная матрица парных сравнений, представленная в приложении И. Относительные веса показателей, получаются путем нормализации каждой комбинационной матрицы попарного сравнения и получения среднего арифметического нормализованной матрицы. Веса показателя и коэффициенты совместимости, полученные в результате попарного сравнения комбинационных матриц, представлены в таблице 4.8. Поскольку индекс несогласованности меньше 0,1, то проведенные сравнения являются приемлемыми, а веса обладают необходимой достоверностью.

Таблица 4.8 – Рассчитанный вес показателей безопасности

Приоритет	Показатель	Вес
1	Количество пешеходных зон	0,0513
2	Количество знаков безопасности	0,0098
3	Количество технических устройств	0,005
4	Количество парковочных зон	0,025
5	Количество нарушений	0,0156
6	Количество опасных участков	0,0073
7	Количество травм	0,1059
8	Количество аварий	0,2039
9	Количество дорожного освещения	0,0140
10	Количество общественного транспорта	0,0098
11	Количество велосипедных зон	0,0075
12	Количество смертей	0,3874
Сумма		1
λ_{max}		12,012
Индекс согласованности (CI)		0,0011
Отношение согласованности (CR)		0,0007

Источник: [приложение И].

4.4 Вычислительные эксперименты с численным методом маршрутизации транспортных средств

Для обеспечения эффективного использования транспортной сети и повышения безопасности движения важно не только предлагать маршруты, но

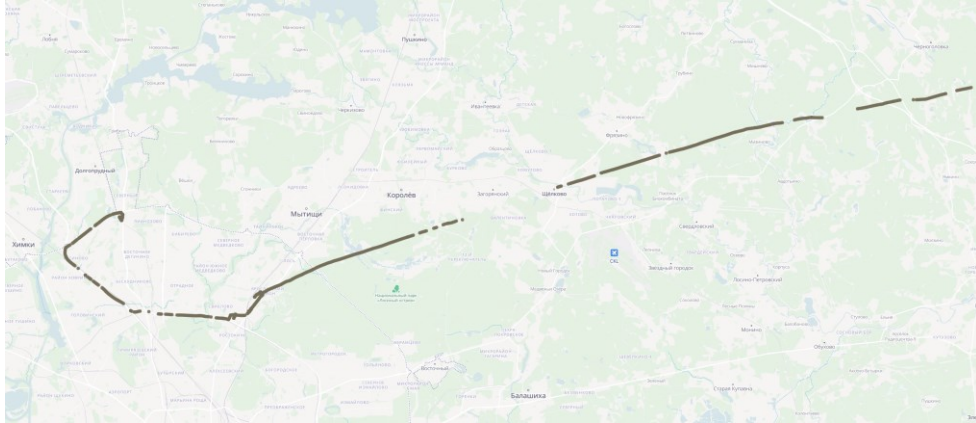
и анализировать поведение участников дорожного движения. Современные информационные сервисы способны обрабатывать большие объемы данных, чтобы выявлять закономерности в движении транспорта и индивидуальные особенности водителей. Такие аналитические подходы позволяют не только оптимизировать маршруты, но и создавать персонализированные рекомендации для пользователей. На основе пользовательских данных при использовании мобильного приложения возможно проанализировать маршрут пользователя и создать «портрет водителя». Для этого необходимо соблюдение следующих условий:

- 1) все точки трека должны принадлежать одному устройству в рамках одной поездки;
- 2) между соседними точками трека не должно быть больших пространственных или временных разрывов (не более 10 секунд между ближайшими точками);
- 3) точность GPS должна быть не менее 30 метров.

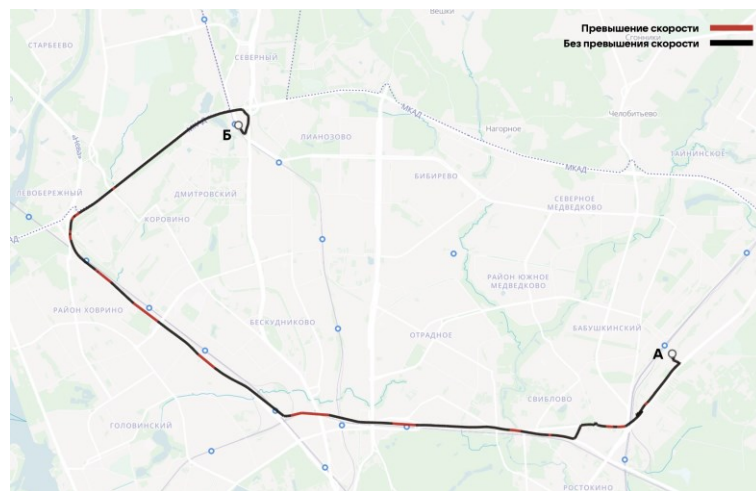
В результате по треку с координатами пользователя (набор GPS-координат с временными метками) возможно восстановить маршрут с привязкой к дорожной сети и проанализировать следующие данные:

- 1) общее время в пути;
- 2) средняя и максимальная скорости;
- 3) превышение скорости;
- 4) геометрия маршрута с привязкой к дорожному графу;
- 5) аномалии на маршруте (подмена координат, резкие скачки местоположения и тому подобное);
- 6) ребра дорожного графа с подробной информацией о прохождении.

Примеры результатов анализа треков представлены на рисунке 4.9.



а



б

а – аномалии пройденного маршрута; б – превышение скорости на маршруте

Источник: составлено автором.

Рисунок 4.9 – Примеры маршрутов с аномалиями

Пример входных данных:

```
{
  "locations": [ // массив объектов (трек) с привязкой ко
    времени
    {
      "lat": 60.09918,
      "lon": 30.2761,
      "ts_device": 1660550761
    },
    {
      "lat": 60.0991,
      "lon": 30.27493,
      "ts_device": 1660550763
    }
  ],
  "costing": "auto", // тип транспорта для построения маршрута
}
```

```

    "gps_accuracy": 10, // точность GPS в метрах
    "speeding_duration": 21, // длительность превышения скорости в
секундах
}

```

Пример данных после анализа:

```

{
  "status": "ok", // статус обработки запроса
  "geometry": [ // геометрия маршрута
    {
      "lat": 60.099245,
      "lon": 30.276089
    },
    {
      "lat": 60.099236,
      "lon": 30.275904
    },
    {
      "lat": 60.099166,
      "lon": 30.274918
    }
  ],
  "route_details": { // расширенная информация о маршруте
    "duration": 168,
    "duration_in_movement": 168,
    "length": 5848.4459,
    "speed_average": 125.323,
    "speed_in_movement": 125.323,
    "speed_max": 154.866
  },
  "speeding": [ // превышения скорости на маршруте
    {
      "geometry_end_index": 68,
      "geometry_begin_index": 66,
      "speed_fact": 242.34479,
      "speed_limit": 110,
      "speeding": 132.34479,
      "ts_end": 1660550838,
      "ts_start": 1660550837
    },
    {
      "geometry_end_index": 72,
      "geometry_begin_index": 70,
      "speed_fact": 134.56799,
      "speed_limit": 110,
      "speeding": 24.56799,
      "ts_end": 1660550843,
      "ts_start": 1660550841
    }
  ],
  "anomalies": [ // аномалии на маршруте
    {

```

```

    "jump_distance": 64.625,
    "jump_duration": 2,
    "lat": 55.650696,
    "lon": 37.700512,
    "ts": 1660572718
  },
  {
    "jump_distance": 44.56425,
    "jump_duration": 3,
    "lat": 55.650528,
    "lon": 37.699868,
    "ts": 1660572721
  },
  {
    "jump_distance": 113.533,
    "jump_duration": 15,
    "lat": 55.650429,
    "lon": 37.698069,
    "ts": 1660572736
  }
],
"matched_points": [ // массив GPS-данных на маршруте
  {
    "dist_from_edge_start": 0.804,
    "edge_index": 0,
    "geometry_index": 1,
    "lat": 55.671941,
    "lon": 37.724477,
    "ts": 1660572364
  },
  {
    "dist_from_edge_start": 0.207,
    "edge_index": 2,
    "speed_fact": 27.919,
    "geometry_index": 4,
    "lat": 55.671472,
    "lon": 37.72421,
    "ts": 1660572371
  }
],
"edges": [ // массив ребер дорожного графа маршрута
  {
    "edge_info": {
      "length": 0.015,
      "speed": 60
    }
  },
  {
    "edge_info": {
      "length": 0.014,
      "speed": 60
    }
  }
],

```

```

{
  "edge_info": {
    "length": 0.025,
    "speed": 70
  }
}
]
}

```

Подобные подходы позволяют создавать интеллектуальные системы маршрутизации, ориентированные на повышение безопасности и удобства для водителей. Анализ аномалий может быть использован для обнаружения проблемных зон, улучшения точности данных и защиты от подделки маршрутов. Персонализированные рекомендации повышают удобство использования дорожной сети и помогают улучшать навыки вождения.

Для тестирования эффективности численного метода маршрутизации транспортных средств использовались смартфоны: iPhone X с шестиядерным процессором и Samsung Galaxy S20 с восьмиядерным процессором. В качестве входных данных использовался протяженный маршрут между Москвой и Санкт-Петербургом. Полученные усредненные результаты после повторного запуска эксперимента в течение 10 дней отображены в таблице 4.9, время прокладывания маршрута в миллисекундах. Программная реализация численного метода представлена в приложении Г.

Таблица 4.9 – Результаты тестирования алгоритма маршрутизации транспортных средств на устройствах

Алгоритм	Samsung Galaxy S20	iPhone X
Алгоритм в один поток	2594 мс	3102 мс
Алгоритм в два потока	1534 мс	1895 мс

Источник: составлено автором.

Таким образом ускорение за счет использования многопоточного режима составило от 39 до 41%.

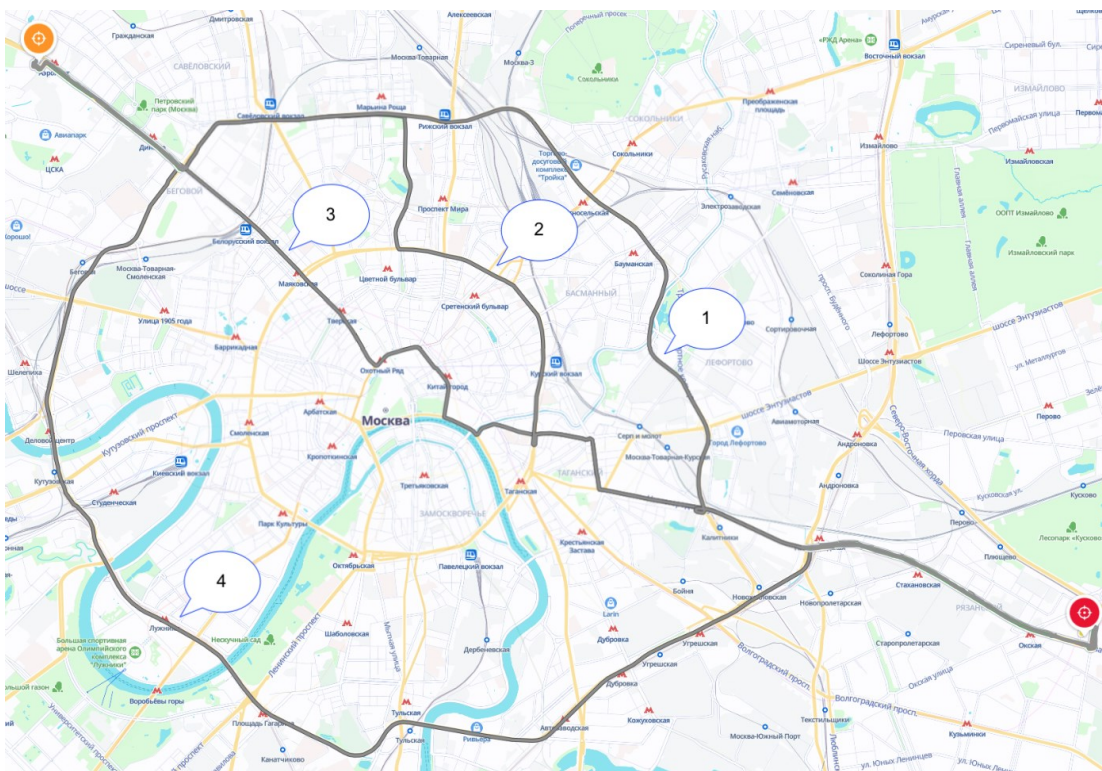
4.5 Вычислительные эксперименты с моделью многокритериальной оценки эффективности маршрута

В рамках экспериментов рассматривались альтернативные маршруты между корпусами Финансового Университета, расположенные по адресам:

- 1) Москва, Ленинградский проспект, д. 49;
- 2) Москва, 4-й Вешняковский проезд, д. 4/3.

Разработанный комплекс программ состоит из нескольких модулей для расчета каждой альтернативы по описанным выше моделям. Модуль поиска альтернатив содержит в себе расчет нескольких маршрутов, по которым можно добраться из начальной точки в конечную с использованием дорог общего пользования – автомагистралей и главные дороги имеют больший вес по сравнению со второстепенными. Для минимизации погрешности при расчете весов дорог алгоритм учитывает не только тип трассы, но и исторические данные о средней скорости движения, полученные в ходе предварительного анализа. Приоритизация автомагистралей в маршрутизации позволяет снизить вероятность непредвиденных задержек, связанных с особенностями локальных дорог, такими как светофоры или сезонные ограничения. При поиске альтернатив (маршрутов) используется алгоритм двухстороннего A^* , на вход которого подаются координаты начальной и конечной точек и данные графа дорог из OpenStreetMap, на выход – маршруты.

Маршруты между двумя точками, изображенные на рисунке 4.10, были проложены с оптимизацией по расстоянию. Оптимизация маршрута производится на основе данных ассессоров по пустому маршруту. Для валидации результатов модуль интегрирован с системой симуляции транспортных потоков, что обеспечивает проверку маршрутов в условиях, приближенных к реальным. Абсолютные значения были получены, когда количество участников движения минимально и они не оказывают влияния на поездку ассессора, например, в будние дни после 12 часов ночи.



Источник: составлено автором.

Рисунок 4.10 – Найденные маршруты между начальной и конечной точками

Данные по объективным и экспертным критериям и подкритериям безопасности используются за период с 01.01.2023 по 31.12.2023. Стоит отметить, что для более качественного анализа рекомендуется использовать данные за больший временной интервал, но без накопленных данных о средней скорости не имеется практического смысла. Анализируемые данные представлены в таблицах 4.10 и 4.11. Мнения экспертов были использованы для присвоения значений критериям и представлены в таблице 4.12.

Таблица 4.10 – Объективные критерии

Маршрут	Расстояние, м.	Время в пути, мин.	Загруженность маршрута	Погодные условия	Стоимость проезда, руб.	Безопасность
1	24200	29,00	0,64	0,66	0	0,962
2	23100	37,50	0,81	0,81	0	0,959
3	22500	39,00	0,85	0,85	0	0,941
4	29400	41,00	0,36	0,37	0	0,974

Источник: составлено автором на основе материалов [21; 27].

Таблица 4.11 – Значения подкритериев критерия безопасности

Маршрут	Максимальное радиоактивное загрязнение	Количество смертей	Количество травм	Количество аварий	Количество опасных участков	Количество велосипедных зон	Количество нарушений	Количество дорожного освещения	Количество пешеходных зон	Количество технических устройств	Количество общественного транспорта	Количество знаков безопасности	Количество парковочных зон
1	0,07	10	24	61	7	8	2947	490	16	34	49	45	9
2	0,08	12	23	73	16	16	4031	450	33	42	62	57	12
3	0,09	8	17	69	13	19	3394	462	29	45	59	68	14
4	0,08	13	21	58	5	4	5106	588	14	22	41	31	8
Сумма	0,36	49	85	261	41	47	15478	1990	92	143	211	201	43

Источник: составлено автором на основе материалов [21; 23-25; 27].

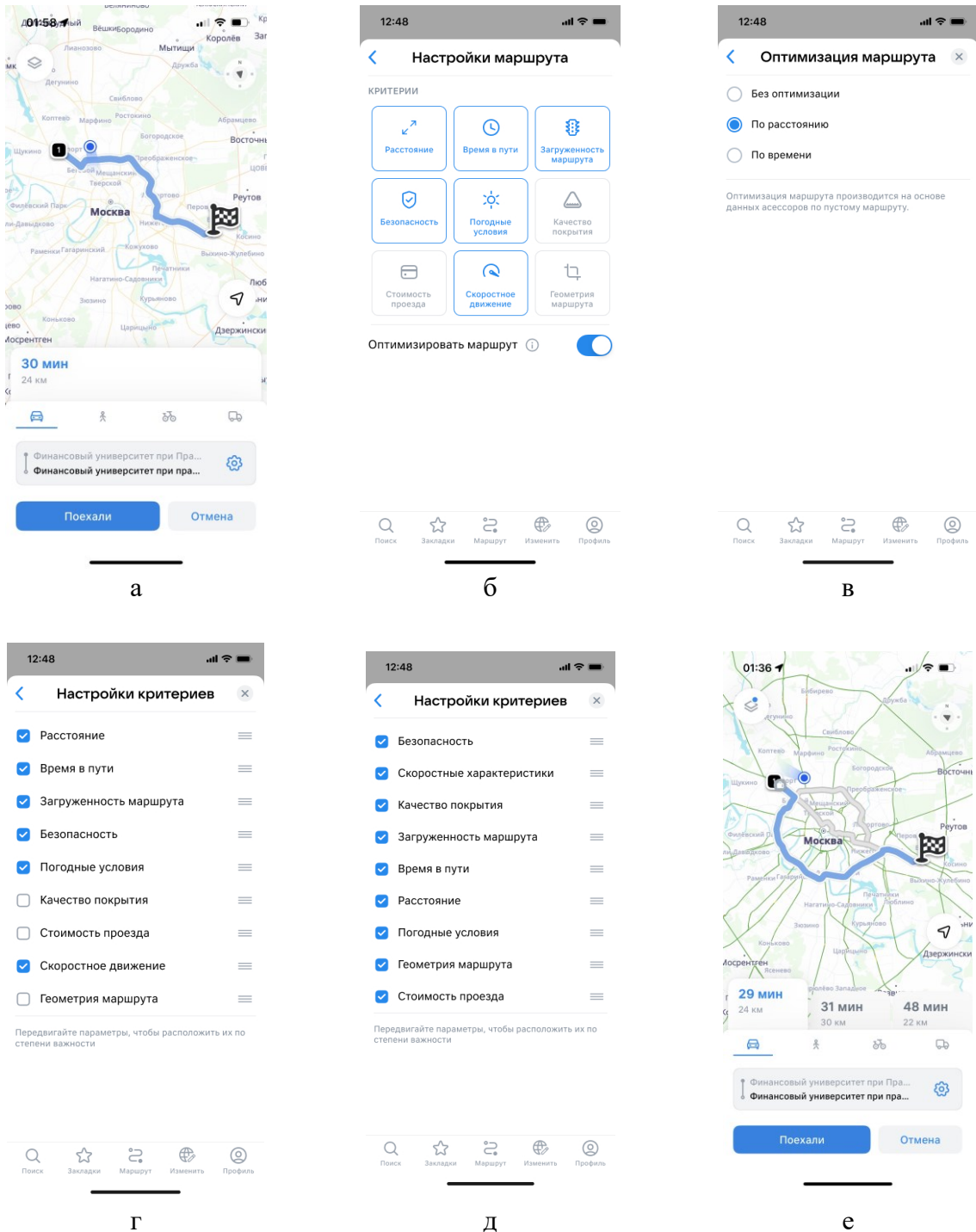
Таблица 4.12 – Экспертные критерии

Маршрут	Скоростные характеристики	Геометрические характеристики	Качество дорожного покрытия
1	6	7	8
2	5	8	7
3	5	9	8
4	7	8	9

Источник: составлено автором.

В таблице 4.13 продемонстрированы веса критериев, рассчитанные на основе субъективных предпочтений ЛПР, представленные в приложении И. Критерии безопасности и стоимости проезда являются наиболее и наименее

важными по сравнению с остальными. На рисунке 4.11 представлены интерфейсы мобильного приложения, с помощью которого можно настроить систему ценностей ЛПР.



- а) прокладывание маршрута между начальной и конечной точками;
 б) активные критерии; в) оптимизация маршрута по расстоянию;
 г) вес критериев; д) подключение всех критериев и изменение их веса;
 е) результат ранжирования альтернатив с учетом веса критериев.

Источник: составлено автором.

Рисунок 4.11 – Интерфейсы разработанного мобильного приложения

Таблица 4.13 – Вес критериев на основе субъективных предпочтений ЛПР

Приоритет	Критерий	Вес
1	Безопасность	0,3070
2	Скоростные характеристики	0,2182
3	Качество покрытия	0,1543
4	Загруженность маршрута	0,1090
5	Время в пути	0,0764
6	Расстояние	0,0533
7	Погодные условия	0,0370
8	Геометрия маршрута	0,0259
9	Стоимость проезда	0,0189
Сумма веса		1
Индекс несогласованности		0,75

Источник: составлено автором.

Модуль загруженности маршрута использует результаты работы модели загруженности дорожной сети, где входными параметрами модели являются анонимные GPS-данные в реальном времени и исторические данные, накапливаемые в процессе работы системы. Актуальные данные используются в моделях загруженности, оценки и краткосрочного прогнозирования. Расчет загруженности маршрута производится согласно параграфу 3.2. Уровень загруженности маршрута описан в формуле (2.7).

Модуль погодных условий представляет модель учета и прогнозирования негативных погодных условий, влияющих на загруженность маршрута, согласно параграфу 2.3. В рамках коротких городских поездок результаты модель почти всегда одинаковы и незначительно влияют на загруженность. Отличия возникают при расчете маршрутов между удаленными районами мегаполиса или на междугородних поездках, например, если на одном маршруте идет дождь, а на другом – нет.

Модуль безопасности рассчитывает коэффициент безопасности маршрута на основе открытых данных о ДТП (с 01.01.2023 по 31.12.2023). На каждый сегмент дороги проецируется авария, ДТП, летальный исход, знаки дорожного движения и технические устройства с известными координатами. Вес критериев безопасности рассчитывается на основе мнения экспертов в

области безопасности дорожного движения в параграфе 3.1. После чего результаты усредняются для всего маршрута.

Стоимость проезда рассчитывается на основе открытых данных о платных трассах и стоимости проезда по предлагаемому маршруту, но для большинства городских поездок равна 0. В Москве необходимо учитывать такие платные дороги, как центральная кольцевая автодорога (далее – ЦКАД), московский скоростной диаметр (далее – МСД) и западный скоростной диаметр (далее – ЗСД), а также иметь актуальные данные о стоимости проезда.

Оценить геометрические характеристики, качество дорожного покрытия и тип дороги вручную за приемлемое время невозможно. Для этого понадобилось бы десятки тысяч часов работы экспертов, постоянно оценивающих новые данные и актуальные события на дорогах, поэтому характеристики оцениваются на основе открытых данных OSM в автоматическом режиме с заданной периодичностью обновления, а затем выборочно верифицируются экспертами. При ремонтных работах или экстренных ситуациях пользователи могут дополнительно сообщить о поврежденном участке дороги. Информация проверяется участниками движения, после чего записывается в хранилище данных и обновляется у всех пользователей. Для объединения значений различных критериев относительный вес каждого умножался на значение важности, а затем суммировались друг с другом по формуле (3.17). После расчета нормализованной матрицы решений была получена взвешенная нормализованная матрица по формуле (3.18). На ее основе были определены идеальные положительные A^+ и отрицательные A^- решения по формуле (3.19). На следующем этапе было получено расстояние каждой альтернативы (маршрута) до положительного d_j^+ и отрицательного d_j^- идеальных решений по формуле (3.20), на основе которых относительная близость была достигнута по отношению к идеальному решению R_j по формуле (3.21). Результатом работы модели многокритериального анализа являются ранги альтернатив, представленные в таблицах 4.14 и 4.15.

Таблица 4.14 – Нормализованная матрица значений критериев предпочтительности для альтернативных маршрутов

Направление улучшения	Критерий								
	max	max	max	min	min	min	max	max	min
Маршрут	Безопасность	Скоростные характеристики	Качество покрытия	Загруженность маршрута	Время в пути	Расстояние	Погодные условия	Геометрия маршрута	Стоимость проезда
1	0,969	0,667	0,915	0,652	0,170	0,218	0,940	0,805	0
2	0,984	0,686	0,884	0,509	0,159	0,213	0,944	0,792	0
3	0,989	0,569	0,823	0,768	0,188	0,202	0,942	0,852	0
4	0,990	0,694	0,956	0,441	0,216	0,307	0,942	0,871	0

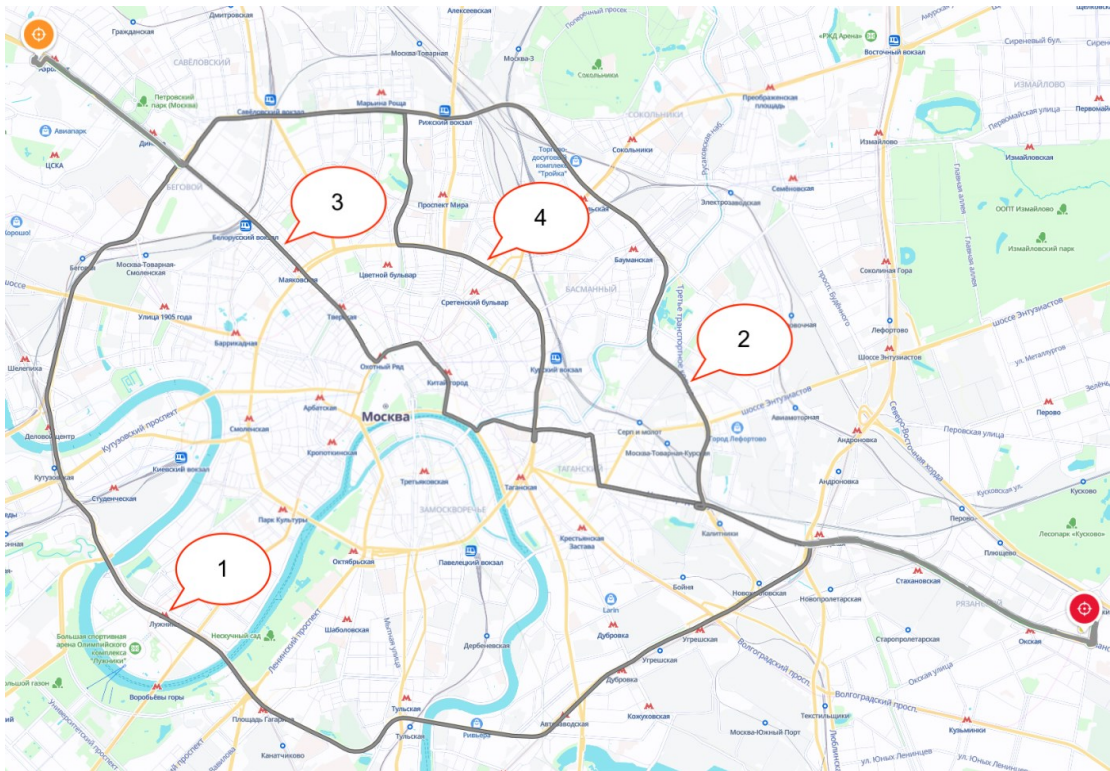
Источник: составлено автором.

Таблица 4.15 – Результат расчета ранга альтернатив

Номер маршрута	d_j^+	d_j^-	R_j	Вес	Относительный вес, в процентах
1	0,037	0,160	0,813	81,3	28,67
2	0,103	0,084	0,451	45,1	15,90
3	0,060	0,171	0,740	74,0	26,09
4	0,032	0,158	0,832	83,2	29,34

Источник: составлено автором.

Чем больше значение R_j , тем альтернатива считается лучше, относительно остальных. По результатам расчетов можно утверждать, что маршрут 4 является лучшим при заданном весе критериев с учетом мнения лица ЛПР, результаты изображены на рисунке 4.12. Окончательное решение по выбору маршрута остается при этом за ЛПР.



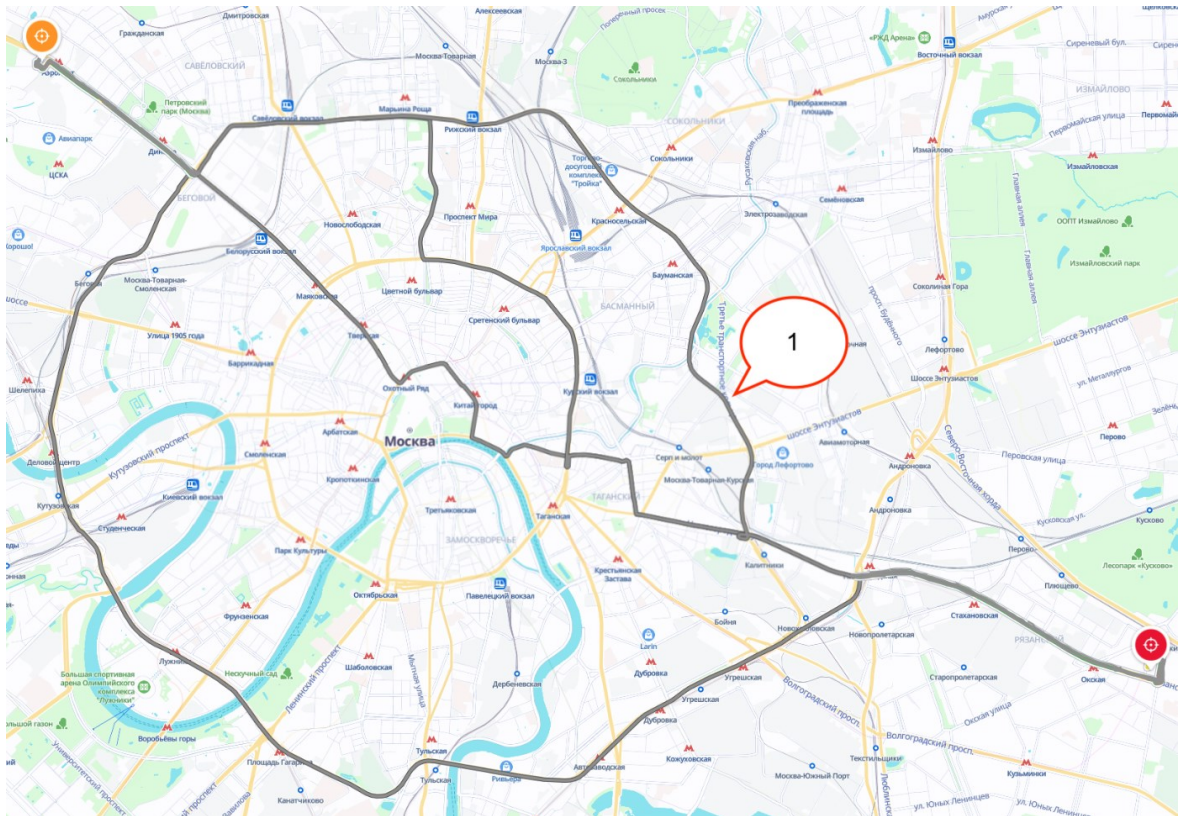
Источник: составлено автором.

Рисунок 4.12 – Результат модели поиска рациональных маршрутов

4.6 Вычислительные эксперименты с когнитивной моделью

В приложении К представлены рассчитанные значения элементов матрицы A для задачи поиска рационального маршрута из четырех альтернатив на основе фреймово-факторной модели. Коэффициенты были получены путем применения процедуры экспертных парных сравнений и расчета собственных значений. Для обозначения приняты следующие сокращения: Π_1 – наименьшее расстояние маршрута; Π_2 – наименьшее время в пути; Π_3 – наибольшие скоростные характеристики; Π_4 – наименее загруженный маршрут; Π_5 – минимальная стоимость проезда; Π_6 – наилучшая геометрия маршрута; Π_7 – наилучшее качество дорожного покрытия; Π_8 – наилучшие погодные условия на маршруте; Π_9 – наиболее безопасный маршрут; M_1 – маршрут 1; M_2 – маршрут 2; M_3 – маршрут 3; M_4 – маршрут 4; c_1 – расстояние маршрута; c_2 – время в пути; c_3 – скоростные характеристики; c_4 – загруженность маршрута; c_5 – стоимость проезда; c_6 – геометрия маршрута; c_7 – качество дорожного покрытия; c_8 – погодные условия;

c_9 – безопасность; V_1 – водитель 1. Результат расчета когнитивной модели представлен на рисунке 4.13.



Источник: составлено автором.

Рисунок 4.13 – Результат поиска рациональных маршрутов с использованием когнитивной модели

В результате применения процедуры расчета была получена матрица A^6 , которая с точностью до погрешности $\varepsilon = 1,081E^{-11}$ остается неизменной. Предельные значения коэффициентов влияния представлены в приложении К. Как видно из таблицы К.2 более целесообразно выбрать маршрут 1, так как он является оптимальным по Парето. Если бы Парето-оптимальное решение не было получено, то следует ввести интегральный фактор или использовать один из методов многокритериального анализа альтернатив.

4.7 Результаты сравнения комплекса программ со сторонними сервисами

Информационные сервисы транспортных сетей являются важной частью современной жизни, которые помогают оценить загруженность на

дорогах, построить маршрут, найти место и так далее. Пользуются такими сервисами не только обычные пользователи с помощью мобильного приложения, но и бизнес, например, при планировании маршрутов для доставки заказов, а также государство для оптимизации расходов, увеличения безопасности и эффективности транспортной сети и строительства новых маршрутов. Наиболее известные известными сервисами являются Яндекс.Карты, 2ГИС и Google Maps. Каждый сервис обладает преимуществами и недостатками, использует собственные модели, алгоритмы и базы данных и представляет схожий функционал. Для объективного сравнения программного обеспечения различных сервисов был проведен эксперимент на одинаковых устройствах в одно и то же время. В таблице 4.16 представлены результаты сравнительного анализа. Поиск рационального маршрута осуществлялся между двумя точками:

- 1) Москва, Ракетный бульвар, д. 13/2;
- 2) Москва, ул. Поварская, д. 22.

Таблица 4.16 – Сравнительный анализ сервисов

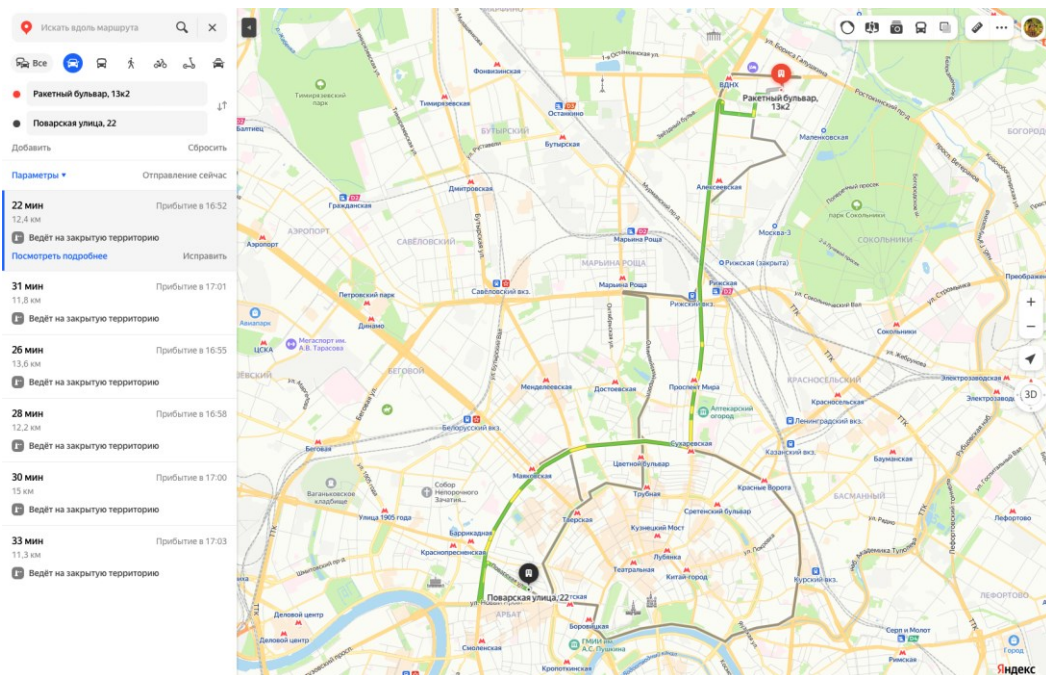
Параметр	Сервис				
	Яндекс.Карты	2ГИС	Google Maps	VK Карты	VK Карты
Модель поиска альтернатив	Нет данных	Нет данных	Нет данных	Модель поиска рационального маршрута	Когнитивная модель поиска рационального маршрута
Количество найденных маршрутов	6	3	3	4	
Минимальное расстояние, км	11,3	13	12,7	12,7	
Минимальное время в пути, мин	22	24	22	28	
Предложенный маршрут	12,4 км 22 мин	13 км 24 мин	12,7 км 22 мин	15,9 км 30 мин	12,7 км 28 мин
Загруженность маршрута, в процентах	14	11	17	18	23

Источник: составлено автором.

По полученным результатам можно сделать следующие выводы. Результаты сервисов отличаются друг от друга и сравнить их напрямую невозможно из-за различий анализируемых данных и отсутствия в открытом доступе применяемых алгоритмов. Тем не менее, возможно провести сравнение по косвенным признакам. Так, при поиске маршрутов между точками – Москва, Ракетный бульвар, д. 13/2 и ул. Поварская, д. 22 сервисы нашли следующее количество маршрутов:

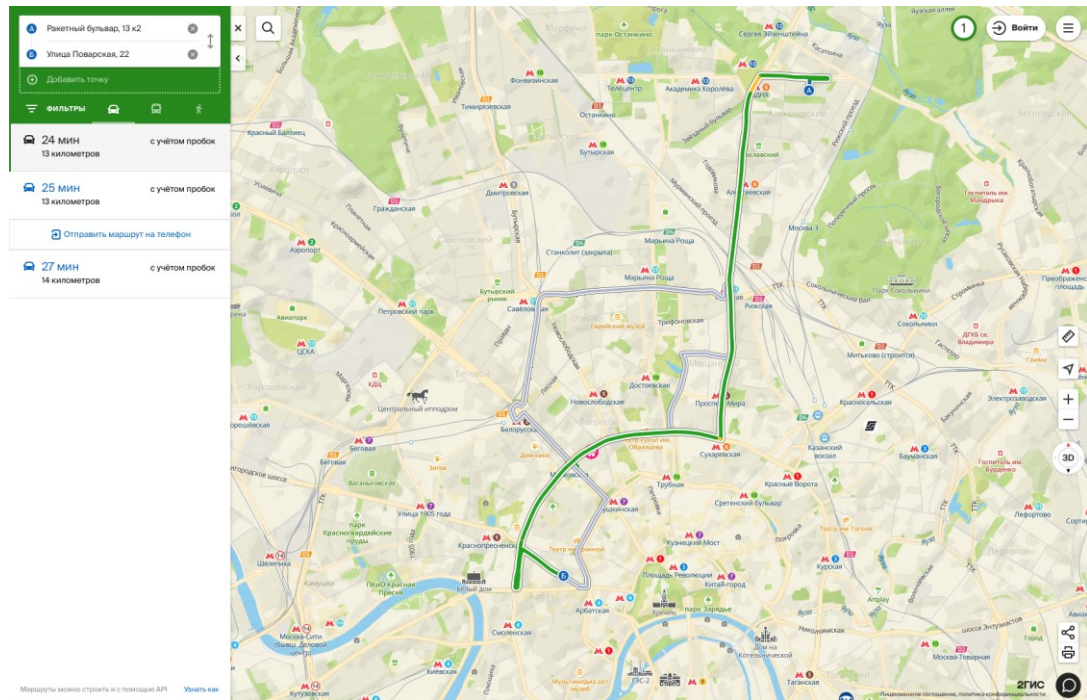
- 1) Яндекс.Карты – 6 маршрутов, рисунок 4.14;
- 2) VK Карты – 4 маршрута, рисунок 4.15;
- 3) 2ГИС – 3 маршрута, рисунок 4.16;
- 4) Google Maps – 3 маршрута, рисунок 4.17.

Сервисы рекомендуют наиболее быстрый маршрут (оптимизация по времени), который не всегда является кратчайшим. Разработанная когнитивная модель также рекомендует наиболее быстрый маршрут, что говорит о сходстве алгоритма поиска альтернатив с остальными сервисами. Модель поиска рационального маршрута ранжирует альтернативы на основе субъективного мнения ЛПР (приложение И) и рекомендует более безопасный маршрут, но с большим расстоянием и временем в пути.

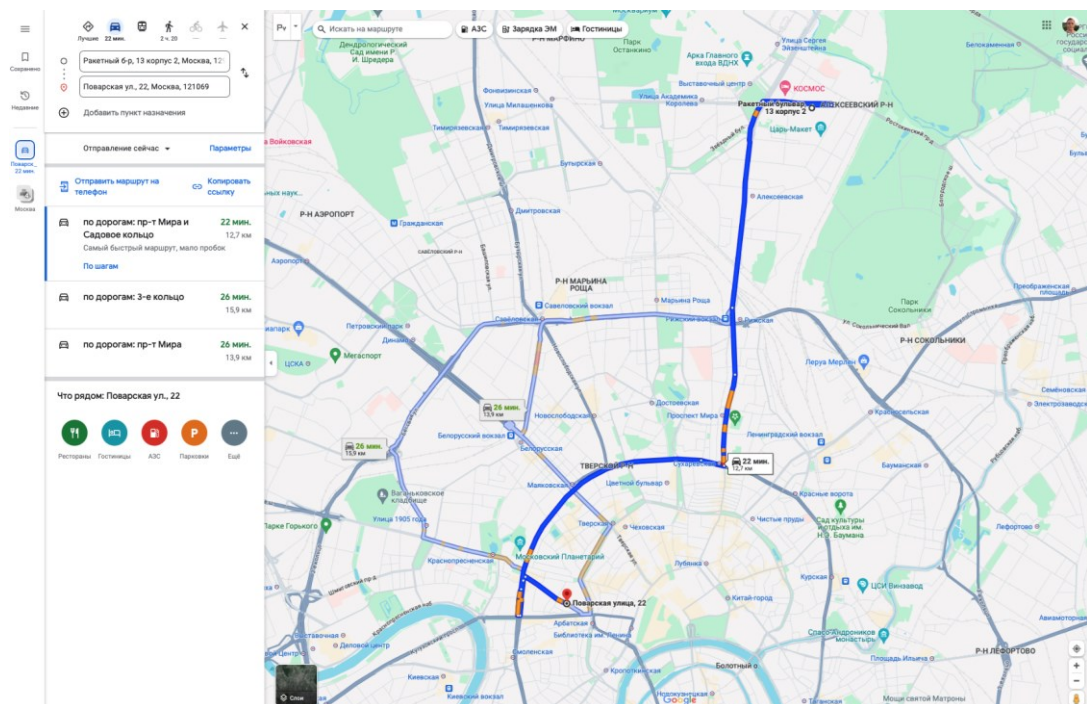


Источник: составлено автором.

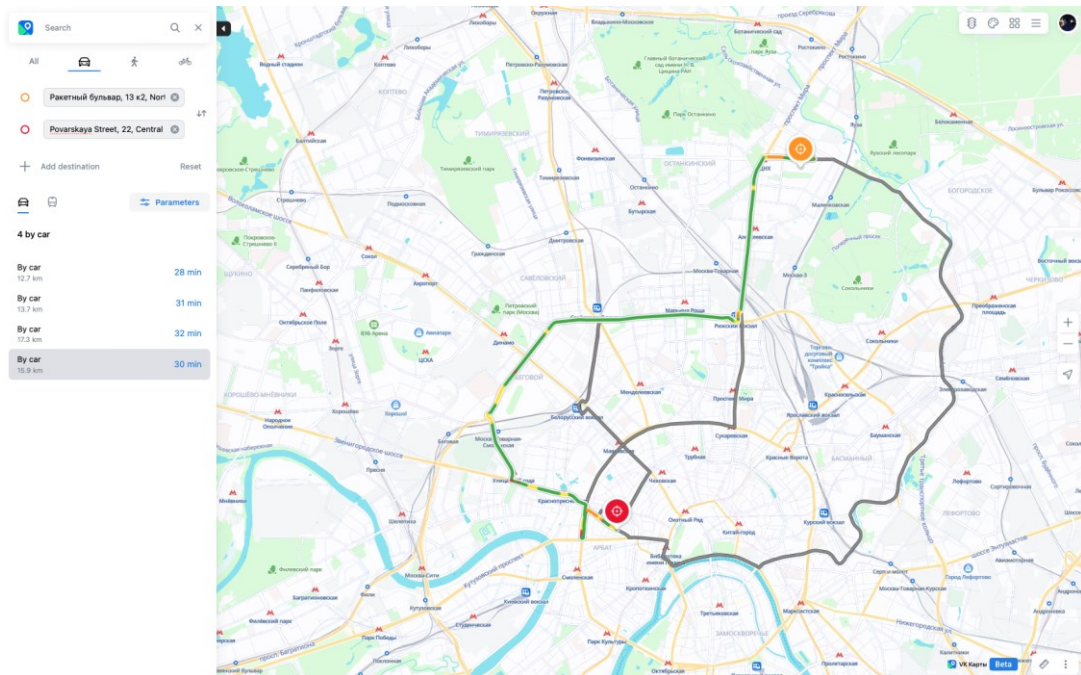
Рисунок 4.14 – Яндекс.Карты



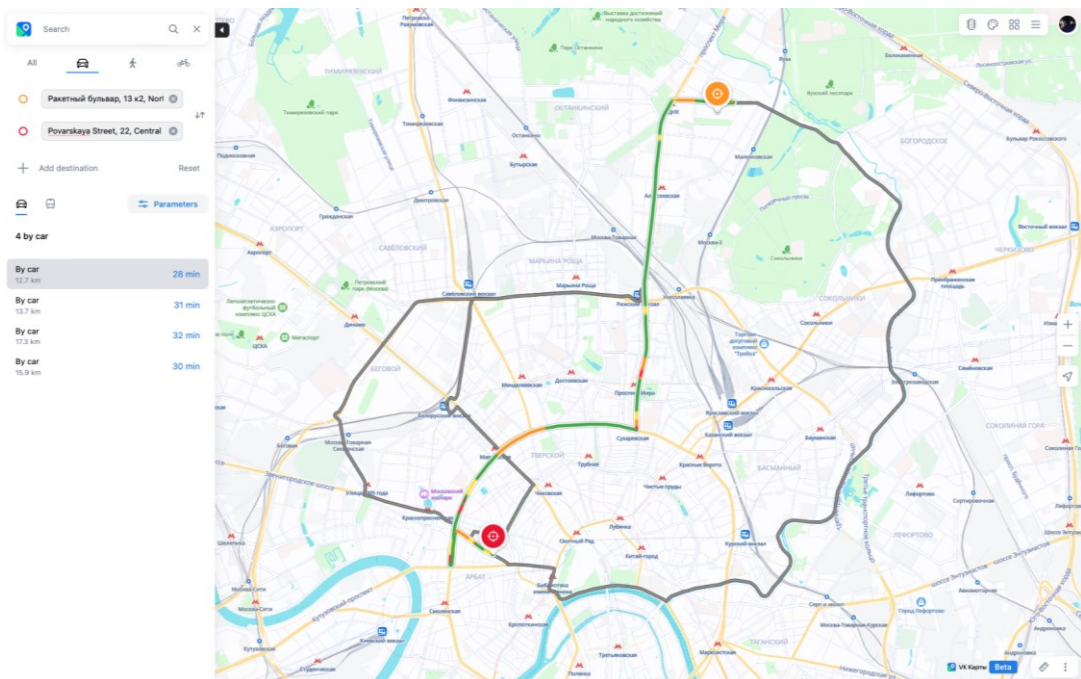
Источник: составлено автором.
Рисунок 4.15 – 2ГИС



Источник: составлено автором.
Рисунок 4.16 – Google Maps

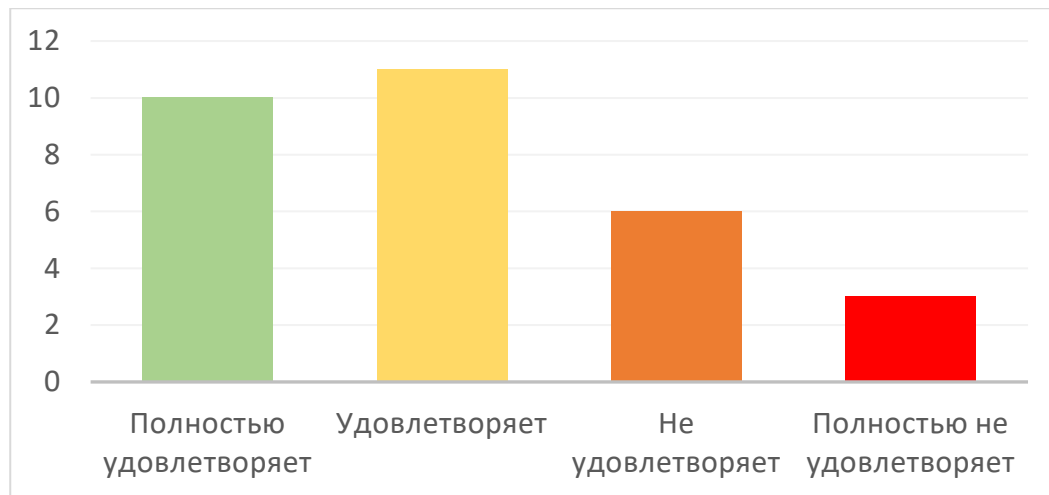


Источник: составлено автором.
Рисунок 4.17 – VK Карты



Источник: составлено автором.
Рисунок 4.18 – VK Карты (когнитивная модель)

Проведено закрытое тестирование комплекса программ, показавшее высокую удовлетворенность потенциальных пользователей. Результат представлен на рисунке 4.19.



Источник: составлено автором.

Рисунок 4.19 – Результаты опроса закрытого тестирования

По результатам тестирования проведен опрос – насколько предлагаемый маршрут удовлетворяет заданным условиям. В опросе участвовали 30 респондентов, более 70% респондентов ответили положительно. Ответы распределились следующим образом: полностью удовлетворяет – 10 респондентов, удовлетворяет – 11, не удовлетворяет – 6, полностью не удовлетворяет – 3.

Таким образом, разработанные модели поиска рационального маршрута на основе моделей многокритериального анализа, машинного обучения и когнитивной фреймово-факторной модели могут улучшить результаты поиска рациональных маршрутов.

Выводы по четвертой главе

В четвертой главе приведено описание комплекса программ и результаты вычислительных экспериментов с созданными моделями и численным методом. Полученные результаты сравнивались с современными сервисами поиска и прокладывания оптимальных маршрутов – Яндекс.Карты, 2ГИС и Google Maps. Сравнительный анализ показал, что при минимальном учете набора критериев (расстояние, время в пути и загруженность маршрута) разработанный сервис не уступает аналогам в результатах поиска и расчета маршрутов, прогнозируемой загруженности и времени движения

по маршруту, а при учете всех критериев – расширяет возможности и результаты существующих сервисов.

Представлена архитектура комплекса программ в форме веб-сервиса. Работоспособность и высокая доступность достигается путем горизонтального масштабирования с использованием кластера серверов, гарантирующих $SLA = 99,99\%$.

По результатам проведенного закрытого тестирования комплекса программ более 70% респондентов положительно оценили его возможности.

Заключение

Итогом диссертационной работы является разработка и апробация комплекса программ поиска рациональных маршрутов в транспортной сети в режиме, приближенному к реальному времени, на основе предпочтений ЛПР.

Разработан комплекс моделей сетевых структур дорожного движения улично-дорожной сети с участниками движения, включающий: а) модель многокритериальной оценки эффективности маршрута, позволяющую рассчитать комплексный показатель эффективности маршрута, в отличие от других моделей ориентированных на принятие решений на основании одного критерия загруженности ребер графа. Так же отличительной особенностью разработанной модели является то, что используются предпочтения, лица принимающего решения (ЛПР), что позволяет построить удовлетворяющий его маршрут, тогда как наиболее распространённые модели для построения маршрутов от компании Яндекс не учитывают предпочтений относительно критериев их оценки; б) модель загруженности сетевой структуры с учетом погодных условий повысившую точность прогнозирования загруженности маршрута на 5-7% в сравнении с классической LSTM за счет выбора архитектуры и гиперпараметров искусственных нейронных сетей LSTM + WSA; в) модель безопасности маршрута, отличающаяся тем что показатель безопасности определяется по разнородным показателям, которые агрегируются с использованием методов теории принятия решений, для расчета используются открытые данные о состоянии дорожного покрытия, средствах регулирования движения (светофоры, переходные переходы), средствах фиксации нарушений (видеокамеры автоматической фиксации нарушений, радары), дорожно-транспортных происшествиях, радиоактивном загрязнении; что позволяет построить маршрут, удовлетворяющий лицу, принимающее решение. В данной модели используются экспертные оценки в форме парных сравнений приоритетов показателей безопасности, что отличается от подхода экспертного сравнения конкретных альтернатив

(альтернативных маршрутов), такое отличие позволяет произвести настройку модели один раз и использовать ее для оценки безопасности произвольного числа маршрутов; г) когнитивная модель расчета показателей оценки маршрута на основе фреймово-факторного графа экспертных оценок, позволяющая приближенно оценить сетевую структуру в случае недостаточности эмпирических данных, что отличает ее от моделей загруженности использующих нейросети и требующих большого объема данных для машинного обучения.

Разработан численный метод поиска альтернативных маршрутов, ускоряющий поиск маршрута на 30% по сравнению с распространённым алгоритмом A^* , что достигается за счет буферизации данных о структуре улично-дорожной сети для двух множеств вершин: первого множества вершин (M_1), просматриваемых на текущей итерации при движении от начальной вершины к конечной, второго множества вершин (M_2), просматриваемых на текущей итерации при движении от конечной вершины к начальной; минимизировано количество обращений к графу на диске; введен усовершенствованный критерий останова алгоритма для данного численного метода: алгоритм продолжает перебирать вершины из множеств M_1 и M_2 после первого их пересечения, что повышает вероятность нахождения оптимального маршрута.

Разработан комплекс программ для оценки и выбора маршрута, использующий большие данные от навигационных приемников, отличающийся от наиболее распространённых навигаторов компаний Яндекс, 2GIS, Google Maps тем, что маршрутизация происходит с учетом предпочтений лица принимающего решения по показателям загруженности и безопасности маршрута. Опрос по результатам тестирования показал, что результаты работы комплекса программ признали приемлемыми более 70% респондентов. Представленная архитектура обеспечивает высокую доступность и отказоустойчивость сервиса на уровне $SLA = 99,99\%$.

Разработанные модели поведения и прогнозирования дополнили существующие исследования и улучшили результаты аналогичных веб-сервисов. По результатам закрытого тестирования более 70% респондентов удовлетворены построенным маршрутом.

Важно отметить, что разработанный инструмент применим к широкому классу сетевых структур, благодаря универсальности подхода. Кроме того, многокритериальная оценка может использоваться не только для оценки существующих маршрутов, но и для маршрутов, которые могут появиться в результате инфраструктурных преобразований, таким образом созданные модели, методы и комплекс программ в перспективе могут быть применены к задачам стратегического транспортного планирования.

Список сокращений и условных обозначений

AHP – Метод анализа иерархий;
ARIMA – Модель авторегрессии скользящего среднего;
ANN – Искусственная нейронная сеть;
CNN – Сверточная нейронная сеть;
DNN – Глубокая нейронная сеть;
GAHP – Метод анализа групповых иерархий;
GCN – Графовая нейронная сеть;
GRU – Управляемый рекуррентный блок;
KNN – Метод k-ближайших соседей;
LSTM – Долгая краткосрочная память;
MAE – Средняя абсолютная ошибка;
MAPE – Средняя абсолютная ошибка в процентах;
MSE – Среднеквадратическая ошибка;
NN – Нейронная сеть;
OSM – OpenStreetMap;
RMSE – Среднеквадратичное отклонение;
RVR – Векторная регрессия релевантности;
SVM – Метод опорных векторов;
ЛПР – Лицо, принимающее решение;
УДС – Улично-дорожная сеть.

Словарь терминов

идеальный маршрут: это маршрут, у которого по всем критериям оценки маршрута указаны наилучшие значения.

критерии оценки качества маршрута: в данной работе используются следующие критерии: расстояние, км; время в пути, мин; стоимость проезда, руб; загруженность маршрута; погодные условия; безопасность; скоростные характеристики; геометрия маршрута; качество дорожного покрытия. Предметом защиты не является конкретный список критериев или показателей. В работе создана методика, которая позволяет работать с такими критериями или показателями, значения которых можно вычислить, для которых можно задать идеальные значения и которые представляют ценность для лица, принимающего решения.

эффективность маршрутизации: это комплексный показатель, который показывает, насколько найденный методикой маршрут близок к идеальному маршруту. Чем ближе, тем выше эффективность. Близость к идеальному маршруту вычисляется с помощью евклидовой метрики в пространстве критериев и показателей оценки качества маршрута формулы (3.20)-(3.21) диссертации.

Список литературы

Книги

1. Арлазаров, В.Л. Информационно-аналитические аспекты в задачах управления : труды института системного анализа Российской академии наук (ИСА РАН) / В.Л. Арлазаров, Н.Е. Емельянов. – Москва : Издательство ЛКИ, 2005. – 384 с. – ISBN 978-5382004860.
2. Ахмадинуров, М.М. Математические модели управления транспортными потоками : монография / М.М. Ахмадинуров, Д.С. Завалищин, Г.А. Тимофеева ; под редакцией А.В. Гасникова. – Екатеринбург : Издательство УрГУПМ, 2011. – 120 с. – 500 экз. – ISBN 978-5-94614-190-1.
3. Бомас, В.В. Поддержка принятия многокритериальных решений по предпочтениям пользователя / В.В. Бомас, В.А. Судаков, К.А. Афонин ; под общей редакцией В.В. Бомаса. – Москва : Издательство МАИ, 2006. – 169 с. – ISBN 5-7035-1705-2.
4. Вилкас, Э.Й. Оптимальность в играх и решениях / Э.Й. Вилкас. – Москва : Издательство Наука, 1990. – 256 с. – ISBN 5-02-014255-7.
5. Вилкас, Э.Й. Решения. Теория, информация, моделирование / Э.Й. Вилкас, Е.З. Майминас. – Москва : Издательство Радио и связь, 1981. – 328 с. – ISBN 30501-130.
6. Гасников, А.В. Введение в математическое моделирование транспортных потоков : учебное пособие / А.В. Гасников, С.Л. Кленов, Е.А. Нурминский, [и др.] ; под редакцией А.В. Гасникова. – Москва : Издательство МЦНМО, 2013. – 429 с. – ISBN 978-5-4439-0040-7.
7. Кастельс, М. Информационная эпоха: экономика, общество и культура / М. Кастельс, О.И. Шкаратан. – Москва : Издательство ГУ ВШЭ, 2000. – 608 с. – ISBN 5-7598- 0069-8.
8. Ларичев О.И. Теория и методы принятия решений, а также хроника событий в волшебных странах. – 3-е издание / О.И. Ларичев. – Москва : Издательство Логос, 2006. – 392 с. – ISBN 5-88439-046-7.

9. Олескин, А.В. Сетевые структуры в биосистемах и человеческом обществе: Децентрализованные распределенные структуры в природе как образцы для самоорганизации и построения эффективного и гуманного общества в информационную эпоху / А.В. Олескин. – Москва : Издательство URSS, 2021. – 304 с. – ISBN 978-5-9710-7436-6.

10. Саати, Т. Принятие решений. Метод анализа иерархий / Т. Саати, Р.Г. Вачнадзе. – Москва : Издательство Радио и связь, 1993. – 320 с. – ISBN 5-256-00443-3.

Диссертации

11. Бабичева, Т.С. Методы математического и имитационного моделирования процессов локального взаимодействия в транспортных системах : специальность 05.13.18 «Математическое моделирование, численные методы и комплексы программ» : диссертация на соискание ученой степени кандидата физико-математических наук / Бабичева Татьяна Сергеевна ; Московский физико-технический институт (Государственный университет). – Долгопрудный, 2015. – 163 с. – Текст : электронный. – URL: https://keldysh.ru/council/3/D00202403/babicheva_diss.pdf (дата обращения: 13.05.2024).

12. Гасников, А.В. Эффективные численные методы поиска равновесий в больших транспортных сетях : специальность 05.13.18 «Математическое моделирование, численные методы и комплексы программ» : диссертация на соискание ученой степени доктора физико-математических наук / Гасников Александр Владимирович ; Московский физико-технический институт (Государственный университет). – Долгопрудный, 2016. – 487 с. – Библиогр.: с. 460-487. – Текст : электронный. – URL: https://old.mipt.ru/upload/iblock/a96/dissertatsiya_gasnikov-a.v..pdf (дата обращения: 15.05.2024).

13. Чернышев, С.В. Модели, методы и алгоритмы эффективного решения задачи маршрутизации транспорта на графах больших размерностей

: специальность 05.13.18 «Математическое моделирование, численные методы и комплексы программ» : диссертация на соискание ученой степени кандидата физико-математических / Чернышов Сергей Владленович ; НИУ «ВШЭ», 2011. – 115 с. – Библиогр.: с. 113-115. – Текст : электронный. – URL: <https://www.dissercat.com/content/modeli-metody-i-algoritmy-effektivnogo-resheniya-zadachi-marshrutizatsii-transporta-na-grafa> (дата обращения: 05.05.2024).

14. Чечина, А.А. Математическое моделирование транспортных потоков на основе теории клеточных автоматов : специальность 05.13.18 «Математическое моделирование, численные методы и комплексы программ» : диссертация на соискание ученой степени кандидата физико-математических наук / Чечина Антонина Александровна ; – 2021. – 136 с. – Библиогр.: с. 130-136. – Текст : электронный. – URL: <https://www.dissercat.com/content/matematiceskoe-modelirovanie-transportnykh-potokov-na-osnove-teorii-kletochnykh-avtomatov> (дата обращения: 01.05.2024).

Авторефераты диссертаций

15. Дорогуш, Е.Г. Математический анализ модели транспортных потоков на автостраде и управления ее состоянием : специальность 01.01.02 «Дифференциальные уравнения, динамические системы и оптимальное управление» : автореферат диссертации на соискание ученой степени кандидата физико-математических наук / Дорогуш Елена Геннадьевна ; Московский государственный университет им. М.В. Ломоносова. – Москва, 2014. – 26 с. – Место защиты : Московский государственный университет им. М.В. Ломоносова.

Электронные ресурсы

16. Алгоритм Витерби : сайт. – URL: https://ru.wikipedia.org/wiki/Алгоритм_Витерби (дата обращения: 13.03.2024). – Текст : электронный.

17. Алгоритмы поиска пути: Алгоритм дейкстры и A^* : сайт. – URL: <https://habr.com/ru/companies/otus/articles/748470> (дата обращения: 09.05.2024). – Текст : электронный.
18. Реализация двустороннего A^* на двух потоках : сайт. – URL: <https://habr.com/ru/companies/vk/articles/731656/> (дата обращения: 16.02.2024). – Текст : электронный.
19. Годовой отчет о выполнении Государственной программы города Москвы «Развитие транспортной системы» за 2022 год / Информационный портал Департамента транспорта и развития дорожно-транспортной инфраструктуры города Москвы : сайт. – Текст : электронный. – URL: <https://www.mos.ru/dt/documents/dokumenty/view/287857220/> (дата обращения: 15.04.2024).
20. Инкрементальный алгоритм привязки GPS-трека к дорожному графу : сайт. – URL: <https://habr.com/ru/companies/vk/articles/157883/> (дата обращения: 11.04.2024). – Текст : электронный.
21. Карта ДТП. Некоммерческий проект со статистическими данными дорожно-транспортных происшествий в России : сайт. – URL: <https://dtp-stat.ru> (дата обращения: 12.05.2024). – Текст : электронный.
22. Картографический онлайн-сервис VK Карты : сайт. – URL: <https://maps.vk.com> (дата обращения: 13.04.2024). – Текст : электронный.
23. Обследование скоростных параметров транспортных потоков на улицах города Москва : сайт. – URL: <https://itetps.hse.ru/news/712278226.html> (дата обращения: 13.05.2024). – Текст : электронный.
24. Статистические данные о ДТП и технических средствах. Официальный сайт ГИБДД России. – Текст : электронный. – URL: <http://stat.gibdd.ru> (дата обращения: 14.05.2024).
25. ФГУП «РАДОН». Онлайн-мониторинг дозы гамма-излучения : сайт. – Текст : электронный. – URL: <https://radon.ru/online-map/> (дата обращения: 10.05.2024).

26. Яндекс Карты, 2GIS, Google Maps – что эффективнее привлекает клиентов : сайт. – URL: <https://revvy.ai/blog/tpost/vu752d4ez1-yandeks-karti-2gis-google-maps-что-эффективнее-привлекает-клиентов> (дата обращения: 10.06.2024). – Текст : электронный.

27. Яндекс.Погода. Данные о текущих и прогнозируемых погодных условиях. Исторические погодные данные : сайт. – Текст : электронный. – URL: <https://yandex.ru/dev/weather/> (дата обращения: 10.03.2024).

28. Яндекс. Маршруты : сайт. – Текст : электронный. – URL: <https://yandex.ru/company/technologies/routes/> (дата обращения: 05.05.2024).

29. Apache Kafka Documentation : сайт. – Текст : электронный. – URL: <https://kafka.apache.org/documentation> (дата обращения: 22.03.2024).

30. Docker Documentation : сайт. – Текст : электронный. – URL: <https://docs.docker.com> (дата обращения: 22.03.2024).

31. Nginx Documentation : сайт. – Текст : электронный. – URL: <https://nginx.org/en/docs> (дата обращения: 22.03.2024).

32. OpenStreetMap. Открытый некоммерческий проект геоданных : сайт. – Текст : электронный. – URL: <https://www.openstreetmap.org> (дата обращения: 14.04.2024).

33. PostgreSQL Documentation : сайт. – Текст : электронный. – URL: <https://www.postgresql.org/docs> (дата обращения: 23.03.2024).

34. Redis Documentation : сайт. – Текст : электронный. – URL: <https://redis.io/docs/latest> (дата обращения: 22.12.2024).

35. Round-robin (алгоритм) : сайт. – URL: [https://ru.wikipedia.org/wiki/Round-robin_\(алгоритм\)](https://ru.wikipedia.org/wiki/Round-robin_(алгоритм)) (дата обращения: 28.03.2024). – Текст : электронный.

36. Federal Highway : сайт. – URL: <https://highways.dot.gov/public-roads/janfeb-1999/traffic-flow-theory> (дата обращения: 29.05.2024). – Текст : электронный.

37. Агафонов, А.А. Оценка и прогнозирование параметров транспортных потоков с использованием композиции методов машинного обучения и моделей прогнозирования временных рядов / А.А. Агафонов, В.В. Мясников // Компьютерная оптика. – 2014. – № 3. Том 38. – С. 539-549. – ISSN отсутствует.

38. Айзерман, М.А. Динамический подход к анализу структур, описываемых графами (основы графодинамики) / М.А. Айзерман, Л.А. Гусев, С.В. Петров, И.Н. Смирнова // Автоматика и телемеханика. – 1977. – № 7. – С. 135-151. – ISSN отсутствует.

39. Антонов, В.Г. Методические подходы к формированию адаптивных структур управления / В.Г. Антонов, И.А. Румянцева, Т.Ю. Кротенко, О.Г. Казеев // Вестник ГУУ. – 2019. – № 9. – С. 15-22. – ISSN отсутствует.

40. Арифджанова, Н.З. Применение искусственного интеллекта для оптимизации маршрутов транспорта / Н.З. Арифджанова // Universum: технические науки. – 2023. – № 5. – С. 45-62. – ISSN отсутствует.

41. Болодурина, И.П. Интеллектуальная модель прогнозирования интенсивности движения транспортных средств на перекрёстке / И.П. Болодурина, Л.М. Анциферова, Л.С. Гришина // Интеллект. Инновации. Инвестиции. – 2022. – № 6. – С. 1-10. – ISSN отсутствует.

42. Данович, Л.М. Прогнозирование исходных данных в динамическом режиме для модели распределения транспортных потоков по сети / Л.М. Данович, Н.А. Наумова // Фундаментальные исследования. – 2016. – № 9. – С. 238-242. – ISSN отсутствует.

43. Джаммул, С.М. Модель идентификации сетевого трафика на основе скрытой Марковской модели / С.М. Джаммул, А.М. Андреев, А.В. Бабиченко // Инженерная физика. – 2018. – № 6. – С. 38-47. – ISSN отсутствует.

44. Ехлаков Р.С. Модель оценки и прогнозирования загруженности транспортной сети в режиме, приближенном к реальному времени / Р.С. Ехлаков // Нейрокомпьютеры: разработка, применение. – 2024. – № 6. Том 26. – С. 126-134. – ISSN 1999-8554.
45. Ехлаков, Р.С. Модель влияния неблагоприятных погодных условий на загруженность транспортной сети / Р.С. Ехлаков, В.А. Судаков // Нелинейный мир. – 2024. – № 4. Том 22. – С. 122-128. – ISSN 2070-0970.
46. Ехлаков, Р.С. Модель многокритериальной оценки маршрутов транспортной сети / Р.С. Ехлаков, А.О. Жуков, В.А. Судаков // Наукоемкие технологии. – 2024. – № 1. – С. 1-12. – ISSN 1999-8465.
47. Ехлаков, Р.С. Многокритериальная оценка безопасности транспортной сети / Р.С. Ехлаков, Т.В. Сивакова, В.А. Судаков // Препринты ИПМ им. М.В. Келдыша РАН. – 2024. – № 38. – С. 1-16. – ISSN 2071-2901.
48. Ехлаков, Р.С. Современные методы моделирования транспортных потоков. Модель многокритериальной оценки рациональности маршрута / Р.С. Ехлаков // Научно-технический вестник Поволжья. – 2023. – № 11. – С. 70-73. – ISSN 2079-5920.
49. Ехлаков, Р.С. Прогнозирование стоимости котировок при помощи LSTM и GRU сетей / Р.С. Ехлаков, В.А. Судаков // Препринты ИПМ им. М.В. Келдыша. – 2022. – № 17. – С. 1-10. – ISSN отсутствует.
50. Коваленко, И.И. Анализ правил комбинирования групповых экспертных оценок в конфликтных ситуациях / И.И. Коваленко, А.В. Швед, Е.С. Пугаченко // Проблемы информационных технологий. – 2014. – № 15. – С. 14-20. – ISSN отсутствует.
51. Куренных, А.Е. О повышении индекса согласованности матрицы парных сравнений в системах поддержки принятия решений / А.Е. Куренных, В.П. Осипов, А.И. Посадский, В.А. Судаков // Препринты ИПМ им. М.В. Келдыша. – 2018. – № 196. – С. 1-15. – ISSN отсутствует.

52. Лысак, И.В. Современное общество как общество сетевых структур / И.В. Лысак, Л.Ф. Косенчук // Информационное общество. – 2015. – № 2. – С. 45-51. – ISSN отсутствует.

53. Маросин, В.С. Сетевые модели планирования в управлении грузовыми автомобильными перевозками / В.С. Маросин, С.А. Гусев // Информационные технологии и инновации на транспорте. – 2019. – С. 65-70. – ISSN отсутствует.

54. Олескин, А.В. Сетевые структуры в биосистемах / А.В. Олескин // Журнал общей биологии. – 2013. – № 2. Том 7. – С. 112-138. – ISSN 0044-4596.

55. Олескин, А.В. Сетевые структуры, иерархии и квазирынки в биосистемах, человеческом обществе и технических информационных системах. Критерии классификации / А.В. Олескин // Сложные системы. – 2014. – № 2 (11). – С. 36-58. – ISSN 2220-8569.

56. Осинцев, Н.А. Многокритериальные методы принятия решений в зеленой логистике / Н.А. Осинцев // Мир транспорта. – 2021. – № 19 (5). – С. 105-114. – ISSN 1992-3252.

57. Родригес, В.С. Многокритериальная оценка альтернатив территориального планирования при строительстве плотин с использованием технологий географических информационных систем / В.С. Родригес, Н.В. Мокрова // Отходы и ресурсы. – 2020. – № 1. Том 7. – С. 1-17. – ISSN 2500-0659.

58. Тишкин, В.Ф. Моделирование транспортных потоков на основе квазигазодинамического подхода и теории клеточных автоматов с использованием суперкомпьютеров / В.Ф. Тишкин, М.А. Трапезникова, А.А. Чечина [и др.] // Компьютерные исследования и моделирование. – 2024. – № 1. Том 16. – С. 175-194. – ISSN отсутствует.

59. Урыков, В.А. Модели транспортного инфраструктурного комплекса / В.А. Урыков, Л.И. Зеленина // Современные научные исследования и инновации. – 2014. – № 12. – С. 1-15. – ISSN 2223-4888.

60. Финогеев, А.А. Методы и средства мониторинга дорожного трафика в среде smart road / А.А. Финогеев, М.В. Деев, А.Г. Финогеев [и др.] // Новые информационные технологии и системы : сборник научных статей по материалам XVII Международной научно-технической конференции. – Пенза : Пензенский государственный университет. – 2020. – С. 91-96. – ISBN отсутствует.

61. Четверушкин, Б.Н. Факторное моделирование для инновационно-активных предприятий / Б.Н. Четверушкин, В.А. Судаков // Математическое моделирование. – 2020. – № 3. Том 32. – С. 115-126. – ISSN отсутствует.

62. Чечина, А.А. Модель транспортных потоков на основе теории клеточных автоматов для решения задач управления движением на городских дорожных сетях / А.А. Чечина, Н.Г. Чурбанова, М.А. Трапезникова // Математическая теория управления и ее приложения. – 2020. – № 2. – С. 154-157. – ISSN отсутствует.

63. Швецов, В.И. Математическое моделирование транспортных потоков / В.И. Швецов // Автоматика и телемеханика. – 2003. – № 11. – С. 1651-1689. – ISSN 0005-2310.

64. Яшина, М.В. Модель потока транспортных средств различных типов на многополосном участке магистрали перед съездом / М.В. Яшина, А.Г. Таташев, А.С. Доткулова [и др.] // Журнал СТАН. – 2022. – № 9. – С. 32-35. – ISSN 0869-7566.

Источники на иностранном языке

65. Abdelgadir, M. Mobility Routing Model for Vehicular Ad-hoc Networks (VANETs), Smart City Scenarios / M. Abdelgadir // Vehicular Communications. – 2017. – № 9. – P. 154-161. – ISSN 2214-2096.

66. Agrawal S. Two-dimensional LWR model for lane-free traffic / S. Agrawal, V. Kanagaraj, M. Treiber // Physica A: Statistical Mechanics and its Applications. – 2023. – № 625. – P. 1-12. – ISSN 0378-4371.

67. Alhadidi, T. A FAHP-VIKOR model for evaluating single point interchange operational performance / T. Alhadidi, A. Alomari // Expert Systems with Applications. – 2024. – № 248. – P. 1-15. – ISSN 0957-4174.
68. Animah, I. Compliance with MARPOL Annex VI regulation 14 by ships in the Gulf of Guinea sub-region: Issues, challenges and opportunities / I. Animah // Transportation Research Part D Transport and Environment. – 2018. – № 62. – P. 441-455. – ISSN 1361-9209.
69. Aydin, N. A hierarchical customer satisfaction framework for evaluating rail transit systems of Istanbul / N. Aydin, E. Celik, A. Gumus // Transportation Research Part A: Policy and Practice. – 2015. – № 77. – P. 61-81. – ISSN 0965-8564.
70. Awasthi, A. Using AHP and Dempster–Shafer theory for evaluating sustainable transport solutions / A. Awasthi // Environmental Modelling & Software. – 2011. – № 26 (6). – P. 787-796. – ISSN 1364-8152.
71. Bao, X. An improved deep belief network for traffic prediction considering weather factors / X. Bao // Alexandria Engineering Journal. – 2021. – № 60 (1). – P. 413-420. – ISSN 1110-0168.
72. Barcelo, J. Models, Traffic Models, Simulation, and Traffic Simulation / J. Barcelo // Fundamentals of Traffic Simulation. – 2010. – № 1. – P. 1-62. – ISSN 0884-8289.
73. Beneicke, J. Empowering Citizens' Cognition and Decision Making in Smart Sustainable Cities // J. Beneicke, A.A. Juan, F. Xhafa [et al.] // IEEE Consumer Electronics Magazine. – 2020. – № 9 (1). – P. 102-108. – ISSN 2162-2248.
74. Bilisik, O. A hybrid fuzzy methodology to evaluate customer satisfaction in a public transportation system for Istanbul / O. Bilisik, M. Erdogan, I. Kaya [et al.] // Total Quality Management & Business Excellence. – 2013. – № 24 (9). – P. 1141-1159. – ISSN 1478-3363.

75. Blasius T. A Strategic Routing Framework and Algorithms for Computing Alternative Paths // T. Blasius, M. Bother, P. Fischbeck [et al.] / Data Structures and Algorithms. – 2020. – № 2. – P. 19-31. – ISSN 0757-1721.

76. Boujelbene, Y. The Performance Analysis of Public Transport Operators in Tunisia Using AHP Method / Y. Boujelbene, A. Derbel // Procedia Computer Science. – 2015. – № 73. – P. 498-508. – ISSN 1877-0509.

77. Brakatsoulas, S. On Map-Matching Vehicle Tracking Data / S. Brakatsoulas, D. Pfoser, R. Salas [et al.] // Conference: Proceedings of the 31st International Conference on Very Large Data Bases. – 2005. – № 1. – P. 853-864. – ISSN 0757-1721.

78. Bratsas, C. Comparison of Machine Learning Methods for the Prediction of Traffic Speed in Urban Places / C. Bratsas, K. Koupidis, J.-M. Salanova [et al.] // Sustainability. – 2020. – № 12 (1). – C. 1-10. – ISSN 2071-1050.

79. Cai, P. A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting / P. Cai // Transportation Research Part C Emerging Technologies. – 2016. – № 62. – P. 21-34. – ISSN 0968-090X.

80. Caramuta, C. Integrated evaluation methodology and its application to freight transport policies in the port of Trieste / C. Caramuta // Transportation Research Procedia. – 2018. – № 30. – P. 119-126. – ISSN 2352-1465.

81. Celik, E. An interval type-2 fuzzy AHP and TOPSIS methods for decision-making problems in maritime transportation engineering: The case of ship loader / E. Celik, E. Akyuz // Ocean Engineering. – 2018. – № 155. – P. 371-381. – ISSN 0029-8018.

82. Chakraborty, S. TOPSIS and Modified TOPSIS: A comparative analysis / S. Chakraborty // Decision Analytics Journal. – 2022. – № 2. – P. 1-14. – ISSN 2772-6622.

83. Chang, Y. Comprehensive Sustainability Evaluation of High-Speed Railway (HSR) Construction Projects Based on Unascertained Measure and

Analytic Hierarchy Process / Y. Chang // Sustainability. – 2018. – № 10. – P. 1-10. – ISSN 2071-1050.

84. Chao, C. Selection of strategic cargo alliance by airlines / C. Chao, K. Kao // Journal of Air Transport Management. – 2015. – № 1. – P. 29-36. – ISSN 0969-6997.

85. Chen, C. Exploiting Spatial-Temporal Relations for Traffic Prediction with Deep Neural Networks / C. Chen, D. Shaojiang, J. Shuyuan [et al.] // Applied Soft Computing. – 2019. – № 5. Volume 78. – P. 712-721. – ISSN отсытсбует.

86. Chen, K. An Improved A* Search Algorithm for Road Networks Using New Heuristic Estimation / K. Chen // Data Structures and Algorithms. – 2022. – № 4. – P. 1-15. – ISSN отсытсбует.

87. Chen, W. Multi-Range Attentive Bicomponent Graph Convolutional Network for Traffic Forecasting / W. Chen // Proceedings of the AAAI Conference on Artificial Intelligence. – 2020. – № 34 (04). – P. 3529-3536. – ISSN 2374-346.

88. Chetverushkin, B.N. Factor model for the study of complex processes / B.N. Chetverushkin, V.A. Sudakov // Doklady Mathematics. – 2019. – №3. Volume 100. – P. 514-518. – ISSN отсытсбует.

89. Chin, K. Identifying passengers' needs in cabin interiors of high-speed rails in China using quality function deployment for improving passenger satisfaction / K. Chin // Transportation Research Part A: Policy and Practice. – 2019. – № 4. Volume 119. – P. 326-342. – ISSN 0965-8564.

90. Chou, C. An evaluation of airline service quality using the fuzzy weighted SERVQUAL method / C. Chou, L. Liu, S. Huang [et al.] // Applied Soft Computing. – 2011. – № 2. Volume 11. – P. 2117-2128. – ISSN 1568-4946.

91. Chung, J. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling / J. Chung, C. Gulcehre, K. Caglar [et al.] // Neural and Evolutionary Computing. – 2014. – № 1. – P. 1-10. – ISSN отсытсбует.

92. Daganzo, C. Remarks on Traffic Flow Modeling and Its Applications / C. Daganzo // Traffic and Mobility. – 1999. – № 1. – P. 105-115. – ISSN отсытсбует.

93. Daganzo, C. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory / C. Daganzo // *Transportation Research Part B: Methodological*. – 1994. – № 1. Volume 28 (4). – P. 269-287. – ISSN 0191-2615.
94. Dechter, R. Generalized best-first search strategies and the optimality of A* / R. Dechter, J. Pearl // *Journal of the ACM*. – 1985. – № 3. Volume 32. – P. 505-536. – ISSN 0037-5487.
95. Ding, R. Application of Complex Networks Theory in Urban Traffic Network Researches / R. Ding, N. Ujang, H. Hamid [et al.] // *Networks and Spatial Economics*. – 2019. – № 1. Volume 19. – P. 1281-1317. – ISSN 1572-9427.
96. Du, S. Traffic flow forecasting based on hybrid deep learning framework / S. Du // *12th International Conference on Intelligent Systems and Knowledge Engineering*. – 2017. – № 12. – P. 1-6. – ISSN 1546-2206.
97. Duleba, S. A dynamic analysis on public bus transport's supply quality by using AHP / S. Duleba, T. Mishina, Y. Shimazaki // *Transport*. – 2012. – № 27 (3). – P. 268-275. – ISSN 1648-3480.
98. Duleba, S. An AHP-ISM approach for considering public preferences in a public transport development decision / S. Duleba // *Transport*. – 2019. – № 1. Volume 34 (3). – P. 662-671. – ISSN 1648-4142.
99. Duleba, S. An Analysis on the Connections of Factors in a Public Transport System by AHP-ISM / S. Duleba // *Transport*. – 2013. – № 28 (4). – P. 404-412. – ISSN 1648-3480.
100. Duleba, S. Sustainable Urban Transport Development with Stakeholder Participation, an AHP-Kendall Model: A Case Study for Mersin / S. Duleba // *Sustainability*. – 2018. – № 10. Volume 10 (8). – P. 22-28. – ISSN 2071-1050.
101. Dushenko, M. Application of a Sustainability Model for Assessing the Relocation of a Container Terminal: A Case Study of Kristiansand Port / M. Dushenko // *Sustainability*. – 2019. – № 8. Volume 11. – P. 1-10. – ISSN 2071-1050.

102. Ebrahimi, H. A system theory based accident analysis model: STAMP-fuzzy DEMATEL / H. Ebrahimi, E. Zarei, M. Ansari [et al.] // Safety Science. – 2024. – № 3. Volume 173. – P. 1-12. – ISSN 0925-7535.

103. Dai, Y.H. Temporal and spatial correlation patterns of air pollutants in Chinese cities / Y.H. Dai, W.X. Zhou // PLoS ONE. – 2017. – № 1. Volume 12. – P. 14-25. – ISSN отсутствует.

104. Evagelopoulos, V. PM10-PM2.5 time 411 series and fractal analysis / V. Evagelopoulos, V. Zoras, S. Triantafyllou [et al.] // Global NEST. – 2006. – № 1. – P. 234-240. – ISSN отсутствует.

105. Feng, S. A fractal analysis of radon migration in discrete fracture network model / S. Feng, Y. Wu, Y. Liu [et al.] // Chemosphere. – 2021. – № 1. – P. 10-18. – ISSN отсутствует.

106. Efrat, A. New similarity measures between polylines with applications to morphing and polygon sweeping / A. Efrat, G. Leonidas, H. Sariel [et al.] // Discrete and Computational Geometry. – 2002. – № 3. Volume 28. – P. 535-569. – ISSN отсутствует.

107. Ехлаков, Р. Modern Methods of Traffic Flow Modeling: A Graph Load Calculation Model Based on Real-Time Data = Современные методы моделирования потока трафика: модель расчета нагрузки на график, основанная на данных в реальном времени / Р. Ехлаков // Intelligent Decision Technologies : proceedings of the 15th KES-IDT 2023 Conference. – Singapore : Springer. – 2023. – P. 302-309. – ISSN 2190-3018.

108. Ехлаков, Р. Multicriteria Assessment Method for Network Structure Congestion Based on Traffic Data Using Advanced Computer Vision = Метод оценки многокритерии для перегрузки сетевой структуры на основе данных трафика с использованием расширенного компьютерного зрения / Р. Ехлаков, Н. Андриянов // Mathematics. – 2024. – № 4. Volume 12. – ISSN 2227-7390.

109. Ехлаков, Р.С. Development and Application of Step-By-Step Addition Algorithm for Modelling the Surface of Radioactive Contamination of the Area = Разработка и применение пошагового алгоритма добавления для

моделирования поверхности радиоактивного загрязнения области / Р.С. Ехлаков, В.В. Татаринов, А.С. Дутов // AIP Conference Proceedings. – 2019. – Issue 1. Volume 2195. – ISSN 1551-7616.

110. Ехлаков, Р.С. Investigation of the Application of Fractal Geometry Methods for the Construction of Radioactive Contamination Zones = Исследование применения методов фрактальной геометрии для построения зон радиоактивного загрязнения / Р.С. Ехлаков, В.В. Татаринов, А.С. Дутов // AIP Conference Proceedings. – 2019. – Issue 1. Volume 2195. – ISSN 1551-7616.

111. Emami, A. Short-term traffic flow prediction based on faded memory Kalman Filter fusing data from connected vehicles and Bluetooth sensors / A. Emami, M. Sarvi, S. Bagloee // Simulation Modelling Practice and Theory. – 2020. – № 12. Volume 102. – P. 1-10. – ISSN 1569-190X.

112. Gao, T. Study of the Competitiveness of Quanzhou Port on the Belt and Road in China Based on a Fuzzy-AHP and ELECTRE III Model / T. Gao // Sustainability. – 2018. – № 5. Volume 10. – P. 1-14. – ISSN 2071-1050.

113. Guitouni, A. Tentative guidelines to help choosing an appropriate MCDA method / A. Guitouni, J.-M. Martel // European Journal of Operational Research. – 1998. – № 4. Volume 109 (2). – P. 501-521. – ISSN 1872-6860.

114. Guner, S. Measuring the quality of public transportation systems and ranking the bus transit routes using multi-criteria decision making techniques / S. Guner // Case Studies on Transport Policy. – 2018. – № 1. Volume 6 (2). – P. 214-224. – ISSN 2213-624X.

115. Hamurcu, M. An Application of Multicriteria Decision-making for the Evaluation of Alternative Monorail Routes / M. Hamurcu, T. Eren // Mathematics. – 2019. – № 2. Volume 7 (16). – P. 20-24. – ISSN 2227-7390.

116. Han, D. Highway Traffic Speed Prediction in Rainy Environment Based on APSO-GRU / D. Han // Journal of Advanced Transportation. – 2021. – № 4. Volume 21. – P. 12-25. – ISSN 2042-3195.

117. He, T. A fuzzy AHP based integer linear programming model for the multi-criteria transshipment problem / T. He, W. Ho, C. Man [et al.] // The

International Journal of Logistics Management. – 2021. – № 9. Volume 23 (1). – P. 159-179. – ISSN 0957-4093.

118. Hochreiter, S. Long Short-Term Memory / S. Hochreiter, J. Schmidhuber // Neural Computation. – 1997. – № 9 (8). – P. 1735-1780. – ISSN 0899-7667.

119. Hsu, W. Evaluating the service requirements of dedicated container terminals: a revised IPA model with fuzzy AHP / W. Hsu // Maritime Policy & Management. – 2015. – № 42 (8). – P. 1-17. – ISSN 1464-5254.

120. Ignaccolo, M. Combining Analytic Hierarchy Process (AHP) with role-playing games for stakeholder engagement in complex transport decisions / M. Ignaccolo // Transportation Research Procedia. – 2017. – № 2. Volume 27. – P. 500-507. – ISSN 2352-1465.

121. Janiak, M. Multiple Criteria Evaluation of Different Redesign Variants of the Public Tram System / M. Janiak // Transportation Research Procedia. – 2014. – № 1. Volume 3. – P. 690-699. – ISSN 2352-1465.

122. Jianyi, L. Using Support Vector Machines for Long-Term Discharge Prediction / L. Jianyi // Hydrological Sciences Journal. – 2020. – № 2. Volume 51 (4). – P. 599-612. – ISSN 2150-3435.

123. Jones, S. Proposed framework for sustainability screening of urban transport projects in developing countries: A case study of Accra, Ghana / S. Jones, M. Tefe, S. Appiah-Opoku // Transportation Research Part A: Policy and Practice. – 2013. – № 3. Volume 49. – P. 21-34. – ISSN 0965-8564.

124. Joubert, J. Multi-agent model of route choice when vehicles are sensitive to road grade / J. Joubert // Procedia Computer Science. – 2017. – № 12. Volume 109. – P. 869-874. – ISSN 1877-0509.

125. Kijewska, K. Application of AHP and DEMATEL Methods in Choosing and Analysing the Measures for the Distribution of Goods in Szczecin Region / K. Kijewska // Sustainability. – 2018. – № 3. Volume 10. – P. 1-10. – ISSN 2071-1050.

126. Kim, Y. Scalable Learning with a Structural Recurrent Neural Network for Short-Term Traffic Prediction / Y. Kim, P. Wang, L. Mihaylova // IEEE Sensors Journal. – 2019. – № 2. Volume 19 (23). – P. 11359-11366. – ISSN 1558-1748.

127. Kockelman, K. Modeling traffic's flow-density relation: Accommodation of multiple flow regimes and traveler types / K. Kockelman // Transportation. – 2001. – № 1. Volume 28. – P. 363-374. – ISSN 1572-9435.

128. Kouziokas, G. Deep Bidirectional and Unidirectional LSTM Neural Networks in Traffic Flow Forecasting from Environmental Factors / G. Kouziokas // Advances in Intelligent Systems and Computing. – 2020. – № 3. Volume 1278. – P. 1-11. – ISSN 0732-0797.

129. Lee, J. Optimal fleet, battery, and charging infrastructure planning for reliable electric bus operations / J. Lee, H. Shon, I. Papakonstantinou, S. Son // Transportation Research Part D: Transport and Environment. – 2021. – № 2. Volume 100. – P. 1-12. – ISSN 1361-9209.

130. Li, H. New Model of Travel-Time Prediction Considering Weather Conditions: Case Study of Urban Expressway / H. Li // Journal of Transportation Engineering, Part A: Systems. – 2020. – № 1. Volume 147 (3). – P. 15-22. – ISSN 473-2907.

131. Li, T. Markov-Based Time Series Modeling Framework for Traffic-Network State Prediction under Various External Conditions / T. Li // Journal of Transportation Engineering, Part A: Systems. – 2020. – № 2. Volume 146 (6). – P. 20-31. – ISSN 2473-2893.

132. Liou, J. Airline safety measurement using a hybrid model / J. Liou, G. Tzeng, H. Chang // Journal of Air Transport Management. – 2007. – № 1. – Volume 13 (4). – P. 243-249. – ISSN 0969-6997.

133. Lin, Z. DeepSTN+: Context-Aware Spatial-Temporal Neural Network for Crowd Flow Prediction in Metropolis / Z. Lin // Proceedings of the AAAI Conference on Artificial Intelligence. – 2019. – № 4. Volume 33 (1). – P. 1020-1027. – ISSN 0732-0797.

134. Lirn, T. Transshipment Port Selection and Decision-making Behaviour: Analysing the Taiwanese Case / T. Lirn, H. Thanopoulou, A. Beresford // International Journal of Logistics Research and Applications. – 2021. – № 6 (4). – P. 229-244. – ISSN 1367-5567.

135. Liu, F. Urban Traffic Flow Prediction Model with CPSO/SSVM Algorithm under the Edge Computing Framework / F. Liu // Wireless Communications and Mobile Computing. – 2020. – № 2. Volume 1. – P. 1-11. – ISSN 1530-8677.

136. Lopez, C. On the Environmental and Social Sustainability of Technological Innovations in Urban Bus Transport: The EU Case / C. Lopez // Sustainability. – 2019. – № 2. Volume 11. – P. 15-31. – ISSN 2071-1050.

137. Lu, H. ST-TrafficNet: A Spatial-Temporal Deep Learning Network for Traffic Forecasting / H. Lu // Electronics. – 2020. – № 19. Volume 9. – P. 112-119. – ISSN 2079-9292.

138. Luca, S. Public engagement in strategic transportation planning: An analytic hierarchy process based approach / S. Luca // Transport Policy. – 2014. – № 10. Volume 33. – P. 110-124. – ISSN 0967-070X.

139. Lupo, T. Handling stakeholder uncertain judgments in strategic transport service analyses / T. Lupo // Transport Policy. – 2013. – № 9. Volume 29. – P. 54-63. – ISSN 0967-070X.

140. Ma, F. Evaluation of urban green transportation planning based on central point triangle whiten weight function and entropy-AHP / F. Ma // Transportation Research Procedia. – 2017. – № 1. Volume 25. – P. 3634-3644. – ISSN 2352-1465.

141. Madushika, I. AHP integrated MILP approach to minimize transportation cost to prioritize distribution requirements / I. Madushika, A. Wijayanayake // International Research Conference on Smart Computing and Systems Engineering. – 2018. – № 2. – P. 1-15. – ISSN 0975-3465.

142. Mahmassani, H. Implementation of a Weather Responsive Traffic Estimation and Prediction System (TrEPS) for Signal Timing at Utah DOT /

H. Mahmassani // Civil and Environmental Engineering. – 2014. – № 1. – P. 19-31. – ISSN 0304-0201.

143. Majumdar, S. Congestion prediction for smart sustainable cities using IoT and machine learning approaches / S. Majumdar // Sustainable Cities and Society. – 2021. – № 12. Volume 64. – P. 10-15. – ISSN 2210-6707.

144. Manca, D. A methodology based on the Analytic Hierarchy Process for the quantitative assessment of emergency preparedness and response in road tunnels / D. Manca // Transport Policy. – 2011. – № 1. Volume 18 (5). – P. 657-664. – ISSN 0967-070X.

145. Mandic, D. Two-phase model for multi-criteria project ranking: Serbian Railways case study / D. Mandic, P. Jovanovic, M. Bugarinovic // Transport Policy. – 2014. – № 19. Volume 36. – P. 88-104. – ISSN 0967-070X.

146. Martin, J. Metaphor in Science / J. Martin, R. Harre // Metaphor: Problems and Perspectives. Brington. – 1982. – № 2. Volume 1 (3). – P. 81-85. – ISSN 0304-0201.

147. Mi, X. Hesitant Fuzzy Linguistic Analytic Hierarchical Process with Prioritization, Consistency Checking and Inconsistency Repairing / X. Mi, X. Wu, M. Tang [et al.] // IEEE Access. – 2019. – № 5. Volume 99. – P. 14-25. – ISSN 2169-3536.

148. Mousavi, M. The application of PROMETHEE multi-criteria decision aid in financial decision making: Case of distress prediction models evaluation / M. Mousavi, J. Lin // Expert Systems with Applications. – 2020. – № 10. Volume 159. – P. 111-129. – ISSN 1873-6793.

149. Nassereddine, M. An integrated MCDM approach to evaluate public transportation systems in Tehran / M. Nassereddine, H. Eskandari // Transportation Research Part A: Policy and Practice. – 2017. – № 12. Volume 106. – P. 427-439. – ISSN 0965-8564.

150. Numeroso, D. Learning heuristics for A* / D. Numeroso, D. Bacciu, P. Velickovic // Artificial Intelligence. – 2022. – № 1. – P. 1-12. – ISSN 0304-0201.

151. Obregon-Biosca, S. Rating road traffic education / S. Obregon-Biosca // *Transportation Research Part F: Traffic Psychology and Behaviour*. – 2018. – № 3. Volume 56. – P. 33-45. – ISSN 1369-8478.
152. Oumaima, E. A stochastic mobility model for traffic forecasting in urban environments / E. Oumaima // *Journal of Parallel and Distributed Computing*. – 2022. – № 2. Volume 165. – P. 142-155. – ISSN 0743-7315.
153. Pan, Y. Infants' MTHFR Polymorphisms and Nonsyndromic Orofacial Clefts Susceptibility / Y. Pan // *Obstetrical & Gynecological Survey*. – 2013. – № 68 (2). – P. 157-159. – ISSN 0754-1801.
154. Park, K. Ship Acquisition of Shipping Companies by Sale & Purchase Activities for Sustainable Growth: Exploratory Fuzzy-AHP Application / K. Park // *Sustainability*. – 2018. – № 5. Volume 10. – P. 1-15. – ISSN 0965-8564.
155. Peng, P. Modelling the competitiveness of the ports along the Maritime Silk Road with big data / P. Peng // *Transportation Research Part A: Policy and Practice*. – 2018. – № 5. Volume 118. – P. 852-867. – ISSN 0965-8564.
156. Peyman, M. Optimization of Vehicular Networks in Smart Cities: From Agile Optimization to Learnheuristics and Simheuristics / M. Peyman, T. Fluechter, J. Panadero [et al.] // *Sensors*. – 2023. – № 10. Volume 23 (1). – P. 15-29. – ISSN 1424-8220.
157. Podvezko, V. The use of AHP and rank correlation methods for determining the significance of the interaction between the elements of a transport system having a strong influence on traffic safety / V. Podvezko // *Transport*. – 2013. – № 28 (4). – P. 389-403. – ISSN 1648-3480.
158. Polson, N. Deep learning for short-term traffic flow prediction / N. Polson // *Transportation Research Part C: Emerging Technologies*. – 2017. – № 2. Volume 79. – P. 1-17. – ISSN 0968-090X.
159. Postorino, M. An application of the Multi-Criteria Decision-Making analysis to a regional multi-airport system / M. Postorino, F. Pratico // *Research in Transportation Business & Management*. – 2012. – № 11. Volume 4. – P. 44-52. – ISSN 2210-5395.

160. Rahaman, M. CAPRA: A contour-based accessible path routing algorithm / M. Rahaman, Y. Mei, M. Hamilton [et al.] // Information Sciences. – 2017. – № 12. Volumes 385-386. – P. 157-173. – ISSN 0020-0255.

161. Rezael, J. Supplier selection in the airline retail industry using a funnel methodology: Conjunctive screening method and fuzzy AHP / J. Rezael, P. Fahim, L. Tavasszy // Expert Systems with Applications. – 2014. – № 22. Volume 41 (18). – P. 8165-8179. – ISSN 0957-4174.

162. Ruiz-Padillo, A. Application of multi-criteria decision analysis methods for assessing walkability: A case study in Porto Alegre, Brazil / A. Ruiz-Padillo // Transportation Research Part D Transport and Environment. – 2018. – № 10. Volume 63. – P. 855-871. – ISSN 1361-9209.

163. Salvador, G. ELECTRE applied in supplier selection – a literature review / G. Salvador, M. Moura, P. Campos [et al.] // Procedia Computer Science. – 2024. – № 5. Volume 22. – P. 1759-1768. – ISSN 1877-0509.

164. Semanjski, I. A Collaborative Stakeholder Decision-Making Approach for Sustainable Urban Logistics / I. Semanjski // Sustainability. – 2019. – № 5. Volume 11. – P. 15-29. – ISSN 2071-1050.

165. Sennaroglu, B. A military airport location selection by AHP integrated PROMETHEE and VIKOR methods / B. Sennaroglu, G. Celebi // Transportation Research Part D: Transport and Environment. – 2018. – № 12. Volume 59. – P. 160-173. – ISSN 1361-9209.

166. Sobhani, M. A framework for analyzing the competitiveness of unconventional modes of transportation in developing cities / M. Sobhani // Transportation Research Part A: Policy and Practice. – 2020. – № 5. Volume 137. – P. 504-518. – ISSN 0965-8564.

167. Song, C. Spatial-Temporal Synchronous Graph Convolutional Networks: A New Framework for Spatial-Temporal Network Data Forecasting / C. Song // Proceedings of the AAAI Conference on Artificial Intelligence. – 2020. – № 34 (1). – P. 914-921. – ISSN 2374-3468.

168. Shiau, T. Evaluating sustainable transport strategies for the counties of Taiwan based on their degree of urbanization / T. Shiau // *Transport Policy*. – 2013. – № 6. Volume 30. – P. 101-108. – ISSN 0967-070X.

169. Sohn, K. A systematic decision criterion for the elimination of useless overpasses / K. Sohn // *Transportation Research Part A: Policy and Practice*. – 2008. – № 1. Volume 42 (8). – P. 1043-1055. – ISSN 0965-8564.

170. Soner, O. Application of AHP and VIKOR methods under interval type 2 fuzzy environment in maritime transportation / O. Soner // *Ocean Engineering*. – 2017. – № 12. Volume 129. – P. 107-116. – ISSN 0029-8018.

171. Sun, T. Bidirectional Spatial–Temporal Network for Traffic Prediction with Multisource Data / T. Sun // *Transportation Research Record: Journal of the Transportation Research Board*. – 2020. – № 11. Volume 2674 (8). – P. 190-212. – ISSN 0361-1981.

172. Tang, J. Characterizing traffic time series based on complex network theory / J. Tang, Y. Wang, F. Liu // *Physica A: Statistical Mechanics and its Applications*. – 2013. – № 24. Volume 392 (18). – P. 4192-4201. – ISSN 0378-4371.

173. Tanimura, R. Prediction of Deceleration Amount of Vehicle Speed in Snowy Urban Roads Using Weather Information and Traffic Data / R. Tanimura // *IEEE 18th International Conference on Intelligent Transportation Systems*. – 2015. – № 12. – P. 2268-2273. – ISBN 2153-0009.

174. Tanner, J. Effect of Weather on Traffic Flow / J. Tanner // *Nature*. – 1952. – № 11. Volume 169. – P. 107-115. – ISSN 0028-0836.

175. Tempelmeier, N. Crosstown traffic – supervised prediction of impact of planned special events on urban traffic / N. Tempelmeier, S. Dietze, E. Demidova // *Geoinformatica*. – 2020. – № 12. Volume 24. – P. 339-370. – ISSN 1573-7624.

176. Tseng, P.-H. Evaluating the key factors of green port policies in Taiwan through quantitative and qualitative approaches / P.-H. Tseng // *Transport Policy*. – 2019. – № 2. Volume 82. – P. 127-137. – ISSN 0967-070X.

177. Tudela, A. Comparing the output of cost benefit and multi-criteria analysis: An application to urban transport investments / A. Tudela, N. Akiki,

R. Cisternas // Transportation Research Part A: Policy and Practice. – 2006. – № 10. Volume 40 (5). – P. 414-423. – ISSN 0965-8564.

178. Valipour, A. Risk Assessment in PPP Projects by Applying Different MCDM Methods and Comparative Results Analysis / A. Valipour, H. Sarvari, J. Tamosaitiene // Rational Decision Making in Risk Management. – 2018. – № 4. Volume 8. – P. 1-19. – ISSN 2076-3387.

179. Vaswani, A. Attention Is All You Need / A. Vaswani // Advances in Neural Information Processing Systems. – 2017. – № 2. – P. 1-15. – ISSN отсытсбует.

180. Vetrian, V. Trace Based Mobility Model for Ad Hoc Networks / V. Vetrian // IEEE Xplore. – 2017. – № 1. – P. 14-25. – ISSN 2160-4886.

181. Wang, H. A traffic data collection and analysis method based on wireless sensor network // H. Wang, M, Ouyang, Q. Meng [et al.] // EURASIP Journal on Wireless Communications and Networking. – 2020. – № 2. – P. 20-31. – ISSN отсытсбует.

182. Wang, J. A Deep Prediction Architecture for Traffic Flow with Precipitation Information / J. Wang // Advances in Swarm Intelligence Notes in Computer Science. – 2020. – № 2. Volume 10942. – ISSN отсытсбует.

183. Wang, J. Traffic speed prediction for urban transportation network: A path based deep learning approach / J. Wang // Transportation Research Part C: Emerging Technologies. – 2019. – № 3. Volume 100. – P. 372-385. – ISSN 0968-090X.

184. Wei, W. An AutoEncoder and LSTM-Based Traffic Flow Prediction Method / W. Wei // Sensors. – 2019. – № 2. Volume 19. – P. 1-14. – ISSN 1424-8220.

185. Wu, Y. A hybrid deep learning based traffic flow prediction method and its understanding / Y. Wu // Transportation Research Part C: Emerging Technologies. – 2018. – № 2. Volume 90. – P. 166-180. – ISSN 0968-090X.

186. Xu, X. Traffic flow prediction based on random forest in severe weather conditions / X. Xu // Journal of Shaanxi Normal University. – 2020. – № 3. Volume 48. – P. 25-31. – ISSN отсуттбует.

187. Xue, F. Prediction of Urban Rail Transit Ridership under Rainfall Weather Conditions / F. Xue // Journal of Transportation Engineering, Part A: Systems. – 2020. – № 2. Volume 146. – P. 1-15. – ISSN 2473-2893.

188. Xue, Y. Traffic Status Evolution Trend Prediction Based on Congestion Propagation Effects under Rainy Weather / Y. Xue // Journal of Advanced Transportation. – 2020. – № 2. – Volume 2020. – P. 1-10. – ISSN отсуттбует.

189. Yuan, X. Deep Learning for Urban Traffic Flow Prediction: A Review and New Directions / X. Yuan // Transportation Research Part C: Emerging Technologies. – 2021. – № 12. Volume 79. – P. 1-17. – ISSN отсуттбует.

190. Yang, Y. A Novel Centrality of Influential Nodes Identification in Complex Networks / Y. Yang, X. Wang, Y. Chen [et al.] // IEEE Access. – 2020. – № 2. Volume 8. – P. 58742-58751. – ISSN 2169-3536.

191. Yao, R. DLW-Net model for traffic flow prediction under adverse weather / R. Yao // Transportmetrica B: Transport Dynamics. – 2022. – № 1. Volume 10. – P. 499-524. – ISSN 2168-0582.

192. Yazdani-Chamzini, A. Tunnel Boring Machine (TBM) selection using fuzzy multicriteria decision making methods / A. Yazdani-Chamzini, S. Yakhchali // Tunnelling and Underground Space Technology. – 2012. – № 2. Volume 30. – P. 194-204. – ISSN 0886-7798.

193. Yedla, S. Multi-criteria approach for the selection of alternative options for environmentally sustainable transport system in Delhi / S. Yedla, R. Shrestha // Transportation Research Part A: Policy and Practice. – 2003. – № 12. Volume 37. – P. 717-729. – ISSN 0965-8564.

194. Yen, J. An algorithm for finding shortest routes from all source nodes to a given destination in general networks / J. Yen // Quarterly of Applied Mathematics. – 1970. – № 10. Volume 27. – P. 526-530. – ISSN 1552-4485.

195. Yeo, G. An Application of the Hierarchical Fuzzy Process to Container Port Competition: Policy and Strategic Implications / G. Yeo, D. Song // Transportation. – 2006. – № 2. Volume 33 (4). – P. 409-422. – ISSN 1572-9435.
196. Yeo, G. Evaluating the competitiveness of the aerotropolises in East Asia / G. Yeo, Y. Wang, C. Chou // Journal of Air Transport Management. – 2013. – № 1. Volume 32. – P. 24-31. – ISSN 0969-6997.
197. Yildirim, V. A geographic information system-based model for economical and eco-friendly high-speed railway route determination using analytic hierarchy process and least-cost-path analysis / V. Yildirim, S. Bediroglu // Expert Systems. – 2019. – № 2. Volume 36. – P. 1-10. – ISSN 0957-4174.
198. Yu, J. Study on the Status Evaluation of Urban Road Intersections Traffic Congestion Base on AHP-TOPSIS Modal / J. Yu, L. Wang, X. Gong // Procedia - Social and Behavioral Sciences. – 2013. – № 2. Volume 96. – P. 609-616. – ISSN 1877-0428.
199. Zak, J. Application of AHP and ELECTRE III/IV Methods to Multiple Level, Multiple Criteria Evaluation of Urban Transportation Projects / J. Zak // Transportation Research Procedia. – 2015. – № 10. Volume 10. – P. 820-830. – ISSN 2352-1457.
200. Zakaria, J. Comfort of Walking in the City Center of Kuala Lumpur / J. Zakaria // Procedia - Social and Behavioral Sciences. – 2015. – № 2. Volume 170. – P. 642-652. – ISSN 1877-0428.
201. Zavadskas, E. Multiple Criteria Decision Making (MCDM) Methods in Economics: An Overview / E. Zavadskas, Z. Turskis // Technological and Economic Development of Economy. – 2011. – № 17 (2). – P. 397-427. – ISSN 2029-4921.
202. Zhang, J. Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction / J. Zhang // Thirty-First AAAI Conference on Artificial Intelligence. – 2017. – № 1. Volume 31. – P. 21-33. – ISSN 2374-3468.
203. Zhang, L. Neural Network based Vehicle Speed Prediction for Specific Urban Driving / L. Zhang // Chinese Automation Congress, 2018. – P. 1798-1803. – ISBN отсуствует.

204. Zhang, Q. Spatio-Temporal Graph Structure Learning for Traffic Forecasting / Q. Zhang // Proceedings of the AAAI Conference on Artificial Intelligence. – 2020. – № 34 (01). – P. 1177-1185. – ISSN 2374-3468.

205. Zhang, W. Impact Analysis of Rainfall on Traffic Flow Characteristics in Beijing / W. Zhang // International Journal of Intelligent Transportation Systems Research Article. – 2019. – № 1. Volume 17. – P. 150-160. – ISSN 1868-8659.

206. Zheng, C. GMAN: A Graph Multi-Attention Network for Traffic Prediction / C. Zheng // Proceedings of the AAAI Conference on Artificial Intelligence. – 2020. – № 34 (01). – P. 1234-1241. – ISSN 2374-3468.

207. Zheng, D. LEA*: An A* Variant Algorithm with Improved Edge Efficiency for Robot Motion Planning // D. Zheng, P. Tsiotras. // Computer Science. Robotics. – 2023. – № 12. – P. 1-13. – ISSN отсyтсtвyет.

208. Zheng, Y. AutoST: Efficient Neural Architecture Search for Spatio-Temporal Prediction / Y. Zheng // KDD '20: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. – 2020. – № 10. – P. 794-802. – ISSN 2154-817X.

209. Zhou, F. Reinforced Spatiotemporal Attentive Graph Neural Networks for Traffic Forecasting / F. Zhou // IEEE Internet of Things Journal. – 2020. – № 7. – P. 6414-6428. – ISSN отсyтсtвyет.

210. Zhu, J. AST-GCN: Attribute-Augmented Spatiotemporal Graph Convolutional Network for Traffic Forecasting / J. Zhu // IEEE Access. – 2021. – № 10. Volume 9. – P. 35973-35983. – ISSN 2169-3536.

211. Zietsman, D. Analytic Hierarchy Process assessment for potential multi-airport systems – The case of Cape Town / D. Zietsman, M. Vanderschuren // Journal of Air Transport Management. – 2014. – № 1. Volume 36. – P. 41-49. – ISSN 0969-6997.

212. Zografos, K. A Multicriteria Decision Making Method for Evaluating Alternative Transportation Projects / K. Zografos, R. Driggs, S. Zanakakis // IFAC Proceedings Volumes. – 1997. – № 4. Volume 30 (8). – P. 137-143. – ISSN 1474-6670.

213. Zubaryeva, A. Spatial multi-criteria assessment of potential lead markets for electrified vehicles in Europe / A. Zubaryeva, C. Thiel, N. Zaccarelli [et al.] // Transportation Research Part A: Policy and Practice. – 2012. – № 1. Volume 46 (9). – P. 1477-1489. – ISSN 0965-8564.

214. Zulkarnaen, H. Viterbi Extraction tutorial with Hidden Markov Toolkit / H. Zulkarnaen, V. Puturuhu // Computer Science. – 2019. – № 2. – P. 1-10. – ISSN отсутствует.

Приложение А

(информационное)

Применение MCDM методов в исследованиях

Таблица А.1 – Применение методов MCDM в транспортных исследованиях за 2013-2023 года

		Метод	Область применения	Цель исследования	Проблема исследования	Результат
1	2	3	4	5	6	7
1	[65]	Fuzzy-АНР	Перевалка грузов	Оценка критериев при выборе порта	В актуальных работах отсутствуют исследования по выбору порта для перевалки грузов	Оптимальным критерием выбора порта для перевалки является географическое положение
2	[66]	АНР	Транспортная система	Критерии выбора транспортной системы Дели с учетом экологической устойчивости	Проблема городского транспортом Дели	Критическими факторами являются окружающая среда, стоимость энергии, технологии, ограничения результата и адаптируемость
3	[68]	АНР	Городской транспорт	Сравнение результатов анализа затрат и выгод городского транспорта	Экономические аспекты системы городского транспорта	Зависимость от актуальной информации на дорогах
4	[69]	АНР	Терминалы и порты	Оценка конкурентоспособности терминальных и контейнерных портов	Конкуренция терминальных и контейнерных портов	Параметры эффективности работы терминала и контейнерного порта

Продолжение таблицы А.1

1	2	3	4	5	6	7
5	[70]	АНР	Эстакады и транспортные развязки	Оценка ликвидации путепроводов в Корее	Важность устранения путепроводов в Корее, которые более не являются полезными	Важнейшими критериями являются эффективность движения, безопасность, экологическая безопасность и устойчивость конструкции
6	[71]	Fuzzy-АНР	Авиаперевозки	Оценка корпоративного имиджа на рынке авиакомпаний	Необходимость внимания к корпоративному имиджу для повышения лояльности клиентов	Качество сервиса и безопасность являются важнейшими факторами на рынке авиакомпаний
7	[75]	Fuzzy-АНР	Авиаперевозки	Оценка качества услуг, предлагаемых международной авиатранспортной отраслью	В актуальных исследованиях отсутствует информация о качестве обслуживания в авиационной отрасли на основе SERVQUAL	Надежность и гарантия являются важными критериями оценки качества обслуживания в отрасли
8	[76]	АНР	Электромобили	Оценка и определение потенциального рынка электромобилей в Европе	Необходимость ранжировать производительность и пригодность различных электромобилей	Основными критериями являются плотность автомобилей, наличие инфраструктуры, ВВП на душу населения, газ, выбросы, государственные льготы и экономия средств
9	[78]	АНР	Аэропорт	Использование методов MCDM для анализа системы аэропортов	Проблема ролей и расположения аэропортовых систем	Местоположение и удобства имеют одинаковый вес

Продолжение таблицы А.1

1	2	3	4	5	6	7
10	[79]	Fuzzy-АНР	Перевалка грузов	Исследование затрат на логистику и обслуживание клиентов	Необходимость построения структуры для снижения затрат на логистику и повышения качества обслуживания	Проблемы в этом секторе можно решить, уделив особое внимание обслуживанию клиентов. Важнейший параметр – надежность выполнения заказов.
11	[80]	Fuzzy-АНР	Логистика	Поиск нового критерия выбора для тоннелепроходческого комплекса	Необходимость проведения анализа для увеличения скорости и сокращения времени	Фактор стоимости имеет решающее значение при выборе тоннелепроходческого комплекса
12	[81]	АНР	Городской транспорт	Оценка общественного автобусного транспорта с учетом мнений различных экспертов	Прозрачность процесса принятия решений, связанного с общественным транспортом, для общественности	Различия в приоритетах различных групп оценки
13	[81]	АНР-ISM	Городской транспорт	Оценка критериев качества снабжения общественным транспортом	Выявление упущенных связей в исходной иерархической структуре	Разработанная модель дает надежные результаты с учетом взаимосвязей и между критериями
14	[82]	TOPSIS и Fuzzy-АНР	Городской транспорт	Изучение измерения удовлетворенности клиентов SERVQUAL	Анализ общественного транспорта для решения множества существующих проблем	Материальные факторы и комиссия являются ключевыми факторами при оценке удовлетворенности клиентов

Продолжение таблицы А.1

1	2	3	4	5	6	7
15	[91]	Fuzzy-АНР	Городской транспорт	Оценка эффективности работы общественного транспорта	Качество услуг на общественном транспорте сопряжено с неопределенностями	Восприятие руководства влияет на общую эффективность
16	[93]	АНР	Устойчивое развитие транспорта	Оценка стратегий устойчивого развития транспорта	Транспортная система Тайваня требует стратегий устойчивого развития	Локальные обстоятельства являются важнейшим фактором
17	[96]	АНР	Автомобильный транспорт	Взаимодействие факторов транспортной системы и дорожно-транспортных происшествий	Системы дорожного движения имеют решающее значение для повышения эффективности транспортной системы	Значимым фактором является взаимодействие между участниками движения
18	[97]	АНР	Городской транспорт	Определение устойчивости проектов городского транспорта	Исследования критериев отбора проектов городского транспорта	Предлагаемая структура подходит для устойчивых транспортных потребностей
19	[98]	Fuzzy-АНР	Различные виды транспорта	Оценка преимуществ транспорта в мегаполисах Восточной Азии	Предыдущие исследования не имеют эмпирических данных	Критичными факторами являются удобство эксплуатации и инфраструктура
20	[99]	АНР	Железнодорожные перевозки	Исследование сербских железных дорог, состоящих из двухэтапных моделей	Необходимость распределения приоритетов в проектах развития после югославских войн	Приоритет отдается реструктуризации старой модели железных дорог, несмотря на плохое техническое состояние

Продолжение таблицы А.1

1	2	3	4	5	6	7
21	[101]	Fuzzy-АНР	Авиаперевозки	Исследование критериев, влияющих на выбор продавца в авиационной отрасли	Существуют противоречивые количественные критерии выбора продавца	Значимым критерием при выборе ритейлера является финансовая стабильность
22	[113]	АНР	Система аэропортов	Оценка и анализ развития систем аэропорта	Поиск факторов, влияющих на территориальную конкурентоспособность развития аэропорта	Кейптауну требуется система одного аэропорта до тех пор, пока объем пассажиров не превысит 27 миллионов в год
23	[119]	Fuzzy-АНР	Железнодорожные перевозки	Создание системы оценки удовлетворенности клиентов общественным транспортом	Проблема удовлетворенности клиентов железнодорожного транспорта	Безопасность и доступность являются ключевыми факторами в оценке удовлетворенности клиентов на железнодорожном транспорте Турции
24	[121]	Fuzzy-АНР	Авиаперевозки	Оценка и выбор грузовых альянсов в авиационной отрасли	Необходимость исследований грузовых альянсов авиационной отрасли	Важные критерии – качество маршрута и частота рейсов, увеличение дохода и увеличение загруженности
25	[123]	TOPSIS и Fuzzy-АНР	Логистика	Оценка логистики 3-го поколения и транспорта	Из-за неопределенности и сложности выбор логистического провайдера затруднен	В зависимости от предлагаемой модели выбор поставщика услуг отражает ожидания

Продолжение таблицы А.1

1	2	3	4	5	6	7
26	[125]	АНР и ELECTRE	Городской транспорт	Выбор проектов городского транспорта	Аксиоматические различия между методами	Расчетные значения расчетных проектов разнообразны
27	[126]	IPA с Fuzzy-АНР	Логистика	Оценка потребностей услуг контейнерных перевозок	Изучение управленческих последствий для улучшения качества обслуживания	Практическая значимость результатов для улучшения качества обслуживания терминалов
28	[127]	АНР и VIKOR	Морские перевозки	Поиск практического решения выбора дизайна	Исследования гибридной теоретической методологии в задачах со множеством параметров	Результат помогает судовладельцам во время постройки судна
29	[128]	АНР	Автобусные перевозки	Системы качества снабжения общественного транспорта	Изучение различий между мнением общественности и лиц, принимающих решения	Существует значительный разрыв между группами оценщиков
30	[131]	АНР, PROMETHEE, VIKOR	Аэропорт	Поиск места для военного аэропорта	Сравнение различных подходов выбора наилучшего местоположения	Используемые методы обеспечивают ту же альтернативу, что и наилучшее местоположение
31	[132]	АНР и линейное программирование	Логистика	Снижение стоимости перевозки	Необходимость достижения наилучшего удовлетворения клиентов и повышения лояльности	Оптимальное удовлетворение клиентов достигается за счет минимизации транспортных расходов

Продолжение таблицы А.1

1	2	3	4	5	6	7
32	[137]	Fuzzy-AHP и TOPSIS	Морские перевозки	Выбор эффективного типа судового погрузчика	Необходимость решения проблемы неопределенности экспортных суждений	Практические преимущества определяются при отправке сухих навалочных грузов
33	[140]	AHP и VIKOR	Морские перевозки	Выбор подходящей конструкции люковой крышки	Модель должна решать проблему неопределенности оценки экспертов	Судопроизводителям оказывается поддержка в принятии решений
34	[141]	AHP с PROMETHEE и VIKOR	Аэропорт	Поиск места для военного аэропорта	Исследование сложной проблемы выбора подходящего местоположения аэропорта	Результаты сравниваются с другими разработками
35	[143]	AHP-TOPSIS	Морские перевозки	Снижение загрязнения морского воздуха	Выделение важных параметров, препятствующих эффективному соблюдению требований при эксплуатации судов	Отсутствие инфраструктуры и всеобъемлющих законов о загрязнении морского воздуха, высокие капитальные и эксплуатационные затраты на решения по снижению содержания серы вошли являются основными показателями
36	[145]	F-AHP	Пешеходные маршруты	Оценка проблемы пешеходной доступности различных участников движения	Необходимость развития пешеходной инфраструктуры	Общественная безопасность и безопасность дорожного движения являются важнейшими критериями

Продолжение таблицы А.1

1	2	3	4	5	6	7
37	[147]	АНР и TOPSIS	Городской транспорт	Оценка нетрадиционных видов транспорта	Проблема общественного транспорта в Бангладеш	Устранение недостатков путем разработки эффективных планов обслуживания общественного транспорта
38	[148]	АНР	Железнодорожные перевозки	Требования пассажиров к качеству обслуживания высокоскоростного железнодорожного транспорта	Требования пассажиров ранжируются на основе динамического анализа	Оценка требований к инфраструктуре железных дорог
39	[150]	АНР	Морские перевозки	Оценка конкурентоспособности портов с учетом 18 факторов	Исследование работы портов и их положения в глобальной транспортной сети	Эффективная оценка обеспечивается ранжированием 99 портов в 51 стране
40	[151]	ГАНР и ELECTRE	Логистика	Оценка конкурентоспособности китайских портов на основе разных критериев	Исследования по внедрению комплексного метода оценки для улучшения стратегических планов портов	Порт Цюаньчжоу считается наименее важным
41	[154]	ГАНР	Логистика	Важность параметров кредита и дебета в логистике	Недостаточность исследований параметра купли-продажи, являющегося приоритетным для судоходных компаний	Рассчитан выгодный вариант корабля

Продолжение таблицы А.1

1	2	3	4	5	6	7
42	[155]	АНР и DEMATEL	Логистика	Процесс принятия решений выбора мер распределения товаров	Сложность реализации решений по развитию мер товародвижения	Взаимосвязь коэффициентов различных операционных показателей
43	[156]	АНР	Логистика	Оценка инноваций городских автобусных компаний, способствующих устойчивому развитию инфраструктуры	Необходимость обсуждения увеличения использования автомобилей	Эффективные решения по развитию логистики
44	[162]	АНР	Железнодорожные перевозки	План строительства железных дорог	Исследования ранжирования связанных факторов и внедрения системы индексов	Освещается наиболее важный проект, подчеркивающий экономические выгоды
45	[163]	АНР	Логистика	Анализ проблем оценки городской логистики	Вовлечение заинтересованных сторон в процесс оценки необходимо для обеспечения более эффективной городской логистики	Обеспечивается высокий уровень адаптации локальных значений
46	[165]	АНР	Логистика	Оценка лучшего маршрута для контейнеровоза	Необходимость исследований поддержки решений методами динамического анализа	Несоответствие между значимостью модели устойчивого развития и практическим применением
47	[166]	АНР	Безопасность на дорогах	Обучение водителей для снижения аварийности	Шесть различных групп должны участвовать в решении проблемы	Мотоциклисты являются самой низкообразованной группой, а пешеходы имеют самый высокий балл образования

Продолжение таблицы А.1

1	2	3	4	5	6	7
48	[178]	FAHP	Логистика	Оценка значимого фактора, связанного с экологической политикой	Необходимо проводить не только количественные, но и качественные интервью	Экологическая политика является важным фактором, а качественный анализ улучшает результат за счет привлечения заинтересованных сторон
49	[179]	AHP	Грузовой транспорт	Перечисление городских портов для усиления взаимосвязи между различными факторами	Определение оптимальной конструкции между факторами	Определены значимые факторы, связанные с улучшением городской инфраструктуры заинтересованных сторон
50	[182]	AHP-ISM	Городской транспорт	Оценка снабжения общественным транспортом	Отсутствие взаимосвязи между факторами разных уровней иерархии	Устойчивые решения проблем общественного транспорта с вовлечением пассажиров в процесс принятия решений с учетом оптимальных пересадок
51	[183]	AHP и Dempster	Транспортное планирование	оценка воздействия интеллектуальных транспортных решений, мультимодальных транспортных решений для совместного использования различного транспорта	Поиск основы объединения данных, поступающих из нескольких источников информации для оценки устойчивости транспортных мер	Выбор критериев оценки, сбор данных и объединение информации, оценка устойчивости города с использованием индекса устойчивости

Продолжение таблицы А.1

1	2	3	4	5	6	7
52	[185]	АНР	Транспортное планирование	Поддержка процесса вовлечения общественности в процесс планирования перевозок	Вовлечение общественности в этап стратегического планирования	Разработана модель для привлечения общественности к стратегическому планированию
53	[186]	АНР и ELECTRE	Городская трамвайная система	Оценка критериев модернизации трамвайных путей в Польше	Необходимость исследования для улучшения сети общественного транспорта в Польше из-за несбалансированности модального разделения, доминирующим компонентом является автобусный транспорт	Многокритериальная оценка вариантов модернизации городской трамвайной системы
54	[187]	АНР-TOPSIS	Качество сервиса	Улучшение качества автобусных перевозок	Необходимость предоставления операторам автобусного транспорта актуальной информации для мониторинга и улучшения качества услуг автобусного транспорта	Студентов в первую очередь волнует удобство пользования общественным транспортом и меньше всего – комфорт
55	[188]	АНР	Экологичный транспорт	Оценка планирования городского экологичного транспорта	Комплексная оценка планирования городского экологичного транспорта	Комбинированный метод дает удовлетворительные результаты оценки и выявляет недостатки планирования

Продолжение таблицы А.1

1	2	3	4	5	6	7
56	[191]	АНР	Транспортное планирование	Исследование проблемы принятия транспортных решений и их оценка	Привлечение большего количества заинтересованных сторон к принятию решений транспортного планирования	Лучшее решение транспортной системы представлено из четырех возможных альтернатив
57	[192]	АНР	Безопасность на дорогах	Оценка эффективности мероприятий по обеспечению готовности к авариям автодорожных туннелей	Необходимость системы поддержки принятия решений для оценки системы реагирования на чрезвычайные ситуации	Оценщик может быть элементом системы оценки рисков вместо внешних организаций
58	[193]	АНР	Городской транспорт	Поиск увеличения государственных расходов и снижения дефицита доходов	Трудности социально-экономической ситуации, дефицит доходов и увеличение государственных расходов в Тунисе	Правила наземного транспорта требуют значительного внимания государства, например, компенсации потери дохода из-за сниженных тарифов

Источник: составлено автором на основе материалов [65-66; 68-71; 75-76; 78-82; 91; 93; 96-99; 101; 113; 119; 121; 123, 125-128; 131-132; 137; 140-141; 143; 145; 147-148; 150-151; 154-156; 162-163; 165-166; 178-179; 182; 183; 185-188; 191-193].

Приложение Б
(информационное)

Сравнение методов и моделей в транспортных исследованиях

Таблица Б.1 – Сравнительный анализ исследований

Источник	Модель	Метрика оценки				
		ACC	MSE	MAE	MAPE	RMSE
[108]	LSTM	94,6	–	–	–	–
[165]	SLC-CNN	–	–	2,22	5,21	4,07
		–	1,44	3,0	2,9	–
		–	2,53	6,7	5,18	–
[169]	RSTAG	–	–	1,22	2,62	2,65
[69]	MRA-BGCN	–	–	1,29	2,90	2,72
[131]	STSGCN	–	–	17,13	10,96	26,86
[170]	AST-GCN	71,93	–	2,7035	–	4,029
[167]	GMAN	–	–	1,34	2,81	2,82
[102]	ST-TrafficNet	–	–	1,26	2,68	2,72
[93]	LSTM & Bi-LSTM	–	–	–	–	6,53
	LSTM	–	–	–	9,31	–
	Bi-LSTM	–	–	–	6,53	–
	SVM	–	–	–	12,39	–
	ANN	–	–	–	15,46	–
[146]	Autoencoder	–	–	16,15	–	26,32
[139]	SVR	–	–	0,469	–	1,024
[98]	Conv+	–	–	22,94	–	336,29
[145]	LSTM	–	13,92	–	–	–
[147]	CNN-RNN	–	–	0,07	19,12	27,918
[78]	LSTM-GRU	–	–	710,05	18,127	–

Источник: составлено автором на основе материалов [69; 78; 93; 98; 102; 108; 131; 139; 145-147; 165; 167; 169-170].

Приложение В

(информационное)

Оценка загруженности дорожного сегмента на Golang

```

1 type ScoreEstimator struct {
2     _MIN_SCORE          int
3     _MAX_SCORE          int
4     _AVG_SPEED_LOWERING_FACTOR float64
5     _MIN_LIVE_SPEED     float64
6     _max_pct_no_live_speeds float64
7     _min_ratio          float64
8     _max_ratio          float64
9     _weighted_speed_ratio float64
10    _no_live_length      float64
11    _live_length         float64
12 }
13
14 func NewScoreEstimator(max_pct_no_live_speeds, min_ratio, max_ratio
15 float64) *ScoreEstimator {
16     return &ScoreEstimator{
17         _MIN_SCORE:          0,
18         _MAX_SCORE:          10,
19         _AVG_SPEED_LOWERING_FACTOR: 0.85,
20         _MIN_LIVE_SPEED:     0.5,
21         _max_pct_no_live_speeds: max_pct_no_live_speeds,
22         _min_ratio:          min_ratio,
23         _max_ratio:          max_ratio,
24     }
25 }
26
27 func (s *ScoreEstimator) AddEdge(avg_speed, live_speed float64, length
28 float64) {
29     if live_speed == 0 {
30         s._no_live_length += length
31         return
32     }
33
34     avg_speed *= s._AVG_SPEED_LOWERING_FACTOR
35     live_speed = s.normalize(live_speed, s._MIN_LIVE_SPEED, avg_speed)
36
37     s._weighted_speed_ratio += length * live_speed / avg_speed
38     s._live_length += length
39 }
40
41 func (s *ScoreEstimator) TrafficScore() *int {
42     if s._live_length == 0 {
43         return nil
44     }
45
46     no_live_pct := s._no_live_length * 100.0 / s._live_length
47     if no_live_pct > s._max_pct_no_live_speeds {
48         return nil
49     }
50
51     ratio := s._weighted_speed_ratio / s._live_length
52     score := s.linearScale(ratio)
53

```

```

54     normalized_score := int(round(s.normalize(score,
55 float64(s._MIN_SCORE), float64(s._MAX_SCORE))))
56     return &normalized_score
57 }
58
59 func (s *ScoreEstimator) linearScale(val float64) float64 {
60     return float64(s._MAX_SCORE) - (val-s._min_ratio)/(s._max_ratio-
61 s._min_ratio)*(float64(s._MAX_SCORE)-float64(s._MIN_SCORE))
62 }
63
64 func (s *ScoreEstimator) normalize(val, min_val, max_val float64) float64
65 {
66     if val < min_val {
67         return min_val
68     }
69     if val > max_val {
70         return max_val
71     }
72     return val
73 }

```

Приложение Г

(информационное)

Поиск альтернативных маршрутов двухсторонним A* на Golang

```

1 package main
2
3 import (
4     "fmt"
5     "math"
6     "sync"
7 )
8
9 type AStar interface {
10     FillTile(p Point, weight int)
11     ClearTile(p Point)
12     FindPath(config AStarConfig, source, target []Point) *PathPoint
13 }
14
15 type AStarConfig interface {
16     IsEnd(p Point, end []Point, end_map map[Point]bool) bool
17     SetWeight(p *PathPoint, fill_weight int, end []Point, end_map
18 map[Point]bool) (allowed bool)
19     PostProcess(p *PathPoint, rows, cols int, filledTiles map[Point]int)
20 (*PathPoint)
21 }
22
23 type gridStruct struct {
24     tileLock    sync.Mutex
25     filledTiles map[Point]int
26
27     rows int
28     cols int
29 }
30
31 func NewAStar(rows, cols int) AStar {
32     return &gridStruct{
33         rows: rows,
34         cols: cols,
35
36         filledTiles: make(map[Point]int),
37     }
38 }
39
40 func (a *gridStruct) FillTile(p Point, weight int) {
41     a.tileLock.Lock()
42     defer a.tileLock.Unlock()
43
44     a.filledTiles[p] = weight
45 }
46
47 func (a *gridStruct) ClearTile(p Point) {
48     a.tileLock.Lock()
49     defer a.tileLock.Unlock()
50
51     delete(a.filledTiles, p)
52 }
53

```

```

54 func (a *gridStruct) FindPath(config AStarConfig, source, target []Point)
55 *PathPoint {
56     var openList = make(map[Point]*PathPoint)
57     var closeList = make(map[Point]*PathPoint)
58
59     source_map := make(map[Point]bool)
60     for _, p := range source {
61         source_map[p] = true
62     }
63
64     a.tileLock.Lock()
65     for _, p := range target {
66         fill_weight := a.filledTiles[p]
67         path_point := &PathPoint{
68             Point:      p,
69             Parent:      nil,
70             DistTraveled: 0,
71             FillWeight:   fill_weight,
72         }
73
74         allowed := config.SetWeight(path_point, fill_weight, source,
75 source_map)
76         if allowed {
77             openList[p] = path_point
78         }
79     }
80
81     a.tileLock.Unlock()
82
83     var current *PathPoint
84     for {
85         current = a.getMinWeight(openList)
86
87         a.tileLock.Lock()
88         if current == nil || config.IsEnd(current.Point, source,
89 source_map) {
90             a.tileLock.Unlock()
91             break
92         }
93         a.tileLock.Unlock()
94
95         delete(openList, current.Point)
96         closeList[current.Point] = current
97
98         surrounding := a.getSurrounding(current.Point)
99
100        for _, p := range surrounding {
101            _, ok := closeList[p]
102            if ok {
103                continue
104            }
105
106            a.tileLock.Lock()
107            fill_weight := a.filledTiles[p]
108            a.tileLock.Unlock()
109
110            path_point := &PathPoint{
111                Point:      p,
112                Parent:      current,

```

```

113             FillWeight:  current.FillWeight + fill_weight,
114             DistTraveled: current.DistTraveled + 1,
115         }
116
117         a.tileLock.Lock()
118         allowed := config.SetWeight(path_point, fill_weight, source,
119 source_map)
120         a.tileLock.Unlock()
121
122         if !allowed {
123             continue
124         }
125
126         existing_point, ok := openList[p]
127         if !ok {
128             openList[p] = path_point
129         } else {
130             if path_point.Weight < existing_point.Weight {
131                 existing_point.Parent = path_point.Parent
132             }
133         }
134     }
135 }
136
137 a.tileLock.Lock()
138 current = config.PostProcess(current, a.rows, a.cols, a.filledTiles)
139 a.tileLock.Unlock()
140
141 return current
142 }
143
144 func (a *gridStruct) getMinWeight(openList map[Point]*PathPoint)
145 *PathPoint {
146     var min *PathPoint = nil
147     var minWeight int = 0
148
149     for _, p := range openList {
150         if min == nil || p.Weight < minWeight {
151             min = p
152             minWeight = p.Weight
153         }
154     }
155     return min
156 }
157
158 func (a *gridStruct) getSurrounding(p Point) []Point {
159     var surrounding []Point
160
161     row, col := p.Row, p.Col
162
163     if row > 0 {
164         surrounding = append(surrounding, Point{row - 1, col})
165     }
166     if row < a.rows-1 {
167         surrounding = append(surrounding, Point{row + 1, col})
168     }
169
170     if col > 0 {
171         surrounding = append(surrounding, Point{row, col - 1})

```

```

172     }
173     if col < a.cols-1 {
174         surrounding = append(surrounding, Point{row, col + 1})
175     }
176
177     return surrounding
178 }
179
180 type Point struct {
181     Row int
182     Col int
183 }
184
185 type PathPoint struct {
186     Point
187     Parent *PathPoint
188
189     Weight      int
190     FillWeight  int
191     DistTraveled int
192
193     WeightData interface{}
194 }
195
196 func (p Point) Dist(other Point) int {
197     return int(math.Abs(float64(p.Row-other.Row)) +
198 math.Abs(float64(p.Col-other.Col)))
199 }
200
201 var _ = fmt.Sprint()
202
203 type pointToPoint struct {
204     VoidPostProcess
205 }
206
207 func NewPointToPoint() AStarConfig {
208     p2p := &pointToPoint{}
209
210     return p2p
211 }
212
213 func (p2p *pointToPoint) SetWeight(p *PathPoint, fill_weight int, end
214 []Point, end_map map[Point]bool) bool {
215     if len(end) != 1 {
216         panic("Invalid end specified")
217     }
218
219     if fill_weight == -1 {
220         return false
221     }
222
223     p.Weight = p.FillWeight + p.DistTraveled + p.Point.Dist(end[0])
224
225     return true
226 }
227
228 func (p2p *pointToPoint) IsEnd(p Point, end []Point, end_map
229 map[Point]bool) bool {
230     if len(end) != 1 {

```

```

231         panic("Invalid end specified")
232     }
233     return p == end[0]
234 }
235
236 type rowToRow struct {
237     VoidPostProcess
238 }
239
240 func NewRowToRow() AStarConfig {
241     r2r := &rowToRow{}
242     return r2r
243 }
244
245 func (r2r *rowToRow) SetWeight(p *PathPoint, fill_weight int, end
246 []Point, end_map map[Point]bool) bool {
247     if len(end) != 1 {
248         panic("Invalid end specified")
249     }
250
251     if fill_weight == -1 {
252         return false
253     }
254
255     p.Weight = p.FillWeight + p.DistTraveled +
256     int(math.Abs(float64(p.Row-end[0].Row)))
257
258     return true
259 }
260
261 func (r2r *rowToRow) IsEnd(p Point, end []Point, end_map map[Point]bool)
262 bool {
263     if len(end) != 1 {
264         panic("Invalid end specified")
265     }
266     return p.Row == end[0].Row
267 }
268
269 type listToPoint struct {
270 }
271
272 type listToPointForward struct {
273     listToPoint
274     VoidPostProcess
275 }
276
277 type listToPointReverse struct {
278     listToPoint
279     ReversePostProcess
280 }
281
282 func NewListToPoint(reverse bool) AStarConfig {
283
284     if reverse {
285         return &listToPointReverse{}
286     } else {
287         return &listToPointForward{}
288     }
289 }

```

```

290
291 func (p2l *listToPoint) SetWeight(p *PathPoint, fill_weight int, end
292 []Point, end_map map[Point]bool) bool {
293     if fill_weight == -1 {
294         return false
295     }
296
297     path_length := len(end)
298
299     min_dist := -1
300     for i, end_p := range end {
301         dist := p.Point.Dist(end_p) + (path_length - i)
302         if min_dist == -1 || dist < min_dist {
303             min_dist = dist
304         }
305     }
306
307     p.Weight = p.FillWeight + p.DistTraveled + min_dist
308
309     return true
310 }
311
312 func (p2l *listToPoint) IsEnd(p Point, end []Point, end_map
313 map[Point]bool) bool {
314     return end_map[p]
315 }
316
317 type VoidPostProcess struct {
318 }
319
320 func (v *VoidPostProcess) PostProcess(p *PathPoint, rows, cols int,
321 filledTiles map[Point]int) (*PathPoint) {
322     return p
323 }
324
325 type ReversePostProcess struct {
326 }
327
328 func (v *ReversePostProcess) PostProcess(p *PathPoint, rows, cols int,
329 filledTiles map[Point]int) (*PathPoint) {
330     var path_prev *PathPoint = nil
331
332     for p != nil {
333         next := p.Parent
334         p.Parent = path_prev
335
336         path_prev = p
337         p = next
338     }
339
340     return path_prev
341 }

```

Приложение Д

(информационное)

Расчет веса параметров методом АНР на Golang

```

1 package main
2
3 import (
4     "fmt"
5     "math"
6     "sort"
7 )
8
9 type Compare struct {
10     Name          string
11     Comparisons   map[string]float64
12     Precision     int
13     RandomIndex   string
14     Iterations    int
15     Tolerance     float64
16     CR            bool
17     Normalize     bool
18     Elements      []string
19     Pairs         map[string]float64
20     Size          int
21     Matrix        [][]float64
22     MissingComparisons map[string]int
23     NodeParent    *Compare
24     NodeChildren  []*Compare
25     NodePrecision int
26     NodeWeights   map[string]float64
27     GlobalWeight  float64
28     LocalWeight   float64
29     ConsistencyRatio float64
30     GlobalWeights  map[string]float64
31     LocalWeights  map[string]float64
32     TargetWeights  map[string]float64
33 }
34
35 type Compose struct {
36     Nodes      []*Compare
37     Hierarchy  map[string][]string
38 }
39
40 func NewCompare(name string, comparisons map[string]float64, precision
41 int, randomIndex string, iterations int, tolerance float64, cr bool)
42 *Compare {
43     comp := &Compare{
44         Name:          name,
45         Comparisons:   comparisons,
46         Precision:     precision,
47         RandomIndex:   randomIndex,
48         Iterations:    iterations,
49         Tolerance:     tolerance,
50         CR:            cr,
51         Normalize:     false,
52         Elements:      nil,
53         Pairs:         nil,

```

```

54         Size:                0,
55         Matrix:              nil,
56         MissingComparisons:  make(map[[2]string]int),
57         NodeParent:          nil,
58         NodeChildren:        nil,
59         NodePrecision:       precision,
60         NodeWeights:         nil,
61         GlobalWeight:        1.0,
62         LocalWeight:         1.0,
63         ConsistencyRatio:    0.0,
64         GlobalWeights:       nil,
65         LocalWeights:        nil,
66         TargetWeights:       nil,
67     }
68     comp.init()
69     return comp
70 }
71
72 func NewCompose() *Compose {
73     return &Compose{
74         Nodes:    make([]*Compare, 0),
75         Hierarchy: nil,
76     }
77 }
78
79 func (comp *Compare) init() {
80     if comp.Normalize {
81         comp.buildNormalizedElements()
82         comp.checkSize()
83         comp.buildNormalizedMatrix()
84     } else {
85         comp.buildElements()
86         comp.checkSize()
87         comp.insertComparisons()
88         comp.buildMatrix()
89     }
90     comp.getMissingComparisons()
91     if comp.MissingComparisons != nil {
92         comp.completeMatrix()
93     }
94     comp.compute()
95 }
96
97 func (comp *Compare) buildElements() {
98     elements := make([]string, 0)
99     for key := range comp.Comparisons {
100         for _, element := range key {
101             if !contains(elements, element) {
102                 elements = append(elements, element)
103             }
104         }
105     }
106     sort.Strings(elements)
107     comp.Elements = elements
108     comp.Pairs = make(map[[2]string]float64)
109     comp.Size = len(elements)
110 }
111
112 func (comp *Compare) buildNormalizedElements() {

```

```

113     elements := make([]string, 0, len(comp.Comparisons))
114     for key := range comp.Comparisons {
115         elements = append(elements, key)
116     }
117     comp.Elements = elements
118     comp.Pairs = make(map[[2]string]float64)
119     comp.Size = len(elements)
120 }
121
122 func (comp *Compare) checkSize() {
123     if !comp.Normalize && comp.CR && ((comp.RandomIndex == "saaty" &&
124 comp.Size > 15) || comp.Size > 100) {
125         msg := fmt.Sprintf("The input matrix is too large and a
126 consistency ratio cannot be computed.")
127         panic(msg)
128     }
129 }
130
131 func (comp *Compare) insertComparisons() {
132     for key, value := range comp.Comparisons {
133         inverseKey := reverseKey(key)
134         comp.Pairs[key] = value
135         comp.Pairs[inverseKey] = 1.0 / value
136     }
137 }
138
139 func (comp *Compare) buildMatrix() {
140     comp.Matrix = make([][]float64, comp.Size)
141     for i := range comp.Matrix {
142         comp.Matrix[i] = make([]float64, comp.Size)
143     }
144     for pair, value := range comp.Pairs {
145         location := [2]int{findIndex(comp.Elements, pair[0]),
146 findIndex(comp.Elements, pair[1])}
147         comp.Matrix[location[0]][location[1]] = value
148     }
149 }
150
151 func (comp *Compare) buildNormalizedMatrix() {
152     comp.Matrix = make([][]float64, comp.Size)
153     for i := range comp.Matrix {
154         comp.Matrix[i] = make([]float64, comp.Size)
155     }
156     for i, value := range comp.Comparisons {
157         comp.Matrix[i][0] = value
158     }
159 }
160
161 func (comp *Compare) getMissingComparisons() {
162     missingComparisons := make(map[[2]string]int)
163     for pair, value := range comp.Pairs {
164         if value == 0 {
165             missingComparisons[pair] = 1
166         }
167     }
168     comp.MissingComparisons = missingComparisons
169 }
170
171 func (comp *Compare) completeMatrix() {

```

```

172     lastIteration := make([]float64, 0, len(comp.MissingComparisons))
173     for _, value := range comp.MissingComparisons {
174         lastIteration = append(lastIteration, float64(value))
175     }
176     difference := math.Inf(1)
177     for difference > comp.Tolerance {
178         comp.minimizeCoordinateValues()
179         currentIteration := make([]float64, 0,
180 len(comp.MissingComparisons))
181         for _, value := range comp.MissingComparisons {
182             currentIteration = append(currentIteration,
183 float64(value))
184         }
185         difference = vectorNorm(vectorSubtract(lastIteration,
186 currentIteration))
187         lastIteration = currentIteration
188     }
189 }
190
191 func (comp *Compare) minimizeCoordinateValues() {
192     lambdaMax := comp.maxEigenValue(comp.Matrix)
193     upperBound := comp.nanMax(comp.Matrix) * 10
194     for comparison := range comp.MissingComparisons {
195         optimalSolution := comp.minimizeScalar(comparison,
196 upperBound)
197         comp.MissingComparisons[comparison] = int(optimalSolution)
198     }
199 }
200
201 func (comp *Compare) maxEigenValue(matrix [][]float64) float64 {
202     eigenValues := make([]complex128, 0, len(matrix))
203     for i := range matrix {
204         for j := range matrix[i] {
205             matrix[i][j] = 1.0 / matrix[i][j]
206         }
207     }
208     return math.Max(eigenValues...)
209 }
210
211 func (comp *Compare) minimizeScalar(comparison [2]string, upperBound
212 float64) float64 {
213     return 0.0
214 }
215
216 func vectorNorm(vector []float64) float64 {
217     sum := 0.0
218     for _, v := range vector {
219         sum += v * v
220     }
221     return math.Sqrt(sum)
222 }
223
224 func vectorSubtract(vector1, vector2 []float64) []float64 {
225     result := make([]float64, len(vector1))
226     for i := range vector1 {
227         result[i] = vector1[i] - vector2[i]
228     }
229     return result
230 }

```

```

231
232 func contains(slice []string, str string) bool {
233     for _, s := range slice {
234         if s == str {
235             return true
236         }
237     }
238     return false
239 }
240
241 func reverseKey(key [2]string) [2]string {
242     return [2]string{key[1], key[0]}
243 }
244
245 func findIndex(slice []string, element string) int {
246     for i, v := range slice {
247         if v == element {
248             return i
249         }
250     }
251     return -1
252 }
253
254 func (comp *Compare) nanMax(matrix [][]float64) float64 {
255     max := math.Inf(-1)
256     for _, row := range matrix {
257         for _, val := range row {
258             if val > max && !math.IsNaN(val) {
259                 max = val
260             }
261         }
262     }
263     return max
264 }

```

Приложение Е

(информационное)

Расчет альтернатив техникой TOPSIS на Golang

```

1 package main
2
3 import (
4     "fmt"
5     "math"
6 )
7
8 type Topsis struct {
9     EvaluationMatrix [][]float64
10    WeightedNormalized [][]float64
11    NormalizedDecision [][]float64
12    M int
13    N int
14    WeightMatrix []float64
15    Criteria []float64
16 }
17
18 func NewTopsis(evaluationMatrix [][]float64, weightMatrix []float64,
19 criteria []float64) *Topsis {
20     t := &Topsis{
21         EvaluationMatrix: evaluationMatrix,
22         M: len(evaluationMatrix),
23         N: len(evaluationMatrix[0]),
24         WeightMatrix: weightMatrix,
25         Criteria: criteria,
26     }
27
28     totalWeight := 0.0
29     for _, weight := range weightMatrix {
30         totalWeight += weight
31     }
32     for i, _ := range weightMatrix {
33         t.WeightMatrix[i] /= totalWeight
34     }
35
36     return t
37 }
38
39 func (t *Topsis) Step2() {
40     t.NormalizedDecision = make([][]float64, t.M)
41     squaredSum := make([]float64, t.N)
42
43     for i := 0; i < t.M; i++ {
44         t.NormalizedDecision[i] = make([]float64, t.N)
45         for j := 0; j < t.N; j++ {
46             squaredSum[j] +=
47 math.Pow(t.EvaluationMatrix[i][j], 2)
48         }
49     }
50
51     for i := 0; i < t.M; i++ {
52         for j := 0; j < t.N; j++ {
53             t.NormalizedDecision[i][j] =
54 t.EvaluationMatrix[i][j] / math.Sqrt(squaredSum[j])
55         }
56     }
57 }

```

```

58
59 func (t *Topsis) Step3() {
60     t.WeightedNormalized = make([][]float64, t.M)
61
62     for i := 0; i < t.M; i++ {
63         t.WeightedNormalized[i] = make([]float64, t.N)
64         for j := 0; j < t.N; j++ {
65             t.WeightedNormalized[i][j] =
66 t.NormalizedDecision[i][j] * t.WeightMatrix[j]
67         }
68     }
69 }
70
71 func (t *Topsis) Step4() {
72     t.WorstAlternatives := make([]float64, t.N)
73     t.BestAlternatives := make([]float64, t.N)
74
75     for i := 0; i < t.N; i++ {
76         if t.Criteria[i] {
77             minVal := t.WeightedNormalized[0][i]
78             maxVal := t.WeightedNormalized[0][i]
79             for j := 1; j < t.M; j++ {
80                 if t.WeightedNormalized[j][i] < minVal {
81                     minVal = t.WeightedNormalized[j][i]
82                 }
83                 if t.WeightedNormalized[j][i] > maxVal {
84                     maxVal = t.WeightedNormalized[j][i]
85                 }
86             }
87             t.WorstAlternatives[i] = minVal
88             t.BestAlternatives[i] = maxVal
89         } else {
90             maxVal := t.WeightedNormalized[0][i]
91             minVal := t.WeightedNormalized[0][i]
92             for j := 1; j < t.M; j++ {
93                 if t.WeightedNormalized[j][i] > maxVal {
94                     maxVal = t.WeightedNormalized[j][i]
95                 }
96                 if t.WeightedNormalized[j][i] < minVal {
97                     minVal = t.WeightedNormalized[j][i]
98                 }
99             }
100             t.WorstAlternatives[i] = maxVal
101             t.BestAlternatives[i] = minVal
102         }
103     }
104 }
105
106 func (t *Topsis) Step5() {
107     t.WorstDistance := make([]float64, t.M)
108     t.BestDistance := make([]float64, t.M)
109     t.WorstDistanceMat := make([][]float64, t.M)
110     t.BestDistanceMat := make([][]float64, t.M)
111
112     for i := 0; i < t.M; i++ {
113         t.WorstDistanceMat[i] = make([]float64, t.N)
114         t.BestDistanceMat[i] = make([]float64, t.N)
115         for j := 0; j < t.N; j++ {
116             t.WorstDistanceMat[i][j] =
117 math.Pow(t.WeightedNormalized[i][j]-t.WorstAlternatives[j], 2)
118             t.BestDistanceMat[i][j] =
119 math.Pow(t.WeightedNormalized[i][j]-t.BestAlternatives[j], 2)
120

```

```

121             t.WorstDistance[i] += t.WorstDistanceMat[i][j]
122             t.BestDistance[i] += t.BestDistanceMat[i][j]
123         }
124
125         t.WorstDistance[i] = math.Sqrt(t.WorstDistance[i])
126         t.BestDistance[i] = math.Sqrt(t.BestDistance[i])
127     }
128 }
129
130 func (t *Topsis) Step6() {
131     t.WorstSimilarity := make([]float64, t.M)
132     t.BestSimilarity := make([]float64, t.M)
133
134     for i := 0; i < t.M; i++ {
135         t.WorstSimilarity[i] = t.WorstDistance[i] /
136 (t.WorstDistance[i] + t.BestDistance[i])
137         t.BestSimilarity[i] = t.BestDistance[i] /
138 (t.WorstDistance[i] + t.BestDistance[i])
139     }
140 }
141
142 func (t *Topsis) Ranking(data []float64) []int {
143     rank := make([]int, len(data))
144     sortedIndices := make([]int, len(data))
145     for i := range sortedIndices {
146         sortedIndices[i] = i
147     }
148     sort.Slice(sortedIndices, func(i, j int) bool { return data[i] <
149 data[j] })
150     for i, index := range sortedIndices {
151         rank[index] = i + 1
152     }
153     return rank
154 }
155
156 func (t *Topsis) RankToWorstSimilarity() []int {
157     return t.Ranking(t.WorstSimilarity)
158 }
159
160 func (t *Topsis) RankToBestSimilarity() []int {
161     return t.Ranking(t.BestSimilarity)
162 }
163
164 func (t *Topsis) Calc() {
165     fmt.Println("Step 1\n", t.EvaluationMatrix, "\n")
166     t.Step2()
167     fmt.Println("Step 2\n", t.NormalizedDecision, "\n")
168     t.Step3()
169     fmt.Println("Step 3\n", t.WeightedNormalized, "\n")
170     t.Step4()
171     fmt.Println("Step 4\n", t.WorstAlternatives, t.BestAlternatives,
172 "\n")
173     t.Step5()
174     fmt.Println("Step 5\n", t.WorstDistance, t.BestDistance, "\n")
175     t.Step6()
176     fmt.Println("Step 6\n", t.WorstSimilarity, t.BestSimilarity, "\n")
177 }
178

```

Приложение Ж

(информационное)

Расчет коэффициентов влияния когнитивной модели на Golang

```

1 package main
2
3 import (
4     "encoding/json"
5     "fmt"
6     "io/ioutil"
7     "math/cmplx"
8     "os"
9
10    "gonum.org/v1/gonum/mat"
11 )
12
13 func main() {
14     var h map[string]interface{}
15     data, _ := ioutil.ReadAll(os.Stdin)
16     json.Unmarshal(data, &h)
17
18     influence := make(map[string]map[string]float64)
19     isTrace := h["is_trace"].(float64) == 1
20
21     edges := h["edges"].([]interface{})
22     for _, e := range edges {
23         edge := e.(map[string]interface{})
24         influences := edge["influence"].(map[string]interface{})
25         for toFactor, hpwc := range influences {
26             if isTrace {
27                 fmt.Printf("to_factor = %s from frame =
28 %v\n", toFactor, edge["from"])
29             }
30             hpwcMap := hpwc.(map[string]interface{})
31             rn := make([]string, 0, len(hpwcMap))
32             for k := range hpwcMap {
33                 rn = append(rn, k)
34             }
35             size := len(rn)
36             pwc := mat.NewDense(size, size, nil)
37             for i := 0; i < size; i++ {
38                 pwc.Set(i, i, 1.0) // Identity matrix
39             }
40
41             var v []float64
42             if _, ok :=
43 hpwcMap[rn[0]].(map[string]interface{}); ok {
44                 for key1, value1 := range hpwcMap {
45                     for key2, value2 := range
46 value1.(map[string]interface{}) {
47                         i := index(rn, key1)
48                         j := index(rn, key2)
49                         val := value2.(float64)
50                         pwc.Set(i, j, val)
51                         pwc.Set(j, i, 1.0/val)
52                     }
53                 }

```

```

54         if isTrace {
55             printMatrix(pwc, 2)
56         }
57         eigen := mat.EigenSym(pwc, true)
58         eigenvalues := eigen.Values(nil)
59         eigenvectors := eigen.VectorsTo(nil)
60
61         maxI := 0
62         maxV := 0.0
63         for i, value := range eigenvalues {
64             if !cmplx.IsNaN(complex(value, 0))
65 && value > maxV {
66                 maxV = value
67                 maxI = i
68             }
69         }
70         if isTrace {
71             fmt.Printf("max eigenvalue =
72 %.2f\n", maxV)
73         }
74         v =
75 eigenvectors.ColView(maxI).RawVector().Data
76         if isTrace {
77             printVector(v, 2)
78         }
79         sum := 0.0
80         for _, x := range v {
81             sum += x
82         }
83         for i := range v {
84             v[i] /= sum
85         }
86     } else {
87         v = make([]float64, size)
88         for i, key := range rn {
89             v[i] = hpwcMap[key].(float64)
90         }
91         if isTrace {
92             printVector(v, 2)
93         }
94     }
95
96     for i, x := range v {
97         if influence[rn[i]] == nil {
98             influence[rn[i]] =
99 make(map[string]float64)
100         }
101         influence[rn[i]][toFactor] = x
102     }
103 }
104 }
105
106 rn := []string{}
107 frames := h["frames"].([]interface{})
108 for _, f := range frames {
109     frame := f.(map[string]interface{})
110     slots := frame["slots"].([]interface{})
111     for _, sl := range slots {
112         slot := sl.(map[string]interface{})

```

```

113         rn = append(rn, slot["code"].(string))
114     }
115 }
116
117 n := len(rn)
118 superMatrix := mat.NewDense(n, n, nil)
119 if isTrace {
120     fmt.Println("A = ")
121 }
122 for i, n1 := range rn {
123     for j, n2 := range rn {
124         if val, ok := influence[n1][n2]; ok {
125             superMatrix.Set(i, j, val)
126         }
127     }
128 }
129 if isTrace {
130     printMatrix(superMatrix, 2)
131 }
132
133 norm := columnSum(superMatrix)
134 dnorm := mat.NewDiagDense(n, norm)
135 superMatrix.DivElem(superMatrix, dnorm)
136
137 i := 0
138 eps := 1.0
139 currMatrix := mat.DenseCopyOf(superMatrix)
140 for {
141     if isTrace {
142         fmt.Printf("h = %d\n", i)
143         fmt.Println("A = ")
144         printMatrix(currMatrix, 2)
145     }
146     perMatrix := mat.DenseCopyOf(currMatrix)
147     currMatrix.Product(currMatrix, superMatrix)
148     norm = columnSum(currMatrix)
149     dnorm = mat.NewDiagDense(n, norm)
150     currMatrix.DivElem(currMatrix, dnorm)
151     delta := mat.NewDense(n, n, nil)
152     delta.Sub(perMatrix, currMatrix)
153     eps = mat.Max(delta)
154     if eps < 0.1e-10 || i > 100 {
155         break
156     }
157     i++
158 }
159
160 hOut := make(map[string]interface{})
161 for i, s := range rn {
162     hOut[s] = currMatrix.At(i, 0)
163 }
164 hOut["epsilon"] = eps
165
166 output, _ := json.MarshalIndent(hOut, "", " ")
167 fmt.Println(string(output))
168 }
169
170 func index(slice []string, value string) int {
171     for i, v := range slice {

```

```

172             if v == value {
173                 return i
174             }
175         }
176         return -1
177 }
178
179 func printMatrix(m *mat.Dense, prec int) {
180     r, c := m.Dims()
181     for i := 0; i < r; i++ {
182         for j := 0; j < c; j++ {
183             fmt.Printf("%.*f\t", prec, m.At(i, j))
184         }
185         fmt.Println()
186     }
187 }
188
189 func printVector(v []float64, prec int) {
190     for _, x := range v {
191         fmt.Printf("%.*f\t", prec, x)
192     }
193     fmt.Println()
194 }
195
196 func columnSum(m *mat.Dense) []float64 {
197     r, c := m.Dims()
198     sums := make([]float64, c)
199     for j := 0; j < c; j++ {
200         for i := 0; i < r; i++ {
201             sums[j] += m.At(i, j)
202         }
203     }
204     return sums
205 }

```

Приложение И

(информационное)

Комбинационная матрица парных сравнений подкритериев «Безопасность»

Таблица И.1 – Нормализованная матрица подкритериев безопасности

a _{ij}	c ₁	c ₂	c ₃	c ₄	c ₅	c ₆	c ₇	c ₈	c ₉	c ₁₀	c ₁₁	c ₁₂
c ₁	1,4	2,0	3,0	4,0	0,25	3,0	0,5	0,25	2,0	3,0	2,0	0,2
c ₂	0,5	1,4	2,0	3,0	0,25	0,5	0,33	0,2	0,33	0,5	0,5	0,16
c ₃	0,33	0,5	1,4	2,0	0,2	0,5	0,25	0,16	0,25	0,33	0,5	0,16
c ₄	0,25	0,33	0,5	1,4	0,16	0,5	0,2	0,14	0,2	0,25	0,33	0,125
c ₅	4,0	4,0	5,0	6,0	1,4	5,0	2,0	1,0	2,0	2,0	3,0	1,0
c ₆	0,33	0,5	0,5	1,0	0,2	1,4	0,25	0,16	0,5	1,0	0,5	0,33
c ₇	3,0	2,0	4,0	5,0	0,5	4,0	1,4	0,5	2,0	3,0	4,0	0,33
c ₈	4,0	5,0	6,0	7,0	1,0	6,0	2,0	1,4	3,0	2,0	3,0	0,25
c ₉	0,5	0,33	0,25	0,2	0,5	0,5	0,5	0,33	1,4	1,0	2,0	0,33
c ₁₀	0,33	0,5	0,33	0,25	0,33	0,33	0,25	0,33	0,5	1,4	2,0	0,33
c ₁₁	0,5	0,5	0,5	0,33	0,33	0,33	0,25	0,33	0,5	0,5	1,4	0,33
c ₁₂	5,0	6,0	7,0	8,0	1,0	3,0	3,0	4,0	4,0	3,0	3,0	1,4
Примечания c₁ – количество пешеходных зон; c₂ – количество знаков безопасности; c₃ – количество технических устройств; c₄ – количество парковочных зон; c₅ – количество нарушений; c₆ – количество опасных участков; c₇ – количество травм; c₈ – количество аварий; c₉ – количество дорожного освещения; c₁₀ – количество общественного транспорта; c₁₁ – количество велосипедных зон; c₁₂ – количество смертей.												

230

Источник: составлено автором.

Таблица И.2 – Взвешенная нормализованная матрица подкритериев безопасности

A _{ij}	c ₁	c ₂	c ₃	c ₄	c ₅	c ₆	c ₇	c ₈	c ₉	c ₁₀	c ₁₁	c ₁₂	c _{ср}	%
c ₁	0,00428	0,00746	0,00843	0,00895	0,00369	0,01029	0,00401	0,00251	0,01039	0,01443	0,00775	0,00369	0,00716	5,13
c ₂	0,00093	0,00162	0,00245	0,00292	0,00160	0,00074	0,00116	0,00087	0,00075	0,00104	0,00084	0,00134	0,00136	0,98
c ₃	0,00044	0,00058	0,00087	0,00140	0,00092	0,00053	0,00062	0,00052	0,00040	0,00050	0,00060	0,00096	0,00069	0,50
c ₄	0,00023	0,00027	0,00031	0,00049	0,00054	0,00037	0,00035	0,00031	0,00023	0,00026	0,00028	0,00051	0,00035	0,25
c ₅	0,03345	0,02915	0,02745	0,02623	0,02881	0,03348	0,03136	0,01960	0,02028	0,01878	0,02269	0,03610	0,02728	1,56
c ₆	0,00054	0,00070	0,00053	0,00084	0,00111	0,00129	0,00076	0,00063	0,00098	0,00182	0,00073	0,00233	0,00102	0,73
c ₇	0,01846	0,01072	0,01616	0,01608	0,01060	0,01971	0,01153	0,00721	0,01492	0,02073	0,02226	0,00885	0,01477	10,59
c ₈	0,03416	0,03720	0,03364	0,03124	0,02941	0,04102	0,03202	0,02001	0,03107	0,01918	0,02317	0,00921	0,02845	20,39
c ₉	0,00112	0,00065	0,00036	0,00023	0,00385	0,00089	0,00210	0,00175	0,00271	0,00251	0,00405	0,00322	0,00195	1,40
c ₁₀	0,00062	0,00081	0,00040	0,00024	0,00214	0,00049	0,00087	0,00146	0,00113	0,00210	0,00338	0,00269	0,00136	0,98
c ₁₁	0,00081	0,00070	0,00053	0,00028	0,00186	0,00043	0,00076	0,00127	0,00098	0,00091	0,00147	0,00234	0,00103	0,75
c ₁₂	0,05886	0,06154	0,05410	0,04922	0,04055	0,02827	0,06622	0,11037	0,05711	0,03966	0,03194	0,05082	0,05406	38,74

Источник: составлено автором.

Приложение К
(информационное)

Комбинационная матрица парных сравнений основных критериев

Таблица К.1 – Нормализованная матрица критериев альтернативных маршрутов

a_{ij}	Расстояние	Время в пути	Загруженность маршрута	Безопасность	Скоростные характеристики	Геометрия маршрута	Качество покрытия	Погодные условия	Стоимость проезда
Расстояние	1	1/2	1/3	1/6	1/5	3	1/4	2	4
Время в пути	2	1	1/2	1/5	1/4	4	1/3	3	5
Загруженность маршрута	3	2	1	1/4	1/3	5	1/2	4	6
Безопасность	6	5	4	1	2	8	3	7	9
Скоростные характеристики	5	4	3	1/2	1	7	2	6	8
Геометрия маршрута	1/3	1/4	1/5	1/8	1/7	1	1/6	1/2	2
Качество покрытия	4	3	2	1/3	1/2	6	1	5	7
Погодные условия	1/2	1/3	1/4	1/7	1/6	2	1/5	1	3
Стоимость проезда	1/4	1/5	1/6	1/9	1/8	1/2	1/7	1/3	1
Сумма	22,083	16,283	11,45	2,829	4,718	36,5	7,593	28,833	45

Источник: составлено автором.

Таблица К.2 – Взвешенная нормализованная матрица критериев альтернативных маршрутов

A_{ij}	Расстояние	Время в пути	Загруженность маршрута	Безопасность	Скоростные характеристики	Геометрия маршрута	Качество покрытия	Погодные условия	Стоимость проезда	C_{cp}	в процентах
Расстояние	0,0453	0,0307	0,0291	0,0589	0,0424	0,0822	0,0329	0,0694	0,0889	0,0533	5,33
Время в пути	0,0906	0,0614	0,0437	0,0707	0,053	0,1096	0,0439	0,104	0,1111	0,0764	7,64
Загруженность маршрута	0,1359	0,1228	0,0873	0,0884	0,0707	0,137	0,0659	0,1387	0,1333	0,109	10,90
Безопасность	0,2717	0,3071	0,3493	0,3535	0,4239	0,2192	0,3951	0,2428	0,2	0,307	30,70
Скоростные характеристики	0,2264	0,2457	0,262	0,1767	0,212	0,1918	0,2634	0,2081	0,1778	0,2182	21,82
Геометрия маршрута	0,0151	0,0154	0,0175	0,0442	0,0303	0,0274	0,022	0,0173	0,0444	0,0259	2,59
Качество покрытия	0,1811	0,1842	0,1747	0,1178	0,106	0,1644	0,1317	0,1734	0,1556	0,1543	15,43
Погодные условия	0,0226	0,0205	0,0218	0,0505	0,0353	0,0548	0,0263	0,0347	0,0667	0,037	3,37
Стоимость проезда	0,0113	0,0123	0,0146	0,0393	0,0265	0,0137	0,0188	0,0116	0,0222	0,0189	1,89

Источник: составлено автором.

Приложение Л

(информационное)

Коэффициенты влияния критериев когнитивной модели

Таблица Л.1 – Исходные значения коэффициентов влияния

a _{ij}	Ц ₁	Ц ₂	Ц ₃	Ц ₄	Ц ₅	Ц ₆	Ц ₇	Ц ₈	Ц ₉	М ₁	М ₂	М ₃	М ₄	с ₁	с ₂	с ₃	с ₄	с ₅	с ₆	с ₇	с ₈	с ₉	В ₁
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Ц ₁	1	0,5	0	0	0	0	0	0	0,2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ц ₂	0,5	1	0,3	0,5	0	0,2	0,8	0,3	0,1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ц ₃	0	0,4	1	0,2	0	0	0,9	0,4	0,4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ц ₄	0	0,6	0,7	1	0	0	0	0,1	0,5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ц ₅	0,1	0,1	0,5	0,5	1	0,1	0,2	0	0,1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ц ₆	0	0,1	0,2	0,1	0	1	0	0	0,2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ц ₇	0	0,4	0,8	0,2	0,5	0	1	0	0,3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ц ₈	0	0,3	0,4	0,2	0	0	0	1	0,2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ц ₉	1	0	0,1	0,2	0,1	0,1	0,3	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
М ₁	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
М ₂	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
М ₃	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
М ₄	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
с ₁	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0,8	0	0	0	0	0	0	0	0
с ₂	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
с ₃	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,1	1	0	0	0	0	0	0,2	0
с ₄	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,8	1	0	0	0	0	0	0
с ₅	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,1	0,2	0	1	0,1	0,2	0	0	0
с ₆	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0,2	0

Продолжение таблицы Л.1

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c ₇	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,4	0,8	0	0	0	1	0	0,1	0
c ₈	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,3	0,2	0	0	0	1	0,2	0
c ₉	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
B ₁	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,5	0	0	0	0	0,3	1
Примечания Ц₁ – наименьшее расстояние маршрута; Ц₂ – наименьшее время в пути; Ц₃ – наибольшие скоростные характеристики; Ц₄ – наименее загруженный маршрут; Ц₅ – минимальная стоимость проезда; Ц₆ – наилучшая геометрия маршрута; Ц₇ – наилучшее качество дорожного покрытия; Ц₈ – наилучшие погодные условия на маршруте; Ц₉ – наиболее безопасный маршрут; М₁ – маршрут 1; М₂ – маршрут 2; М₃ – маршрут 3; М₄ – маршрут 4; с₁ – расстояние маршрута; с₂ – время в пути; с₃ – скоростные характеристики; с₄ – загруженность маршрута; с₅ – стоимость проезда; с₆ – геометрия маршрута; с₇ – качество дорожного покрытия; с₈ – погодные условия; с₉ – безопасность; В₁ – водитель 1.																							

Источник: составлено автором.

Таблица Л.2 – Взвешенная нормализованная матрица критериев альтернативных маршрутов

A _{ij}	Ц ₁	Ц ₂	Ц ₃	Ц ₄	Ц ₅	Ц ₆	Ц ₇	Ц ₈	Ц ₉	M ₁	M ₂	M ₃	M ₄	c ₁	c ₂	c ₃	c ₄	c ₅	c ₆	c ₇	c ₈	c ₉	B ₁
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Ц ₁	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ц ₂	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ц ₃	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ц ₄	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ц ₅	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ц ₆	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ц ₇	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ц ₈	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ц ₉	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M ₁	0,33	0,29	0,20	0,31	0,2	0,21	0,18	0,19	0,22	1	0	0	0	0	0	0	0	0	0	0	0	0	0
M ₂	0,24	0,12	0,15	0,22	0,2	0,16	0,18	0,16	0,15	0	1	0	0	0	0	0	0	0	0	0	0	0	0
M ₃	0,21	0,11	0,19	0,21	0,2	0,14	0,12	0,15	0,14	0	0	1	0	0	0	0	0	0	0	0	0	0	0
M ₄	0,22	0,28	0,19	0,19	0,2	0,19	0,16	0,15	0,22	0	0	0	1	0	0	0	0	0	0	0	0	0	0
c ₁	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
c ₂	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
c ₃	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
c ₄	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
c ₅	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
c ₆	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
c ₇	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
c ₈	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
c ₉	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B ₁	0,76	0,72	0,64	0,71	0,39	0,61	0,55	0,62	0,71	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Источник: составлено автором.

Приложение М
(информационное)

Свидетельство о государственной регистрации программы для ЭВМ

РОССИЙСКАЯ ФЕДЕРАЦИЯ



RU2024615278

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ
ГОСУДАРСТВЕННАЯ РЕГИСТРАЦИЯ ПРОГРАММЫ ДЛЯ ЭВМ

Номер регистрации (свидетельства):
2024615278
Дата регистрации: **05.03.2024**
Номер и дата поступления заявки:
2024613173 13.02.2024
Дата публикации и номер бюллетеня:
05.03.2024 Бюл. № 3
Контактные реквизиты:
eromacheb@gmail.com

Автор(ы):
Ехлаков Роман Сергеевич (RU)
Правообладатель(и):
Ехлаков Роман Сергеевич (RU)

Название программы для ЭВМ:

«Программа поддержки принятия финансовых решений на основе сетевых структур»

Реферат:

Программа предназначена для расчета финального решения о выдаче финансовых продуктов заемщику на основе оценки различных параметров и заложенной модели бизнес-риска. Параметры были получены различным путем: заполнение анкеты пользователем, получение данных от бюро кредитных историй, вендоров мобильной связи и др. Заложенная модель риска гибко настраивается в зависимости от потребностей бизнеса, риск-аналитиков и финансового портфеля, доступного в момент времени. Все параметры, настройки, полученные данные и результаты записываются в базу данных (PostgreSQL). Конечным результатом программы является один из вариантов рекомендации: автоматическая выдача денежных средств, ручная выдача денежных средств (необходима дополнительная ручная проверка со стороны бизнеса), отказ в выдаче денежных средств. Выходные данные могут быть представлены, как в формате JSON для REST API, в формате SQL INSERT для записи в БД, в формате AMQP для брокера сообщений. Программа может быть полезна для разработчиков финансовых продуктов, рассчитывающих риски на основе множества параметров.

Язык программирования: PHP 8.1

Объем программы для ЭВМ: 1,2 МБ