

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ И РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА»
(САМАРСКИЙ УНИВЕРСИТЕТ)

ИНСТИТУТ СИСТЕМ ОБРАБОТКИ ИЗОБРАЖЕНИЙ РАН –
ФИЛИАЛ ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО УЧРЕЖДЕНИЯ «ФЕДЕРАЛЬНЫЙ
НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЙ ЦЕНТР «КРИСТАЛЛОГРАФИЯ И ФОТОНИКА»
РОССИЙСКОЙ АКАДЕМИИ НАУК

На правах рукописи

Агафонов Антон Александрович

**МЕТОДЫ И АЛГОРИТМЫ ОБРАБОТКИ
ГЕТЕРОГЕННОЙ ИНФОРМАЦИИ И АДАПТИВНОГО УПРАВЛЕНИЯ
В ИНТЕЛЛЕКТУАЛЬНОЙ ТРАНСПОРТНОЙ СИСТЕМЕ**

2.3.1. Системный анализ, управление и обработка информации, статистика

Диссертация на соискание учёной степени
доктора технических наук

Научный консультант:

Мясников Владислав Валерьевич,
доктор физико-математических наук,
доцент

Самара – 2023

ОГЛАВЛЕНИЕ

Введение	7
1 Математические методы построения интеллектуальных транспортных систем	17
1.1 Современные тенденции развития транспортной отрасли	17
1.2 Современные методы решения задач анализа и прогнозирования транспортного потока	26
1.3 Современные методы решения задач директивного управления транспортным потоком	32
1.3.1 Управление сигналами светофорных объектов	33
1.3.2 Построение траектории движения транспортных средств	39
1.3.3 Совместное управление сигналами светофоров и траекториями движения транспортных средств	41
1.4 Современные методы решения задач косвенного управления транспортным потоком и информирования в ИТС	43
1.4.1 Краткосрочное прогнозирование движения общественного транспорта	43
1.4.2 Решение навигационных задач	45
1.4.3 Маршрутизация транспортных средств	49
1.5 Обеспечение информационной безопасности в ИТС	51
1.5.1 Аутентификация транспортных средств	51
1.5.2 Обнаружение аномального поведения транспортных средств	53
1.6 Современные тенденции развития интеллектуальных транспортных систем	56
1.7 Постановка цели и задач исследования	62
2 Алгоритмические средства решения задачи прогнозирования в ИТС	64
2.1 Прогнозирование динамики транспортного потока с использованием актуальных и статистических данных	65
2.1.1 Основные обозначения и постановка задачи	66
2.1.2 Оценка параметров транспортных потоков по данным GPS/ГЛОНАСС наблюдений	68
2.1.3 Общая схема предлагаемого алгоритма	71

2.1.4	Экспериментальные исследования.....	86
2.2	Прогнозирование параметров транспортного потока с использованием подхода к обработке больших данных	96
2.2.1	Основные обозначения и постановка задачи	96
2.2.2	Модель прогнозирования параметров транспортного потока.....	97
2.2.3	Разбиение графа на подграфы.....	100
2.2.4	Реализация в MapReduce	103
2.2.5	Экспериментальные исследования.....	104
2.3	Прогнозирование параметров транспортного потока с использованием графовых нейронных сетей	109
2.3.1	Методология решения задачи прогнозирования	109
2.3.2	Модель прогнозирования параметров транспортного потока.....	113
2.3.3	Экспериментальные исследования.....	115
2.4	Прогнозирование параметров гетерогенного транспортного потока.....	122
2.4.1	Методология решения задачи прогнозирования	122
2.4.2	Экспериментальные исследования.....	126
2.5	Выводы и результаты второго раздела	128
3	Алгоритмические средства решения задачи директивного управления транспортным потоком в ИТС	130
3.1	Основные определения	131
3.2	Управление сигналами светофоров на основе обучения с подкреплением.....	133
3.2.1	Постановка задачи	133
3.2.2	Методология решения задачи адаптивного управления на основе обучения с подкреплением	135
3.2.3	Экспериментальный исследования алгоритма адаптивного управления на основе обучения с подкреплением.....	138
3.3	Управление сигналами светофоров на основе метода максимального взвешенного потока.....	142
3.3.1	Метод максимального потока на основе детерминированной прогнозной модели	143

3.3.2	Метод максимального взвешенного потока на основе модели глубокой нейронной сети	146
3.3.3	Метод максимального взвешенного потока в гетерогенном потоке транспортных средств	148
3.3.4	Экспериментальные исследования метода адаптивного управления на основе максимального взвешенного потока	150
3.4	Совместное управление сигналами светофоров и траекториями движения транспортных средств	160
3.4.1	Метод управления траекториями движения с учетом адаптивного управления сигналами светофоров	160
3.4.2	Метод совместного управления траекториями движения транспортных средств и адаптивного управления сигналами светофоров.	171
3.4.3	Экспериментальные исследования методов совместного управления.	174
3.5	Выводы и результаты третьего раздела.....	184
4	Алгоритмические средства решения задач косвенного управления транспортным потоком и информирования в ИТС.....	186
4.1	Прогнозирование движения отдельных транспортных средств	188
4.1.1	Прогнозирование движения общественного транспорта.....	189
4.1.2	Прогнозирование движения подключенных транспортных средств....	198
4.2	Решение навигационных задач в ИТС.....	202
4.2.1	Постановка задачи нахождения надежного пути.....	204
4.2.2	Алгоритм нахождения надежного пути на основе прямого вычисления сверток	206
4.2.3	Алгоритм нахождения надежного пути с использованием параметрически заданных устойчивых распределений вероятностей.....	213
4.2.4	Алгоритм нахождения оптимального пути на общественном транспорте	221
4.2.5	Экспериментальные исследования алгоритмов нахождения надежного пути	225
4.3	Маршрутизация подключенных транспортных средств в сети	232

4.3.1	Алгоритм резервирования маршрутов движения транспортных средств	232
4.3.2	Алгоритм резервирования маршрутов движения транспортных средств в стохастической транспортной сети	237
4.3.3	Алгоритм резервирования маршрутов движения транспортных средств в гетерогенном транспортном потоке	241
4.3.4	Экспериментальные исследования алгоритмов маршрутизации подключенных транспортных средств в сети	245
4.4	Обеспечение информационной безопасности в контексте ИТС.....	253
4.4.1	Аутентификация транспортных средств.....	254
4.4.2	Обнаружение аномального поведения транспортных средств.....	264
4.4.3	Обнаружение аномального поведения транспортных средств в колонне	284
4.5	Выводы и результаты четвертого раздела.....	291
5	Архитектура и реализация программного комплекса кооперативной ИТС	292
5.1	Требования к программному комплексу	292
5.2	Архитектура программного комплекса с использованием принципов обработки больших данных.....	293
5.2.1	Понятие больших данных	293
5.2.2	Инструментарий для обработки больших данных	294
5.2.3	Архитектура систем обработки больших данных	296
5.2.4	Программное обеспечение для работы с большими данными	301
5.2.5	Архитектура и реализация программного комплекса кооперативной интеллектуальной транспортной системы	305
5.3	Программный модуль краткосрочного прогнозирования параметров транспортного потока.....	311
5.3.1	Назначение программного модуля.....	311
5.3.2	Описание логической структуры.....	311
5.3.3	Схема работы программного модуля.....	312
5.3.4	Логическая модель программного модуля	313

5.3.5	Физическая модель данных программного модуля.....	315
5.3.6	Программный интерфейс модуля.....	316
5.4	Программный модуль адаптивного управления транспортным потоком путем светофорного регулирования	318
5.4.1	Назначение программного модуля.....	318
5.4.2	Логическая модель программного модуля	319
5.4.3	Физическая модель данных программного модуля.....	321
5.4.4	Процедура обработки данных.....	321
5.5	Программный модуль расчета оптимального пути на общественном транспорте .	322
5.5.1	Назначение программного модуля.....	322
5.5.2	Исходные данные.....	324
5.5.3	Формирование модели транспортной системы города	326
5.5.4	Описание логической структуры.....	327
5.5.5	Описание работы программного модуля.....	330
5.5.6	Физическая модель данных программного модуля.....	336
5.5.7	Программный интерфейс модуля.....	338
5.5.8	Взаимодействие с клиентскими приложениями	340
5.6	Программный модуль расчета оптимального (в том числе – надежного) пути на индивидуальном транспорте в стохастической сети.....	343
5.6.1	Назначение программного модуля.....	343
5.6.2	Логическая модель программного модуля	345
5.6.3	Физическая модель данных программного модуля.....	348
5.6.4	Программный интерфейс модуля.....	348
5.7	Выводы и результаты пятого раздела.....	353
	Заключение.....	355
	Список сокращений и обозначений.....	357
	Список литературы.....	358
	Приложение А Использование результатов диссертации	394

ВВЕДЕНИЕ

Диссертационная работа посвящена разработке методов, алгоритмов и программных средств, направленных на решение проблемы повышения эффективности использования транспортной инфраструктуры путем управления транспортным потоком с гетерогенным составом транспортных средств, а также движением отдельных транспортных средств в кооперативных интеллектуальных транспортных системах.

Актуальность темы

В настоящее время транспортные системы являются неотъемлемой частью человеческой деятельности. Постоянный рост дорожного трафика, особенно в крупных городах, приводит к значительному увеличению затрат (времени, стоимости) на осуществление корреспонденций участниками дорожного движения, а также к увеличению вредных выбросов в атмосферу и ухудшению экологической обстановки. Для улучшения ситуации во многих странах используются различные стратегии: замена классических транспортных средств на гибриды и электромобили, территориальное и временное зонирование доступных территорий передвижения и стоянки, развитие альтернативных средств передвижения, развитие средств совместного использования транспортных средств (каршеринг, информационные системы попутчиков), оптимизация существующей транспортной инфраструктуры с целью повышения эффективности ее использования.

Одной из современных тенденций является развитие информационно-коммуникационных технологий и их проникновение в различные сферы жизни человека. Развитие интернета вещей, автомобильных самоорганизующихся сетей (VANET), подключенных (англ.: connected vehicles) и автономных (англ.: autonomous vehicles) транспортных средств привело к увеличению числа источников данных, которые могут использоваться для повышения эффективности решения задач анализа, прогнозирования и управления транспортным потоком: данные с камер видеонаблюдения на перекрестках и детекторов транспортного потока, траектории движения транспортных средств из навигационных приложений, информация о маршруте, положении и скорости движения подключенных и автономных транспортных средств и т.д. Одновременно с ростом числа источников, лавинообразно растет и объем доступных для анализа данных, что позволяет использовать методы машинного обучения и подходы к обработке «больших данных» для решения транспортных задач.

В настоящее время разработка и внедрение интеллектуальных транспортных систем (ИТС) является ключевым способом повышения эффективности использования дорожно-

транспортной инфраструктуры. ИТС предназначены для решения задач в различных транспортных областях, таких как управление транспортными потоками, планирование маршрутов движения общественного транспорта, разработка навигационных и логистических сервисов, сервисов информирования граждан о прибытии общественного транспорта и т.д. Учитывая указанные тенденции, существующие системы анализа, прогнозирования и управления дорожным движением как часть интеллектуальных транспортных систем (ИТС, англ.: ITS - Intelligent Transportation System) также должны претерпевать существенные изменения, в т.ч. в научном плане.

Разработка кооперативной интеллектуальной транспортной системы, решающей задачи анализа, прогнозирования и управления транспортным потоком с гетерогенным (смешанным) составом транспортных средств, включающим подключенные и/или автономные транспортные средства и управляемые водителями транспортные средства, позволит:

- повысить эффективность решения задач анализа и прогнозирования параметров гетерогенного транспортного потока за счет использования данных о маршрутах движения подключенных транспортных средств, их положении и скорости движения;

- повысить эффективность решения задач директивного управления дорожным движением в транспортных сетях с гетерогенным составом транспортных средств за счет адаптивного управления транспортным потоком на регулируемых перекрестках или совместного управления сигналами светофорных объектов и движением подключенных и/или автономных подключенных транспортных средств, что позволит уменьшить общую загруженность дорожной сети, потребление топлива и выбросы продуктов сгорания, особенно в городских районах;

- повысить эффективность решения задач косвенного управления и информирования, таких как прогнозирование времени движения подключенных или общественных транспортных средств, построение маршрутов движения отдельных транспортных средств или перераспределение транспортных потоков путем маршрутизации всех транспортных средств или их отдельной части.

Существующие интеллектуальные транспортные системы используют традиционные источники данных. В то же время, согласно аналитическому отчету НТИ «Автонет», по прогнозам, к 2025 году в эксплуатации будет более 400 миллионов подключенных автомобилей, а размер рынка достигнет 121 млрд. долларов США. Также, по мнению аналитиков, к 2030 году доля новых транспортных средств со встроенной связью достигнет 96%. Кроме того, в России, как и во всем мире, существуют

стратегические планы развития интеллектуальных транспортных систем, что также подтверждает актуальность представленной проблемы. Проблема развития ИТС в России отражена в национальном проекте РФ «Безопасные и качественные автомобильные дороги» на 2020-2024 гг., мероприятия 3.3.1 – 3.3.5 «Внедрение интеллектуальных транспортных систем, предусматривающих автоматизацию процессов управления дорожным движением в городских агломерациях». Кроме того, разрабатываемые методы и подходы отражены в федеральном проекте «Искусственный интеллект» и «Национальной стратегии развития искусственного интеллекта на период до 2030 года» (указ президента РФ от 10.10.2019 № 490). В федеральном проекте нашли отражение шесть задач, среди которых поддержка научных исследований, разработка и развитие программного обеспечения, повышение доступности и качества данных. Также в России ведется разработка федерального проекта «Инфраструктура беспилотных и подключенных транспортных средств», в рамках которого предусмотрено развитие автомобильного, железнодорожного и других видов транспорта с автономным управлением, а также создание инфраструктуры для автономного и подключенного транспорта.

Учитывая все изложенные выше тезисы, можно говорить о безусловной актуальности как темы диссертационной работы в целом, так и отдельных выбранных направлений исследований в частности.

Цель и задачи исследований

Целью исследования является повышение эффективности использования транспортной инфраструктуры путем управления транспортным потоком с гетерогенным составом транспортных средств, а также движением отдельных транспортных средств в кооперативных интеллектуальных транспортных системах.

Для достижения поставленной цели в диссертационной работе решаются следующие задачи:

- 1) проведение системного анализа задачи управления транспортным потоком с гетерогенным составом транспортных средств;
- 2) разработка алгоритмических средств решения задачи краткосрочного прогнозирования параметров гетерогенного транспортного потока в транспортной сети с использованием методов машинного обучения и подходов к обработке больших данных на основе гетерогенной информации (актуальных и статистических данных) о состоянии транспортного потока и движении отдельных транспортных средств;

3) разработка алгоритмических средств решения задачи директивного управления транспортным потоком на регулируемых перекрестках путем независимого и/или координированного адаптивного управления сигналами светофоров и движением подключенных автономных транспортных средств в транспортной сети;

4) разработка алгоритмических средств решения задач косвенного управления транспортным потоком и информирования в ИТС:

- разработка алгоритмов решения задачи прогнозирования движения отдельных транспортных средств с использованием методов машинного обучения;

- разработка алгоритмов решения навигационной задачи нахождения надежного пути в стохастической зависящей от времени транспортной сети;

- разработка алгоритмов маршрутизации подключенных транспортных средств в транспортной сети в интеллектуальной транспортной системе на основе численного метода резервирования маршрутов, позволяющего достичь транспортного равновесия в сети;

5) разработка и реализация программного комплекса кооперативной ИТС, решающей задачи анализа, прогнозирования и управления транспортным потоком с гетерогенным составом транспортных средств с использованием подходов к обработке больших данных;

б) проведение экспериментальных исследований разработанных методов на натуральных и модельных данных, анализ результатов и сравнение с существующими решениями.

Объект исследования

Объектом исследования являются кооперативные интеллектуальные транспортные системы.

Предмет исследования

Предметом исследования являются методы управления транспортным потоком и движением отдельных транспортных средств, направленные на повышение эффективности использования существующей транспортной инфраструктуры.

Методы исследований

В диссертационной работе используются методы машинного обучения и искусственного интеллекта, теории вероятностей и статистического анализа, теории графов, методы оптимизации.

Научная новизна работы

1) предложен комплекс алгоритмических средств (математический метод и алгоритмы) решения задачи краткосрочного прогнозирования параметров транспортного потока в транспортной сети с использованием графовых сверточных нейронных сетей и подходов к обработке больших данных на основе гетерогенной информации о состоянии транспортного потока и движении отдельных транспортных средств.

2) предложен комплекс алгоритмических средств решения задачи директивного управления транспортным потоком, включая:

- метод адаптивного светофорного управления транспортным потоком на основе максимизации взвешенного потока транспортных средств с использованием алгоритмов оценки транспортного потока на основе детерминированной модели прогнозирования движения транспортных средств (не требующей настройки/обучения) и на основе обучаемой модели глубокой нейронной сети регрессионного вида;

- алгоритм адаптивного светофорного управления транспортным потоком с использованием подхода на основе машинного обучения с подкреплением, учитывающий как наблюдаемые, так и прогнозные параметры, описывающие состояние транспортного потока;

- метод адаптивного управления транспортным потоком на регулируемых перекрестках путем координированного управления сигналами светофоров и траекториями движения подключенных автономных транспортных средств в транспортной сети;

3) предложен комплекс алгоритмических средств решения задачи косвенного управления транспортным потоком и информирования в ИТС:

- алгоритм определения надёжного пути в зависящей от времени стохастической (транспортной) сети, учитывающий информацию о пространственной и временной корреляции сегментов дорожной сети, текущую и прогнозную информацию о состоянии транспортного потока; ускоренная модификация алгоритма с использованием распределения Леви;

- алгоритмы краткосрочного прогнозирования времени движения отдельных транспортных средств, учитывающие гетерогенную информацию о транспортной ситуации, прямо или косвенно влияющую на прогнозируемое время движения;

- алгоритм маршрутизации подключенных транспортных средств в транспортной сети на основе численного метода резервирования маршрутов, учитывающий стохастические свойства транспортной сети; модификация алгоритма для его применения в гетерогенном транспортном потоке;

4) разработана архитектура и реализован программный комплекс кооперативной интеллектуальной транспортной системы, решающей задачи анализа, прогнозирования и управления транспортным потоком с гетерогенным составом транспортных средств с использованием подходов к обработке больших данных.

Практическая значимость работы

Разработанные решения могут быть использованы в составе кооперативной интеллектуальной транспортной системы и позволяют повысить точность прогнозирования параметров транспортного потока, снизить временные затраты на совершение транспортных корреспонденций, потребление топлива и выбросы продуктов сгорания, т.е. в целом повысить эффективности использования транспортной инфраструктуры.

Разработанный программный модуль, входящий в состав программного комплекса, позволяет решать задачу прогнозирования времени прибытия общественных транспортных средств на остановочные пункты с учетом актуальных и статистических данных о движении отдельных транспортных средств в частности и состояния транспортных потоков в целом. Программный модуль используется для информирования пассажиров о времени прибытия общественных транспортных средств на остановочные пункты в г. Самара, что подтверждается актом внедрения АО «Самара-Информспутник». Прогнозная информация, предоставляемая разработанным программным комплексом, доступна для пассажиров на сайте транспортного оператора г.Самара (tosamara.ru) или с использованием мобильного приложения «Прибывалка-63».

Реализация результатов работы

Результаты диссертации использованы при выполнении ряда госбюджетных и хоздоговорных НИР в АО «Самара-Информспутник», в т.ч. работ по созданию информационно-справочной системы городского округа Самара в части разработки интернет-портала Транспортного оператора Самары (www.tosamara.ru), проекта РФФИ № 13-07-12103-офи-м «Анализ и прогнозирование транспортных потоков на основе комплексного использования космической навигационной информации, данных дистанционного зондирования Земли и систем видеонаблюдения», программы фундаментальных исследований Президиума РАН «Фундаментальные проблемы информатики и информационных технологий» (проект 2.12), проекта РФФИ № 16-37-00055 мол-а «Адаптивные методы краткосрочного прогнозирования параметров транспортных потоков», проекта РФФИ № 18-29-03135-мк «Быстрые алгоритмы и высокопроизводительные вычисления в задачах обработки больших данных для анализа,

предсказания и управления движением в интеллектуальных транспортных системах», проекта РФФИ № 18-07-00605-а «Методы и алгоритмы централизованного управления автономными транспортными средствами в интеллектуальных транспортных системах», проекта ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технического комплекса России на 2014-2020 годы» по теме «Персональный цифровой автотранспортный помощник» (соглашение № 075-15-2019-062), проекта РНФ № 21-11-00321 «Методы и алгоритмы совместного и координированного управления сигналами светофоров и подключенными автономными транспортными средствами в транспортной сети».

Апробация работы

Основные результаты диссертации были представлены на международных научных конференциях: 18th IEEE International Conference on Intelligent Transportation Systems (Лас-Пальмас-де-гран-Канария, Испания, 2015), 5th International Conference on Analysis of Images, Social networks, and Texts (Екатеринбург, Россия, 2016), 2016 IEEE Intelligent Vehicle Symposium (Гетеборг, Швеция, 2016), 21st IEEE International Conference on Computational Science and Engineering (Бухарест, Румыния, 2018), 19th International Conference on Intelligent Data Engineering and Automated Learning (Мадрид, Испания, 2018), 2018 The 2nd International Conference on Mechanical, System and Control Engineering (Самара, Россия, 2018), 4th International Conference on Intelligent Transportation Engineering (Сингапур, 2019), 16th International Symposium on Neural Networks (Москва, Россия, 2019), международная научная конференция «Проблемы управления и моделирования в сложных системах» (Самара, Россия, 2019), 16th Conference on Computer Science and Intelligence Systems (Болгария, виртуальный формат, 2021), 3rd International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (Россия, виртуальный формат, 2021), 2022 IEEE International Multi-Conference on Engineering, Computer and Information Sciences (Россия, виртуальный формат, 2022), 4th International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (Липецк, Россия, 2022), научная конференция «Транспортные потоки на сетях» (Россия, виртуальный формат, 2023), Международная конференция и молодежная школа «Информационные технологии и нанотехнологии» (Самара, Россия, 2015-2023).

Публикации

По теме диссертационной работы автором опубликовано 75 работ, в т.ч. 57 работ в изданиях, индексируемых в базах данных Scopus и Web of Science (WoS), получено 8 свидетельств о государственной регистрации программы для ЭВМ.

Личный вклад автора

Личный вклад автора состоит в участии на всех этапах процесса исследования, включая постановку задач, разработку комплекса алгоритмических средств (математических методов и алгоритмов) решения задач управления и информирования, получение исходных данных, постановку экспериментов, апробацию результатов исследования, разработку архитектуры и реализацию программного комплекса. Выбор концепции диссертационной работы и направления исследований проводился совместно с научным консультантом. Программная реализация отдельных алгоритмов, модулей, проведение экспериментов и подготовка публикаций по выполненной работе выполнялась совместно с другими соавторами.

Соответствие паспорту специальности

Область исследования соответствует пункту 4 – Разработка методов и алгоритмов решения задач системного анализа, оптимизации, управления, принятия решений, обработки информации и искусственного интеллекта; 9 – Разработка проблемно-ориентированных систем управления, принятия решений и оптимизации технических объектов; 10 – Методы и алгоритмы интеллектуальной поддержки при принятии управленческих решений в технических системах направлений исследований паспорта научной специальности 2.3.1. Системный анализ, управление и обработка информации, статистика.

Структура диссертации

Диссертация состоит из введения, пяти глав, заключения, списка литературы (427 наименований), одного приложения. Работа изложена на 397 страницах, содержит 179 рисунков, 44 таблицы.

На защиту выносятся

1) комплекс алгоритмических средств (математический метод и алгоритмы) решения задачи краткосрочного прогнозирования параметров транспортного потока в транспортной сети с использованием графовых сверточных нейронных сетей и подходов к обработке больших данных на основе гетерогенной информации о состоянии

транспортного потока и движения отдельных транспортных средств, отличающийся от известных повышенной точностью;

2) комплекс алгоритмических средств решения задачи директивного управления транспортным потоком, позволяющий снизить среднее время транспортных корреспонденций и средний расход топлива по сравнению с ранее известными методами:

- метод адаптивного светофорного управления транспортным потоком на основе максимизации взвешенного потока транспортных средств с использованием алгоритмов оценки транспортного потока на основе детерминированной модели прогнозирования движения транспортных средств (не требующей настройки/обучения) и на основе обучаемой модели глубокой нейронной сети регрессионного вида;

- алгоритм адаптивного светофорного управления транспортным потоком с использованием подхода на основе машинного обучения с подкреплением, учитывающий как наблюдаемые, так и прогнозные параметры, описывающие состояние транспортного потока;

- метод адаптивного управления транспортным потоком на регулируемых перекрестках путем координированного управления сигналами светофоров и траекториями движения подключенных автономных транспортных средств в транспортной сети;

3) комплекс алгоритмических средств решения задачи косвенного управления транспортным потоком и информирования в интеллектуальных транспортных системах:

- алгоритм определения надёжного пути в зависящей от времени стохастической (транспортной) сети, учитывающий информацию о пространственной и временной корреляции сегментов дорожной сети, текущую и прогнозную информацию о состоянии транспортного потока; ускоренная модификация алгоритма с использованием распределения Леви, отличающаяся от известных алгоритмов повышенной производительностью;

- алгоритмы краткосрочного прогнозирования времени движения отдельных транспортных средств, учитывающие гетерогенную информацию о транспортной ситуации, прямо или косвенно влияющую на прогнозируемое время движения, отличающиеся от известных алгоритмов повышенной точностью;

- алгоритм маршрутизации подключенных транспортных средств в транспортной сети на основе численного метода резервирования маршрутов, учитывающий стохастические свойства транспортной сети; модификация алгоритма для его применения в гетерогенном транспортном потоке;

4) программный комплекс кооперативной интеллектуальной транспортной системы, решающей задачи анализа, прогнозирования и управления транспортным потоком с гетерогенным составом транспортных средств с использованием подходов к обработке больших данных;

5) результаты экспериментальных исследований, подтвердившие эффективность разработанных методов, алгоритмов и реализованного программного комплекса.

1 Математические методы построения интеллектуальных транспортных систем

1.1 Современные тенденции развития транспортной отрасли

В настоящее время транспортные системы являются неотъемлемой частью человеческой деятельности. Постоянный рост дорожного трафика, особенно в крупных городах, приводит к значительному увеличению затрат (времени, стоимости) на осуществление корреспонденций участниками дорожного движения, а также к увеличению вредных выбросов в атмосферу и ухудшению экологической обстановки. Согласно данным [1] в странах Евросоюза на долю транспортной системы приходится в среднем около 5% валового внутреннего продукта (ВВП). В России в 2021 году на эту отрасль экономики пришлось 6% ВВП [2]. При этом на экономические показатели транспортной отрасли экономики существенное отрицательное влияние оказывает рост интенсивности транспортного потока. Согласно аналитическому отчету компании INRIX [3], прогнозируемые потери экономики Великобритании вследствие заторов на дорогах к 2030 году составят 21,4 миллиарда фунтов стерлингов в год. Помимо этого, заторы на дорогах также пагубно влияют на здоровье участников движения [4], способствуют загрязнению окружающей среды [5] и повышают потребление топлива участниками движения [6]. Проблема дорожных заторов также актуальна и для России. По данным Единой межведомственной информационно-статистической системы [7], количество зарегистрированных транспортных средств в Российской Федерации за последние 10 лет увеличилось примерно на 35%. Всего в Российской Федерации на конец 2021 года зарегистрировано более 57 млн. транспортных средств [7]. Такая тенденция роста количества транспортных средств обуславливает необходимость в разработке новых эффективных средств анализа, прогнозирования и контроля транспортного потока, в т.ч. в составе интеллектуальных транспортных систем [8, 9].

Под **интеллектуальной транспортной системой (ИТС)** понимается система управления, интегрирующая современные информационные и телематические технологии и предназначенная для автоматизированного поиска и принятия к реализации максимально эффективных сценариев управления транспортно-дорожным комплексом региона, конкретным транспортным средством (ТС) или группой транспортных средств с целью обеспечения заданной мобильности населения, максимизации показателей использования дорожной сети, повышения безопасности и эффективности транспортного процесса, комфортности для водителей и пользователей транспорта [8].

Обеспечение максимальной эффективности функционирования дорожно-транспортного комплекса страны является одной из основных задач транспортной системы. Развитие интеллектуальных транспортных систем является одним из ключевых способов повышения эффективности использования транспортной инфраструктуры и совершенствования организации дорожного движения путем управления транспортными потоками и транспортными средствами, а также своевременного информирования и управления действиями в условиях инцидентов, нештатных и чрезвычайных ситуаций [10]. Цель транспортной системы «умного города» – предоставить жителям ряд безопасных, удобных и экологически чистых транспортных средств. Умные городские транспортные системы используют передовые технологии, такие как датчики дорожного движения и интеллектуальные системы управления дорожным движением, для оптимизации транспортного потока и уменьшения заторов. Интеллектуальные подключенные транспортные средства (CVs) являются одним из важнейших элементов системы управления дорожным движением. CVs – это транспортные средства, оснащенные передовыми технологиями связи и датчиками, которые позволяют им обмениваться информацией с другими транспортными средствами, пешеходами и инфраструктурой (Vehicle-to-Everything, V2X-коммуникации).

Формально, согласно [11], Vehicle-to-Everything (V2X) – процесс взаимодействия ТС с любыми объектами, которые могут повлиять на транспортное средство, для взаимного обмена информацией посредством беспроводной связи. **Подключенное транспортное средство** – транспортное средство, которое обменивается данными с другими транспортными средствами и устройствами, сетями и сервисами, охватывающими дорожную инфраструктуру [11]. **Беспилотное (автономное) транспортное средство** – высоко- или полностью автоматизированное транспортное средство, функционирующее без вмешательства человека [11]. **Кооперативная интеллектуальная транспортная система** – ИТС, основанная на технологиях V2X (cooperative intelligent transport system, C-ITS) [11].

Начиная с появления первых автомобилей с бензиновым двигателем внутреннего сгорания в конце 1890-х годов, автомобильная отрасль постоянно развивается (рисунок 1).



Рисунок 1 – Основные этапы развития автомобильной отрасли

Среди ключевых факторов развития, важных в контексте рассматриваемой в диссертационной работе теме исследования, следует выделить следующие:

- конец 1980-х годов – появление первых навигаторов;
- 1996 г. – разработка первого подключенного автомобиля компанией General Motors;
- начало 2000-х годов – появление автомобильных самоорганизующихся сетей (VANETs), массовое использование GPS-навигаторов;
- 2014 г. – разработка классификации автоматизации автомобилей Сообществом автомобильных инженеров (SAE) [12].

К настоящему времени выделяется 6 уровней автоматизации транспортных средства (ТС):

- 1) уровень 0: нет автоматизации, водитель может получать лишь определенные предупреждения;
- 2) уровень 1, «помощь водителю»: автоматизированная система помогает водителю, деля с ним контроль над управлением ТС, например, водитель контролирует руль, система – скорость (система круиз-контроль) или наоборот, при этом могут возникать предупреждения о сходе с полосы;
- 3) уровень 2, «частичная автоматизация»: частичная автоматизация вождения, при которой автоматизированная система выполняет ускорение, торможение, рулевое управление в виде перестроения с полосы на полосу, при этом человек контролирует процесс и осуществляет маневрирование;
- 4) уровень 3, «условная автоматизация»: автоматизация условного вождения, представляющая собой вождение ТС автоматизированной системой, но водитель должен вмешаться в экстренных ситуациях;

5) уровень 4, «широкая автоматизация»: высокая автоматизация вождения обеспечивает контроль над автомобилем в пределах городской местности или в местах, для которых существуют подробные карты;

6) уровень 5, «полная автоматизация»: полная автоматизация обеспечивает контроль над ТС в любых условиях, человеческого вмешательства не требуется.

Первые автомобили с официально подтвержденным уровнем автоматизации 3 появились в 2021 году. Согласно отчету [13] появление полностью транспортных средств с полной автоматизацией (уровень 5) ожидается к 2035 году.

Рассмотрим указанные тенденции развития автомобильной отрасли подробнее.

Согласно принятой Стратегии развития автомобильной промышленности Российской Федерации на период до 2025 г. [14], ключевыми глобальными трендами развития автомобильной отрасли, обладающими наибольшим потенциалом влияния на развитие транспортных систем, являются повышение автономности транспортных средств и внедрение сетевых (телекоммуникационных) технологий в транспортных системах. Похожие направления развития выделены и в европейской стратегии развития кооперативных ИТС [15]. Согласно стратегии, в настоящее время в транспортном секторе ожидается значительная трансформация существующих решений в разработке и внедрении интеллектуальных транспортных систем благодаря широкому распространению цифровых технологий и увеличению объемов данных. Обмен данными между различными участниками транспортной системы позволит согласовать транспортный спрос и предложение, что приведет к более эффективному распределению ресурсов.

Как уже подчеркивалось ранее, одной из ключевых тенденций развития транспортных систем является повышение автономности транспортных средств с целью разработки полностью автономных (беспилотных) автомобилей (уровни 4 и 5 автоматизации в соответствии с классификацией Сообщества автомобильных инженеров (SAE)). Разработкой автономных ТС занимаются многие мировые компании [16], включая как лидеров автомобильной индустрии (Volvo [17], Tesla [18], BMW), так и технологические компании из сектора информационных технологий (Google, Waymo, Alibaba). В России лидерами рынка беспилотных автомобилей являются компании Яндекс [19] и Cognitive Technologies.

В настоящее время развитие автономных ТС достигло уровня, при котором движение может выполняться без вмешательства человека, хотя и в строго определенных условиях. Одна из основных проблем на пути к полной автономии заключается в том, что окружающая среда (включая правила движения, погодные условия и т.д.) в значительной

степени влияет на уровень автономности, что не позволяет применять разработанные решения в любых условиях. Еще одним препятствием на пути к реализации проектов по созданию беспилотных автомобилей является отсутствие законодательной базы, регулирующей перемещение таких транспортных средств по дорогам общего пользования [20].

Второй доминирующей тенденцией в развитии транспортных систем является разработка подключенных автотранспортных средств. Развитие подключенных транспортных средств включает в себя разработку телематических систем, внедрение протоколов обмена данными (выделенная связь ближнего действия) для автомобильного транспорта, а также проведение научных исследований в области кибербезопасности, больших данных, искусственного интеллекта. В России, в ходе реализации проекта российской сервисной телематической платформы «Автодата» в рамках национальной технологической инициативы «Автонет» было завершено тестирование технологий на базе V2X в ряде регионов [21]. Система позволит беспилотным автомобилям общаться с инфраструктурой, другими транспортными средствами, светофорами, перекрестками и иными дорожными объектами. С прикладной точки зрения развитие подключенных и/или автономных транспортных средств позволяет обмениваться информацией между объектами дорожной инфраструктуры и транспортными средствами в реальном времени, что может использоваться для повышения эффективности методов прогнозирования транспортного потока и управления движением, в т.ч., путем координированного управления сигналами светофоров и движением транспортных средств [22, 23].

Разработка автономных и подключенных транспортных средств позволит генерировать большие объемы данных о движении транспортных средств, мобильности участников движения, что приводит к необходимости разработки новых методов обработки и анализа данных, в т.ч. методов на основе искусственного интеллекта и машинного обучения [24-26]. Наиболее популярные решаемые задачи включают в себя прогноз распространения транспортного потока [27], управление транспортным потоком [28], прогнозирование движения общественного транспорта [29] и другие.

Состояние отрасли можно охарактеризовать по нескольким показателям: количество публикаций в научных журналах, количество патентов на технологии, объем финансирования НИР и др. Согласно отчету «Artificial Intelligence Index Report 2021» Стэнфордского университета [30], наблюдается рост публикационной активности по теме «Искусственный интеллект» (рисунок 2), причем наибольший рост показали публикации

по темам машинное обучение (cs.LG) и робототехника (cs.RO) (рисунок 3). В лидерах остаются публикации по темам компьютерного зрения (cs.CV) и анализ языка (cs.CL).

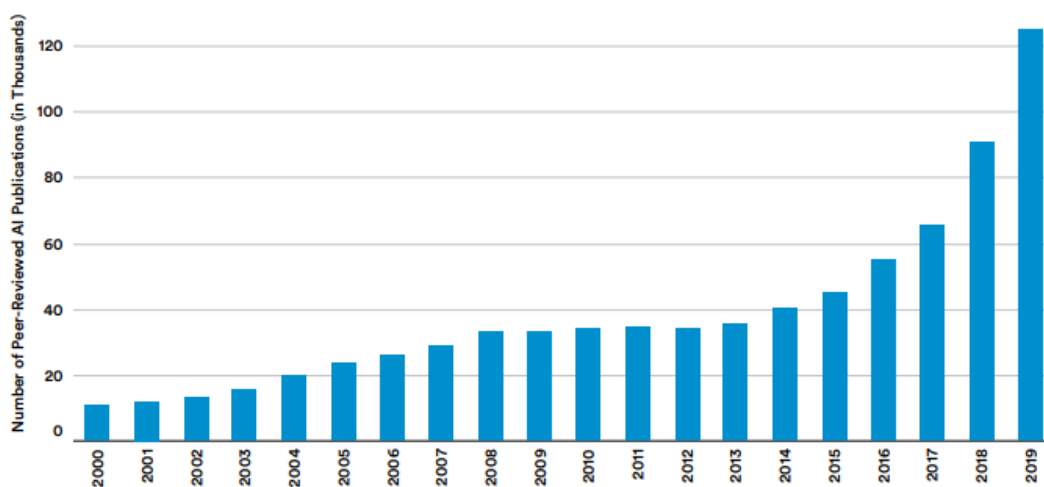


Рисунок 2 – Число рецензируемых публикаций (в тысячах)

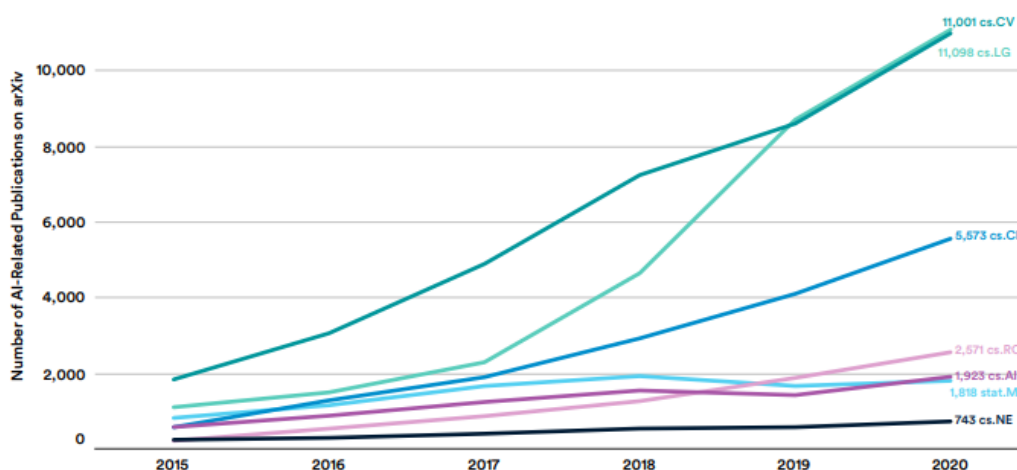


Рисунок 3 – Число публикаций на arXiv по темам

Кроме того, согласно данным [31], наблюдается значительный рост по более узким темам, связанным с тематикой настоящей диссертационной работы: разработка автономных и подключенных транспортных средств (рисунок 4).

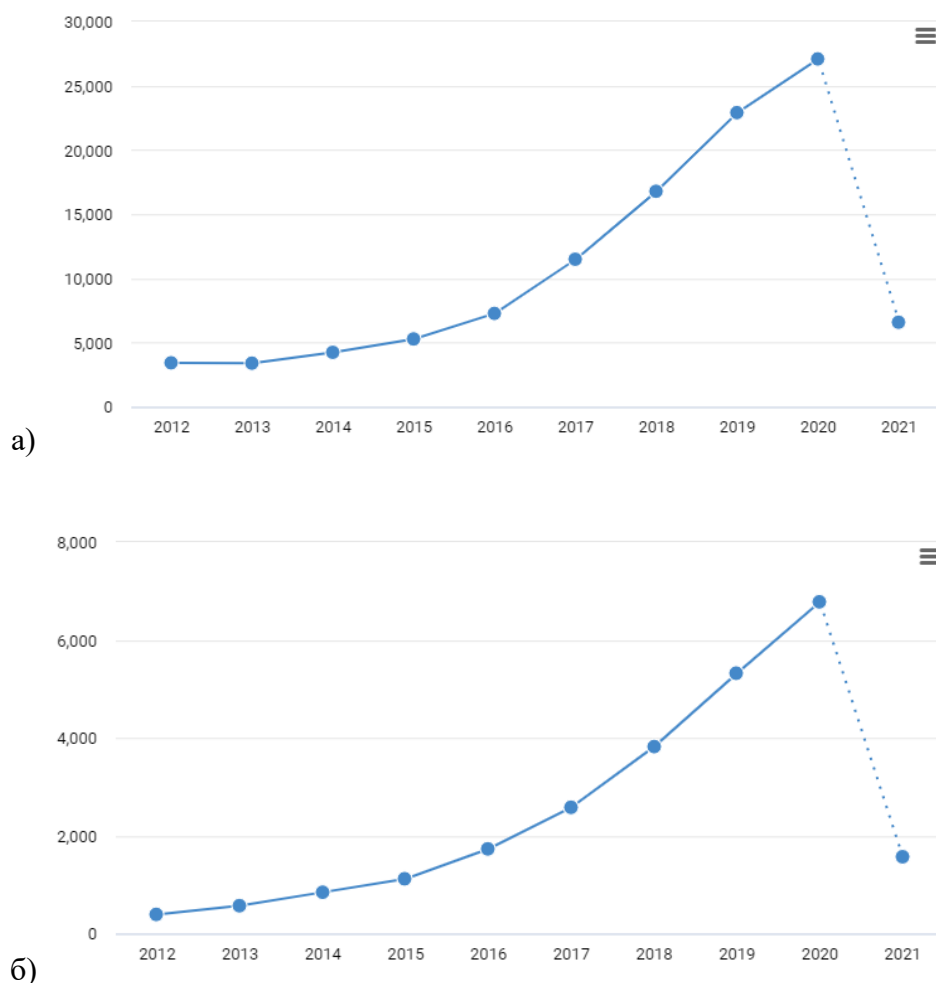


Рисунок 4 – Число публикаций по темам «Автономные ТС» (а) и «Подключенные ТС» (б)

Указанные тенденции в развитии автономных и подключенных транспортных средств, увеличение источников и объёма данных как о движении транспортных средств в частности, так и о городской мобильности и состоянии транспортной инфраструктуры в целом, а также развитие методов машинного обучения позволяют с уверенностью прогнозировать трансформацию традиционных ИТС к т.н. кооперативным ИТС. Кооперативные ИТС позволят обеспечить взаимодействие подключенных / автономных транспортных средств друг с другом и объектами инфраструктуры для обмена информацией и координации своих действий. Ожидается, что координированное движение (в т.ч., оповещение о потенциально опасных ситуациях, получение рекомендаций по оптимальной скорости движения и т.д.), обеспечиваемое цифровой коммуникацией, значительно повысит безопасность дорожного движения, эффективность и комфорт вождения, помогая водителю принимать правильные решения и адаптироваться к дорожной ситуации. Связь между транспортными средствами, инфраструктурой и другими участниками дорожного движения также имеет решающее значение для повышения

безопасности автоматизированных транспортных средств и их полной интеграции в общую транспортную систему. Примером такой интеграции в настоящее время может служить движение грузового транспорта в колонне (ТС автоматически и безопасно следуют друг за другом на очень коротком расстоянии). Таким образом, особую научную и прикладную значимость приобретает разработка методов управления дорожным движением, основанная на взаимодействии интеллектуальной транспортной инфраструктуры с подключенными и/или автономными транспортными средствами, в т.ч. для решения задач прогнозирования и оценки транспортного потока, управления движением транспортных средств, координированного решения навигационных и логистических задач.

Можно выделить следующие задачи, решаемые в транспортных системах:

- задачи анализа, описывающие распределение транспортных потоков в сетях с известной конфигурацией;
- задачи прогнозирования, описывающие последствия изменения в транспортной сети, либо прогнозирующие изменения транспортных потоков во времени;
- задачи оптимизации и управления, используемые для оптимизации функционирования транспортных сетей.

Эти задачи решаются на различных уровнях ИТС, которые схематично показаны на рисунке 5.

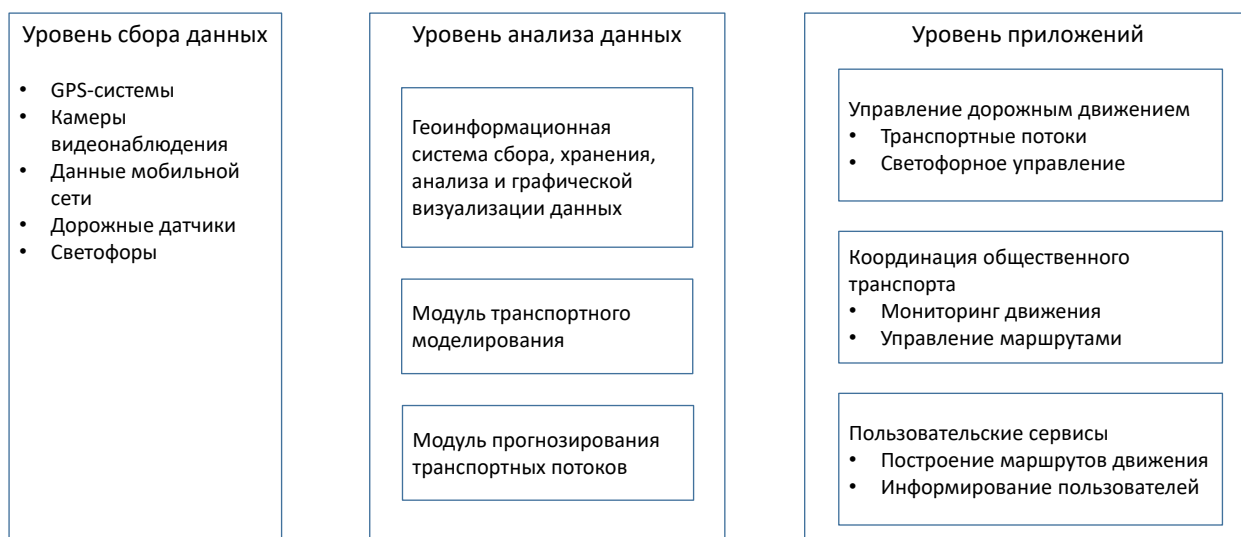


Рисунок 5 – Уровни в архитектуре ИТС

В данной диссертационной работе представлены разработанные математические методы и алгоритмы задач анализа, прогнозирования и управления в кооперативных ИТС, учитывающие гетерогенный состав транспортных средств (таблица 1).

Таблица 1 – Решаемые в рамках диссертационной работы задачи

Объект	Анализ	Прогноз	Управление и оптимизация
Транспортное средство	+	+	+
Транспортный поток	+	+	+
Инфраструктура	+	+	+

Рассматриваемые в рамках кооперативных ИТС задачи можно условно отобразить на временной диаграмме, показанной на рисунке 6. В рамках данной диссертационной работы рассматриваются задачи, относящиеся к блоку «Интеллектуальные транспортные системы» (показаны зеленым цветом на диаграмме). Задачи, связанные с моделированием транспортных потоков как ключевого этапа при проектировании транспортной системы умных городов в целом и улично-дорожных сетей в частности, в рамках данной диссертационной работы не рассматриваются. Также за рамками настоящей работы остаются задачи, связанные с автономным вождением отдельных транспортных средств, а также обеспечение информационной безопасности отдельного транспортного средства.



Рисунок 6 – Задачи, рассматриваемые в рамках ИТС

Поскольку диссертационная работа затрагивает каждое из указанных направлений, приводимый ниже обзор современного состояния составлен из нескольких разделов, каждый из которых отражает современное состояние конкретного направления исследований. В частности, в разделе 1.2 представлен обзор современного состояния исследования задач анализа и прогнозирования транспортного потока. Раздел 1.3 рассматривает задачу директивного управления транспортным потоком путем светофорного управления на регулируемых перекрестках. В разделе 1.4 рассматриваются задачи косвенного управления транспортным потоком и информирования пользователей – участников дорожного движения, в частности, навигационные задачи и задачи прогноза событий. В разделе 1.5 рассматривается задача обеспечения информационной безопасности коммуникаций в кооперативных ИТС. Наконец, в разделе 1.6 представлены современные тенденции развития интеллектуальных транспортных систем.

1.2 Современные методы решения задач анализа и прогнозирования транспортного потока

Под моделированием (статическим прогнозированием) транспортных потоков понимают задачу построения математической модели, способной адекватно описывать параметры функционирования транспортной сети (интенсивности движения на всех элементах сети, потоки, средние скорости движения и т.д.), поведение участников транспортного потока, а также прогнозировать их изменения. Для анализа распространения транспортного потока могут использоваться модели, описывающие транспортные потоки с разной степенью детализации: макроскопические, мезоскопические, микроскопические модели.

Макроскопические модели транспортных потоков описывают динамику движения в усреднённых терминах: плотность потока, средняя скорость, поток и др. Как правило, модели данного типа выводятся из аналогии между транспортным потоком и потоком сплошных сред, поэтому их часто называют гидродинамическими. В [32] рассматривается задача математического моделирования транспортных потоков на основе теории равновесия, а также приводятся макроскопические и микроскопические модели движения. Вывод равновесных моделей распределения транспортных потоков в больших сетях, а также численные методы поиска равновесия в таких моделях представлен авторами в [33]. В [34] авторы представили модель равновесного распределения транспортных потоков, которая состоит из двух стадий: расчета матрицы корреспонденций и равновесного распределения транспортных потоков по путям. Первая стадия основана на энтропийной

модели, которая описывает потребности в объемах передвижения из одного района в другой, вторая – на принципе равновесия Нэша–Вардропа (каждый водитель выбирает кратчайший для себя путь), которая описывает распределение потребностей в перемещениях по возможным путям.

В [35] рассматриваются модели и алгоритмы распределения транспортных потоков в сети крупного города или агломерации. Проводится сравнительный анализ алгоритмов поиска равновесного распределения в транспортной сети. Для решения неоднозначности распределения соответствий по путям в равновесии используются энтропийные модели.

В работах [36, 37] представлена статистическая модель транспортных потоков, которая моделирует движение групп транспортных средств на автомагистралях значительной протяженности с использованием фундаментальной диаграммы транспортного потока. Предложенный подход объединяет достоинства макроскопического и микроскопического моделирования: высокую точность и низкие вычислительные затраты. В статье [38] описано математическое моделирование транспортных потоков в большой автомобильной сети. Используется статистическая модель транспортных потоков для полномасштабного моделирования функционирования больших транспортных систем в течение длительных интервалов времени. Предложен способ повышения эффективности движения на автомагистрали через ограничение входного потока путем управления въездами.

В работе [39] предложена обобщенная гидродинамическая модель транспортного потока второго порядка, которая основывается на выражении для относительной скорости распространения дорожного затора (волны сжатия). Было показано, что предложенная модель обобщает известные модели транспортного потока второго порядка.

Краткосрочное прогнозирование транспортного потока решает задачу прогнозирования состояния транспортных потоков на основе текущей и архивной информации о параметрах транспортных потоков. Большинство исследований по этой тематике сосредоточены на разработке методов для моделирования характеристик транспортных потоков (например, плотности или скорости потока). Обзор методов краткосрочного прогнозирования транспортного потока представлен в [40-42]. В статье [42] проведён анализ современного состояния исследований краткосрочного прогнозирования параметров транспортных потоков по различным критериям, включая используемые технологии сбора данных, модели прогнозирования, прогнозируемые характеристики транспортных потоков и т.д. В [43, 44] авторы исследовали различные модели глубокого

обучения на предмет их применимости в современных ИТС, описали проблемы и возможные направления будущих исследований.

Учитывая неоднородность и динамические свойства транспортных потоков, а также сложные нелинейные взаимодействия участников дорожного движения друг с другом и с транспортной инфраструктурой, можно говорить о высокой степени изменчивости характеристик транспортных потоков в сети. Кроме того, состояние транспортных потоков на определённом сегменте сети сильно зависит от состояния потоков на смежных с ним сегментах, что затрудняет прогнозирование, особенно для краткосрочного интервала времени. Очевидно, что краткосрочное прогнозирование параметров транспортных потоков является сложной задачей, и, как следствие, для её решения в течение последних десятилетий было предложено много моделей, методов и алгоритмов. Известные подходы к прогнозированию используют различные эмпирические и теоретические методы, которые в целом могут быть классифицированы по трём категориям:

1) параметрические методы [40, 45], включая модели временных рядов (модель авторегрессии скользящего среднего ARMA, интегрированную модель ARIMA [46], а также её модификацию с учётом сезонности SARIMA [47-49], векторную модель авторегрессии VARMA [50]), модели пространства состояний (например, фильтр Калмана [51-54]), модели на основе динамического распределения трафика (DTA) [55] и т.д.;

2) непараметрические методы [56], включая модели искусственных нейронных сетей [57-60], метод k ближайших соседей (kNN) [61, 62], включая подходы, основанные на обработке больших данных [63, 64], метод опорных векторов [65-67], модель байесовской сети [68, 69] и др.;

3) гибридные методы, сочетающие параметрические и непараметрические методы [70, 71], например, комбинация модели ARIMA с другими моделями для повышения точности прогноза [72], объединение статистических методов и моделей нейронных сетей [73], а также комбинация моделей прогнозирования с методами предобработки данных [74, 75].

Современные научные исследования сосредоточены на использовании методов искусственного интеллекта и глубокого машинного обучения для решения задачи краткосрочного прогнозирования транспортного потока. Гибридная модель, сочетающая сверточную нейронную сеть и модель долгой краткосрочной памяти (LSTM), была исследована в [76-78]. В [79] авторы предложили глубокую нейронную сеть с механизмом внимания. Гибридная модель, объединяющая рекуррентные слои с механизмом внимания, исследовалась в [80]. В [81, 82] авторы использовали сверточную сеть с механизмом

внимания для краткосрочного прогнозирования. В [83] исследован ансамбль моделей, объединяющий ARIMA, LSTM, составные автоэнкодеры и учитывающий текущую транспортную ситуацию и долгосрочные тенденции. В [84] авторы использовали генеративно-состязательные сети для решения задачи прогнозирования. В предложенной схеме учитывались пространственные и временные зависимости в данных, обрабатываемые сверточными слоями (CNN) и слоями LSTM для повышения точности прогноза.

Основываясь на успешном применении операторов свертки в распознавании образов и анализе изображений, развитие получили графовые нейронные сети, использующие свертки на графах [85-87]. В [88] авторы преобразовывали графовые данные в структуры типа регулярная сетка, чтобы использовать обычные сверточные операции. Традиционные сверточные нейронные сети способны эффективно извлекать локальные паттерны, но могут применяться только для структур данных, представляющих собой регулярные сетки. Свертка на графах обобщает традиционную свертку и может использоваться для обработки данных на графовых структурах. Графовые нейронные сети используют матрицу смежности графов или матрицу Кирхгофа для описания структуры графа [85]. В [89] авторы представили полуконтролируемое обучение на графовых данных для задач классификации. Обобщенные графовые нейронные сети, основанные на использовании графового оператора Лапласа, были предложены в [86]. Их оптимизация с использованием полиномов Чебышева была исследована в [90]. Пространственно-временные графовые сверточные сети были предложены в [91] для решения задачи прогнозирования транспортного потока как временного ряда, однако в работе не учитывались пространственно-временные корреляции в данных. В [92] авторы предложили объединить операцию свертки на графах с нейронными сетями LSTM (долгой краткосрочной памятью) для выявления паттернов и прогнозирования транспортного потока. Однако авторы рассматривали только текущие данные, не учитывая архивные наблюдения. В [93-95] графовая нейронная сеть была расширена с использованием механизма внимания. Авторы предложили использовать три компонента для моделирования текущих, ежедневных и еженедельных периодических паттернов распространения транспортного потока. В [92, 96] использовалась комбинация графовой сверточной нейронной сети с LSTM слоем, учитывающая дневные и недельные паттерны распространения потоков. В [97] было реализовано несколько вариантов LSTM с блоком свертки на графе в рамках фреймворка энкодер-декодер для моделирования пространственно-временных зависимостей.

Все описанные методы имеют как преимущества, так и недостатки при работе в разных условиях, поэтому сложно сделать вывод, что один метод значительно превосходит

другие во всех режимах прогнозирования. Это объясняется в том числе и тем фактом, что точность моделей прогнозирования, которые обучаются и проверяются на небольших специальных наборах данных, зависит и от характеристик транспортных потоков в используемых выборках [27]. Кроме того, большинство существующих методов работают в автономном режиме, и поэтому имеют ограничения на используемые вычислительные ресурсы, а также возможности хранения и обработки данных. Как следствие, некоторые исследования были посвящены разработке систем прогнозирования, основанных на технологиях обработки больших данных [98, 64].

Следует отметить, что рассмотренные модели, методы и алгоритмы используют традиционные источники данных (детекторы транспортного потока, видеокамеры) для получения усредненных значений параметров транспортного потока, которые далее используются в качестве входных данных моделей для решения задачи краткосрочного прогнозирования транспортного потока.

Для использования данных от подключенных и/или автономных транспортных средств с целью повышения эффективности решения задачи краткосрочного прогнозирования транспортного потока был проведен обзор методов предварительной (упреждающей) маршрутизации отдельных транспортных средств. Предварительная маршрутизация учитывает маршруты движения транспортных средств (в т.ч. подключенных) для оценки планируемой загрузки транспортных сегментов. Планируемая загрузка сегментов дорожной сети, входящих в маршрут движения транспортного средства, используется для оценки скорости прохождения сегмента. В [99] авторы предложили децентрализованный подход к маршрутизации транспортных средств, основанный на мультиагентных системах. Когда транспортное средство выбирает маршрут, по которому оно должно следовать, оно информирует агентов инфраструктуры о выбранном маршруте. Эта информация используется для улучшения оценки маршрутов других агентов. В [100] авторы исследовали подход, при котором сегменты дорожной сети делятся на слоты в пространственной и временной областях, и каждое транспортное средство резервирует слот для выбранного маршрута. Считалось, что скорость автомобиля зависит от количества зарезервированных слотов на сегменте. В [101] авторы представили обзор соотношений скорости и плотности транспортного потока и описали новую обобщенную модель, основанную на микроскопическом моделировании движения. В [102] было предложено использовать логистические функции для описания модели скорость-плотность транспортного потока. В [103] авторы обсуждали фундаментальные стохастические диаграммы транспортных потоков и стохастические зависимости скорости от плотности. В

данных работах предполагается, что для оценки загрузки сегментов дорожной сети и маршрутизации транспортных средств имеется полная информация обо всех маршрутах движения в сети (т.е. доля подключенных транспортных средств в общем потоке составляет 100%).

Наконец, на третьем этапе анализа современного состояния исследований работ, посвященных краткосрочному прогнозированию транспортного потока, были рассмотрены статьи, которые используют данные от подключенных транспортных средств для решения задач анализа и прогнозирования.

В [104] рассматривается задача локализации транспортного средства. Авторами предложен метод объединения данных с различных датчиков, включая глобальные координаты, полученные из системы глобального позиционирования (GPS), и относительные координаты из инерциального измерительного блока (IMU) мобильного телефона, для определения траектории движения транспортного средства. Полученная траектория корректируется за счет применения вероятностной байесовской модели.

В [105] данные от подключенных транспортных средств использовались для оценки измерений параметров транспортного потока, которые используются для решения задачи прогнозирования. Предложенная схема использовалась для нахождения экономической стратегии управления транспортным средством, проезжающим перекресток [106]. Проблема минимизации времени и расхода топлива при проезде перекрестка также исследовалась в [107], было предложено использовать модель прогнозирования транспортного потока и модель управления светофорами для расчета оптимальной скорости подключенного транспортного средства на перекрестке. В [108] траектории транспортных средств типа CV использовались для оценки эффективности стратегий управления светофорами.

В [109] траектории подключенных транспортных средств использовались для оценки скорости движения на сегментах дорожной сети и генерации вектора признаков, который использовался в качестве входных данных нейронной сети. Аналогичный подход для обновления данных о состоянии транспортного потока с использованием информации от транспортных средств типа CV был использован в [110]. Авторы применили фильтр Калмана для обновления оценки значений параметров транспортного потока, модифицированные данные были использованы в качестве входных данных для модели LSTM. В [111] авторы рассматривали проблему построения профиля скорости путем оценки будущих траекторий транспортных средств и обновления профиля скорости с

учетом прогнозируемых данных скорости с использованием метода взвешенного скользящего среднего.

Обзор показывает, что большая часть работ использует традиционные источники данных (детекторы транспортного потока, видеокамеры) для получения и прогнозирования усредненных значений параметров транспортного потока. Данные от подключенных транспортных средств использовались или для оценки/уточнения значений параметров транспортного потока, или для прямой маршрутизации этих транспортных средств. Разработка методов и алгоритмов прогнозирования, учитывающих как традиционные данные о состоянии транспортных потоков (усредненные значения скорости/плотности, данные с видеокамер на перекрестках), так и данные от подключенных транспортных средств, определяет научную значимость представленного в диссертационной работе направления исследования.

1.3 Современные методы решения задач директивного управления транспортным потоком

Тенденция роста количества транспортных средств обуславливает необходимость в разработке новых эффективных средств контроля транспортного потока, в т.ч., в составе интеллектуальных транспортных систем. Одним из наиболее распространенных способов решения проблем с заторами на дорогах является применение регулируемого управления на перекрестках. Существуют различные методы управления светофорами. Один из простейших подходов к управлению светофорным объектом основан на переключении фаз по фиксированному, заранее определенному расписанию. Ввиду того, что интенсивность движения транспортных средств с течением времени часто меняется, такой подход малоэффективен. Более сложные алгоритмы максимизируют пропускную способность перекрестка, используя различную информацию о текущей дорожной ситуации на прилегающих к перекрестку проезжих частях.

В связи с развитием автономных транспортных средств перспективным подходом к решению задач, связанных с эффективным контролем транспортного потока, является контроль движения подключенных автономных транспортных средств на регулируемых светофорами перекрестках. Возможность автономного ТС осуществлять обмен информацией с другими участниками движения и дорожной инфраструктурой позволяет безопасно управлять траекторией движения такого транспортного средства на перекрестке. Совместное управление сигналами светофоров и траекториями движения транспортных средств позволит снизить временные затраты на осуществление транспортных

корреспонденций, потребление топлива и повысить эффективность использования транспортной инфраструктуры в целом.

Обзор современного состояния исследований по теме совместного управления сигналами светофоров и траекториями движения транспортных средств состоит из трех этапов. На первом этапе рассмотрены основные подходы к управлению сигналами светофоров, включая детерминированные методы и методы на основе машинного обучения с подкреплением. Далее проведен обзор литературы, посвященный задаче построения траектории движения транспортных средств. На третьем этапе рассмотрены работы, посвященные решению задачи совместного управления.

1.3.1 Управление сигналами светофорных объектов

Возможность использования систем адаптивного управления движением транспортных средства посредством сигналов/тактов светофоров используется с начала 80х годов прошлого века. С этого момента широкое распространение на практике получили системы SCOOT [112, 113] и SCATS [114], которые использовали информацию о числе «проехавших» мимо соответствующего детектора транспортных средств, и на основании заранее разработанного эмпирического плана переключали на «следующую» фазу светофора в рамках определённого цикла работы светофора. Подобная жёсткая схема адаптации оказалась недостаточно эффективной, что привело к появлению других методов управления. Дополнительной причиной этому является также значительно возросшее количество источников данных, которые могут использоваться для решения задачи координированного управления сигналами/фазами светофоров: данные с камер видеонаблюдения, траектории движения из навигационных приложений, информация от подключенных транспортных средств, обменивающихся данными с объектами инфраструктуры. Одновременно с ростом числа источников, в последнее время лавинообразно возрос и объём доступных для анализа данных, что позволяет для решения задач управления светофорами в интеллектуальных транспортных системах использовать методы машинного обучения и обработки «больших данных» [64]. Ниже представлен краткий обзор основных подходов и современных методов, используемых для управления сигналами/тактами светофоров.

Фазы светофорного регулирования могут выбираться как произвольно, так и в рамках некоторой предопределённой последовательности фаз – светофорного цикла. Одной из наиболее простых систем управления – контроллеров – является широко распространённый способ Uniform, подразумевающий переключение фаз светофора через

заранее фиксированные промежутки времени и выбор фаз из детерминированного светофорного цикла. Если выбор фазы светофорного регулирования происходит на основании данных движения транспортных средств, то говорят о системе адаптивного управления дорожными сигналами. При этом следует различать системы, где светофорный цикл предопределён (то есть адаптивно выбирается только момент переключения, как в вышеуказанных системах SCOOT и SCATS), и где выбор следующей фазы производится без ограничений. Системы подобного типа можно условно разделить на эмпирические детерминированные алгоритмы и алгоритмы на основе машинного обучения. Последние для управления движением активно используют методы обучения с подкреплением (англ.: RL – reinforcement learning). Среди них в последнее время наибольшее внимание уделяется методам с использованием глубоких нейронных сетей (DNN).

Одной из первых работ, посвященных проблеме управления сигналами/фазами светофорного объекта, можно считать работу Вебстера [115]. В работе был предложен метод расчета длины светофорного цикла и фазового цикла для одиночного перекрестка. Развитие метода для использования стратегий координированного управления несколькими перекрестками было продолжено в [116, 117]. Ранние методы управления светофорами использовали так называемые стратегии с фиксированным временем, при которых длительность светофорного цикла и распределение фаз фиксировались и не зависели от состояния движения на сегментах сети, что ограничивает применимость моделей [118]. Следующим этапом развития методов управления светофорами было применение адаптивных схем управления, зависящих от наблюдаемого транспортного потока на перекрестке. Такие схемы позволяют выбирать новую фазу светофорного цикла в зависимости от, например, длины очереди транспортных средств на перекрестке [119]. В [120] авторы предложили адаптивную схему управления, учитывающую введенную величину «давления», которая определяется как разность между количеством въезжающих и выезжающих транспортных средств на перекрестке. Данный метод, согласно работе [121], смог превзойти более современные подходы на небольшой синтетической сети. В [122] был предложен адаптивный алгоритм управления, основанный на выборе фазы светофорного цикла, прогнозируемый транспортный поток для которой будет максимальным.

Эмпирические методы управления, основанные на правилах, использовали набор предопределенных правил для управления сигналами светофора, таких как управление с фиксированным временем [115, 116], активируемое управление [118, 123, 124] и адаптивное управление [125, 126, 120]. Методы, основанные на правилах, просты и легко

реализуемы, но они могут быть негибкими и неэффективными в средах, где транспортная ситуация быстро меняется.

Второй класс традиционных методов управления – это методы, основанные на оптимизации [127, 128]. Эти методы рассматривают задачу управления сигналами светофоров как многокритериальную задачу оптимизации и используют алгоритмы математической оптимизации для нахождения оптимальной синхронизации сигналов с учетом различных ограничений и задач, таких как минимизация задержек движения и выбросов и максимизация пропускной способности дорог. В [129] рассматривается математическая модель управления светофорами на перекрестках, основанная на системе нелинейных уравнений в конечных разностях. Управление транспортным потоком осуществляется путем согласования активных фаз светофоров. В работе [130] рассматривается задача оптимального светофорного управления транспортным потоком на городской многополосной дороге. Используется макроскопический подход, в котором транспортный поток описывается нелинейной гиперболической системой второго порядка. Авторами было установлено, что предложенный подход обеспечивает высокую эффективность адаптивного управления светофорами для предотвращения перегруженности автомагистрали. Показано, что использование адаптивного управления на въездах на МКАД может уменьшить или устранить дорожные заторы на самой трассе за счет образования небольших очередей на въездах. Методы, основанные на оптимизации, более гибкие и могут обрабатывать изменяющиеся условия движения, но они сложны в вычислительном отношении и могут потребовать точных моделей среды.

Третья группа методов использует модельно-предикативное управления (MPC, Model Predictive Control) для прогнозирования различных характеристик транспортного потока, которые будут использоваться для оптимизации фаз сигналов светофоров [131-135]. В [133] авторы использовали прогнозирование плотности транспортного потока для расчета оптимального цикла фаз. В [131] было предложено использовать простую макроскопическую модель транспортного потока в качестве входных данных для модели управления. Модель точечной очереди была рассмотрена в [132] для минимизации длины очереди транспортных средств на перекрестке. В [135] авторы прогнозировали время прибытия каждого транспортного средства на перекресток на основе модели глубокой нейронной сети и использовали эту информацию для максимизации транспортного потока через перекресток.

Более детальное рассмотрение представленных и других ставших уже «классическими» эмпирических методов адаптивного и неадаптивного управления сигналами светофоров можно найти в обзорах [118, 136, 137].

В последние два десятилетия для адаптивного управления сигналами/фазами светофоров были использованы различные методы искусственного интеллекта и машинного обучения: методы на основе нечёткой логики [138], интеллекта роя [139], генетические алгоритмы [140-142], методы на основе обучения с подкреплением [143-152] и др. Среди них решения на основе обучения с подкреплением являются, по мнению многих авторов, наиболее перспективными.

Для удобства изложения далее представлено описание принципов методов обучения с подкреплением (далее – RL-методов) в типовой нотации.

Пусть S обозначает множество состояний некоторого объекта (для задачи управления сигналами/фазами светофора под состоянием может пониматься расположение и характеристики движения транспортных средств в окрестности перекрёстка или какие-либо производные числовые характеристики от указанных данных), A – множество возможных действий. Упрощённо, задача состоит в построении такого отображения $S \rightarrow A$, которое для каждого конкретного состояния $s \in S$ давало бы то действие $a \in A$, которое необходимо предпринять в указанном состоянии в некотором смысле оптимальным образом. Построение указанного отображения производится путём обучения, то есть анализа некоторой последовательности вида:

$$S_{start} = s_0, a_0, s_1, a_1, \dots, s_{N-2}, a_{N-2}, s_{N-1} = S_{terminal}.$$

Учитывая возможную статистическую природу перехода состояний при конкретных действиях, для решения указанной проблемы обычно используют аппарат Марковских процессов принятия решений:

$$\langle S, A, P, R, \gamma \rangle, \quad (1)$$

где P определяет вероятности перехода из состояния в состояние при конкретных действиях в некоторый момент времени t , т.е.:

$$P_a(s_{prev}, s_{next}) = P_a(s_{t+1} = s_{next} | s_t = s_{prev}, a_t = a).$$

Величина $r_t \equiv R_a(s_t, s_{t+1})$ характеризует «награду» за переход из состояния s_t в состояние s_{t+1} под действием a , а коэффициент γ используется для формальной записи целевой функции RL-задачи в виде:

$$R = \sum_{t=0}^{\infty} r_t \gamma^t. \quad (2)$$

В рамках представленного формализма (1) решение заключается в нахождении т.н. политики/стратегии $\pi(s, a) \in R[0,1]$, которая бы численно характеризовала в каждый момент времени и состояния s_t предпочтительность соответствующего действие a_t . Типовое решение в этом случае заключается в построении т.н. Q-функции, которая задаётся как ожидаемое значение целевой функции (2) для конкретной пары состояние-действие:

$$Q^\pi(s, a) = E\{R | s_t = s, a_t = a, \pi\}$$

и удовлетворяет уравнению Беллмана.

RL-методы можно разделить на несколько типов в зависимости от подходов к обучению: методы, основанные на оптимизации функции полезности, методы, основанные на оптимизации политики, и методы класса «actor-critic», сочетающие оба подхода.

Метод Q-обучения (Q-learning) – это метод, основанный на оптимизации функции полезности, который оценивает ожидаемое вознаграждение за выполнение определенного действия в определенном состоянии с использованием Q-значений [153]. Этот подход можно использовать в сценариях с небольшим количеством состояний и действий. Чтобы справиться с проблемой многомерного пространства состояний, в подходе Deep Q-Networks (DQN) было предложено объединить Q-обучение с глубокими нейронными сетями для представления Q-значений [154-159]. Комбинация модели DQN с алгоритмом управления исследовалась в [154]. В [155] авторы предложили совместную модель на основе глубокого RL-обучения, которая управляет сигналами светофоров в регионе с использованием нескольких региональных агентов и централизованного глобального агента. RL-алгоритм, который использует параметры транспортного потока для изучения оптимальной политики, описан в [156]. В [157] авторы предложили модель DQN, представление пространства состояний и формулировку награды для оптимизации управления светофором в гетерогенном потоке с подключенными транспортными средствами. В подходе Double DQN было предложено использовать две отдельные нейронные сети для оценки Q-

значений и выбора наилучшего действия [160, 161]. Этот подход решает проблему переобучения и демонстрирует более стабильное поведение в сложных сценариях.

Второй класс RL-алгоритмов включает методы на основе оптимизации политики. Алгоритмы на основе политики напрямую оптимизируют политику, которая представляет собой распределение вероятностей пар действие-состояние, путем обновления параметров политики на основе градиента ожидаемого вознаграждения [162-164]. В [162] авторы исследовали алгоритм глубокого детерминированного градиента политики. В [163] исследовался алгоритм проксимальной оптимизации политики (PPO, Proximal Policy Optimization) с переменными временными интервалами для фаз светофора. В [164] авторы предложили модифицированный алгоритм PPO, направленный на лучшую адаптацию к условиям дорожного движения.

Методы класса «actor-critic» представляют собой комбинацию алгоритмов, основанных на оптимизации функций полезности и политики. В этих методах actor представляет собой функцию политики, которая используется для выбора действий, а critic представляет функцию полезности, которая оценивает действия, совершаемые агентом [121, 150, 165-167]. В [150] авторы представили многоагентный RL-метод для адаптивного регулирования, основанный на алгоритме A2C (Advantage Actor-Critic). В [166] авторы интегрировали модель глубокой нейронной сети, которая оценивает состояние окружающей среды по серии изображений перекрестка, в модель actor-critic. Оценка различных представлений состояния для задачи адаптивного управления с использованием асинхронного алгоритма A2C (A3C, Asynchronous Advantage Actor-Critic) была представлена в [121].

Применение RL-методов для решения проблемы адаптивного управления сигналами светофоров требует, наряду с выбором конкретного метода и, при использовании, структуры DNN сети, информации о том, как описывается состояние дорожного движения в текущий момент времени и как определяется «награда». Собственно, указанные выше работы по использованию RL-методов для адаптивного управления светофором [143-151] именно этим и отличаются. Среди возможных вариантов описания состояния можно выделить [125]: длину очереди, время ожидания, количество ТС, величину задержки, скорость, длительность фазы светофора и саму фазу, загруженность, относительные позиции ТС и др. Среди возможных вариантов задания величины «награды» можно выделить [125]: длину очереди, время ожидания, изменение задержки, скорость, количество остановок, пропускную способность, частоту смены сигнала, давление и др.

Для обучения RL-методов используют следующие системы моделирования: AIMSUN, CORSIM, MATSim, Paramics, VISSIM, GLD, SUMO, CityFlow, AIM [125, 168].

Несмотря на активное развитие методов на основе обучения с подкреплением для задачи адаптивного управления сигналами светофоров, эти методы имеют ряд недостатков. Во-первых, обучение RL-алгоритмов может потребовать много времени и вычислительных ресурсов, особенно для больших и сложных реальных сценариев. Во-вторых, RL-алгоритмы часто требуют точных моделей среды для изучения оптимальной политики. Однако в реальных сценариях получение точной и полной информации о состоянии окружающей среды может оказаться сложной задачей. Наконец, может быть сложно обобщить RL-алгоритмы на новые ситуации, особенно когда пространства состояний и действий велики или когда среда значительно меняется.

1.3.2 Построение траектории движения транспортных средств

На городских перекрестках транспортные средства, прибывающие на красный сигнал светофора, часто останавливаются максимально близко к стоп-линии перекрестка, а затем начинают разгоняться, когда светофор загорится зеленым. Такой характер движения увеличивает время в пути и расход топлива транспортных средств, а также снижает пропускную способность перекрестка. При соответствующем планировании траектории автономные транспортные средства могут заранее снизить скорость, чтобы избежать остановок и очередей у перекрестка, снизить расход топлива. Применение методов и алгоритмов формирования траектории движения автономных подключенных транспортных средств направлено на улучшение различных характеристик транспортного потока. В качестве таких характеристик часто используются: безопасность [169], комфорт [170], эффективность [171], расход топлива [172], воздействие на окружающую среду [173].

В [174] описана модель прогнозирования времени прохождения очереди транспортных средств на регулируемом перекрестке. С помощью этой модели авторы предложили подход к выбору оптимального скоростного режима для сокращения количества остановок подключенных транспортных средств.

В работах [175, 176] предложены методы построения траекторий движения автономных подключенных транспортных средств на нерегулируемых перекрестках. В рамках данной группы методов управление движением транспортных средств осуществляется не только на входящей в перекресток дорожной полосе, но и непосредственно на перекрестке. Такой подход к управлению движением транспортных средств допускает возможность резкого торможения транспортных средств с целью

избегания столкновения, так как характер движения ТС до въезда в контролируемую зону перекрестка не учитывается. Так же стоит отметить, что данная группа методов не допускает наличие на перекрестке управляемых человеком транспортных средств и пешеходных переходов.

Авторы [177] предложили эвристический алгоритм, который позволяет эффективно сглаживать траектории потока транспортных средств, приближающихся к регулируемому перекрестку, путем управления их скоростью движения. Этот алгоритм формирует для каждого транспортного средства на перекрестке траекторию движения, состоящую из нескольких сегментов аналитических квадратичных кривых. Таким образом, он эффективно строит большое количество траекторий транспортных средств с учетом накладываемых физических ограничений, траекторий движения впереди идущего транспортного средства и сигналов светофорного объекта на перекрестке. В работе [169] данный алгоритм используется для построения оптимальных траекторий движения. В этой работе целевая функция оптимизации, используемая для построения оптимальных траекторий, учитывает расход топлива, время движения по дорожной полосе и безопасность движения.

Способ управления движением транспортного средства, при котором минимизируется расход топлива и негативное воздействие на окружающую среду, называют эко-вождением (eco-driving) [178]. Динамический подход к эко-вождению представлен в [179]. Авторы разработали алгоритм формирования тригонометрического профиля скорости транспортного средства. Данный алгоритм не только минимизирует расход топлива и количество выбросов углекислого газа в атмосферу, но и не допускает резкого торможения/ускорения, что способствует повышению комфорта езды. В [178] разработана модель эко-вождения для гибридных электромобилей в условиях перегруженного городского движения. При формировании траектории движения транспортных средств данная модель учитывает прогнозируемую загруженность перекрестка.

В [180] сформулирована двухэтапная модель, которая оптимизирует траекторию продольного и поперечного поведения автономных транспортных средств вдоль регулируемой магистрали в условиях смешанного движения.

1.3.3 Совместное управление сигналами светофоров и траекториями движения транспортных средств

Заключительная часть обзора современного состояния исследований по теме управления транспортным потоком в ИТС посвящена анализу литературы, рассматривающей задачу координированного управления движением автономных/подключенных транспортных средств и сигналами светофоров.

Обзор методов совместного управления в среде с подключенными транспортными средствами был представлен в [181, 182]. Авторы представили обзор существующих подходов, типовых сценариев управления транспортным потоком, а также стратегий активного и косвенного управления, и пришли к выводу, что координированное управление в гетерогенном транспортном потоке может значительно повысить эффективность управления дорожным движением. В [183] авторы представили обзор методов и алгоритмов управления сигналами светофора в условиях гетерогенного транспортного потока, включая подходы, основанные на автономном управлении перекрестком и оптимизации сигналов.

В [135] авторы предложили метод кооперативного управления светофором, в котором алгоритм оптимизации траектории учитывает предсказанную фазу светофора. Однако адаптивный алгоритм управления не учитывал траектории движения транспортных средств.

Авторы [184] представили договорно-приоритетную систему управления транспортом. В предлагаемой системе управления транспортом транспортные средства типа CAV обмениваются информацией о своем текущем состоянии движения и состоянии ближайших неподключенных транспортных средств друг с другом и с центром управления движением. Эта информация используется для оптимизации транспортного потока в транспортной сети системой управления. В [185] авторы представили метод решения задачи оптимизации транспортного потока и применили его для оптимизации энергопотребления транспортных средств в транспортной сети. В работе предполагается, что все транспортные средства в транспортной системе являются автономными, а оптимизация транспортного потока достигается за счет взаимодействия адаптивного алгоритма управления сигналами светофора и алгоритма построения траектории движения транспортного средства.

Один из методов построения траекторий движения транспортных средств в гетерогенном транспортном потоке предложен в [186]. Однако представленный подход использовал предопределенный светофорный цикл. В [187] предлагается система, которая включает в себя как управление сигналами светофоров, так и управление траекториями

движения транспортных средств. Предлагаемая структура объединяет компоненты восприятия, прогнозирования, планирования и оптимизации и направлена на повышение эффективности управления гетерогенным транспортным потоком. На первом этапе алгоритм, основанный на рекуррентной нейронной сети LSTM, использовался для прогнозирования поведения водителя. На втором этапе эта информация использовалась в качестве входных данных для модели оптимизации сигналов светофоров. Однако в алгоритме не предполагалась смена полос движения, а экспериментальные исследования алгоритма проводились на сценарии, содержащем один изолированный перекресток. В [181] обобщены методы управления сигналами светофоров, активного управления автономными подключенными транспортными средствами и косвенного управление обычными транспортными средствами для изучения стратегии координированного управления несколькими перекрестками для гетерогенного транспортного потока. Согласно [188], решение задачи координированного управления позволит значительно повысить эффективность использования транспортной инфраструктуры.

В [189] авторы предложили метод совместного управления в условиях гетерогенного транспортного потока для оптимизации синхронизации фаз сигналов светофора и траекторий движения транспортных средств типа CAV для снижения потребления топлива. Однако экспериментальное исследование метода было проведено на синтетическом сценарии с одним перекрестком. В [190] предложена методология координированного управления сигналами светофора и оптимизации траектории, сформулированная в виде смешанно-целочисленной нелинейной программы. Предлагаемый подход применим только к перекресткам с движением с левым поворотом.

Приведенный обзор современного состояния движения подтверждает, что задача координированного управления движением автономных/подключенных транспортных средств и сигналами светофоров является актуальной проблемой. Существующие исследования показывают, что координированное управление в гетерогенном транспортном потоке может значительно повысить эффективность управления дорожным движением. Однако существующие подходы к решению этой задачи обладают рядом недостатков, например, рассматривают только перекрестки определенной структуры, не учитывают адаптивное изменение сигналов светофоров, либо проводят исследования на синтетических сценариях моделирования малого размера.

1.4 Современные методы решения задач косвенного управления транспортным потоком и информирования в ИТС

Прогнозирование и директивное управление транспортными потоками являются ключевыми задачами, решаемыми в ИТС, т.к. позволяют повысить эффективность использования транспортной инфраструктуры, уменьшить время совершения транспортных корреспонденций, расход топлива и уровень загрязнения окружающей среды. Однако для пользователей – участников дорожного движения не менее важным является эффективное решение задач косвенного управления транспортным потоком и информирования в ИТС: точное прогнозирование движения общественного транспорта, решения навигационных задач и т.д. Обзор современного состояния исследования по этим проблемам приведен в следующих подразделах.

1.4.1 Краткосрочное прогнозирование движения общественного транспорта

В последнее десятилетие проблемы прогнозирования движения общественного транспорта являются одними из наиболее популярных практических задач в интеллектуальных транспортных системах. Как следствие, для их решения были разработаны различные модели, методы и алгоритмы, которые можно разделить на следующие группы:

- статистические модели на основе архивных данных [191];
- модели линейной регрессии [192, 193];
- непараметрические регрессионные модели [194, 56];
- модели временных рядов [195];
- модели искусственных нейронных сетей [196, 197];
- модели на основе метода опорных векторов [198, 199];
- гибридные модели.

В ранних подходах к прогнозированию времени прибытия общественного транспорта использовались модели на основе архивных данных [191] и модели линейной регрессии [192, 193]. Основным недостатком таких моделей является их медленная реакция на изменения дорожной ситуации, что делает результаты этих моделей недостоверными в случае возникновения дорожных заторов или дорожно-транспортных происшествий. Однако их простота, как относительно потребляемых вычислительных ресурсов, так и в подготовке данных, обеспечивает их широкое практическое применение. При проведении исследований эти модели используются только в качестве базовых для сравнения с предлагаемыми алгоритмами.

Непараметрические регрессионные модели просты из-за отсутствия оцениваемых параметров. Одной из популярных моделей непараметрической регрессии является метод k ближайших соседей [194, 56, 64]. Однако требование большого размера выборки накладывает ограничение на использование этого метода в режиме реального времени. В [200] авторы использовали кластерный подход для оценки распределения времени прохождения дорожных сегментов транспортными средствами.

Модели временных рядов предполагают существование тренда в наборе данных и прогнозируют следующие значения, исходя из тренда [195]; таким образом, эти модели очень зависимы от сходства между архивными и реальными данными.

Фильтры Калмана, благодаря своим способностям поддерживать состояние между измерениями, использовались как в качестве независимых моделей прогнозирования времени прибытия общественного транспорта [201], так и в сочетании с другими моделями [202]. Из-за линейности этих моделей они являются вычислительно простыми, но ограничение в представлении и прогнозировании сложных нелинейных пространственно-временных отношений одновременно является и их недостатком. Другим недостатком фильтра Калмана является неспособность оценки состояния для нескольких последовательных дорожных сегментов и / или временных шагов.

Последние исследования заявляют о хороших результатах, достигнутых в решении транспортных проблем с использованием методов машинного обучения: моделей искусственных нейронных сетей (ИНС) и метода опорных векторов (SVM). Модель искусственной нейронной сети [196, 197, 203] является одной из наиболее часто используемых моделей для прогнозирования времени прибытия. Эти модели способны моделировать сложные нелинейные отношения между временем прохождения дорожных сегментов и независимыми переменными, которые характеризуют транспортный поток. Вычислительная сложность полносвязных нейронных сетей ограничивает количество нейронов в сети и сложность распознаваемых паттернов движения. Как следствие, некоторые исследования были посвящены разработке гибридных моделей. В [202] представлена комбинированная модель ИНС с фильтром Калмана для снижения вычислительной сложности и быстрой адаптации к реальной дорожной ситуации. В [204] комбинировался метод k -средних и ИНС. В [203] применялся Байесовский подход для выбора нейронных сетей с различными архитектурами, итоговая оценка прогноза формировалась как взвешенная сумма элементарных прогнозов.

Метод опорных векторов [198, 205, 206] – это метод машинного обучения, используемый для решения задач классификации и регрессии. В нескольких работах было

показано, что метод опорных векторов обеспечивает схожую точность прогнозирования по сравнению с моделями искусственных нейронных сетей [207], но эти модели имеют высокую вычислительную сложность. В [199] предложена гибридная модель, объединяющая SVM с фильтром Калмана. В [206] использовался генетический алгоритм для подбора параметров SVM.

Другие подходы к решению задачи нахождения времени прибытия общественного транспорта включают использование иерархической регрессии [208], метода случайного леса [209].

В настоящее время, благодаря развитию технологий машинного обучения, многие исследователи делают упор на моделях глубокого обучения. В [29] используется сеть глубокого обучения с долгой краткосрочной памятью (LSTM), учитывающая пространственно-временные корреляции данных. Модель LSTM для прогнозирования времени движения по автомагистрали была использована в [210, 211]. В [212] для прогноза движения предложено использовать сверточные нейронные сети на графах.

Несмотря на популярность рассматриваемой проблемы, в большинстве работ рассматривается малое количество признаков, описывающих пространственные и временные характеристики дорожной ситуации (например, скорость на текущем и предыдущем сегментах транспортной сети). Кроме того, сравнение алгоритмов проводится на различных наборах данных, поэтому нельзя сделать однозначный вывод о том, что одна модель всегда превосходит другие. В то же время, использование разнородной информации, прямо или косвенно описывающей транспортную ситуацию, включая данные реального времени и статистические данные о движении транспортных средств (в т.ч. подключенных) позволит повысить качество прогнозирования времени прибытия общественного транспорта.

1.4.2 Решение навигационных задач

В последние десятилетия проблеме нахождения кратчайшего пути в транспортных сетях было посвящено много исследований, в основном фокусирующихся на нахождении наименьшего ожидаемого времени движения при различных допущениях. Обычно проводят следующую классификацию моделей в зависимости от предполагаемого типа веса дорожного сегмента:

- константный вес;
- зависящий от времени;
- стохастический;

- стохастический зависящий от времени.

Классические модели рассматривают время прохождения дорожного сегмента как константное или зависящее от времени. В зависящих от времени стохастических сетях время прохождения сегментов представляется как случайная величина с функцией распределения, зависящей от времени [213-215].

Алгоритмы маршрутизации в стохастических сетях предоставляют пользователям либо априорный оптимальный путь [216] или адаптивный маршрут движения [217]. В первом случае оптимальный путь определяется до момента отправления и не изменяется в процессе движения. При нахождении адаптивного маршрута путь может измениться при достижении очередного узла сети (перекрестка дорожной сети) после получения актуальной информации о времени прохождения сегментов дорожной сети.

В зависимости от используемого критерия оценки надежного кратчайшего пути используемые модели могут быть классифицированы следующим образом:

1) модели с наименьшим ожидаемым временем движения (Least Expected Time – LET) [214, 218, 213]. Данные модели рассматривают ожидаемое время прохождения сегментов дорожной сети как критерий оценки для сравнения возможных путей;

2) модели α -надежного пути. Целевая функция модели – минимизация интервала времени, необходимого для обеспечения прибытия в конечную вершину (пункт назначения) к выбранному моменту времени с заданной вероятностью α [216, 219, 220];

3) модели с наибольшей надежностью движения [217, 221, 222]. Задача заключается в максимизации вероятности прибытия в конечную вершину (пункт назначения) в течение заранее определенного интервала времени (бюджета поездки). Часто эта проблема обозначается как SOTA (Stochastic On-Time Arrival).

Приведённые три вида целевых функций могут быть обобщены двумя видами задач: задача наименьшего ожидаемого времени движения (least expected time path problem – LETPP) и задача надёжного кратчайшего пути (reliable shortest path problem – RSPP). Путь с априорным (или адаптивным) наименьшим ожидаемым временем движения может быть не подходящим для движения с наименьшим риском, т.к. может иметь высокую дисперсию времени движения. Эмпирические исследования показывают, что участники движения стараются не только снизить время движения, но и дисперсию времени движения [223].

Рассмотрим указанные модели подробнее. В [214] была рассмотрена задача нахождения пути с наименьшим ожидаемым временем движения в зависящих от времени стохастических сетях. В [213] была определена политика маршрутизации, определяющая, какой узел сети необходимо выбрать на следующем шаге с учетом актуального времени

прохождения дорожных сегментов и текущего момента времени. Многокритериальный алгоритм A^* для решения LET-задачи был предложен в [224]. Однако, зачастую необходимо не только минимизировать ожидаемое время в пути, но и учитывать потенциальный риск опоздания, так как путь с наименьшим ожидаемым временем движения может иметь высокую дисперсию.

Задача нахождения α -надежного кратчайшего пути была рассмотрена в [225]. Время прохождения дорожных сегментов задавалось в виде коррелированных случайных величин с логнормальным законом распределения. Для нахождения надежного пути предложен гибридный генетический алгоритм на основе метода сопоставления моментов для аппроксимации параметров распределения времени прохождения пути.

В статье [226] исследовалась проблема нахождения k -надежных кратчайших путей в стохастических сетях в условиях неопределенности времени прохождения. Задача k -надежных кратчайших путей расширяет классическую задачу k -кратчайших путей без циклов на стохастические сети, явно рассматривая надежность времени движения. Применялась концепция отклонения в пути для нахождения α -надежных кратчайших путей, для улучшения быстродействия использовался модифицированный алгоритм A^* .

В статье [221] авторы сформулировали SOTA-проблему как задачу стохастического динамического программирования, для ее решения был применен стандартный метод последовательных аппроксимаций. Однако данный метод обладает плохой сходимостью.

В статье [217] предложено точное решение SOTA-проблемы для сетей, в которых время прохождения сегментов является положительной величиной. Как и в работе [221], одним из этапов алгоритма является вычисление свертки, что является основной вычислительно сложной задачей. В общем виде свертка не может быть вычислена аналитически, и поэтому требуется схема дискретной аппроксимации.

В [227] было предложено решение задачи нахождения априорного надежного пути с наибольшей надежностью движения, представлена адаптивная схема дискретизации для повышения эффективности свертки.

Двухэтапный алгоритм решения задачи нахождения надежного пути предложен в [228]. На первом этапе оценивались верхние и нижние границы надежности времени движения, учитывая наименьшее ожидаемое время движения. На основе оцененного диапазона надежности было установлено эффективное правило доминирования и монотонное свойство целевой функции. На втором этапе использовался многокритериальный подход к установлению меток для определения наиболее надежного пути при различных сценариях принятия рисков.

В [229] проблема смоделирована в виде Марковского процесса принятия решений. Предложен метод q-обучения, для определения функции полезности использовалась динамическая нейронная сеть. Заявлено, что метод может хорошо масштабироваться для крупномасштабных дорожных сетей.

Задача нахождения надежного пути в расширенной постановке – минимизация функции риска перерасхода бюджета поездки – рассматривается в [230]. Представлены эффективные процедуры решения (на основе быстрого преобразования Фурье и алгоритма построения выпуклой оболочки) для вычисления приближенно-оптимальных стратегий навигаций.

В работе [231] рассмотрена задача нахождения надежного пути в двух постановках. Формулируются эквивалентные задачи смешанного целочисленного линейного программирования, для их решения используется метод Лагранжевых релаксаций. Комбинированный метод Лагранжевых релаксаций для решения задачи маршрутизации предложен в [232]. В качестве критерия оптимизации рассматривается минимизация суммы среднего времени и среднеквадратического отклонения времени движения с учетом корреляции времени прохождения дорожных сегментов.

Повышения качества работы алгоритма навигации и снижения ошибки прогноза времени движения можно достичь, прогнозируя транспортные потоки на сегментах дорожной сети [64]. В работе [233] предложена модификация решения [217], учитывающая актуальную и прогнозную информацию о транспортных потоках в сети.

Несколько исследований были посвящены ускорению работы алгоритма решения задачи SOTA. В статье [234] авторы представили несколько методов ускорения алгоритма решения SOTA-проблемы, включая усовершенствованные алгоритмы вычисления свертки с помощью быстрого преобразования Фурье и алгоритмы вычисления свертки с нулевой задержкой, а также методы определения оптимального порядка вычисления стратегии навигации. В [235] описана эвристика поиска адаптивного маршрута движения в стохастической сети. Представленный метод позволяет обеспечить наиболее вычислительно эффективную стратегию нахождения пути для общих распределений вероятностей. В [236] рассмотрены стохастические варианты двух методов предварительной обработки графа для решения задачи нахождения детерминированного кратчайшего пути, которые можно адаптировать к проблеме SOTA.

Стратегия распараллеливания задачи с использованием графического процессора предложена в [237].

Несмотря на большое количество работ, посвященных данной проблеме, разработка эффективного алгоритма, который позволит быстро находить надежный путь в крупномасштабной транспортной сети, остается предметом исследований.

1.4.3 Маршрутизация транспортных средств

Существующие алгоритмы поиска пути, реализованные в интеллектуальных транспортных системах, картографических сервисах или бортовых навигационных системах в основном позволяют находить кратчайший маршрут на основе текущей информации о распределении транспортных потоков и не позволяют быстро реагировать на изменение дорожной ситуации и предлагать альтернативный маршрут движения. Кроме того, предоставление схожих маршрутов движения и информации о состоянии транспортных потоков разным водителям может приводить к образованию новых заторов, т.к. большинство водителей будут выбирать менее загруженные маршруты движения. Такое поведение, как показано в [238], приводит к колебаниям состояния сети и ухудшает транспортную ситуацию в целом.

Постепенное развитие подключенных и автономных транспортных средств позволяет решать задачу минимизации времени движения с точки зрения эффективного распределения транспортных средств в сети. Такое распределение позволит уменьшить уровень дорожных заторов и сократить общее время поездок в сети.

Существуют различные подходы к классификации систем маршрутизации транспортных средств. В [239] исследуются статические и динамические, детерминированные и стохастические, реактивные и прогнозные, а также централизованные и децентрализованные системы построения маршрутов. Обзор существующих методов управления дорожными заторами представлен в [240]. В статье изучаются многоагентные системы, показано преимущество гибридных систем, в которых принятие решений распределено между транспортными средствами и объектами транспортной инфраструктуры, что позволяет учитывать пользовательские предпочтения при построении маршрута, основываясь на данных о распределении транспортных потоков в сети.

В реактивных многоагентных системах предполагается, что маршрутизация транспортных средств – агентов – может быть выполнена наилучшим образом, основываясь на текущей информации о состоянии транспортных потоков. В отличие от реактивных систем, проактивные многоагентные системы прогнозируют условия движения, учитывая планируемые маршруты движения агентов – транспортных средств [99, 241, 242]. При

возникновении дорожных заторов выполняется перестроение маршрутов движения. В [99] рассматривается децентрализованный подход к маршрутизации транспортных средств на основе многоагентных систем. После нахождения маршрута движения каждый агент – транспортное средство информирует инфраструктурные агенты о предполагаемом маршруте движения. Эта информация используется для улучшения оценки маршрутов движения других агентов. В [243] авторы предлагают несколько эвристических методов перестроения маршрутов для предотвращения образования дорожных заторов, включая: случайный выбор пути из k кратчайших путей; распределение потока транспортных средств по различным маршрутам; максимизация энтропии маршрутов.

В [241] предложен метод резервирования маршрутов, который заключается в разбиении сегментов дорожной сети на слоты в пространственной и временной областях и резервировании областей для каждого транспортного средства. В соответствии с этим методом, дорожный сегмент доступен для движения только в том случае, если количество зарезервированных слотов не превышает критическую плотность транспортного потока на сегменте. Однако время прохождения дорожного сегмента считается постоянным и не зависит от количества транспортных средств на сегменте. Схожий подход был использован в [242], однако скорость прохождения сегмента считалась зависящей от количества зарезервированных слотов на сегменте. В [244] для выбора маршрутов движения в многоагентной системе применяется муравьиный алгоритм.

Обучение с подкреплением для решения задачи управления движением автономных транспортных средств на перекрёстке используется в [245, 246]. Несколько работ посвящено решению задачи нахождения системного оптимума распределения транспортных потоков с использованием аппарата теории игр [247-249]. Однако представленные алгоритмы являются вычислительно сложными и не могут быть использованы для маршрутизации транспортных средств в режиме реального времени.

Практически все рассмотренные работы считают время прохождения дорожных сегментов не зависящим от количества зарезервированных слотов / планируемых маршрутов движения, проходящих по указанным сегментам. В то же время, скорость движения на дорожном сегменте напрямую зависит от плотности транспортного потока на сегменте. В [101] сделан обзор основных соотношений скорости-плотности потока и представлена новая обобщенная модель, основанная на микроскопическом моделировании движения транспортных средств. В [102] предложено использовать логистические функции для описания соотношения скорости-плотности потока. Зависимость дисперсии скорости прохождения дорожного сегмента от плотности потока исследовалась в [250]. В [103]

рассматриваются стохастические фундаментальные диаграммы транспортных потоков и стохастические соотношения скорости-плотности.

1.5 Обеспечение информационной безопасности в ИТС

Одним из ключевых компонентов кооперативной ИТС является обмен данными между подключенными транспортными средствами и/или объектами дорожной инфраструктуры посредством автомобильных самоорганизующихся сетей (VANET). Передача сообщений в такой открытой среде как VANET ведет к необходимости рассмотрения ключевых проблем безопасности. Аутентификация, конфиденциальность данных, целостность данных, доступность данных – важнейшие компоненты безопасности в VANET.

Для обеспечения информационной безопасности необходимо обеспечивать защиту как от внешних угроз путем аутентификации участников коммуникаций, так и от внутренних угроз. При защите от внутренних угроз предполагается, что злоумышленником является аутентифицированное и авторизованное транспортное средство, поэтому одним из основных подходов в этом случае является обнаружение аномального поведения участников движения. В разделе 1.5.1 приводится обзор современного состояния исследований, посвященных аутентификации участников движения в сети VANET для защиты от внешних угроз. Исследования, посвященные защите от внутренних угроз путем обнаружения аномального поведения транспортных средств, представлены в разделе 1.5.2.

1.5.1 Аутентификация транспортных средств

Обеспечение надежной идентификации и аутентификации – одна из основных проблем информационной безопасности при внедрении сетей VANET. Все существующие узлы в сети должны быть аутентифицированы перед доступом к любым доступным службам. Аутентификация узлов сети исключает действие неопознанных злоумышленников, дает возможность оценить надежность полученных сообщений и обеспечивает конфиденциальность в сети. Любая атака, связанная с процессом идентификации или аутентификации, подвергает серьезной опасности всю сеть.

Проблемы безопасности в VANET были рассмотрены в [251]. Авторы представили анализ угроз и предложили схему архитектуры безопасности. В [252] авторы представили обзор современного состояния VANET, представили коммуникационные архитектуры VANET, описали и классифицировали существующие проблемы безопасности. Схемы аутентификации рассмотрены в [253]. Авторы классифицировали эти схемы на три

категории: аутентификация на основе криптографии, на основе подписей и с использованием методов проверки сообщений.

Традиционный способ аутентификации в системах связи – использование алгоритмов симметричной или асимметричной криптографии. В схемах на основе асимметричной инфраструктуры открытых ключей (PKI) создание открытых / закрытых ключей и сертификата выполняется центром сертификации. Каждое транспортное средство может использовать свой закрытый ключ для создания цифровой подписи сообщения, которое может быть проверено получателем с помощью открытого ключа. Несмотря на популярность этой схемы, она имеет серьезные ограничения, такие как накладные расходы на вычисления и обмен сообщениями, проблемы с отзывом сертификатов и т.д. [251, 254]. В симметричных схемах общий ключ используется для шифрования и дешифрования сообщения [255, 256]. По сравнению с асимметричными схемами, симметричная криптография работает быстрее, но требует знания секретного общего ключа. Производительность схемы аутентификации на основе PKI и схемы аутентификации с симметричным ключом сравнивалась в [257].

В схеме подписи на основе идентификации идентификатор транспортного средства может использоваться в качестве открытого ключа, тогда как закрытый ключ генерируется с помощью специального генератора секретных ключей [258, 259]. Схема аутентификации с сохранением условной конфиденциальности, основанная на криптографии на основе идентификации, была предложена в [260]. В схемах аутентификации на основе групповой подписи каждое транспортное средство в группе может подписывать сообщения от имени всей группы, что позволяет достичь анонимности [260, 261].

В схемах аутентификации на основе инфраструктуры придорожных блоков [262, 263] транспортные средства аутентифицируют друг друга с использованием придорожной инфраструктуры. В [264] авторы предложили схему, которая объединяет схемы на основе инфраструктуры придорожных блоков и устройства защиты от несанкционированного доступа.

В нескольких статьях авторы рассматривали гибридные подходы. В [265] было предложено объединить подходы, основанные на псевдонимах, и подходы, основанные на групповых подписях. В [266] авторы использовали пакетную проверку сообщений и агрегирование подписей для разработки схемы аутентификации с условным сохранением конфиденциальности.

1.5.2 Обнаружение аномального поведения транспортных средств

Традиционные подходы к обеспечению безопасности в сетях, такие как инфраструктура открытых ключей, сертификаты и подписи, используются для защиты от внешних типов угроз, в том время как для аутентифицированных и авторизованных недобросовестных транспортных средств эти подходы фактически бесполезны.

В настоящее время описано большое количество моделей обнаружения аномального поведения на основе доверия в различных сетях. Во многих используются алгоритмы на основе нечеткой логики, коллаборативной фильтрации и теории графов [267, 268, 269, 270, 271]. Каждая из этих моделей учитывает различные факторы: доверие на основе рекомендаций, прямое доверие и др.

Большинство схем обнаружения аномального поведения учитывают прямые факторы доверия для создания моделей оценок. Система обнаружения аномального поведения для обеспечения безопасности информации в одноранговых сетях [267] производит сбор двух прямых коэффициентов доверия: коэффициент средней задержки и коэффициент качественного уровня доставки пакетов. В основе этой системы лежит нечеткая логика оценки доверия с использованием этих факторов. В последующем данная система была модифицирована посредством использования теории графов для оценивания уровня доверия к узлам. В модели обнаружения аномального поведения [268], которая так же использует нечеткую логику, рассматриваются и другие факторы, связанные с вопросом корректности сообщений при их получении, такие как оценка правдоподобности отправителя, опыт прямых взаимодействий и время жизни сообщения. В качестве способа оценки точности определения местоположения события в данной модели используется так называемый туманный узел. Данный узел может выступать в качестве авторитетного узла, при помощи которого можно обнаруживать нарушителей в автомобильных сетях VANET.

В моделях обнаружения аномального поведения, помимо прямых факторов доверия, учитываются обратная связь и рекомендации. Большое количество моделей на основе доверия в последнее время делают упор именно на угрозах безопасности самой сети VANET. В [269] представлена схема обнаружения аномального поведения в VANET, нацеленная на противодействие таким атакам, как атака в сговоре и атака on-off. Расчет уровня доверия происходит в двух форматах, т.н. функциональное и рекомендательное доверия. Модифицированная версия данной модели [270] объединяет два формата доверия и позволяет строить сети VANET, которые будут противодействовать различным видам атак. Данная модель предполагает установление надежных и безопасных путей связи посредством выбора доверенных транспортных средств – ретрансляторов. Также модели

обнаружения аномального поведения рассматриваются в различных сценариях взаимодействия, например, в движении автомобильной колонны. При этом сценарии использование схемы доверительных рекомендаций [271] может помочь исключить выбор «плохих» авторитетных транспортных средств. Уровень доверия для транспортных средств в этой модели вычисляется на основе обратной связи с авторитетными транспортными средствами. Дополнительно внедрен итеративный алгоритм, который осуществляет фильтрацию для противодействия атакам в сговоре и атаке новичка.

Последние исследования вопроса безопасности в сетях VANET вводят новые алгоритмы оценки уровня доверия и факторы доверия. В [272] дополнительно вводятся так называемые социальные факторы связи между транспортными средствами: дружба, прямое или косвенное соседство. Многие модели предполагают наличие локального и глобального доверия между участниками сети [273]. Помимо социальных связей, некоторые модели вводят роли транспортных средств, которые определяют начальные уровни доверия [274]. Транспортные средства делятся на два вида: обычные транспортные средства и заведомо доверенные (официальные) транспортные средства (автомобили скорой помощи, пожарные и полицейские автомобили). Изначальный уровень доверия у обычных транспортных средств ниже, чем у официальных транспортных средств. Модель [275] принимает во внимание, что большая часть решений, которые имеют отношение к управлению транспортными средствами принимают именно водители. Укрепление доверия между транспортными средствами происходит на основе весового коэффициента, который учитывает честность водителей транспортных средств. Схема обнаружения аномального поведения на основе доверия AATMS [276] предполагает вычисление локальных и глобальных доверий транспортных средств в сети. AATMS отчасти основан на модели IWOTV [273]. IWOTV использует на алгоритмы VehicleRank и BayesRank. Эти алгоритмы, как и в AATMS, вычисляют локальные и глобальные доверия на основе Байесовского вывода и алгоритма PageRank соответственно. Однако вычисление доверия и обнаружение аномального поведения транспортных средств в алгоритме AATMS выполняется централизованно с использованием доверенных объектов инфраструктуры.

Частным случаем сценария движения, требующего определение аномального поведения транспортных средств, является движение транспортных средств в колонне. Автомобильная колонна – это особый вид автомобильного кооперативного вождения. Его можно описать как скоординированное движение группы транспортных средств, сообщающихся друг с другом для поддержания оптимальной политики управления с точки зрения относительного расстояния и скорости между транспортными средствами.

Разработка оптимальной политики управления является наиболее изученной проблемой в исследованиях кооперативного движения транспортных средств в колонне. Адаптивный круиз-контроль – одна из самых ранних автономных систем, которая была разработана для поддержания желаемого расстояния по отношению к предшествующему транспортному средству на полосе движения [277, 278]. В работе [277] авторы провели сравнительное исследование характеристик автономных систем управления с постоянным временным промежутком и кооперативных продольных систем управления, использующих связь между транспортными средствами. Проблема поддержания желаемого расстояния была исследована в работе [278]. Стратегии кооперативного адаптивного круиз-контроля основаны на использовании беспроводной связи между транспортными средствами для обеспечения совместного управления [279, 280]. В работе [281] представлена стохастическая оптимальная стратегия управления как для адаптивного круиз-контроля, так и для кооперативного адаптивного круиз-контроля, основанная на политике постоянного временного разрыва.

Другим подходом к проектированию оптимальных стратегий управления является задача консенсуса, заключающаяся в координации группы транспортных средств на основе обмена информацией при наличии ограничений и неопределенностей [282, 283]. Распределенный протокол управления был исследован в работе [282]. Авторы также доказали устойчивость движения в колонне с помощью теоремы Ляпунова-Разумихина. Аналогичный консенсусный алгоритм второго порядка был изучен в работе [284]. В работе [285] авторы предложили распределенную адаптивную стратегию совместного управления, которая использует информацию, поступающую от подключенных транспортных средств, для достижения синхронизации с лидером. Консенсусный подход третьего порядка также был исследован в работе [286].

Связь между транспортными средствами является ключевым аспектом оптимальных стратегий управления при совместном движении в колонне. Как и другие сети, автомобильные сети подвержены различным угрозам безопасности [287, 288]. В работе [289] авторы рассмотрели некоторые основные угрозы безопасности и атаки, которые могут быть использованы в сетях VANET, и представили подходы к обеспечению информационной безопасности.

Многие исследования посвящены проблемам безопасности в системах движения транспортных средств в колонне. В работе [290] авторы рассматривали различные типы вредоносных угроз, исследовали атаки прикладного уровня (спуфинг и фальсификация сообщений) и атаки сетевого уровня (отказ в обслуживании и пакетная передача). Была

предложена совместная стратегия управления, основанная на алгоритме голосования, для усиления защиты колонны транспортных средств. Однако предложенный алгоритм применим только в том случае, когда транспортные средства движутся с постоянной скоростью. В работе [271] авторы сосредоточились на разработке надежной схемы рекомендаций по обслуживанию колонны транспортных средств, для избегания выбора вредоносных транспортных средств при голосовании. Механизм обнаружения аномалий в реальном времени с использованием законов физики (кинематики) наряду со слиянием данных для проверки информации, полученной от лидера, был представлен в работе [291].

Анализ атак на адаптивный алгоритм круиз-контроля и два метода обнаружения атак, основанные на физических ограничениях и алгоритмах машинного обучения, были представлены в работе [292]. В работе [293] авторы смоделировали различные атаки в случае использования кооперативного адаптивного круиз-контроля показали их влияние на движение транспортных средств.

1.6 Современные тенденции развития интеллектуальных транспортных систем

В заключительном разделе, посвященном обзору современного состоянию исследований, рассматриваются основные этапы создания интеллектуальных транспортных систем и тенденции их развития.

В [294] представлены этапы развития ИТС начиная с 20-х годов прошлого века. В основных главах рассматриваются социально-экономическая среда, политические факторы и государственные программы развития ИТС, а также исследования и технологические разработки, характерные для каждого периода времени.

Автомобильная культура начала формироваться в начале 20 века. Вся история развития ИТС направлена на увеличение безопасности, уменьшение дорожных заторов, повышение эффективности транспортной инфраструктуры и мобильности участников дорожного движения. В более ранние годы развития ИТС основной тенденцией было удобство использования и мобильность. В середине 1960-х начали проводиться исследования в области бесконтактной навигации по маякам с помощью информационно-маршрутной системы General Motor для помощи водителю (DAIR). Автомобиль с DAIR мог отправить экстренное сообщение в сервисный центр, включая информацию о дорожных условиях. Система опиралась на датчики, расположенные через равные промежутки времени вдоль дороги. DAIR включала в себя панель дисплея на приборной панели автомобиля, которая показывала бы предупреждающие сообщения о дорожных инцидентах, и имела навигационную систему, которая могла направлять водителя по

заранее определенному маршруту. Также в 1960-х усилия исследователей были сосредоточены на создании первых мобильных роботов, которые затем стали основой для создания автономных ТС, в частности, функции навигации, сенсорики и локализации использовались при создании подключенных и автоматизированных транспортных средств. Начиная с 1970-х годов, в США началось внедрение технологии автоматического определения местоположения транспортного средства (AVL) раннего поколения, использующей придорожные маяки в качестве средства отслеживания местоположения. До 1980-х годов активно развивались центры управления дорожным движением, которые собирали и обрабатывали данные о системе движения на магистралях, включая погодные условия, скорость движения, дорожные заторы, дорожно-транспортные происшествия и другие события. С появлением спутников была создана глобальная система позиционирования (GPS). В 1980-х годах встал вопрос экологии. Вопросы безопасности и охраны окружающей среды становились все более приоритетными в транспортной политике. В это десятилетие был введен термин «интеллектуальная система автомобильных дорог» (IVHS), в которой описывалась группа технологий (включая обработку информации, связь, управление и электронику), которые подключают транспортные средства к инфраструктуре для повышения безопасности и эффективности транспортных систем. Разрабатывались различные датчики и видеокамеры. В 1990-х годах в США формировались стандарты, протоколы и шаблоны по развитию автоматизированной системы автомобильных дорог. Наиболее заметные проекты десятилетия, включающие автоматизированные дорожные системы: TravTek, Pathfinder, концепция навигации для водителей и транспортных средств (ADVANCE), Guidestar, INFORM, проект Denver Smart Bus, FAST-TRAC и HELP. В 2000-е года сотовая связь стала неотъемлемой частью ИТС, начали стремительно развиваться подключенные транспортные средства, а также облачные технологии. Для улучшения безопасности была создана инициатива Integrated Vehicle-Based Safety Systems (IVBSS) для разработки и тестирования интегрированных систем безопасности как на легковых транспортных средствах, так и на коммерческих грузовиках. С конца 2000-х годов началась активная разработка мобильных приложений для управления транспортным потоком с улучшенными системами определения географического положения и навигации. Многие крупные компании начинают инвестировать в технологии автоматизации. Так, например, Google сосредоточился на одной из наиболее сложных сфер в мире автоматизации – вождении в городе. Далее Volvo, BMW, Toyota, General Motors и другие автомобильные компании инвестируют в развитии автономных транспортных средств. Также в настоящее время развиваются «умные города».

Стратегический план [295] на период с 2020 года по 2025 год, разработанный Министерством транспорта США, содержит следующие направления развития ИТС:

- 1) определение и оценка новых технологических альтернатив для транспортной системы;
- 2) координация и руководство исследованиями и разработками в области ИТС в общественных интересах;
- 3) демонстрация преимуществ новых технологий ИТС для улучшения транспортной системы;
- 4) поддержка внедрения проверенных технологий, подходов и политик ТС во всей транспортной системе;
- 5) сохранение технологий и политик ИТС для реализации всего потенциала преимуществ на всех видах наземного транспорта.

В стратегическом плане рассматриваются цели каждой стратегии и их описание.

В [296] производится подробный обзор существующих ИТС, в качестве главного пути решения проблемы дорожных заторов предлагается улучшение обслуживания общественного транспорта и снижение количества личного транспорта. Так, например, предлагается использовать следующие существующие подсистемы ИТС:

- 1) усовершенствованная информационная система для путешественников (ATIS), предоставляющую информацию о прибытии и отправлении ТС, прогнозирование доступности маршрутов, загруженности дорог, предварительной информации для сокращения времени ожидания в пробках;
- 2) усовершенствованная система общественного транспорта (APTS), предназначенная для повышения эффективности общественного транспорта с использованием интеллектуального управления расписанием в соответствии с уровнем загруженности в различных районах;
- 3) усовершенствованная система управления дорожным движением (ATMS), используемая органами регулирования дорожного движения для мониторинга и управления транспортным потоком;
- 4) система управления чрезвычайными ситуациями (EMS);
- 5) система электронного сбора платы за проезд (EFC);
- 6) система сбора данных о дорогах (HDC), предоставляющая информацию о сценарии дорожного движения на основе устройств позиционирования;
- 7) системы управления дорожным движением (TMS), собирающие информацию с различного оборудования для повышения общей эффективности транспорта;

8) система сбора данных о транспортных средствах (VDC), собирающая параметры ТС для прогнозирования технического обслуживания и долговечности.

Также в статье рассматриваются подсистемы ИТС для комфортного использования транспортных средств:

- 1) система прогнозирования и мониторинга наличия мест в транспорте;
- 2) система мониторинга и оповещения об избыточном количестве пассажиров в общественном транспорте;
- 3) системы поиска мест для парковки ТС;
- 4) системы отслеживания общественного транспорта.

В работе [297] представлен детальный обзор научных публикаций по теме ИТС за период с 1997 по 2019 года. В работе отмечается рост опубликованных научно-исследовательских статей в области ИТС с каждым годом, что подтверждает актуальность данной темы исследования. Большая часть исследований, проводимых в области ИТС, связана с прогнозированием, при этом для решения данной задачи используются передовые вычислительные модели обработки больших данных, а также методы машинного обучения и искусственного интеллекта. В рамках ИТС данные методы успешно применяются для прогнозирования транспортных потоков, движения отдельных транспортных средств, распознавания дорожных знаков, обнаружения инцидентов, оптимизации и безопасности движения в транспортной сети.

В [298] представлен обзор использования больших данных в задачах в области ИТС. Для решения задач мониторинга, детектирования, прогнозирования выделяются различные источники данных, такие как смарт карты, видеокамеры, GPS-треки, подключенные и автономные транспортные средства, сети VANET, пассивные данные и другие источники. Также в статье рассматриваются различные виды обучения нейронных сетей на основе больших данных и платформы больших данных такие как Apache Hadoop и Apache Spark. В работе утверждается, что аналитика больших данных оказывает глубокое влияние на разработку ИТС, делая ее более безопасной и эффективной.

В [299] представлен обзор интеллектуальных транспортных систем, использующих подход к управлению, основанный на данных (англ.: data-driven ITS), обсуждаются функциональные возможности ключевых компонентов и некоторые проблемы развертывания. Основным компонентом таких ИТС является машинное зрение. Более того, системы поддерживают использование данных, полученных от различных источников. Также в статье рассматриваются проблемы обучения нейросетевых моделей: зашумленность данных, высокая размерность данных, разреженное обучение и

гетерогенное обучение. В качестве решения данных проблем предлагается использовать очистку данных с использованием методов анализа данных для обнаружения и удаления шума и выявления его распределения, для уменьшения размерности – многомерное обучение, неотрицательную матричную факторизацию и уменьшение размерности ядра, для решения проблемы гетерогенного обучения и сбора данных от разнородных источников предлагается использовать модели машинного обучения для поиска общего пространства признаков или трансферное обучение [300].

В [301, 302] авторами представлен обзор различных видов нейронных сетей, используемых для решения задач в области ИТС: сверточные нейронные сети, многослойный персептрон, сети с механизмом внимания, графовые нейронные сети, автоэнкодеры, рекуррентные нейронные сети и т.д. Производится обзор таких задач как прогнозирование пункта назначения, транспортного потока, прогнозирование тяжести дорожно-транспортного происшествий, прогнозирование способа передвижения, управление сигналами светофора и т.д.

Обзор развития ИТС в Китае представлен в [303, 304]. В качестве ключевых трендов развития отмечены исследования в области распознавания поведения водителей, сбора информации о состоянии транспортного средства и разработка стратегии управления транспортным средством, а также исследования, направленные на безопасность ТС и ИТС, синергию автомобильно-дорожной сети, интегрирование систем управления дорожным движением и аварийных систем, совместное вождение для движения транспортных средств в колонне и интеллектуальное управление транспортом.

В работе [305] рассматриваются сетевые решения от компании Siemens. В качестве ключевых возможностей отмечены синхронизация светофоров на перекрестках, предоставление водителям информации о дорожно-транспортной ситуации, предоставление информации транспортным службам о состоянии транспортных потоков с разделением по типам трафика и другие.

В [306] рассматриваются вопросы применения ИТС в современных мегаполисах, а также использование мировых практик при создании российских ИТС. В частности, внедрение ИТС в Москве включает в себя решения по интеграции системы управления парковочным пространством, освещением, мониторинг уровня различных типов загрязнения окружающей среды, контроль за соблюдением ПДД и фиксация нарушений. При этом в качестве основного источника данных используются системы видеонаблюдения. В качестве базовых решений выделяются следующие:

1) интегрированные интеллектуальные решения для видеомониторинга компаний Axis и Citilog, предназначенные для автоматического обнаружения инцидентов и сбора данных о дорожном движении, управления перекрестками, транспортными потоками. Интегрированные решения повышают безопасность на дорогах, снижают риски и увеличивают скорость движения ТС;

2) продукция компании Hikvision для детектирования дорожных событий, таких как определение дорожных заторов, езда по встречной полосе, обнаружение пешеходов, определение оставленных предметов на дороге, детектирование нарушения скоростного режима, соблюдения правил ПДД. Кроме того, решения включают в себе системы для интеллектуальной парковки, обеспечивающие сбор данных о парковочных местах, соблюдение правил дорожного движения, вследствие чего обеспечивается более удобный трафик, поиск парковочных мест и детектирование нарушений правил парковки.

В отчете [307] представлены статистические данные до и после использования ИТС в России в период с 2019 по 2024 года. С каждым годом наблюдается рост дорожных сетей, которые соответствует нормативным требованиям, снижение доли перегруженных дорог и доли дорожно-транспортных происшествий. В статье предлагается создание Единой цифровой интеллектуальной транспортной системы агломерации для повышения средней скорости движения транспортных средств, сокращения времени задержек движения, уменьшения износа дорожных покрытий, снижения расхода топлива и снижения массы выбросов вредных веществ, уменьшения количества ДТП.

В [308] предложены научные методы и подходы к обоснованию жизненных циклов локальных проектов ИТС с применением принципов оптимизации на основе построения психофизиологических моделей поведения участников дорожного движения. Пример систем косвенного управления транспортными потоками послужил основой для формирования проектов ИТС.

В [309] авторы предложили методологию системного анализа формирования инфраструктуры интеллектуального управления транспортными процессами с научным обоснованием стратегических решений в интеллектуальной транспортной геоинформационной системе. В [310] представлена математическая модель сетцентрической зональной системы управления транспортными процессами города, позволяющей достичь адаптивного управления транспортным средством по отношению к изменениям внешней среды и нештатным ситуациям благодаря информации о динамическом состоянии зоны управления.

Наконец, в [311] рассматриваются тенденции развития ИТС. Авторами дается прогноз на использование как подключенных ТС, так и автономных ТС, рассматриваются преимущества и недостатки их использования, приводятся примеры задач, которые могут быть эффективно решены с использованием новых данных, такие как прогнозирование транспортного потока, моделирование движения, уменьшение заторов, безопасное вождение, управление ТС и светофорами. В статье делается вывод, что в скором времени ожидается переход к гетерогенному транспортному потоку, включающему подключенными ТС и управляемыми водителем ТС. Поскольку характеристики подключенных ТС существенно отличаются от управляемых водителями ТС, то важно понимать характеристики потока в таких средах для использования в ИТС. Расширенные модели транспортных потоков будут необходимы как на микроскопическом, так и на макроскопическом уровне. На микроскопическом уровне необходима разработка новых моделей следования за лидером, которые будут учитывать параметры, связанные с подключенными ТС (например, ненадежная связь, задержка сигнала, протоколы вождения в режиме движения в колонне, доля подключенных ТС и т.д.). На макроскопическом уровне характеристики, связанные с подключенными ТС, следует учитывать при разработке модели потоков на сетевом уровне, чтобы помочь в региональном мониторинге и планировании (например, мониторинг уровня загруженности района, разработка схемы ценообразования на дорогах и т.д.).

1.7 Постановка цели и задач исследования

Проведенный обзор подтвердил актуальность рассматриваемой в рамках диссертационной работы задачи построения кооперативной интеллектуальной транспортной системы, решающей задачи анализа, прогнозирования и эффективного управления транспортным потоком с гетерогенным составом транспортных средств, а также движением отдельных транспортных средств. Существующие решения задач анализа, прогнозирования и управления транспортным потоком основываются на традиционных источниках данных: датчиках, детекторах транспортного потока, видеокамерах. В то же время, ожидаемый переход к транспортному потоку с гетерогенной структурой, включающему подключенные и управляемые водителями транспортные средства, увеличение числа источников данных и их объема, а также развитие методов машинного обучения и искусственного интеллекта, требует как разработки отдельных математических методов, моделей и алгоритмов решения отдельных задач в транспортных

сетях, так и реализации ИТС, объединяющей эти решения для повышения эффективности использования транспортной инфраструктуры.

Целью проводимого в диссертационной работе исследования является повышение эффективности использования транспортной инфраструктуры путем управления транспортным потоком с гетерогенным составом транспортных средств, а также движением отдельных транспортных средств в кооперативных интеллектуальных транспортных системах.

Для достижения поставленной цели в диссертационной работе решаются следующие задачи:

1) разработка алгоритмических средств решения задачи краткосрочного прогнозирования параметров гетерогенного транспортного потока в транспортной сети с использованием методов машинного обучения и подходов к обработке больших данных на основе актуальных и статистических данных о состоянии транспортного потока и движении отдельных транспортных средств;

2) разработка алгоритмических средств решения задачи директивного управления транспортным потоком на регулируемых перекрестках путем независимого и/или координированного управления сигналами светофоров и движением подключенных автономных транспортных средств в транспортной сети;

3) разработка алгоритмических средств решения задач косвенного управления транспортным потоком и информирования в ИТС:

- разработка алгоритмов решения задачи прогнозирования движения отдельных транспортных средств с использованием методов машинного обучения;

- разработка алгоритмов решения навигационной задачи нахождения надежного кратчайшего пути в стохастической зависящей от времени транспортной сети;

- разработка алгоритмов маршрутизации подключенных транспортных средств в транспортной сети в интеллектуальной транспортной системе на основе численного метода резервирования маршрутов, позволяющего достичь транспортного равновесия в сети;

4) разработка и реализация программного комплекса кооперативной интеллектуальной транспортной системы, решающей задачи анализа, прогнозирования и управления транспортным потоком с гетерогенным составом транспортных средств с использованием подходов к обработке больших данных;

5) проведение экспериментальных исследований разработанных методов и алгоритмов на натуральных и модельных данных, анализ результатов и сравнение с существующими решениями.

2 Алгоритмические средства решения задачи прогнозирования в ИТС

В данном разделе рассматривается задача прогнозирования транспортного потока в ИТС. Эффективное решение большинства транспортных задач зависит от точности данных о дорожном движении и распространении транспортных потоков. Информация о текущем и прогнозном распределении транспортных потоков необходима для решения ряда ключевых задач, рассматриваемых в ИТС: директивное управление транспортными потоками посредством светофорного регулирования, косвенное управление транспортным потоком и информирование, в т.ч. решение навигационных и логистических задач, прогнозирования времени прибытия общественного транспорта и т.д.

Появление новых источников данных и развитие методов их обработки позволяет повысить эффективность решения задачи краткосрочного прогнозирования транспортных потоков, что, в свою очередь, ведет к повышению качества решения указанных задач. В качестве перспективных методов обработки данных следует выделить методы и подходы к обработке «больших» данных, а также методы машинного обучения, в частности, аппарат искусственных нейронных сетей. В качестве перспективных источников данных следует выделить подключенные и/или автономные транспортные средства, которые могут предоставлять информацию о маршруте, положении и скорости движения.

В данном разделе предложен комплекс алгоритмических средства решения задачи прогнозирования, учитывающий как традиционные данные о состоянии транспортных потоков (усредненные значения скорости/плотности, данные с видеокамер на перекрестках), так и данные от подключенных транспортных средств. В частности, в разделе 2.1 представлен алгоритм прогнозирования динамики транспортного потока с использованием актуальных и статистических данных. В разделе 2.2 предложена модификация алгоритма прогнозирования параметров транспортного потока, использующего подход MapReduce к обработке данных больших объемов. Алгоритм прогнозирования на основе графовой нейронной сети описан в разделе 2.3.

Алгоритм, представленный в разделе 2.1, предлагается применять в ситуациях, когда задача прогнозирования решается для УДС малого размера и/или при недостаточном объеме статистических данных, описывающих состояние транспортных потоков. При увеличении объемов данных и возможности использования инфраструктуры обработки больших данных предлагается применение алгоритма, описанного в разделе 2.2. Наконец, при накоплении достаточного объема данных для обучения графовой нейронной сети, а

также неизменности топологии улично-дорожной сети, возможно использование алгоритма, описанного в разделе 2.3.

В разделе 2.4 предложен метод, предназначенный для прогнозирования гетерогенного транспортного потока, состоящего из подключенных и управляемых водителем транспортных средств. Экспериментальные исследования разработанного метода и алгоритмов представлены в заключительной части каждого раздела. Выводы и результаты данного раздела представлены в подразделе 2.5.

2.1 Прогнозирование динамики транспортного потока с использованием актуальных и статистических данных

Данный раздел посвящен решению задачи анализа и прогнозирования транспортных потоков в сети крупного города. В качестве исходных данных для решения указанной задачи используются данные GPS/ГЛОНАСС о местоположении отдельных транспортных средств. Уточнение местоположения отдельных транспортных средств по графу УДС необходимо для решения ряда прикладных задач, включая навигационные задачи и задачи прогнозирования времени движения отдельных транспортных средств.

Проецируя полученную информацию о положении транспортных средств на граф транспортной сети города, а также используя дополнительную фильтрацию, можно получить оценку отдельных параметров транспортных потоков (раздел 2.1.2). Эти параметры используются для краткосрочного (в пределах часа) прогнозирования изменения ситуации в транспортной сети города.

Предлагаемый алгоритм прогнозирования использует несколько этапов для построения прогноза. На первом этапе (раздел 2.1.3.1) предлагается декомпозировать транспортный граф на некоторое число подграфов по территориальному признаку. На втором этапе (раздел 2.1.3.2) для описания пространственно-временного состояния распределения транспортных потоков в получаемых подграфах используется метод снижения размерности, основанный на методе главных компонент. На третьем этапе (раздел 2.1.3.3) формируется элементарный прогноз, рассчитываемый с использованием известных скалярной и векторной моделей Бокса-Дженкинса. На четвертом этапе (раздел 2.1.3.4) для каждого из подграфов формируется несколько элементарных прогнозов с использованием метода опорных векторов и метода потенциальных функций. На пятом этапе (раздел 2.1.3.5) производится построение прогноза для каждого из подграфов с использованием адаптивной линейной комбинации полученных элементарных прогнозов. На шестом, заключительном, этапе (раздел 2.1.3.6) производится расчёт прогнозных

параметров транспортных потоков во всей транспортной сети города как линейной комбинации данных для подграфов.

В разделе 2.1.4 проводится экспериментальное исследование эффективности предложенного алгоритма прогнозирования на примере решения соответствующей задачи для области транспортной сети города Самара, даётся сравнение результатов прогнозирования с другими способами построения прогнозов.

2.1.1 Основные обозначения и постановка задачи

Примем в качестве математической модели УДС ориентированный граф $G = (V, E)$, ребра $w \in E$ которого соответствуют реальным участкам (сегментам) УДС, а вершины $q \in V$ представляют собой разделяющие участки дорог узлы. Направление ребра определяет направление движения ТС на соответствующем участке сети, параметр транспортного потока на конкретном участке сети определим как функцию $v: E \times T \rightarrow R$, которая в конкретный момент времени $t \in T$ для конкретного ребра $w \in E$ определяет его значение $v(w, t)$. В качестве параметров транспортного потока на ребре могут выступать следующие величины:

- скорость потока;
- среднее время прохождения ТС сегмента сети (величина, обратная к скорости потока);
- плотность потока;
- поток (собственно величина потока).

В дальнейшем изложении под параметром транспортного потока будет подразумеваться любая из указанных величин.

Геометрическое расположение сегмента $w \in E$ транспортной сети определим в виде двузначной функции вида $\bar{x}^w(\tau)$ ($\tau \in [0,1]$) такой, что координаты $(x_0^w(0), x_1^w(0))$ и $(x_0^w(1), x_1^w(1))$ определяют физическое расположение точек начала и конца соответствующего сегмента УДС, а геометрическое местоположение остальных точек можно получить, положив параметр в интервале $\tau \in (0,1)$.

Исходными данными для оценок параметров транспортного потока выступают данные GPS/ГЛОНАСС измерений, которые поступают с различных ТС как актуальная информация, то есть информация, поступающая в режиме реального времени. Формально, эти данные могут быть представлены как последовательность пар физических координат следующего вида:

$$\left(p_0^i(t_j), p_1^i(t_j) \right)_{\substack{i \in \mathfrak{Z} \\ j \in N}} \quad (3)$$

где $\mathfrak{Z} \subseteq N$ – определяет множество условных номеров ТС, которые поставляют GPS/ГЛОНАСС -данные о своём местоположении;

j – задаёт порядковый номер поступившего GPS/ГЛОНАСС сигнала.

Существенным моментом в данных (3) является то, что поступающие GPS/ГЛОНАСС данные о координатах являются неточными, что вызвано погрешностями навигационных систем. То есть координаты в (3) можно интерпретировать в виде:

$$p_0^i(t_j) = P_0^i(t_j) + \delta_0, \quad p_1^i(t_j) = P_1^i(t_j) + \delta_1 \quad (4)$$

где пара $(P_0^i(t_j), P_1^i(t_j))$ определяет истинное расположение ТС, а (δ_0, δ_1) – вектор искажений.

В связи с этим, практически для всех поступающих данных (то есть пар i, j) следующее утверждение оказывается неверным:

$$\exists w, \tau: \left(p_0^i(t_j), p_1^i(t_j) \right) = \left(x_0^w(\tau), x_1^w(\tau) \right) \quad (5)$$

Более того, в тех редких случаях, когда соотношение (5) всё же оказывается верным, реальное расположение ТС может не совпадать с полученным. То есть следующая импликация может быть неверной:

$$\left(p_0^i(t_j), p_1^i(t_j) \right) = \left(x_0^w(\tau), x_1^w(\tau) \right) \wedge \left(P_0^i(t_j), P_1^i(t_j) \right) = \left(x_0^{\tilde{w}}(\tilde{\tau}), x_1^{\tilde{w}}(\tilde{\tau}) \right) \Rightarrow w = \tilde{w} \wedge \tau = \tilde{\tau}.$$

Указанные недостатки данных GPS/ГЛОНАСС сигналов приводят к необходимости дополнительной обработки данных для получения более точных оценок реального местоположения ТС.

Учитывая введённые обозначения, формальная постановка задачи получения краткосрочного прогноза для заданного параметра транспортного потока УДС может быть сделана следующим образом: имея заданный граф УДС с множеством рёбер \mathbf{E} и актуальные (и исторические) данные о положении ТС в виде (1), рассчитать оценку (спрогнозировать) параметров транспортного потока $v(w, t)$ для всех $w \in \mathbf{E}$ и $t = t^* + n\Delta (n = \overline{1, N})$.

В приведённой формулировке N – число формируемых прогнозов, расположенных регулярно с временным интервалом Δ , величина t^* – текущий момент времени. Прогнозный

горизонт в этом случае определяется величиной $N\Delta$, которая для интересующего нас краткосрочного прогноза имеет порядок одного часа.

В свою очередь, построение алгоритма краткосрочного прогнозирования параметров транспортных потоков для УДС заключается:

- в определении математической модели, задающей вид преобразования указанных выше данных (3) в значения прогнозных величин. Математическая модель определяется с точностью до ряда параметров;

- в определении способа (алгоритма) настройки (оценки) параметров указанной выше математической модели по данным реальных GPS/ГЛОНАСС наблюдений (3).

Предварительным, но необходимым этапом предлагаемого алгоритма, является преобразование исходных данных к виду, удобному для обработки. Суть такого преобразования – переход от набора данных (3), косвенно характеризующих параметры транспортных потоков в сети в конкретные моменты времени (прошлого), к собственно значениям параметров в эти моменты:

$$v(w, t), \quad w \in E, \quad t = t^* - n\Delta \quad (n = 0, 1, \dots) \quad (6)$$

Предлагаемый способ получения (6) на основании данных (3) представлен в следующем разделе.

2.1.2 Оценка параметров транспортных потоков по данным GPS/ГЛОНАСС наблюдений

Предлагаемый способ построения актуальных оценок параметров транспортных потоков (6) по данным GPS/ГЛОНАСС наблюдений (3) состоит из двух этапов:

- построение оценок $(\hat{P}_0^i(t_j), \hat{P}_1^i(t_j))$ местоположений ТС в конкретные моменты времени, для которых соотношения (5) заведомо выполняются и учитываются ограничения, налагаемые поступательным и/или маршрутным движением отдельных ТС;

- расчёт величин (6) по полученным оценкам местоположений.

Ниже рассмотрены оба этапа подробнее.

2.1.2.1 Алгоритмы оценки текущего положения ТС на графе дорожной сети

Суть предлагаемого ниже решения заключается в проецировании данных (3) на граф УДС с учётом ограничений, налагаемых поступательным движением ТС. Определение местоположения произвольного ТС на графе УДС заключается в выборе того местоположения, которое наиболее «вероятно» с учётом истории движения этого ТС.

Поскольку история движения ТС оказывает влияние на оценки не только текущего, но и предшествующего положения ТС, окончательная оценка местоположений ТС производится с запаздыванием на заранее определённое число j_c поступающих сигналов. Таким образом, формально, задача оценки местоположения i -го ТС заключается в нахождении величин $(\hat{P}_0^i(t_j), \hat{P}_1^i(t_j))$ по набору последовательных измерений $\{(p_0^i(t_{j+j}), p_1^i(t_{j+j}))\}_{j=-j_c}^{j_c}$ с учётом структуры графа УДС \mathbf{E} .

Рассмотрим множество допустимых маршрутов в графе для i -го ТС в j -ый момент времени, то есть всех маршрутов графа УДС, включающих одно из ребер множества:

$$\left\{w: w \in \mathbf{E} \wedge \min_{\tau \in [0,1]} \left((p_0^i(t_j) - x_0^w(\tau))^2 + (p_1^i(t_j) - x_1^w(\tau))^2 \right) \leq \rho_c \right\}.$$

Для определённости будем считать, что указанное множество допустимых маршрутов состоит из S маршрутов: $\{\tilde{W}_s\}_{s=0}^{S-1}$. Обозначим $P(\tilde{W}_s)$ – вероятность (априорную) выбора произвольным ТС конкретного s -го маршрута. Также определим проекции поступающих измерений местоположения ТС на каждый из допустимых маршрутов:

$$(\tilde{w}_j^s, \tilde{\tau}_j^s) = \arg \min_{\substack{w \in \tilde{W}_s \\ \tau \in [0,1]}} \left((p_0^i(t_j) - x_0^w(\tau))^2 + (p_1^i(t_j) - x_1^w(\tau))^2 \right) \quad (7)$$

и введём величину, характеризующую общее отклонение последовательности измерений от соответствующего маршрута:

$$\varepsilon_s = \sqrt{\sum_{j=-j_c}^{j_c} \left((p_0^i(t_{j+j}) - x_0^{\tilde{w}_{j+j}^s}(\tilde{\tau}_{j+j}^s))^2 + (p_1^i(t_{j+j}) - x_1^{\tilde{w}_{j+j}^s}(\tilde{\tau}_{j+j}^s))^2 \right)}.$$

Заметим, что в случае если минимальное расстояние в выражении (7) для какого-либо допустимого маршрута превышает пороговую величину ρ_c , то этот маршрут исключается из множества допустимых. При этом, если множество допустимых маршрутов оказалось пустым, то позиция i -го ТС в момент времени t_j полагается неопределённой, а текущий маршрут движения i -го ТС (на графе) – завершённым в момент t_{j-1} . С момента t_{j+1} начинается построение нового маршрута.

Введём также величину, характеризующую смену направления движения ТС в конкретном маршруте \tilde{W}_s :

$$\psi_s = \sum_{j=1}^{j_c} |r_j^s - r_{j-1}^s| I(r_j^s - r_{j-1}^s < -\rho_c).$$

В последнем выражении величина r_j^s – расстояние вдоль маршрута графа \tilde{W}_s между положениями ТС $(\tilde{w}_0^s, \tilde{\tau}_0^s)$ и $(\tilde{w}_j^s, \tilde{\tau}_j^s)$, $I(\cdot)$ – индикатор соответствующего события.

Тогда в качестве оценки $(\hat{P}_0^i(t_j), \hat{P}_1^i(t_j))$ местоположения ТС принимается величина $(\tilde{w}_j^{s^*}, \tilde{\tau}_j^{s^*})$, где номер s^* искомого маршрута определяется из выражения:

$$s^* = \arg \min_{s=0, S-1} (1 - P(\tilde{W}_s)) (\gamma_0 \varepsilon_s + \gamma_1 \psi_s).$$

Величины $P(\tilde{W}_s)$ в начале работы алгоритма полагаются одинаковыми; в процессе работы алгоритма эти величины полагаются пропорциональными соответствующим относительным частотам выбора маршрутов. Весовые коэффициенты $\gamma_0, \gamma_1 \in \mathbf{R}_+$ подбираются эмпирически, в настоящей работе использованы значения $\gamma_0 = \gamma_1 = 1$.

2.1.2.2 Оценка параметров транспортных потоков по актуальным данным

Динамическая (привязанная ко времени) оценка параметров транспортного потока может быть получена по информации о положении отдельных транспортных средств $(\hat{P}_0^i(t_j), \hat{P}_1^i(t_j))_{\substack{i \in \mathfrak{S} \\ j \in \mathfrak{N}}}$. Для удобства дальнейшего изложения определим величины $w(i, j), \tau(i, j)$, значения которых определим из условия выполнения следующих равенств:

$$\begin{aligned} x_0^{w(i,j)}(\tau(i,j)) &= \hat{P}_0^i(t_j), \\ x_1^{w(i,j)}(\tau(i,j)) &= \hat{P}_1^i(t_j). \end{aligned}$$

Для оценки $v_{time}^w(t)$ времени прохождения ребра/сегмента w по величинам $w(i, j), \tau(i, j)$ могут использоваться различные методы и алгоритмы. Так, для оценки времени движения подключенных ТС в данной работе используется модель на основе ГНС (раздел 4.1.2), для оценки движения общественных транспортных средств используется модель на основе рекуррентной нейронной сети (раздел 4.1.1). Эта величина позволяет также получить оценку скорости потока на соответствующем сегменте по очевидной формуле:

$$v_{speed}^w(t) = \frac{|w|}{v_{time}^w(t)}.$$

Оценка оставшихся двух параметров может быть задана следующими выражениями:

- плотность потока (L – число полос движения):

$$v_{density}^w(t) = \frac{\kappa}{L|w|} |\{i \in \mathfrak{Z}: \exists j \in \mathbf{N}(t_{j-1} < t) \wedge (t_j \geq t) \wedge (w(i, j) = w)\}|;$$

- поток (число ТС в единицу времени, T – интервал наблюдения):

$$v_{flow}^w(t) = \frac{\kappa}{T} |\{i \in \mathfrak{Z}: (w(i, j) \neq w) \wedge (w(i, j + j) = w) \wedge (t_j < t) \wedge (t_{j+j} \in [t, t + T]) \wedge j \in \mathbf{N}\}|.$$

В приведённых выражениях величина $\kappa \geq 1$ – эмпирически подобранный коэффициент пропорциональности, определяющий отношение общего числа ТС к числу ТС с датчиками GPS/ГЛОНАСС.

2.1.3 Общая схема предлагаемого алгоритма

Общее описание предлагаемого алгоритма, представляющее его основные компоненты и идеи, но не детализирующее его до математической модели и алгоритмов, представлено ниже. Предполагается, что предварительная обработка, описанная в предыдущем разделе и заключающаяся в оценке актуальных параметров транспортных потоков по данным GPS/ГЛОНАСС наблюдений, выполнена, а входная актуальная и исторически доступная (архивная) информация для алгоритма представлена в виде наборов значений параметра транспортного потока для всех сегментов сети:

$$v(w, t), \quad w \in \mathbf{E}, \quad t = t^* - n\Delta \quad (n = 0, 1, \dots),$$

где t^* – текущий момент времени.

Выходной информацией алгоритма являются краткосрочные прогнозные значения параметра транспортного потока для всех сегментов сети:

$$v(w, t), \quad w \in \mathbf{E}, \quad t = t^* + n\Delta \quad (n = \overline{1, N}).$$

Предлагаемый алгоритм состоит из следующих этапов:

1) разбиение графа УДС на пересекающиеся подграфы по территориальному признаку и формирование вектора признаков (вектора значений параметров потоков) по каждому из них;

2) снижение размерности вектора признаков для каждого подграфа путём устранения пространственно-временной зависимости значений параметров потоков;

3) построение набора элементарных прогнозов для каждого из подграфов. В предлагаемом решении используется три варианта элементарных прогнозов. В первом случае используется ряд элементарных прогнозов, основанных на методе потенциальных функций с мерой близости векторов признаков подграфов, вводимой по аналогии с методом

билатеральной фильтрации. Для этого варианта прогноза возможна ситуация, когда прогноз может не быть сформирован по причине финитности выбранных ядер. Эта ситуация используется на следующем этапе предлагаемого алгоритма как дополнительная (управляющая) информация. Во втором варианте используется элементарный прогноз, основанный на методе опорных векторов – SVM (support vector machine). В третьем варианте используются элементарные прогнозы, основанные на классических скалярных и векторных моделях временных рядов Бокса-Дженкинса. Результатом работы каждого элементарного прогноза является прогнозный вектор параметров потоков сети или (для первого варианта) указание на невозможность построения прогноза;

4) агрегация элементарных прогнозов, построенных для каждого из подграфов УДС, с использованием адаптивной линейной комбинации полученных элементарных прогнозов. Адаптивность вводится путём учёта дополнительной (управляющей) информации, возникающей при невозможности построения отдельных элементарных прогнозов для метода потенциальных функций – в этом случае линейная комбинация элементарных прогнозов формируется для сокращённого набора построенных прогнозов;

5) расчёт окончательных значений прогнозных параметров транспортных потоков всей УДС в виде линейной комбинации прогнозов для всех подграфов УДС. В данном случае изменениям (усреднениям) подвергаются только те параметры, которые соответствуют сегментам УДС (рёбрам графа), попавшим одновременно в несколько подграфов.

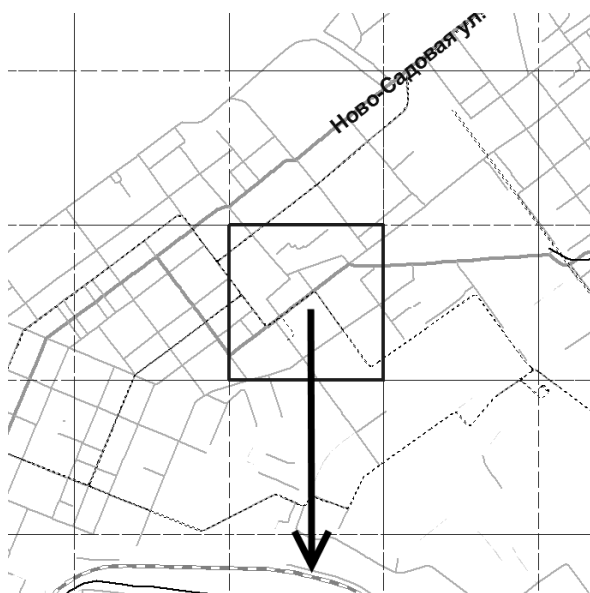
2.1.3.1 Представление графа сети с использованием подграфов. Вектор признаков подграфа

Выбираемый способ разбиения графа УДС на подграфы должен удовлетворять ряду плохо формализуемых требований, которые условно могут быть сформулированы следующим образом:

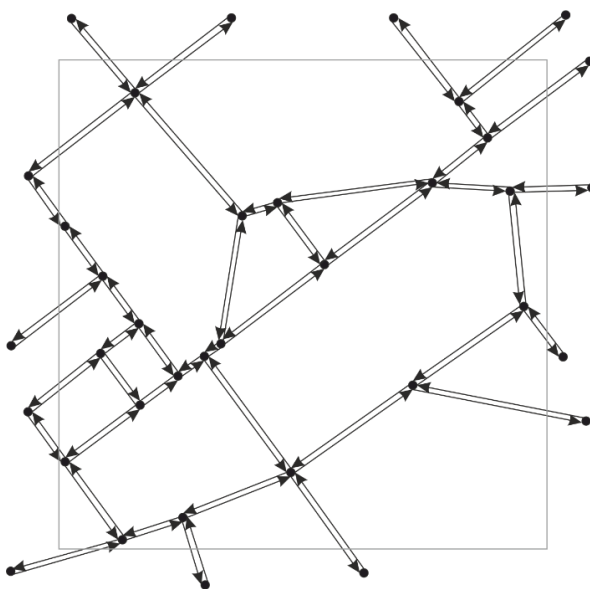
- способ должен быть регулярным;
- подграфы УДС должны быть связными;
- подграфы УДС должны быть примерно одинакового размера (по количеству рёбер и по размеру занимаемой площади);
- набор сегментов УДС, попадающих в подграфы, должен быть "компактно" расположен (то есть территориально близко);
- число рёбер в подграфах не должно быть малым – для реальных графов УДС оно должно составлять от нескольких десятков до нескольких сотен;

- простота введения меры территориальной близости между двумя подграфами УДС (используется в последующих разделах).

Учитывая приведённые выше требования, предлагается следующий способ представления графа УДС с использованием подграфов, иллюстрированный на рисунке 7.



а) исходный граф сети;



б) сформированный подграф сети;

Рисунок 7 – Иллюстрация к способу представления графа УДС с использованием подграфов

Выбирается число формируемых подграфов в количестве $K_0 \cdot K_1$, и в подграф с номером $k = k_0 K_1 + k_1$ ($k_0 = \overline{0, K_0 - 1}$; $k_1 = \overline{0, K_1 - 1}$), обозначаемый ниже E_k ,

относятся те сегменты из множества \mathbf{E} , координаты одной из вершин которых попадают в соответствующую прямоугольную область Π_{k_0, k_1} :

$$\mathbf{E}_{k_0 K_1 + k_1} \equiv \{w \in \mathbf{E}: \bar{x}^w(0) \in \Pi_{k_0, k_1} \vee \bar{x}^w(1) \in \Pi_{k_0, k_1}\},$$

где

$$\begin{aligned} \Pi_{k_0, k_1} &\equiv \left[x_0^{\min} + \frac{k_0}{K_0} \Delta^0, x_{\min}^0 + \frac{k_0 + 1}{K_0} \Delta^0 \right] \times \left[x_1^{\min} + \frac{k_1}{K_1} \Delta^1, x_1^{\min} + \frac{k_1 + 1}{K_1} \Delta^1 \right], \\ x_s^{\min} &= \min_{\substack{\zeta=0,1, \\ w \in W}} x_s^w(\zeta), & x_s^{\max} &= \max_{\substack{\zeta=0,1, \\ w \in W}} x_s^w(\zeta), \\ \Delta^s &= x_s^{\max} - x_s^{\min}, \quad s = 0, 1. \end{aligned}$$

Число подграфов по вертикали и горизонтали K_0, K_1 выбирается эмпирически, но так, чтобы удовлетворять требованию по числу рёбер.

Если при полученном способе разбиения нарушается связность отдельных подграфов УДС, выполняется расширение области для добавления несвязанных сегментов УДС к одному или нескольким подграфам.

Каждый получаемый подграф \mathbf{E}_k УДС для дальнейшей обработки представляется в виде своего описания – вектора признаков, характеризующего потоки подграфа в конкретный момент времени. Формирование указанного вектора признаков производится следующим образом.

Пусть $\{w_s^k\}_{s=0}^{S^k-1}$ – набор рёбер конкретного выбранного подграфа \mathbf{E}_k в количестве S^k штук, упорядоченный определённым образом (способ упорядочивания значения не имеет). Тогда вектор признаков, используемый в качестве описателя этого подграфа, имеет вид:

$$\bar{v}_M^k(t) = \begin{pmatrix} v(w_0, t) \\ \dots \\ v(w_0, t - M\Delta) \\ \dots \\ v(w_{S^k-1}, t) \\ \dots \\ v(w_{S^k-1}, t - M\Delta) \end{pmatrix}, \quad (8)$$

где $M > 1$ – число используемых в векторе признаков архивных значений параметров транспортных потоков для каждого сегмента сети.

Для удобства дальнейшего изложения, процесс получения описания для конкретного подграфа запишем в виде операции проекции:

$$\bar{v}_M^k(t) = \bar{v}_M(t)|_{E_k}, \quad k = \overline{0, K-1}. \quad (9)$$

2.1.3.2 Снижение размерности описания подграфа УДС с учётом пространственной и временной избыточности данных о потоках

Представление состояния подграфа сети в виде вектора признаков (8) обладает существенной информационной избыточностью. Прямое подтверждение информационной избыточности по отношению к пространственному распределению транспортных потоков (то есть избыточности представления подграфа в виде вектора текущих значений параметров транспортных потоков $\bar{v}_0^k(t)$) дано в работе [312], где отмечено, что объём основной информации о распределении транспортного потока составляет не более 10% от объёма данных в сети. Другим, но косвенным подтверждением того же тезиса о наличии значительной избыточности в данных, является использование вычислительных процедур распределения транспортных потоков в статических и динамических сетях на основании матрицы корреспонденции существенно меньшего (чем количество сегментов в УДС) размера. Ниже предлагается решение, которое для снижения размерности вектора описания ситуации использует как информационную избыточность, связанную с информацией о пространственном распределении транспортных потоков, так и избыточность, связанную с информацией о временном распределении потоков.

Предлагаемый способ снижения размерности заключается в переходе от исходного представления (8) к сокращённому представлению с использованием небольшого числа компонент, получаемых с использованием метода главных компонент (РСА – Principal Component Analysis) [313].

В контексте данной работы метод РСА состоит из следующих шагов (выполняется для каждого подграфа E_k УДС отдельно):

1) вычисляется оценка ковариационной матрицы векторов признаков $\bar{v}_M^k(t)$ (реализации вектора соответствуют \tilde{N} временным моментам в прошлом):

$$C^k = \frac{1}{\tilde{N}} \sum_{n=0}^{\tilde{N}-1} (\bar{v}_M^k(t - \Delta n) - \bar{v}^k) (\bar{v}_M^k(t - \Delta n) - \bar{v}^k)^T, \quad (10)$$

где средний вектор

$$\bar{v}^k = \frac{1}{\tilde{N}} \sum_{n=0}^{\tilde{N}-1} \bar{v}_M^k(t - \Delta n).$$

2) вычисляются собственные значения $\lambda_1^k, \dots, \lambda_{MS^k}^k$ и собственные векторы ковариационной матрицы C^k . Производится их упорядочивание так, что $\lambda_1^k \geq \lambda_2^k \geq \dots \geq \lambda_{MS^k}^k \geq 0$;

3) вычисляется квадрат относительной ошибки представления вектора (6) заданным числом главных компонент как отношение остаточной дисперсии к выборочной дисперсии:

$$\delta_{k,r}^2 = \frac{\sum_{\ell=r+1}^{MS^k} \lambda_{\ell}^k}{\sum_{\ell=1}^{MS^k} \lambda_{\ell}^k}, \quad (11)$$

4) по относительной ошибке $\delta_{k,r}$ выбирается число R главных компонент, обеспечивающих требуемую точность представления $\delta_{threshold}^2$;

5) формируется матрица главных компонент M^k размера $MS^k \times R$ из собственных векторов ковариационной матрицы C^k , соответствующих первым R собственным числам;

б) вычисляется новый вектор признаков со сниженной размерностью как проекция исходного вектора признаков на главные компоненты:

$$\bar{v}_M^k(t) = (M^k)^T \bar{v}_M^k(t). \quad (12)$$

Результатом работы метода снижения размерности оказывается вектор $\bar{v}_M^k(t)$, состоящий из $R (R < MS^k)$ компонент, которые совокупно характеризуют текущее состояние (с предысторией на M отсчётов по времени) подграфа E_k УДС в конкретный момент времени t .

Полученный вектор описания $\bar{v}_M^k(t)$ вместе с исходным $\bar{v}_M^k(t)$ описанием подграфа УДС используются для построения набора элементарных прогнозов.

2.1.3.3 Алгоритмы прогнозирования транспортных потоков с использованием временных рядов

Решение задачи прогнозирования транспортных потоков с использованием временных рядов обычно производится с использованием одной из следующих моделей:

- интегрированная модель авторегрессии–скользящего среднего (ARIMA) [48] и её модификация с учётом сезонных компонент – сезонная модель ARIMA [314, 56];
- векторная модель VARMA [315, 316];
- пространственно-временная модель STARMA [317, 318].

В данном разделе для построения отдельных элементарных прогнозов и исследования их эффективности были использованы сезонная модель ARIMA и векторная модель VARMA. Их краткое описание представлено ниже.

Модель ARIMA

В модели ARIMA прогнозируется значение параметра транспортного потока на сегменте дорожной сети, основываясь только на предыдущих значениях этого параметра на том же сегменте. Развитием и обобщением модели ARIMA на ряды, в которых имеется периодическая сезонная компонента, является сезонная модель ARIMA. Сезонная модель ARIMA с порядком модели $(p, d, q) \times (P, D, Q)_s$ имеет вид:

$$\begin{aligned} \left(1 - \sum_{i=1}^p \varphi_i L^i\right) \left(1 - \sum_{i=1}^P \Phi_i L^{i \cdot s}\right) (1-L)^d (1-L^s)^D v(w, t) \\ = \left(1 + \sum_{i=1}^q \theta_i L^i\right) \left(1 + \sum_{i=1}^Q \Theta_i L^{i \cdot s}\right) \varepsilon(t), \end{aligned} \quad (13)$$

где $L^i v(w, t) = v(w, t - i\Delta)$ – лаговый оператор;

$\varepsilon(t)$ – стационарный временной ряд (скалярный);

$\varphi, \theta, \Phi, \Theta$ – параметры модели (скалярные).

Модель VARMA

Векторная модель VARMA позволяет учитывать в прогнозе значения параметра транспортного потока на конкретном сегменте значения параметров на соседних сегментах. Основное соотношение, определяющее модель VARMA с порядком (p, q) , имеет вид:

$$\left(I - \sum_{i=1}^p \Phi_i L^i\right) \bar{v}(t) = \left(I + \sum_{i=1}^q \Theta_i L^i\right) \bar{\varepsilon}(t), \quad (14)$$

где

$$\bar{v}(t) = (v(w_0, t), v(w_1, t), \dots, v(w_{S-1}, t))^T$$

– это вектор текущих значений параметров потоков на некотором наборе сегментов УДС. Для предложенного ранее разбиения графа сети на подграфы в качестве указанного вектора на конкретном k -ом подграфе используется вектор $\bar{v}_0^k(t)$, определённый в (6). Значение других величин в приведённом выражении:

L^i – лаговый оператор,

$\bar{\varepsilon}(t)$ – стационарный временной ряд (векторный),

Φ, Θ – параметры модели (матрицы).

Общим недостатком моделей временных рядов является необходимость дополнительной обработки ситуаций отсутствия данных.

Результат прогнозирования, получаемый при применении одного из представленных выше методов прогнозирования временных рядов для всех сегментов в конкретном k -ом подграфе сети, будем в дальнейшем обозначать следующим образом:

$$\bar{v}_0^k(t + n\Delta) = TS(k, M, t, n\Delta).$$

2.1.3.4 Алгоритмы прогнозирования транспортных потоков с использованием методов машинного обучения

Среди методов машинного обучения для решения задачи кратковременного прогнозирования транспортных потоков обычно используются следующие методы и алгоритмы:

- алгоритм линейной регрессии;
- регрессия методом опорных векторов (Support Vector Regression – SVR);
- метод потенциальных функций;
- метод ближайших соседей;
- нейронные сети.

В данном разделе для построения отдельных элементарных прогнозов и исследования их эффективности были использованы метод опорных векторов и метод потенциальных функций. Их краткое описание представлено ниже.

Регрессия методом опорных векторов (SVR)

Метод опорных векторов является методом нелинейной регрессии, основная идея которого заключается в переводе исходных векторов признаков в пространство более высокой размерности и использование в новом пространстве модели линейной регрессии. Часто метод опорных векторов представляют в следующем виде:

$$v(w, t + n\Delta) = \sum_{m=0}^{\tilde{N}-1} \beta_m K(\bar{v}_M^k(t), \bar{v}_M^k(t - m\Delta)) + \beta_{\tilde{N}}, \quad (15)$$

где $\{\beta_m\}_{m=0}^{\tilde{N}}$ – параметры модели;

$K(\cdot, \cdot)$ – нелинейная функция ядра.

Метод является вычислительно сложным из-за большой размерности векторов признаков, поэтому для его использования вместо исходных векторов $\bar{v}_M^k(t)$ в настоящей работе используются вектора описания $\bar{\vartheta}_M^k(t)$. Тогда метод опорных векторов, определяющий расчёт параметра транспортного потока для конкретного сегмента w , примет вид:

$$v(w, t + n\Delta) = \sum_{m=0}^{\tilde{N}-1} \beta_n^{w, n\Delta} K(\bar{\vartheta}_M^k(t), \bar{\vartheta}_M^k(t - m\Delta)) + \beta_{\tilde{N}}^{w, n\Delta}.$$

Коэффициенты $\{\beta_n^{w, n\Delta}\}_{n=0}^{\tilde{N}}$ для метода SVR определяются в результате решения двойственной задачи математического программирования.

Наиболее распространёнными ядрами для метода SVR являются следующие:

- полиномиальное:

$$K(\bar{\vartheta}_M^k(t), \bar{\vartheta}_M^k(t - m\Delta)) = (\bar{\vartheta}_M^k(t)^T \bar{\vartheta}_M^k(t - m\Delta) + 1)^d;$$

- радиальная базисная функция:

$$K(\bar{\vartheta}_M^k(t), \bar{\vartheta}_M^k(t - m\Delta)) = \exp(-\gamma \|\bar{\vartheta}_M^k(t) - \bar{\vartheta}_M^k(t - m\Delta)\|)^d, \gamma > 0;$$

- сигмоид:

$$K(\bar{\vartheta}_M^k(t), \bar{\vartheta}_M^k(t - m\Delta)) = \tanh(\beta_0 (\bar{\vartheta}_M^k(t)^T \bar{\vartheta}_M^k(t - m\Delta)) + \beta_1)^d.$$

Результат прогнозирования, получаемый в результате применения метода SVR для всех сегментов в конкретном k -ом подграфе сети, будем в дальнейшем обозначать следующим образом:

$$\bar{v}_0^k(t + n\Delta) = SVR(k, M, t, n\Delta).$$

Метод потенциальных функций

Метод потенциальных функций позволяет прогнозировать значения параметров транспортного потока с учётом близости векторов описания $\bar{\vartheta}_M^k(t)$ в разные моменты времени. Общее соотношение, характеризующее вычислительную процедуру получения прогнозного значения, для нашей задачи имеет вид (значение « \emptyset » в приводимой ниже формуле соответствует факту, что результат прогноза не определён/отсутствует):

$$\begin{aligned} \bar{v}_0^k(t + n\Delta) = PF_\sigma(k, M, t, n\Delta) &\equiv \\ &\equiv \begin{cases} \frac{\sum_{m=0}^{\bar{N}-1} \bar{v}_0^k(t - m\Delta + n\Delta) R_\sigma(\bar{\vartheta}_M^k(t), \bar{\vartheta}_M^k(t - m\Delta))}{\sum_{m=0}^{\bar{N}-1} R_\sigma(\bar{\vartheta}_M^k(t), \bar{\vartheta}_M^k(t - m\Delta))}, & \sum_{m=0}^{\bar{N}-1} R_\sigma(\bar{\vartheta}_M^k(t), \bar{\vartheta}_M^k(t - m\Delta)) > 0, \\ \emptyset, & \sum_{m=0}^{\bar{N}-1} R_\sigma(\bar{\vartheta}_M^k(t), \bar{\vartheta}_M^k(t - m\Delta)) = 0. \end{cases} \end{aligned} \quad (16)$$

где $R_\sigma(\bar{\vartheta}_M^k(t), \bar{\vartheta}_M^k(t - m\Delta))$ – ядро формируемой оценки, монотонно убывающее по мере увеличения расхождения между векторами $\bar{\vartheta}_M^k(t), \bar{\vartheta}_M^k(t - m\Delta)$.

В работе предлагается использовать следующую функцию, которая учитывает близость векторов описания не только в том же подграфе, в котором стоит прогноз, но и в «соседних» к k -ому H подграфам:

$$R_\sigma(\bar{\vartheta}_M^k(t), \bar{\vartheta}_M^k(t - m\Delta)) = \begin{cases} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\rho^2}{2\sigma^2}\right), & \rho \leq 4\sigma, \\ 0, & \rho > 4\sigma. \end{cases} \quad (17)$$

где $\rho = \|\bar{\vartheta}_M^k(t) - \bar{\vartheta}_M^k(t - m\Delta)\|^2 + \frac{1}{H} \sum_{h=0}^{H-1} \|\bar{\vartheta}_M^h(t) - \bar{\vartheta}_M^h(t - m\Delta)\|^2$.

Параметр σ определяет максимальное расстояние между векторами описания, которые будут использоваться в прогнозе. Варьируя значение параметра, будем получать различные значения прогнозов параметров транспортного потока.

В выбранном варианте использования – с финитным ядром (16)-(17) – метод потенциальных функций имеет специфику: при малых значениях параметра σ может быть формально получен результат со значением, обозначенным символом « \emptyset », то есть факт отсутствия результата. Такая ситуация возникает, если для текущего состояния сети в исторических данных не окажется ни одного близкого прототипа. В предлагаемом решении эта ситуация используется для построения адаптивной вычислительной процедуры агрегации элементарных прогнозов, описание которой приведено ниже.

2.1.3.5 Алгоритм адаптивной линейной комбинации элементарных прогнозов

Алгоритм адаптивной линейной комбинации элементарных прогнозов применяется для построения окончательного прогноза параметров транспортного потока для каждого из подграфов с использованием адаптивной линейной комбинации полученных элементарных прогнозов. Адаптивность комбинации вводится путём анализа фактов появления неопределённых прогнозных значений в наборе алгоритмов прогнозирования,

использующих рассмотренный ранее метод потенциальных функций. Ниже предлагаемый способ построения адаптивной комбинации представлен более формально.

Пусть $\{\sigma_q\}_{q=0}^{Q-1}$ ($\sigma_q \in \mathbf{R}_+$) монотонно убывающая последовательность чисел: $\sigma_q > \sigma_{q+1}$ ($q = \overline{0, Q-2}$). Учитывая вид соотношений (16)-(17), очевидны следующие утверждения:

$$\begin{aligned} R_{\sigma_q} \left(\bar{\vartheta}_M^k(t), \bar{\vartheta}_M^k(t - m\Delta) \right) = 0 &\Rightarrow R_{\sigma_{q+1}} \left(\bar{\vartheta}_M^k(t), \bar{\vartheta}_M^k(t - m\Delta) \right) = 0, \\ PF_{\sigma_q}(k, M, t, n\Delta) = \emptyset &\Rightarrow PF_{\sigma_{q+1}}(k, M, t, n\Delta) = \emptyset. \end{aligned}$$

Тогда при Q выбранных для метода потенциальных функций ядер с различными $\{\sigma_q\}_{q=0}^{Q-1}$, удовлетворяющими указанному выше ограничению, возможна всего $(Q + 1)$ различная ситуация, когда отдельные прогнозы дают определённые/неопределённые значения.

Идея алгоритма адаптивной линейной комбинации элементарных прогнозов заключается в построении независимых линейных комбинаций элементарных прогнозов (линейной регрессии элементарных прогнозов) для каждой из $(Q + 1)$ возможных ситуаций.

Предлагаемый вариант адаптивной комбинации может быть формально представлен в следующем виде:

$$\begin{aligned} \bar{v}_0^k(t + n\Delta) = & \\ = & \begin{cases} \alpha_0^0 TS(k, M, t, n\Delta) + \alpha_1^0 SVR(k, M, t, n\Delta), & PF_{\sigma_0}(k, M, t, n\Delta) = \emptyset, \\ \alpha_0^{\bar{q}} TS(k, M, t, n\Delta) + \alpha_1^{\bar{q}} SVR(k, M, t, n\Delta) + & \\ + \sum_{q=0}^{\bar{q}-1} \alpha_{q+2}^{\bar{q}} PF_{\sigma_q}(k, M, t, n\Delta), & \bar{q} = 1 + \arg \max_{q=\overline{0, Q-1}} (PF_{\sigma_q}(k, M, t, n\Delta) \neq \emptyset). \end{cases} \end{aligned} \quad (18)$$

В представленной адаптивной линейной комбинации настройка требуется для следующего набора вещественных коэффициентов $\{\alpha_q^{\bar{q}}\}_{q=\overline{0, Q}; q=0, \bar{q}+1}$.

2.1.3.6 Вычислительная процедура расчёта прогнозных параметров транспортных потоков

Заключительным этапом предлагаемого алгоритма является расчёт прогнозных параметров транспортных потоков для графа всей УДС как линейной комбинации

прогнозных данных (18) для подграфов. Данная операция оказывается необходимой, поскольку для некоторых из сегментов, попавших одновременно в несколько подграфов, существует несколько прогнозных значений. Искомое прогнозное значение параметра транспортного потока для сегмента w в момент времени t предлагается вычислять по следующей формуле:

$$\hat{v}(w, t) = \frac{\sum_{k=0}^{K-1} \bar{v}_0^k(t)|_w}{\sum_{k=0}^{K-1} \bar{v}_0^k(t)||_w}, \quad (19)$$

где $\bar{v}_0^k(t)|_w$ – значение параметра транспортного потока на конкретном сегменте w , полученное для k -го подграфа:

$$\bar{v}_0^k(t)|_w = \begin{cases} v(w_r, t), & w = w_r \in \mathbf{E}_k; \\ 0, & \text{иначе.} \end{cases}$$

Значение, как видно из формулы, полагается нулевым, если сегмент не входит в конкретный подграф.

Значение $\bar{v}_0^k(t)||_w$ играет роль индикатора вхождения сегмента в конкретный подграф и определяется по формуле:

$$\bar{v}_0^k(t)||_w = \begin{cases} 1, & w = w_r \in \mathbf{E}_k; \\ 0, & \text{иначе.} \end{cases}$$

2.1.3.7 Математическая модель краткосрочной динамики транспортных потоков

В окончательном виде математическая модель краткосрочного прогнозирования параметров транспортных потоков по актуальным и историческим данным может быть представлена следующим набором формальных преобразований (ниже величина t^* означает текущий момент времени, а $t^* + n\Delta$ – момент времени, на который прогноз формируется).

Этап 1: проекция потоков на графе УДС на подграфы по формуле (9):

$$\bar{v}_M^k(t^*) = \bar{v}_M(t^*)|_{\mathbf{E}_k}, \quad k = \overline{0, K-1}.$$

Этап 2: снижение размерности описания подграфов по формуле (12):

$$\bar{\vartheta}_M^k(t^*) = (M^k)^T \bar{v}_M^k(t^*), \quad k = \overline{0, K-1}.$$

Этап 3: расчёт набора элементарных прогнозов для состояния сети в будущий момент времени $\bar{v}_0^k(t^* + n\Delta)$ с использованием следующих методов ($n = \overline{1, N}$):

- метода прогнозирования временных рядов $TS(k, M, t^*, n\Delta)$, формально определяемого соотношениями (13) или (14);

- метода опорных векторов $SVR(k, M, t^*, n\Delta)$, выполняющего преобразование (15);

- метода потенциальных функций $\left\{PF_{\sigma_q}(k, M, t^*, n\Delta)\right\}_{q=0}^{Q-1}$, выполняющего преобразование в виде (16).

Этап 4: построение адаптивной линейной комбинации элементарных прогнозов, полученных на предыдущем шаге, и получение набора прогнозов для отдельных подграфов $\{\bar{v}_0^k(t^* + n\Delta)\}_{k=0}^{K-1}$. Формальное выражение для предлагаемой адаптивной линейной комбинации приведено в формуле (18).

Этап 5: расчёт прогнозных параметров транспортных потоков для графа всей УДС по формуле (19):

$$\hat{v}(w, t^* + n\Delta) = \frac{\sum_{k=0}^{K-1} \bar{v}_0^k(t^* + n\Delta)|_w}{\sum_{k=0}^{K-1} \bar{v}_0^k(t^* + n\Delta)\|_w}, \quad w \in E.$$

2.1.3.8 Численный метод настройки математической модели краткосрочной динамики транспортных потоков

Математическая модель краткосрочного прогнозирования параметров транспортного потока, представленная в предыдущем разделе, определена с точностью до набора параметров. Параметры можно условно разбить на две группы, первая из которых содержит параметры, определяемые пользователем или постановкой задачи. Вторая группа параметров – это параметры модели и входящих в её состав алгоритмов/методов. Принципиальным отличием второй группы параметров от первой является необходимость их определения (оценки) по историческим данным состояний УДС. Ниже приведены составы указанных групп и для параметров второй группы представлено описание численных методов их оценки.

Параметры, определяемые пользователем или постановкой задачи

К указанной группе параметров относятся следующие:

- шаг дискретизации по времени Δ ;
- горизонт прогноза ΔN , определяющий число прогнозных значений N ;
- число маршрутов $|\Omega|$ и их расположение $\{W_m\}_{m \in \Omega}$;

- максимально допустимая погрешность ρ_c при определении координат для GPS/ГЛОНАСС датчика;
- количества K_0, K_1 областей по вертикали и горизонтали соответственно, подразделяющих территорию на области, а граф УДС на подграфы;
- число M используемых при описании состояния подграфа архивных значений параметров транспортного потока;
- требуемая точность представления вектора состояния подграфа УДС главными компонентами $\delta_{threshold}^2$;
- вид и параметры ядер $K(\cdot, \cdot)$ для метода SVR;
- набор максимальных среднеквадратических отклонений $\{\sigma_q\}_{q=0}^{Q-1}$ для метода потенциальных функций.

Параметры, подлежащие оценке по выборочным данным

- $\{M^k\}_{k=0}^{K-1}$ – матрица главных компонент;
- $(p, d, q) \times (P, D, Q)_s$ – порядок сезонной модели ARIMA;
- $\phi, \theta, \Phi, \Theta$ – параметры сезонной модели ARIMA (скалярные);
- (p, q) – порядок модели VARMA;
- Φ, Θ – параметры модели VARMA (векторные);
- $\{\beta_n^{w, n\Delta}\}_{n=0}^{\tilde{N}}$ – коэффициенты для метода SVR;
- $\{\alpha_{\tilde{q}}\}_{\tilde{q}=0, \tilde{Q}; q=0, \tilde{q}+1}$ – коэффициенты адаптивной линейной комбинации элементарных

прогнозов.

Численные методы оценки параметров модели

Оценки матриц главных компонент основаны на оценках ковариационных матриц векторов признаков подграфов, выражения для которых приведены в (10). Описание метода выбора порядка и настройки параметров модели временных рядов (сезонной ARIMA и векторной VARMA) представлены в монографиях [319, 320] и для краткости изложения здесь не приводятся. По той же причине опускается описание метода решения задачи оценки коэффициентов для метода опорных векторов, который детально описан в работе [321] и заключается в решении двойственной задачи математического программирования. Наконец, способ получения коэффициентов адаптивной линейной комбинации сводится к независимому решению задач оценки коэффициентов для обычной линейной регрессии, выполняемой методом наименьших квадратов (МНК), в количестве $(Q + 1)$ штук (для

каждой из возможных ситуаций, описание которых дано в соответствующем подразделе выше). Каждая из указанных МНК задач сводится в конечном итоге к решению системы нормальных уравнений, явный вид которого известен из теории оценивания и здесь, для краткости изложения, также не приводится.

Существенным моментом при оценивании всех указанных параметров является способ формирования выборочных данных, которые используются в качестве прецедентов обработки в вычислительных процедурах оценки. Ниже представлены описания способов формирования выборочных данных для различных методов. При описании этих способов используются несколько новых обозначений.

Обозначим как Θ множество исторических моментов времени t , для которых относительно всех сегментов УДС известны параметры транспортного потока, то есть величины:

$$\begin{aligned} v(w, t + n\Delta), \quad n = \overline{1, N}, \\ v(w, t - m\Delta), \quad m = \overline{0, M}. \end{aligned}$$

Из множества Θ выделим два непересекающихся подмножества Θ^* , Θ^{**} :

$$\begin{aligned} \Theta^* \subseteq \Theta, \quad \Theta^{**} \subseteq \Theta; \\ \Theta^* \cap \Theta^{**} = \emptyset. \end{aligned}$$

Подмножество Θ^* будем использовать для оценки параметров, а Θ^{**} – для проверки качества настройки модели. Также обозначим $t_0 = \min_{t \in \Theta^*} t$, и пусть $\tilde{N} = \arg \max_{\{n \in \mathbb{N}: t_0 + n\Delta \in \Theta^*\}} n$.

Обучающая и тестовая выборки для модели ARIMA

Для каждого сегмента $w \in E$ коэффициенты оцениваются независимо. В качестве обучающей выборки используется последовательность вида:

$$v(w, t_0), v(w, t_0 + \Delta), \dots, v(w, t_0 + \tilde{N}\Delta),$$

Для проверки качества модели выборка формируется аналогичным образом, но используется подмножество временных отсчётов Θ^{**} .

Обучающая и тестовая выборки для модели VARMA

Для каждого подграфа E_k коэффициенты оцениваются независимо. В качестве обучающей выборки используется последовательность вида:

$$\bar{v}_0^k(t_0), \bar{v}_0^k(w, t_0 + \Delta), \dots, \bar{v}_0^k(w, t_0 + \tilde{N}\Delta).$$

Для проверки качества модели выборка формируется аналогичным образом, но используется подмножество временных отсчётов Θ^{**} .

Обучающая и тестовая выборки для метода SVR

Для каждого сегмента w подграфа E_k коэффициенты оцениваются независимо. В качестве обучающей выборки используется набор (множество) прецедентов (вход;выход), формируемый следующим образом:

$$\left\{ \left(\bar{v}_M^k(t_0 + \ell\Delta); v(w, t_0 + \ell\Delta + n\Delta) \right) \right\}_{\ell=0}^{\tilde{N}},$$

где n – задаваемый и фиксированный порядковый номер требуемого прогноза.

Для проверки качества модели выборка формируется аналогичным образом, но используется подмножество временных отсчётов Θ^{**} .

Обучающая и тестовая выборки для адаптивной линейной регрессии

Для каждого подграфа E_k коэффициенты оцениваются независимо. В качестве обучающей выборки используется набор (множество) прецедентов (вход;выход), формируемый следующим образом:

$$\left\{ \left(\begin{array}{l} TS(k, M, t, n\Delta), \\ SVR(k, M, t, n\Delta), \\ PF_{\sigma_0}(k, M, t, n\Delta), \\ \dots \\ PF_{\sigma_{Q-1}}(k, M, t, n\Delta); \\ v(w, t + n\Delta) \end{array} \right) \right\}_{\substack{t=t_0+\ell\Delta; \\ \ell=0, \tilde{N}}}.$$

где n – задаваемый и фиксированный порядковый номер требуемого прогноза.

Для проверки качества модели выборка формируется аналогичным образом, но используется подмножество временных отсчётов Θ^{**} .

2.1.4 Экспериментальные исследования

Цели проводимых экспериментальных исследований были следующие.

Эксперимент 1: определение числа главных компонент в методе PCA, которые с достаточной точностью позволяют описать вектор пространственно-временного распределения транспортного потока в подграфах УДС.

Эксперимент 2: выбор используемого в адаптивной комбинации алгоритма прогнозирования временных рядов (ARIMA или VARMA).

Эксперимент 3: оценка эффективности предлагаемой адаптивной комбинации и сравнение её с качеством отдельных алгоритмов.

Эксперимент 4: определение зависимости времени работы предлагаемой адаптивной комбинации от числа главных компонент, используемых в методе PCA для описания вектора пространственно-временного распределения потоков.

Экспериментальные исследования разработанного метода проводились для области УДС г. Самары. Рассматриваемая часть дорожной сети состоит из 3387 сегментов. Количество ОТС, подключённых к системе мониторинга – более 1500, новые координаты положения ОТС поступают с усреднённой периодичностью в 30 секунд.

Для экспериментальных исследований производилось разбиение графа дорожной сети на подграфы по территории размером 1 км^2 . Каждый подграф содержал в среднем 50 ребер. Число используемых в векторе признаков архивных значений параметров транспортного потока для каждого сегмента сети $M = 6$, значение временного интервала $\Delta = 10$ минут, т.е. вектор признаков содержит архивные данные за последний час.

Ниже представлены результаты экспериментов.

2.1.4.1 Эксперимент 1. Определение числа главных компонент в методе PCA

В этом эксперименте проверялась гипотеза, что объём основной информации о пространственно-временном распределении транспортных потоков составляет малую часть от всего объёма данных. Для этого исследовалась зависимость доли остаточной дисперсии, рассчитываемой по правой части формулы (11), в зависимости от числа используемых главных компонент. График зависимости показан на рисунке 8.

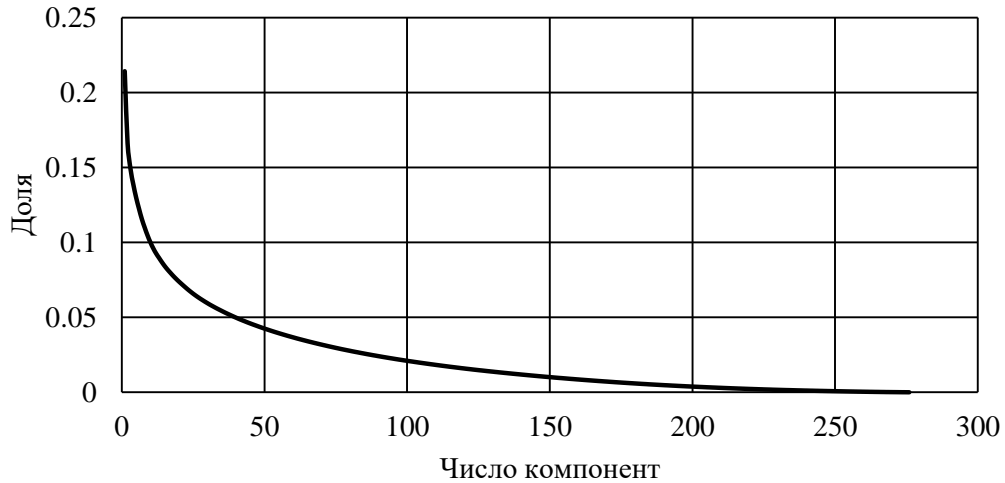


Рисунок 8 – Зависимость доли остаточной дисперсии от числа выбираемых главных компонент

При использовании первых 40 главных компонент вектора признаков (при типовом числе компонент вектора в 300 элементов) доля остаточной дисперсии не превышает 5%. Именно эта величина в 5% была использована в экспериментах далее.

2.1.4.2 Эксперимент 2. Выбор алгоритма прогнозирования временных рядов

Сравнение моделей временных рядов ARIMA и VARMA проводилось на выборке, состоящей из значений параметров транспортных потоков на дорожных сегментах за 12 будних дней методом перекрёстной проверки. Сравнение проводилось по критериям среднеквадратической ошибки (СКО), средней абсолютной ошибке (MAE) и средней относительной ошибке (MAPE):

$$СКО = \frac{1}{N} \sqrt{\sum_{i=1}^N (\hat{v}_i - v_i)^2},$$

$$MAE = \frac{1}{N} \sum_{i=1}^N |\hat{v}_i - v_i|,$$

$$СКО = \frac{1}{N} \sum_{i=1}^N \frac{|\hat{v}_i - v_i|}{v_i},$$

где \hat{v}_i – прогнозное значение параметра транспортного потока;

v_i – реальное значение параметра транспортного потока;

N – объем выборки.

Результаты сравнения на обучающей и контрольной выборках приведены в таблицах 2 и 3.

Таблица 2 – Сравнение алгоритмов на обучающей выборке

–	СКО, с	MAE, с	MAPE
ARIMA	49,12	23,74	0,3344
VARMA	46,99	23,67	0,3389

Таблица 3 – Сравнение алгоритмов на контрольной выборке

–	СКО, с	MAE, с	MAPE
ARIMA	49,82	24,083	0,338
VARMA	49,68	24,856	0,355

Модели временных рядов показали схожие результаты на контрольной выборке, лучший результат по быстрдействию показала модель VARMA, поэтому в качестве метода прогнозирования временных рядов в модели адаптивной комбинации предлагается использовать эту модель.

2.1.4.3 Эксперимент 3. Оценка эффективности

В рамках этого направления исследовалась зависимость величины СКО прогнозных значений параметров транспортных потоков для предлагаемого метода адаптивной комбинации и отдельных алгоритмов прогнозирования, входящих в состав его модели, от горизонта прогноза. Для простоты анализа результата далее представлены результаты анализа для одного подграфа УДС. В качестве прогнозируемого параметра транспортного потока используется время прохождения сегмента УДС.

В первую очередь оценивалась эффективность прогнозов методом потенциальных функций с разными значениями параметра σ . Исследование качества прогнозов проводилось на выборке, состоящей из значений параметров транспортного потока на дорожных сегментах за 24 будних дня. Исследование проводилось методом перекрёстной проверки, размер одной части контрольной выборки составлял один день.

Доля полученных прогнозов от общего объема обучающей и контрольной выборок для выбранных σ приведена в таблице 4.

Таблица 4 – Доля непустых прогнозов методом потенциальных функций

–	σ_0	σ_1	σ_2	σ_3
Обучающая выборка	1	0,781	0,613	0,407
Контрольная выборка	1	0,783	0,617	0,412

График зависимости средней абсолютной и средней относительной ошибок прогнозов методом потенциальных функций (PF) на обучающей выборке показан на рисунках 9 и 10.

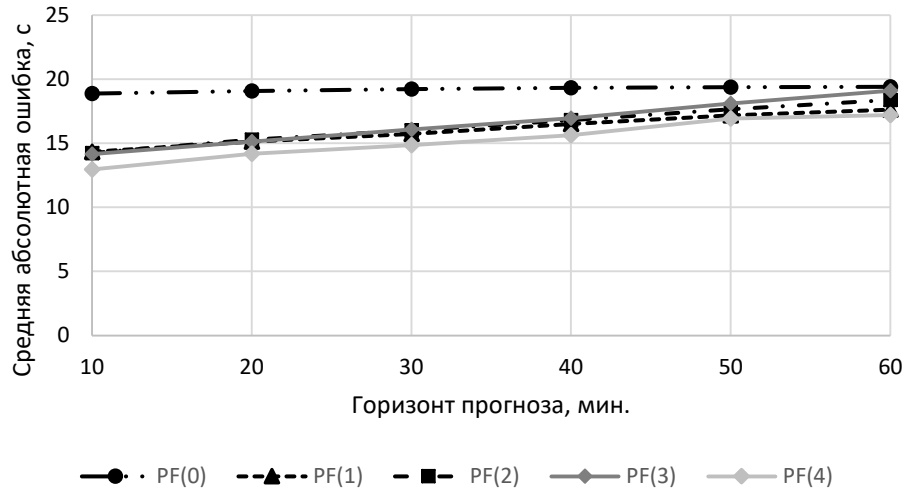


Рисунок 9 – Зависимость MAE от горизонта прогноза на обучающей выборке

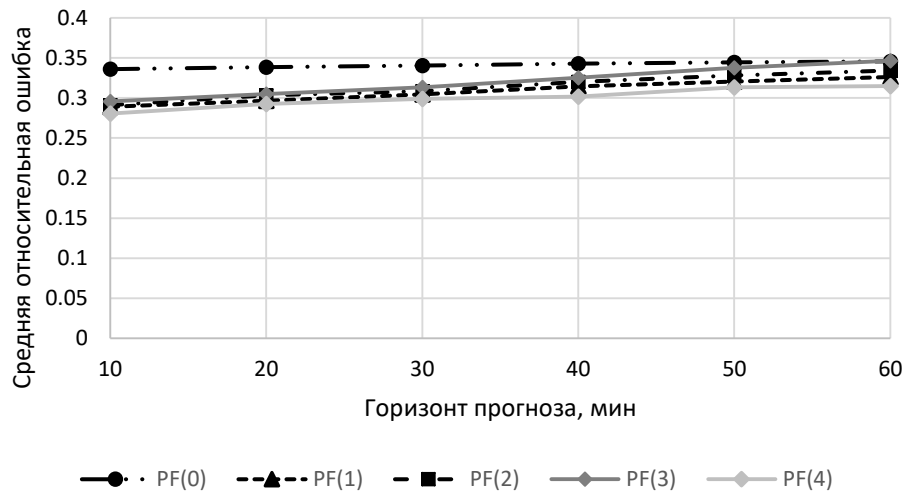


Рисунок 10 – Зависимость MARE от горизонта прогноза на обучающей выборке

Худший результат, как и предполагалось, показал метод потенциальных функций с наибольшим значением параметра σ_0 (PF(0) на графиках). При увеличении горизонта прогноза качество прогноза ухудшается, причем увеличение σ может снижать ошибку. Это объясняется, по-видимому, нестабильной дорожной ситуацией в рассматриваемой области. На контрольной выборке зависимости ошибок имеют схожий вид.

Далее исследовалось качество модели адаптивной комбинации по сравнению с элементарными прогнозами. В качестве элементарных прогнозов использовались прогнозы

методом опорных векторов (на графиках – SVR), моделью временных рядов VARMA (на графиках – TS) и методом потенциальных функций с наибольшим значением σ (PF(0)), дающим результат прогноза для всех участков УДС.

Графики зависимости СКО, средней абсолютной и средней относительной ошибок от горизонта прогноза на обучающей выборке показаны на рисунках 11, 12 и 13.

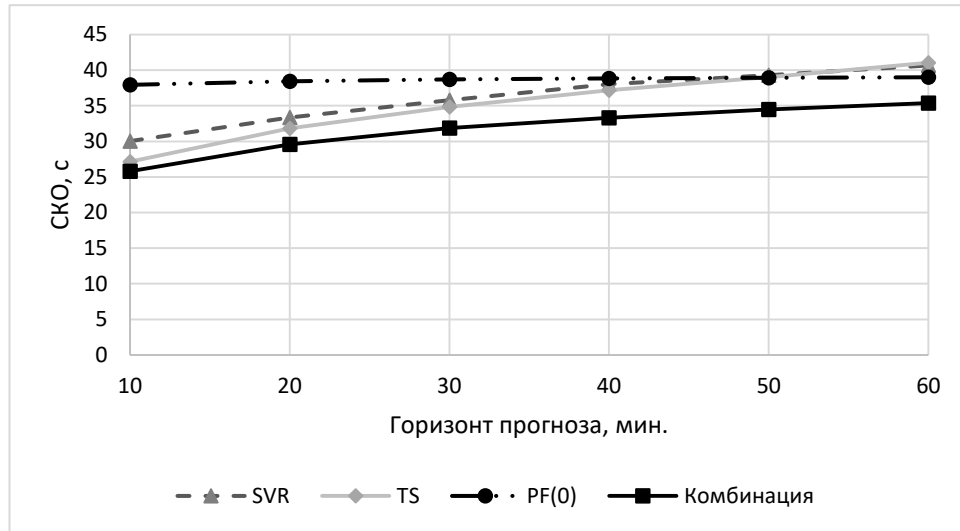


Рисунок 11 – Зависимость СКО от горизонта прогноза на обучающей выборке

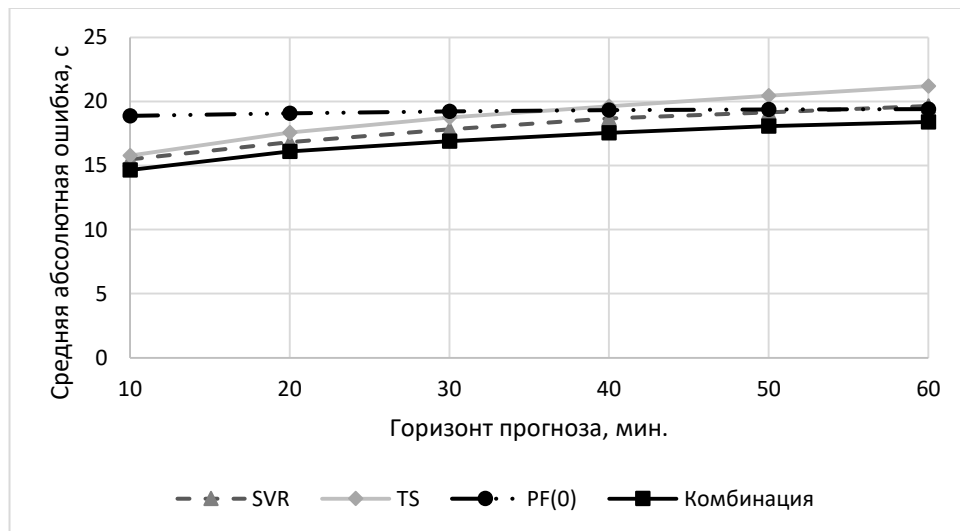


Рисунок 12 – Зависимость MAE от горизонта прогноза на обучающей выборке

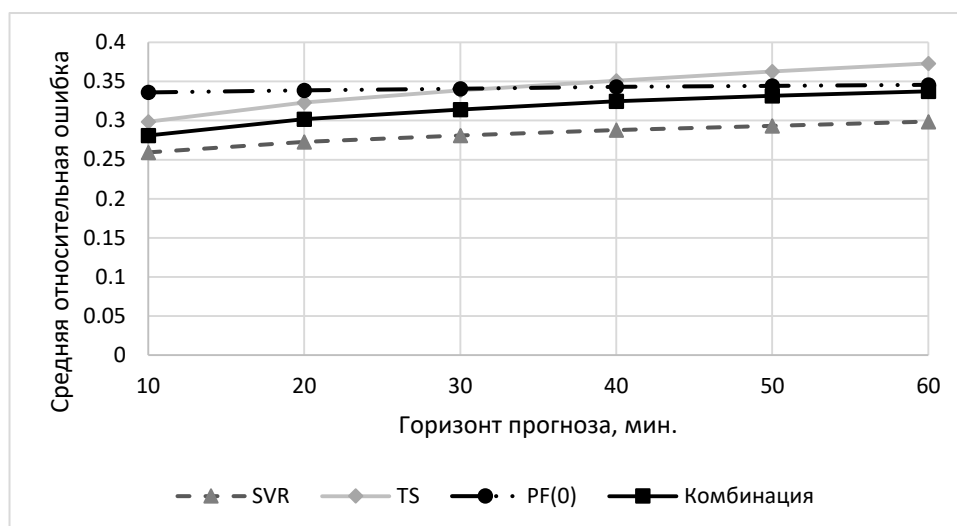


Рисунок 13 – Зависимость MAPE от горизонта прогноза на обучающей выборке

Модель адаптивной комбинации элементарных прогнозов показала лучшие результаты по критериям СКО и средней абсолютной ошибке, но уступает элементарному прогнозу методом опорных векторов по критерию средней относительной ошибки.

Графики зависимости СКО, средней абсолютной и средней относительной ошибок от горизонта прогноза на контрольной выборке показаны на рисунках 14, 15 и 16.

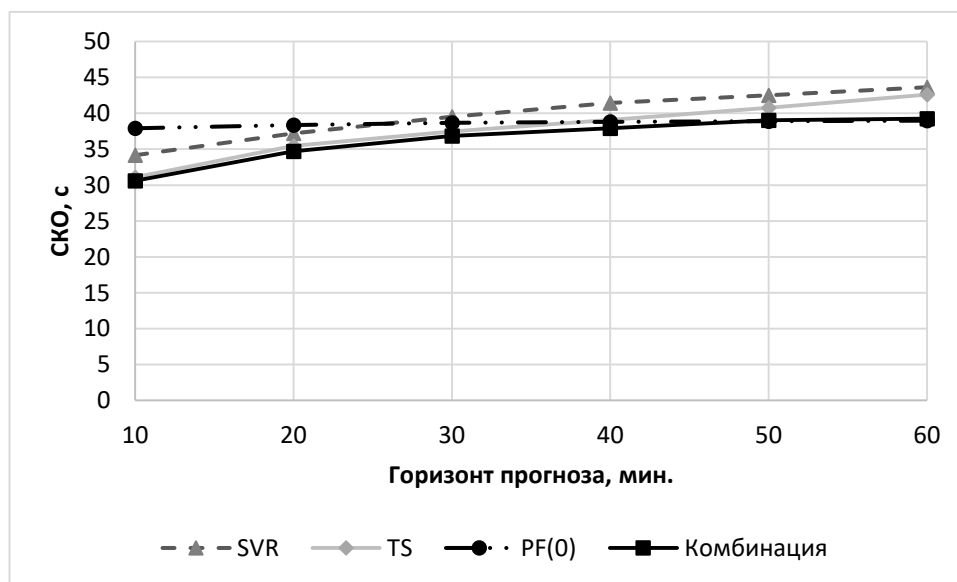


Рисунок 14 – Зависимость СКО от горизонта прогноза на контрольной выборке

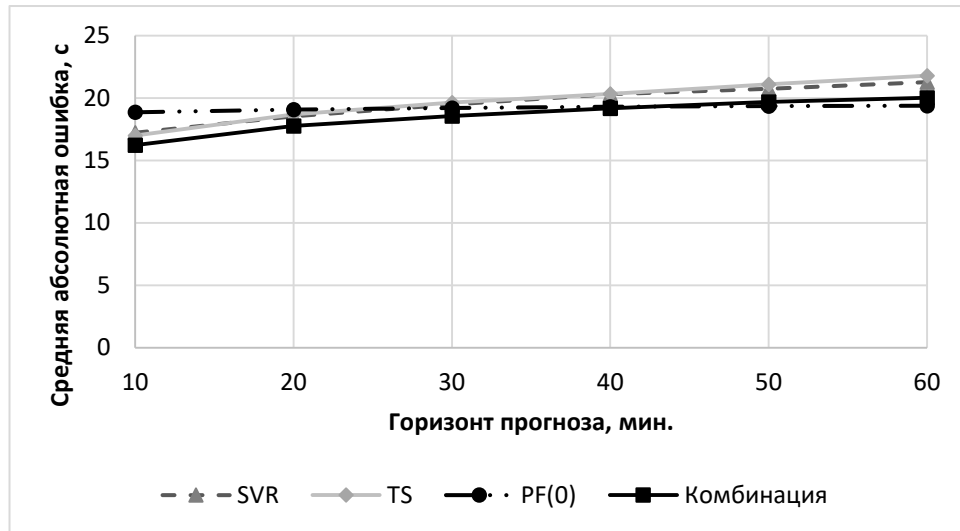


Рисунок 15 – Зависимость MAE от горизонта прогноза на контрольной выборке

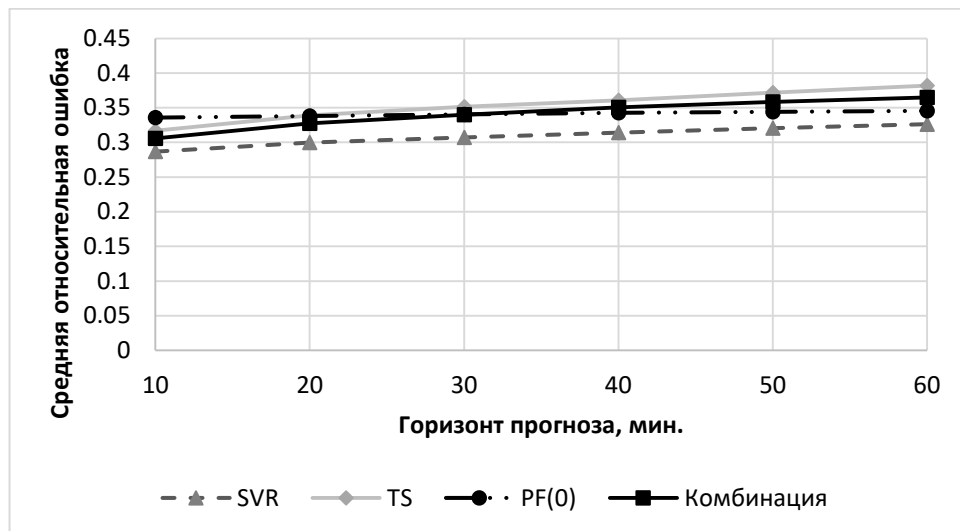


Рисунок 16 – Зависимость MAPE от горизонта прогноза на контрольной выборке

На контрольной выборке адаптивная модель показала схожие результаты.

На представленных выше графиках исследовалось качество работы модели на каждом из дорожных сегментов в отдельности. Часто такие сегменты имеют среднее время проезда меньше минуты, что объясняет большое значение средней относительной ошибки в расчетах. К тому же, для коротких сегментов оценивание реальных параметров транспортных потоков по данным GPS\ГЛОНАСС-наблюдений уже может содержать ошибки. Поэтому интерес представляет прогноз ситуации на нескольких сегментах по ходу движения.

Графики зависимости СКО, средней абсолютной и средней относительной ошибок от горизонта прогноза на контрольной выборке для нескольких сегментов УДС показаны на рисунках 17, 18 и 19.

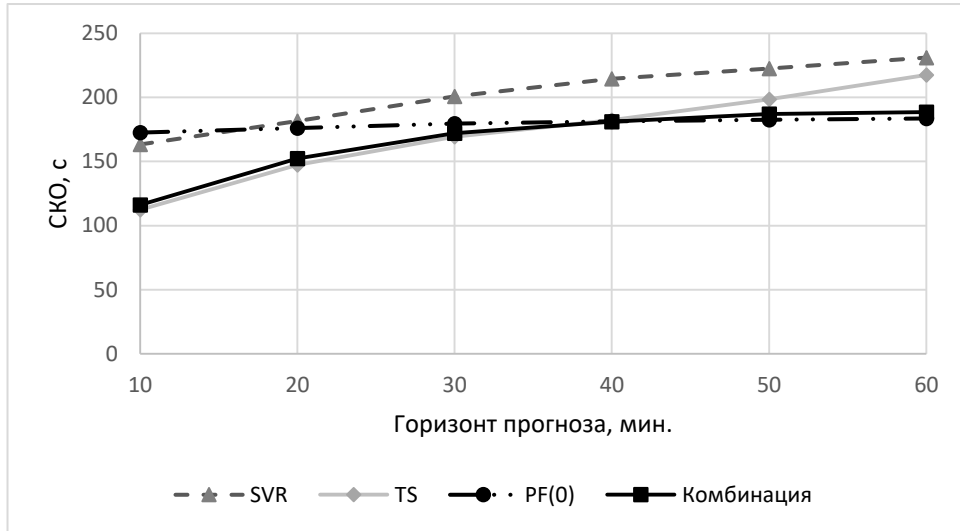


Рисунок 17 – Зависимость СКО от горизонта прогноза на контрольной выборке

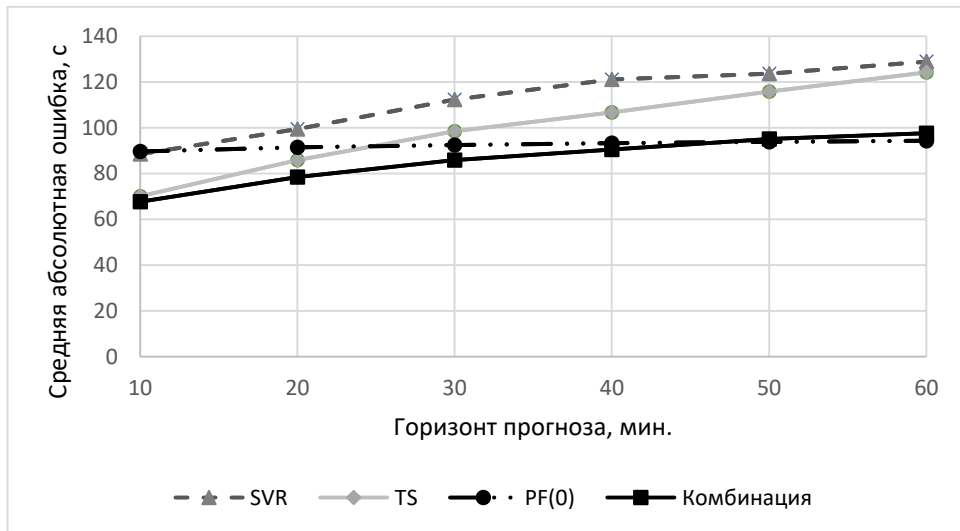


Рисунок 18 – Зависимость MAE от горизонта прогноза на контрольной выборке

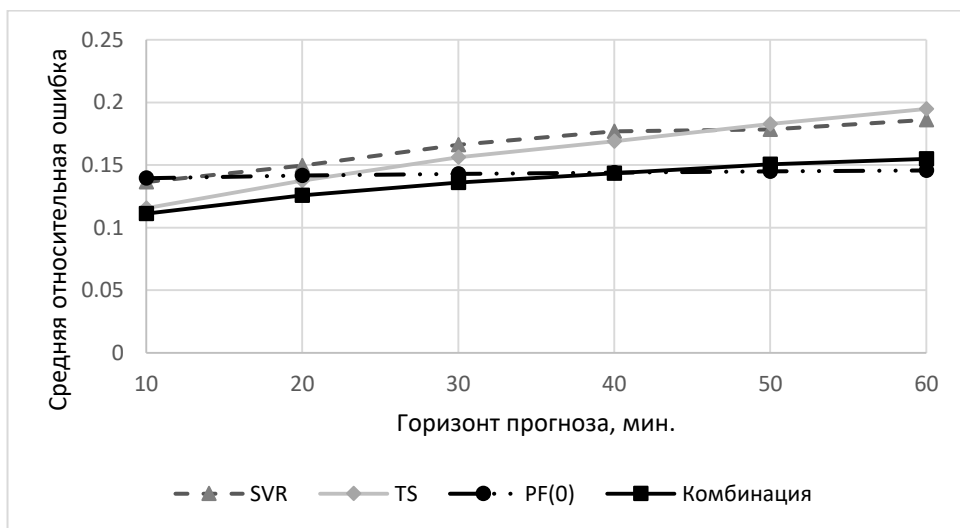


Рисунок 19 – Зависимость MAPE от горизонта прогноза на контрольной выборке

При прогнозе параметров транспортных потоков на нескольких сегментах УДС в совокупности элементарный прогноз методом опорных векторов уступил модели адаптивной комбинации по критериям средней абсолютной и средней относительной ошибок.

Из представленных результатов видно, что модель адаптивной комбинации даёт более качественный результат практически на всём горизонте прогноза при прогнозировании параметров транспортных потоков для нескольких сегментов УДС. Средняя абсолютная ошибка модели составляет около 2 минут при горизонте прогноза в 1 час.

Более детальный анализ результатов показал, что наибольший вклад в величины ошибок вносят те моменты состояния УДС, которые соответствуют заторам на сегменте. Данный факт делает актуальным как разработку способов предварительной фильтрации данных для обучения и оценки эффективности методов, что используется в некоторых существующих работах, так и собственно методов «обнаружения» подобных ситуаций с последующим изменением метода прогнозирования.

2.1.4.4 Эксперимент 4. Определение зависимости времени расчёта прогноза от числа главных компонент

В этом эксперименте исследовалась зависимость времени работы предлагаемой адаптивной комбинации алгоритмов от числа используемых главных компонент, обеспечивающих необходимую точность представления исходных данных. График зависимости времени работы от относительного числа оставленных в представлении вектора признаков параметров транспортного потока показан на рисунке 20. В эксперименте прогнозировалось значение параметров потока для 70 подграфов УДС при горизонте прогноза 1 час. Характеристики используемой ПЭВМ: процессор Intel Core i5-3740 3.20 GHz, оперативная память 8 ГБ, жёсткий диск 1 ТБ, ОС – Windows 8.1.

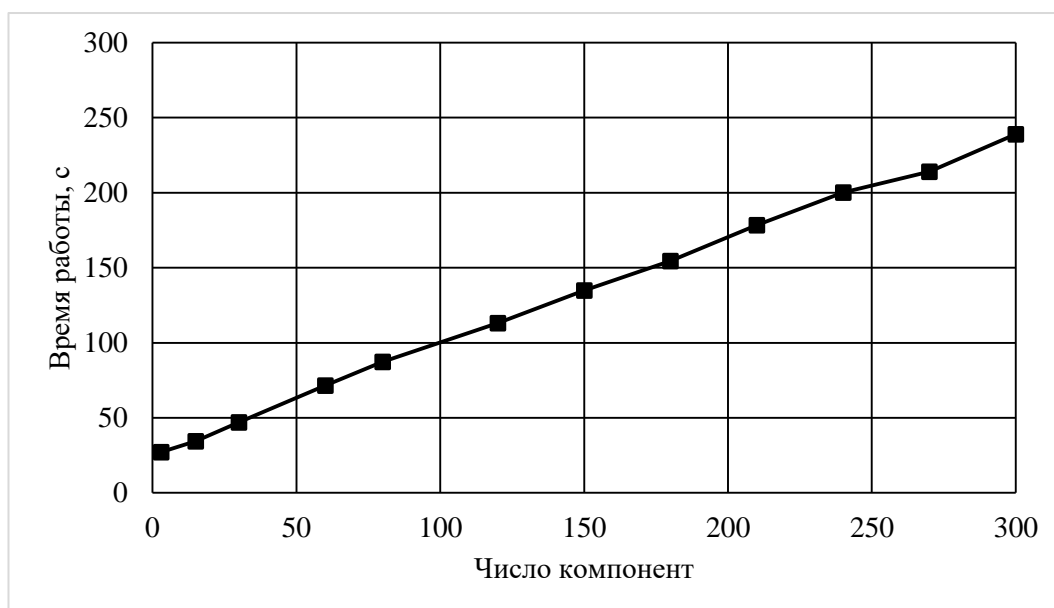


Рисунок 20 – Зависимость времени работы от числа оставленных компонент

Детальный анализ времени работы отдельных алгоритмов показал, что большую часть времени работы алгоритма комбинации занимает оценка параметров транспортного потока методом опорных векторов.

2.2 Прогнозирование параметров транспортного потока с использованием подхода к обработке больших данных

В данном разделе предложена модификация алгоритма прогнозирования, представленного в разделе 2.1. Данная модификация направлена на возможность использования алгоритма при обработке «больших» данных, а также для повышения точности прогнозирования. Предлагаемая модификация алгоритма, подробно описанного в разделе 2.2.2, основывается на непараметрическом регрессионном методе k ближайших соседей. В разделе 2.2.3 представлен модифицированный метод разбиения графа УДС на подграфы с учетом информации о распределении транспортных потоков в сети. Реализация представленного алгоритма с использованием подхода MapReduce описана в разделе 2.2.4. Экспериментальные исследования представленного алгоритма для транспортной сети г. Самары представлены в разделе 2.2.5.

2.2.1 Основные обозначения и постановка задачи

Как и в предыдущем разделе, улично-дорожную сеть будем рассматривать как ориентированный граф $G = (V, E)$, в котором вершины V , $N_V = |V|$ соответствуют

перекрёсткам дорожной сети, рёбра E , $N_E = |E|$ соответствуют сегментам дорожной сети между перекрёстками.

Обозначим $x_t^i \in R$ – наблюдаемый параметр транспортного потока (ТП) на ребре $i \in E$ в момент времени t , $x^i = (x_{t_{cur}-T}^i, \dots, x_{t_{cur}}^i)$ – набор наблюдаемых параметров ТП на ребре $i \in E$ в момент времени t_{cur} за T временных интервалов, X_t – набор наблюдаемых параметров для всех ребер графа в момент времени t , X – набор наблюдаемых параметров за T временных интервалов. В качестве параметра транспортного потока могут выступать следующие величины:

- 1) средняя скорость транспортного потока;
- 2) плотность потока;
- 3) поток (собственно величина потока).

Учитывая введённые обозначения, формальная постановка задачи получения краткосрочного прогноза для заданного параметра транспортного потока может быть сделана следующим образом:

Имея заданный граф $G = (V, E)$ и последовательность X наблюдаемых значений параметров транспортного потока, рассчитать оценку параметров транспортного потока \hat{X}_t через T_{ph} временных интервалов: $\hat{X}_{t+1}, \hat{X}_{t+2}, \dots, \hat{X}_{t+T_{ph}} = f(X; G)$.

2.2.2 Модель прогнозирования параметров транспортного потока

Предлагаемое в работе решение задачи краткосрочного прогнозирования транспортных потоков основывается на непараметрическом регрессионном методе k ближайших соседей.

Для применения метода необходимо решить следующие задачи:

- 1) определить вектор признакового описания транспортного потока;
- 2) определить подходящую метрику расстояния для определения близости между векторами признаков, описывающих текущие характеристики транспортного потока и архивные данные;
- 3) определить функцию вычисления прогноза по набору ближайших соседей.

Решение каждой задачи описано в следующих подразделах.

2.2.2.1 Вектор признаков

Выбор вида вектора признаков в методе k ближайших соседей зависит от конкретного приложения метода. Для решения задачи прогнозирования транспортного

потока целесообразно использовать вектор признаков, учитывающий пространственные и временные корреляции параметров транспортного потока.

В работе [98] в качестве вектора признаков предложено использовать значение параметров транспортных потоков на текущем, предыдущем и следующем сегменте дорожной сети за T временных интервалов:

$$(x_{t-T}^j, \dots, x_{t-1}^j, x_t^j, x_{t-T}^{j-1}, \dots, x_{t-1}^{j-1}, x_t^{j-1}, x_{t-T}^{j+1}, \dots, x_{t-1}^{j+1}, x_t^{j+1}). \quad (20)$$

Однако такое описание не учитывает транспортную ситуацию на смежных сегментах. Кроме того, в некоторых случаях предыдущий / следующий дорожный сегмент не может быть определён однозначно. Поэтому для описания транспортных потоков предлагается формировать вектор признаков с учётом характеристик транспортного потока в пространственно-компактном подграфе графа дорожной сети.

В работе предложен следующий способ формирования вектора признаков:

1) граф улично-дорожной сети разбивается на пространственно компактные непересекающиеся подграфы $\{G_j\}_{j=1, \dots, K}$. Набор ребер конкретного подграфа G_j зависит от выбранного алгоритма разбиения графа. Для каждого подграфа j формируется вектор признаков X^j из параметров транспортного потока

$$X^j = (x^i)_{i \in G_j}. \quad (21)$$

2) выполняется снижение размерности исходного вектора признаков X^j путём устранения пространственно-временной зависимости значений параметров потоков и формируется итоговый вектор признаков подграфа j , состоящий из M_j компонент: $y^j = (y_0^j, \dots, y_{M_j-1}^j)$, $M_j < |G_j|$.

3) результирующий вектор признаков S_i для каждого сегмента $i \in E$ формируется из исходного вектора признаков x^i для сегмента i и вектора признаков y^j для подграфа j , которому принадлежит сегмент:

$$S_i = (x^i, y^j), \quad i \in E. \quad (22)$$

Способы разбиения графа транспортной сети на подграфы дорожной сети подробно описаны в разделе 2.2.3.

2.2.2.2 Мера близости

Для определения близости между векторами признаков необходимо определить подходящую метрику расстояния. В литературе описаны разные варианты определения меры расстояния между векторами, в т.ч. евклидово расстояние, расстояние Махаланобиса, расстояние Хэмминга.

В данной работе используется взвешенное евклидово расстояние с учётом тренда, предложенное в работе [98], модифицированное для использования вектора признаков, описывающего подграфы транспортной сети $\{X\}$:

$$d(S_i, \bar{S}_i) = d^{link}(x^i, \bar{x}^i) + \gamma d^{pca}(y^j, \bar{y}^j), \quad (23)$$

$$d^{link}(x, \bar{x}) = \alpha \sqrt{\sum_{t=1}^T \beta^{T-t+1} (x_t - \bar{x}_t)^2} + (1 - \alpha) \sqrt{\sum_{t=2}^T \sum_{\delta=1}^{t-1} ((x_t - x_\delta) - (\bar{x}_t - \bar{x}_\delta))^2} \quad (24)$$

$$d^{pca}(y, \bar{y}) = \sqrt{\sum_{n=1}^M (y_n - \bar{y}_n)^2} \quad (25)$$

где $0 \leq \alpha \leq 1$, $0 < \beta \leq 1$, $0 \leq \gamma \leq 1$ – коэффициенты;

T – количество временных интервалов в векторе признаков;

N – количество элементов в векторе признаков, описывающего подграф;

$d(x, \bar{x}^i)$ – расстояние между вектором признаков, описывающим текущее распределение транспортных потоков, и i -м архивным вектором признаков;

S – значение вектора признаков, описывающего текущий транспортный поток;

\bar{S}_i – значение вектора признаков, описывающего i -ый архивный транспортный поток;

x_t, \bar{x}_t^i – значения векторов признаков, описывающих соответственно текущие и архивные значения транспортного потока на заданном дорожном сегменте за временной интервал t ;

y_n, \bar{y}_n^i – n -ые значения векторов признаков, описывающих соответственно текущее и архивное состояние транспортного потока в подграфе.

2.2.2.3 Функция прогнозирования

Традиционным подходом для оценки значения при использовании метода k ближайших соседей в задаче регрессии является выбор среднего или средневзвешенного значения по k ближайшим к оцениваемому векторам признаков [56].

Функция прогнозирования по средневзвешенному значению ближайших векторов признаков имеет вид:

$$\hat{X}_{T+1} = \sum_{k=1}^K \frac{d_k^{-1}}{\sum_{k=1}^K d_k^{-1}} X_{T+1}^k \quad (26)$$

где \hat{X}_{T+1} – прогнозное значение параметра транспортного потока в момент времени $T + 1$;

X_{T+1}^k – значение параметра транспортного потока k -го ближайшего соседа в соответствующий момент времени;

d_k – расстояние между вектором признаков, описывающим текущее состояние транспортных потоков, и k -м ближайшим соседом.

В работе используется комбинированная функция прогнозирования, учитывающая средневзвешенное значение векторов признаков и тренд прогноза:

$$\hat{X}_{T+1} = \theta \sum_{k=1}^K \frac{d_k^{-1}}{\sum_{k=1}^K d_k^{-1}} X_{T+1}^k + (1 - \theta) \left(X_T + \frac{1}{KT} \sum_{k=1}^K \sum_{t=1}^T (X_{T+1}^k - X_t^k) \right) \quad (27)$$

где $0 \leq \theta \leq 1$ – коэффициент;

K – количество ближайших соседей.

2.2.3 Разбиение графа на подграфы

В работе используются несколько способов кластеризации транспортной сети:

1) кластеризация по территориальному признаку: в один подграф попадают рёбра графа, соответствующие сегментам дорожной сети, принадлежащим указанной прямоугольной области;

2) кластеризация с учётом матрицы расстояний: для каждого ребра графа в соответствующий ему подграф попадают рёбра, находящиеся на расстоянии от выбранного ребра, не превышающем заданное;

3) кластеризация на основе информации о характеристиках транспортных потоков в течение определённого периода времени.

2.2.3.1 Кластеризация по территориальному признаку

Данный способ кластеризации был подробнее рассмотрен в разделе 2.1.3.1. Для удобства изложения, в данном разделе приведено его краткое описание.

Введём дополнительные обозначения. Пусть ребру графа $i \in E$ соответствует сегмент дорожной сети e_i с известными координатами начала и конца сегмента $x_{start} = (x_{start}^0, x_{start}^1)$ и $x_{end} = (x_{end}^0, x_{end}^1)$ соответственно.

Тогда кластеризация графа по территориальному признаку может быть описана следующим образом:

- 1) выбирается число формируемых подграфов в количестве M_0, M_1 ;
- 2) в подграф G_m^{bbox} с номером $m = m_0 M_0 + m_1$, ($m_0 = \overline{0, M_0 - 1}$; $m_1 = \overline{0, M_1 - 1}$) относятся те рёбра $i \in E$, координаты одной из вершин соответствующих дорожных сегментов которых попадают в соответствующую прямоугольную область Π_{m_0, m_1}

$$G_m^{bbox} \equiv \{i \in E: x_{start}^i \in \Pi_{m_0, m_1} \vee x_{end}^i \in \Pi_{m_0, m_1}\}$$

где

$$\begin{aligned} \Pi_{m_0, m_1} &\equiv \left[x_{min}^0 + \frac{m_0}{M_0} \Delta^0, x_{min}^0 + \frac{m_0 + 1}{M_0} \Delta^0 \right] \times \left[x_{min}^1 + \frac{m_1}{M_1} \Delta^1, x_{min}^1 + \frac{m_1 + 1}{M_1} \Delta^1 \right], \\ x_{min}^s &= \min_{v=\{start, end\}, i \in E} x_v^{s, i}, \quad x_{max}^s = \max_{v=\{start, end\}, i \in E} x_v^{s, i}, \\ \Delta^s &= x_{max}^s - x_{min}^s, \quad s = 0, 1. \end{aligned}$$

Число подграфов по вертикали и горизонтали M_0, M_1 выбирается эмпирически. Будем считать, что каждый сегмент дорожной сети может попасть только в один подграф.

Данный способ обладает следующими недостатками:

- 1) прогноз характеристик транспортных потоков на граничных сегментах дорожной сети может быть неточным;
- 2) при формировании подграфов не учитываются характеристики транспортных потоков.

В следующих подразделах рассмотрены два способа выделения подграфов, исправляющие указанные недостатки.

2.2.3.2 Кластеризация с учётом матрицы расстояний

Определим расстояние $r(i, j)$ между двумя рёбрами графа $i \in E$ и $j \in E$ как длину кратчайшего пути от начальной вершины ребра i, x_{start}^i до конечной вершины ребра j, x_{end}^j в невзвешенном графе (т.е. количество рёбер в кратчайшем пути). Расстояние между всеми вершинами графа определяется на основе матрицы смежности графа.

Тогда каждому ребру графа $i \in E$ ставится в соответствие подграф G_i^{dist} по следующему правилу:

$$G_i^{dist} = \{j \in E : r(i, j) \leq R\},$$

где R – эмпирически выбираемое максимальное расстояние, определяющее размер подграфа.

При данном разбиении разные подграфы транспортной сети будут содержать общие рёбра графа.

2.2.3.3 Кластеризация с учётом транспортных потоков

Для кластеризации транспортной сети на несколько однородных регионов с учётом транспортных потоков используется алгоритм Normalized Cut (NCut), который позволяет эффективно выделять главные компоненты и гарантирует пространственную компактность форм сегментов.

Нахождение точного минимума целевой функции в алгоритме NCut является NP-полной задачей, однако дискретное решение может быть аппроксимировано вещественным путём решения обобщённой задачи поиска собственных значений. Кластеризация транспортной сети с помощью алгоритма NCut может быть описана следующим образом:

1) используя граф транспортной сети, установить вес ребра $w(i, j)$ как меру сходства двух сегментов дорожной сети $i \in E, j \in E$:

$$w(i, j) = \begin{cases} \exp(- (V_i - V_j)^2), & r(i, j) < R_{max} \\ 0, & \text{иначе.} \end{cases}$$

2) решить эквивалентную систему собственных векторов для второго наименьшего собственного значения;

3) дискретизировать собственный вектор, соответствующий второму наименьшему собственному значению, разделить граф на два подграфа;

4) повторить процесс рекурсивно для отдельных подграфов, если необходимо.

Результат кластеризации обозначим как $\{G_i^{flow}\}$.

В следующем разделе описана реализация предложенной в данной работе модели краткосрочного прогнозирования транспортного потока с использованием модели распределённых вычислений MapReduce.

2.2.4 Реализация в MapReduce

В процессе решения задачи прогнозирования транспортного потока с помощью предложенной в данной работе модели используется большой объём архивных и текущих данных. Для повышения эффективности производимых вычислений, использующих такой объём информации, предлагаемая модель была реализована с помощью фреймворка Apache Spark [322], использующего модель распределённых вычислений MapReduce [323].

Apache Spark – фреймворк с открытым исходным кодом для реализации распределённой обработки неструктурированных и слабоструктурированных данных, входящий в экосистему проектов Hadoop. В отличие от классического обработчика из ядра Hadoop, реализующего двухуровневую модель MapReduce с дисковым хранилищем, использует специализированные примитивы для рекуррентной обработки в оперативной памяти, благодаря чему позволяет получать значительный выигрыш в скорости работы для некоторых классов задач, в частности, возможность многократного доступа к загруженным в память пользовательским данным делает библиотеку привлекательной для алгоритмов машинного обучения.

В рамках модели MapReduce обработка данных происходит параллельно на нескольких вычислительных узлах. Работа MapReduce состоит из трёх основных этапов: Map, Shuffle и Reduce. На рисунке 21 представлена схема работы предлагаемой модели на основе MapReduce.

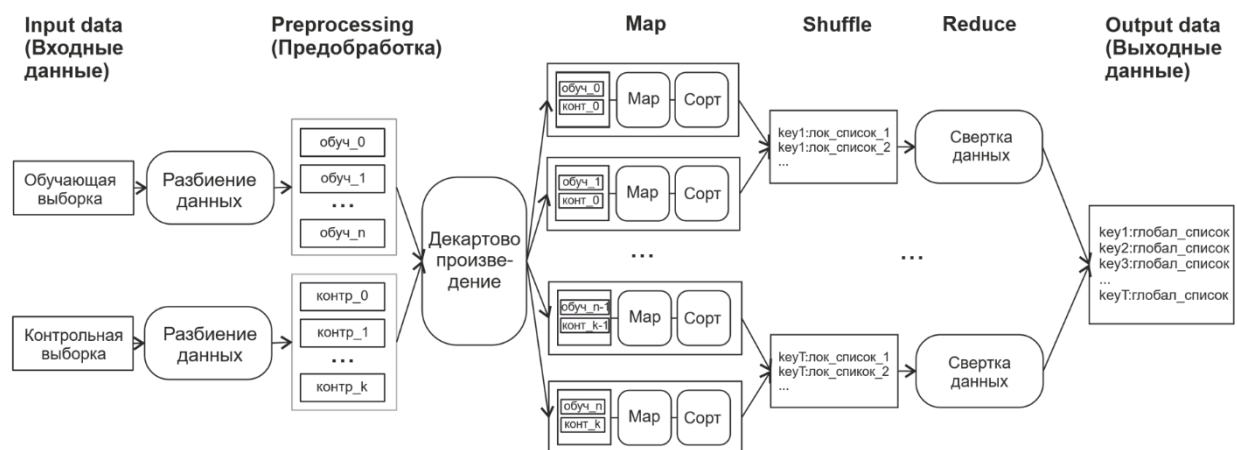


Рисунок 21 – Схема работы предлагаемой модели на основе MapReduce

Как видно из рисунка 21, на первом шаге осуществляется подготовка входных данных для Map этапа. Сначала происходит разбиение исторических и тестовых данных на разделы. Оптимальное количество таких разделов зависит от объёма обрабатываемой информации и количества вычислительных узлов. Затем формируются упорядоченные пары разделов исторических и тестовых данных с помощью декартова произведения. Далее на Map шаге к каждой паре разделов применяется map функция, которая возвращает промежуточный набор пар ключ/значение – тестовый элемент/локальный список k ближайших соседей. На шаге Shuffle осуществляется группировка пар ключ-значение и их передача функции reduce. На заключительном Reduce шаге для каждого элемента тестовых данных множество списков локальных k ближайших соседей преобразуется в результирующий (глобальный) список k ближайших соседей. Полученные списки k ближайших соседей впоследствии используются для нахождения прогнозируемой величины транспортного потока.

Результаты оценки эффективности и масштабируемости предлагаемой модели прогнозирования на основе модели распределённых вычислений MapReduce представлены в разделе 2.2.5.

2.2.5 Экспериментальные исследования

В разделе 2.1.4 экспериментальные исследования проводились для области УДС г. Самары. В данном разделе экспериментальные исследования разработанной модели проводились для всей транспортной сети г. Самары, граф которой включает в себя 26018 дорожных сегментов. В качестве исходных данных для проведения экспериментальных исследований использовались значения средней скорости движения транспортного потока (в км/ч), полученные за 50 дней, начиная с 27 июля 2018 г.

В ходе проведения экспериментальных исследований было произведено сравнение предложенной модели при использовании различных алгоритмов разбиения графа, сезонной модели временных рядов SARIMA и модели TDUD-KNN, представленной в работе [98]. В основе модели TDUD-KNN также лежит метод k ближайших соседей с вектором признаков (20), который содержит в себе информацию о транспортном потоке на заданном и соседних сегментах. Исследования качества прогноза данных моделей проводилось для каждого дня исходных данных, при этом в качестве архивных данных выступали данные о средней скорости транспортного потока, полученные за весь названный выше период времени (50 дней), за исключением рассматриваемого дня.

Для оценки качества прогноза предложенной модели использовались следующие метрики: средняя абсолютная ошибка (MAE) и средняя абсолютная ошибка в процентах (MAPE):

$$MAE = \frac{1}{n} \sum_{t=1}^n |V_t - \hat{V}_t| \quad (28)$$

$$MAPE = \frac{1}{n} \sum_{t=1}^n \frac{|V_t - \hat{V}_t|}{V_t} \times 100\% \quad (29)$$

где V_t – реальное значение величины транспортного потока на временном интервале t ;

\hat{V}_t – спрогнозированное значение величины транспортного потока на временном интервале t ;

n – общее количество прогнозов.

В первую очередь были проведены эксперименты по выбору параметра k (числа соседей) в предложенной модели. Эксперименты проводились на части выборки за 30 дней. Зависимость ошибки MAPE от параметра k представлена на рисунке 22.

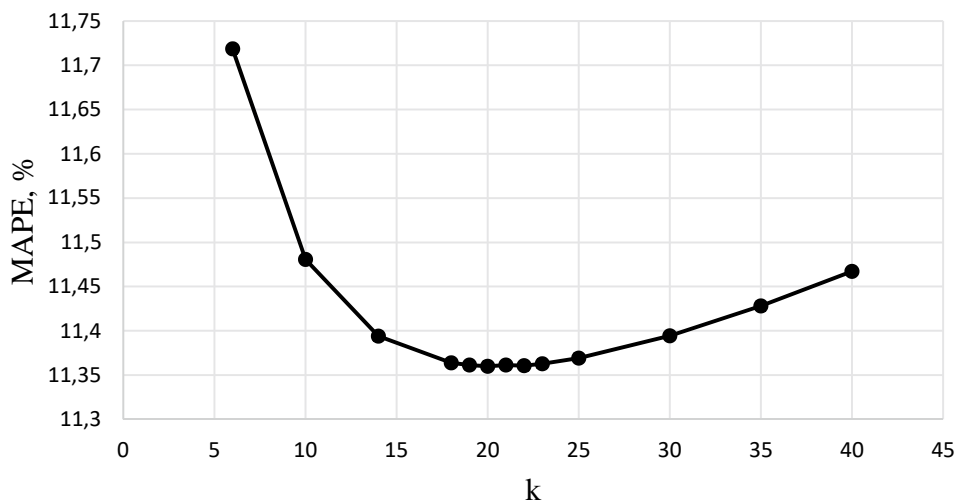


Рисунок 22 – Зависимость MAPE от числа соседей k

Лучший результат был показан для значения $k = 20$, далее в экспериментах использовалось это значение.

В таблице 5 представлено сравнение ошибок MAE и MAPE для предложенной модели при различных способах кластеризации транспортной сети (G^{box} , G^{dist} , G^{flow}), модели SARIMA и модели TDUD-KNN.

Таблица 5 – Сравнение моделей

Модель прогнозирования	MAE	MAPE
G^{box}	2,654	11,45
G^{dist}	2,653	11,42
G^{flow}	2,646	11,4
TDUD-KNN	2,732	11,76
SARIMA	2,677	11,64

Анализ представленных результатов показывает, что предлагаемая в данной работе модель даёт более качественный результат прогноза, чем известная ранее модель. Согласно полученным результатам, все алгоритмы кластеризации графа показали схожее качество прогноза, с небольшим преимуществом лучший результат был показан при использовании алгоритма кластеризации графа транспортной сети с учётом информации о транспортных потоках.

Кроме того, можно сделать вывод, что по критерию средней относительной ошибки (MAE) предложенная в данном разделе модификация алгоритма прогнозирования позволила улучшить результаты, показанные базовым алгоритмом (раздел 2.1).

На рисунках 23, 24 представлены результаты оценки качества прогноза предложенной модели с алгоритмом кластеризации с учётом транспортных потоков (NCut), моделей SARIMA и TDUD-KNN за две недели. Как видно из рисунков, качество прогноза предложенной модели превосходит качество прогноза ранее известной модели в каждый из рассматриваемых дней. Результаты предложенной модели и модели SARIMA близки, однако точность предложенной модели может быть увеличена за счёт накопления большего объёма обучающей выборки.

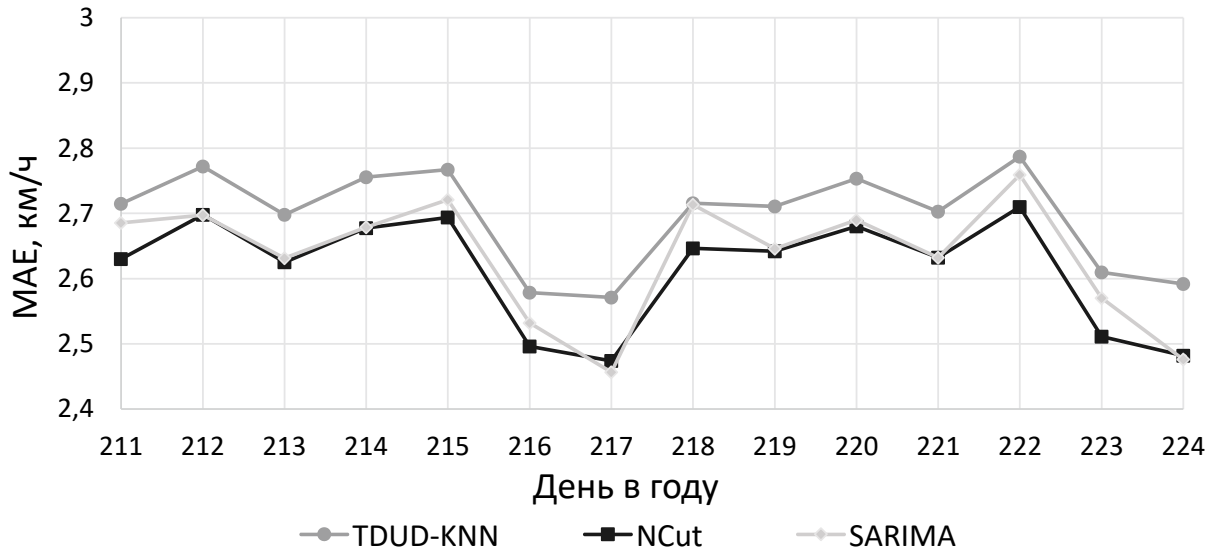


Рисунок 23 – Сравнение значений MAE предложенной модели, моделей SARIMA и TDUD-KNN

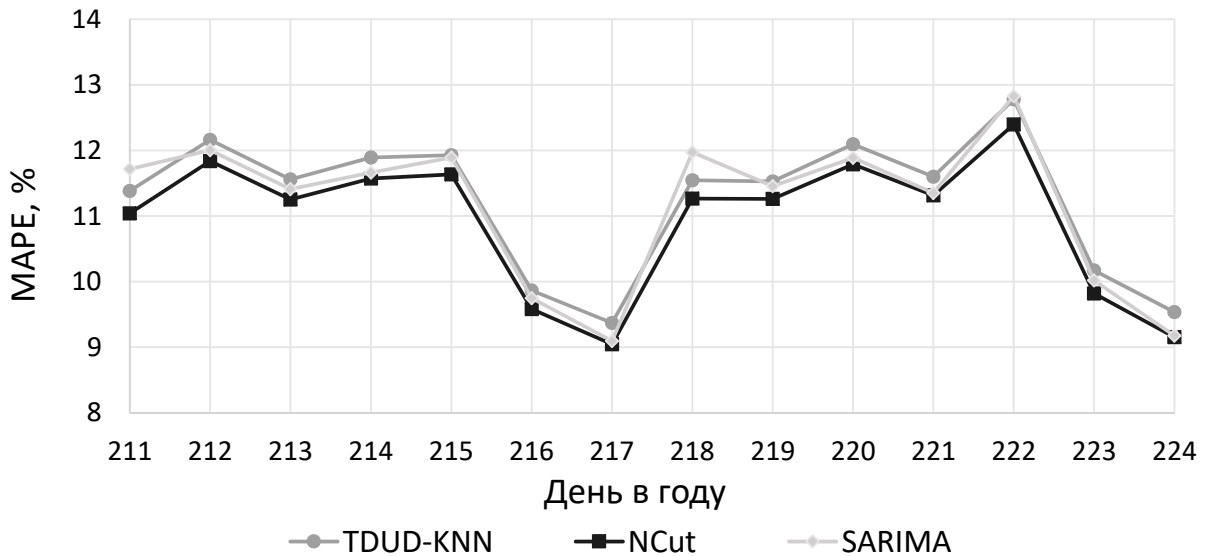


Рисунок 24 – Сравнение значений MAPE предложенной модели, моделей SARIMA и TDUD-KNN

Анализируя полученные результаты, можно сделать вывод, что в выходные дни ошибка прогноза заметно снижается при использовании каждой из рассматриваемых моделей. Это объясняется тем, что транспортный поток в выходные и рабочие дни существенно различается, в выходные дни транспортная ситуация стабильнее и предсказывается с большей точностью.

Для оценки масштабируемости и эффективности разработанной модели, в основе которой лежит модель распределённых вычислений MapReduce, был проведён ряд экспериментов. Вычисления проводились на кластере из шести компьютеров с CPU Core i5-3450, 8 Гб RAM. Для прогнозирования использовалась предложенная модель с учётом информации о транспортных потоках для кластеризации графа. Время работы модели усреднялось за 5 измерений.

Время работы предложенной модели (в секундах) для прогнозирования скорости транспортных потоков на всех сегментах дорожной сети для одного интервала времени в зависимости от числа вычислительных узлов показано в таблице 6.

Таблица 6 – Сравнение времени работы модели

Число узлов	1	2	3	4	5	6
Время, с	346	176	139	101	88	74

Учитывая, что период обновления данных о состоянии транспортных потоков составляет 10 минут, можно сделать вывод, что модель позволяет прогнозировать состояние транспортных потоков в режиме реального времени.

Далее исследовалась масштабируемость предложенной модели, которая показывает, как меняется производительность системы при пропорциональном увеличении объёма обрабатываемой информации и мощности системы.

$$Scaleup = \frac{T_s}{T_{pp}} \quad (30)$$

где T_s – время, затрачиваемое на решение заданной задачи на одном вычислительном узле;
 T_{pp} – время, затрачиваемое на решение задачи, размер входных данных которой увеличен в p раз, на p вычислительных узлах.

Чем ближе отношение (30) к единице, тем лучшей масштабируемостью обладает система. Результаты представлены на рисунке 25.

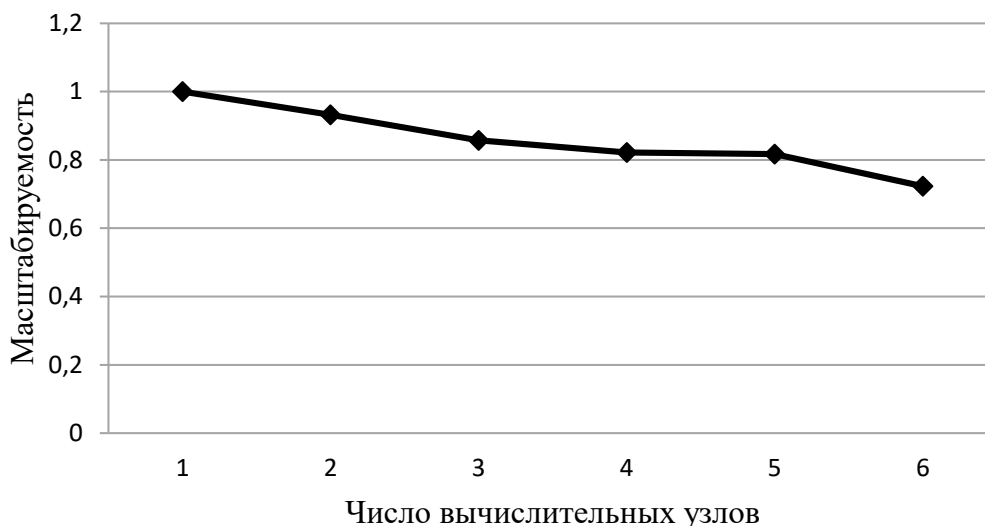


Рисунок 25 – Масштабируемость

На основе анализа полученных результатов можно сделать вывод, что предлагаемая модель прогнозирования, использующая модель распределённых вычислений MapReduce, обладает хорошей масштабируемостью, что позволяет эффективно использовать данную модель при обработке большого объёма данных.

2.3 Прогнозирование параметров транспортного потока с использованием графовых нейронных сетей

Предложенный в разделе 2.2 подход был модифицирован для повышения точности прогнозирования параметров транспортного потока в пространственно-компактной области с использованием методов машинного обучения, и, в частности, аппарата графовых сверточных нейронных сетей. Традиционные сверточные нейронные сети способны эффективно извлекать локальные паттерны, но могут применяться только для структур данных, представляющих собой регулярные сетки. Свертка на графах обобщает традиционную свертку и может использоваться для обработки данных на графовых структурах.

2.3.1 Методология решения задачи прогнозирования

2.3.1.1 Постановка задачи

Задача прогнозирования транспортного потока заключается в оценке значений параметров транспортного потока в определенное время (или интервал времени) в будущем

с использованием текущих и архивных значений параметров транспортного потока и топологии дорожной сети.

Введем следующие обозначения. Улично-дорожную сеть будем рассматривать в виде ориентированного графа $G = (V, E, W)$ с вершинами V и ребрами E , где число вершин $N = |V|$. В зависимости от данных каждая вершина может представлять перекресток дорог, дорожное соединение или дорожный детектор. $W \in R^{N \times N}$ обозначает взвешенную матрицу смежности графа G .

Пусть $x_t^i \in R$ обозначает наблюдаемый параметр транспортного потока $i \in V$ в момент времени t . Набор наблюдаемых параметров для всех вершин графа в данный момент t обозначим как:

$$\mathbf{X}_t = (x_t^1, x_t^2, \dots, x_t^N) \in \mathbb{R}^N. \quad (31)$$

Значения всех параметров транспортного потока за τ временных интервалов обозначим как

$$\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_\tau)^T \in \mathbb{R}^{N \times \tau}. \quad (32)$$

Прогнозируемое значение параметров транспортного потока в будущий момент времени t обозначим как

$$\hat{\mathbf{X}}_t = (\hat{x}_t^1, \hat{x}_t^2, \dots, \hat{x}_t^N) \in \mathbb{R}^N. \quad (33)$$

Задачу прогнозирования транспортного потока можно определить следующим образом: имея граф $G = (V, E, W)$ и последовательность \mathbf{X} наблюдаемых значений параметров транспортного потока, рассчитать оценку (спрогнозировать) параметров транспортного потока $\hat{\mathbf{X}}_t$ через T_{ph} временных интервалов:

$$\hat{\mathbf{X}}_{t+1}, \hat{\mathbf{X}}_{t+2}, \dots, \hat{\mathbf{X}}_{t+T_{ph}} = f(\mathbf{X}; G), \quad (34)$$

где t – текущий момент времени.

В следующих подразделах описаны основные концепции предлагаемой гибридной пространственно-временной сверточной нейронной сети на графе.

2.3.1.2 Свертка в пространственной области

Свертка в пространственной области позволяет обрабатывать локализованные пространственные зависимости, представленные в транспортных сетях. Традиционные сверточные операторы, обрабатывающие двумерные или трехмерные данные на основе регулярной сетки, не подходят для обработки графовых данных о состоянии транспортного потока. Для решения этой проблемы были разработаны обобщенные сверточные операторы, основанные на спектральной теории графов [86].

Общий подход к определению сверточного оператора для графовых данных основан на матрице смежности. Рассмотрим матрицу Лапласа графа:

$$L = D - W.$$

В нормализованном виде его можно определить следующим образом:

$$L = I - D^{-\frac{1}{2}} W D^{\frac{1}{2}},$$

где I – единичная матрица;

D – диагональная матрица, состоящая из узловых степеней:

$$D_{ii} = \sum_j W_{ij}$$

Разложение по собственным значениям матрицы Лапласа можно определить следующим образом:

$$L = U \Lambda U^T$$

где Λ – диагональная матрица;

U – Фурье-базис.

Используя эту нотацию, сверточный оператор для данных на основе графа определяется как результат умножения сигнала $x \in \mathbb{R}^N$ на графе с ядром g_θ [324]:

$$g_\theta * x = g_\theta(L)x = g_\theta(U \Lambda U^T)x = U g_\theta(\Lambda) U^T x.$$

Прямое вычисление операции свертки может быть вычислительно сложной задачей, особенно для крупномасштабных графов. Для локализации фильтра g_θ , уменьшения количества параметров и уменьшения времени вычислений было предложено обобщить свертку на графе с помощью полиномов Чебышева [324]:

$$g_{\theta} * x = g_{\theta}(\mathbf{L})x = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{\mathbf{L}})x,$$

где параметр θ_k – вектор полиномиальных коэффициентов,

$$\tilde{\mathbf{L}} = \frac{2}{\lambda_{max}} \mathbf{L} - I - \text{масштабированная матрица Лапласа};$$

λ_{max} – максимальное собственное значение матрицы Лапласа.

Полиномы Чебышева рекурсивно определяются следующим образом:

$$T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x),$$

$$T_0(x) = 1, T_1(x) = x.$$

2.3.1.3 Свертка во временной области

Чтобы объединить информацию из разных временных интервалов и учесть временные корреляции в параметрах транспортного потока, после свертки на графе в пространственной области применяется операция свертки во временной области. Свертка во временной области выполняется для данных на регулярной сетке с использованием стандартного сверточного оператора. Расчеты на r -м слое нейронной сети можно описать следующим образом:

$$x^r = ReLU\left(\Phi * (ReLU(g_{\theta} * x^{r-1}))\right),$$

где $*$ – сверточный оператор;

Φ – параметр сверточного ядра во временной размерности;

x – исходный входной временной ряд или результат вычислений на предыдущем слое нейронной сети;

$ReLU$ – активационная функция.

2.3.1.4 Long Short-Term Memory (LSTM)

Нейронная сеть с долгой краткосрочной памятью (LSTM) – это особый тип рекуррентной нейронной сети (RNN) [325] с возможностью передачи состояния ячейки с предыдущего временного шага на следующий. Нейронная сеть LSTM может использоваться для обработки долгосрочных зависимостей, особенно при обработке последовательных данных.

На рисунке 26 схематично представлена структура ячейки LSTM. В схеме x – входное значений, C – состояние ячейки, h – выходное значение, t – временной шаг.

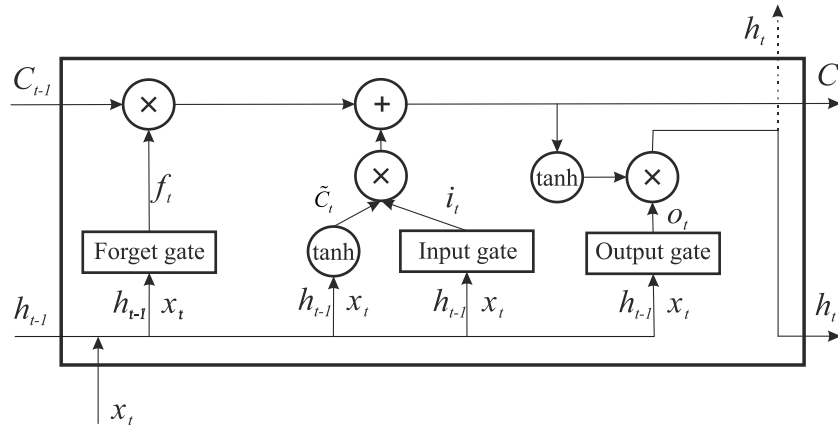


Рисунок 26 – Структура ячейки LSTM

В следующем разделе описывается предложенная модель.

2.3.2 Модель прогнозирования параметров транспортного потока

Предложенная модель краткосрочного прогнозирования параметров транспортного потока на основе графовой нейронной сети представляет собой гибридную модель глубокого обучения, которая объединяет слои пространственно-временной свертки на графах с операторами LSTM. Для использования модели далее будет представлено описание входного вектора признаков, который объединяет текущие, дневные и недельно-периодические компоненты в значениях параметров транспортного потока, а также описана архитектура предлагаемой модели.

2.3.2.1 Входной вектор признаков

Чтобы описать транспортный поток в дорожной сети, определим вектор признаков для всех вершин графа G . Согласно (31), набор наблюдаемых параметров транспортного потока для всех вершин графа в момент времени t обозначается как $X_t \in \mathbb{R}^N$. В данном разделе предлагается использовать вектор признаков, который учитывает дневные и недельно-периодические компоненты в данных о транспортном потоке.

Пусть наблюдаемые значения параметров транспортного потока поступают с частотой q записей в день. Обозначим текущий момент времени как t_0 . Горизонт прогноза (т.е. количество прогнозируемых записей) обозначим как T_{ph} . Чтобы определить вектор признаков, рассмотрим следующие временные ряды:

1) текущий временной ряд, учитывающий наблюдаемые значения параметров транспортного потока за последние T_h временных интервалов:

$$X^c = (X_{t_0-T_h+1}, X_{t_0-T_h+2}, \dots, X_{t_0});$$

2) временной ряд, учитывающий дневные паттерны. Транспортные потоки имеют дневную периодичность (например, утренние и вечерние часы пик). Для учета дневной периодичности будут рассматриваться временные ряды следующего вида:

$$\mathbf{X}_d^{day_past} = (\mathbf{X}_{t_0-d*q-T_h+1}, \mathbf{X}_{t_0-d*q-T_h+2}, \dots, \mathbf{X}_{t_0-d*q}), d = \overline{1, T_d},$$

$$\mathbf{X}_d^{day_sh} = (\mathbf{X}_{t_0-d*q-T_h+T_{ph}+1}, \mathbf{X}_{t_0-d*q-T_h+T_{ph}+2}, \dots, \mathbf{X}_{t_0-d*q+T_{ph}}), d = \overline{1, T_d},$$

где T_d – число дней;

3) временной ряд, учитывающий недельные паттерны. Часто распределение транспортного потока различается по дням недели (особенно в будние / выходные дни). Для учета недельной периодичности в данных будут рассматриваться временные ряды следующего вида:

$$\mathbf{X}_w^{week_past} = (\mathbf{X}_{t_0-7*w*q-T_h+1}, \mathbf{X}_{t_0-7*w*q-T_h+2}, \dots, \mathbf{X}_{t_0-7*w*q}), w = \overline{1, T_w},$$

$$\mathbf{X}_w^{week_sh} = (\mathbf{X}_{t_0-7*w*q-T_h+T_{ph}+1}, \mathbf{X}_{t_0-7*w*q-T_h+T_{ph}+2}, \dots, \mathbf{X}_{t_0-7*w*q+T_{ph}}), w = \overline{1, T_w},$$

где T_w – число недель.

Учитывая рассматриваемые временные ряды, вектор признаков имеет следующий вид:

$$\mathbf{X}^{fv} = (\mathbf{X}^c, \{\mathbf{X}_d^{day_past}\}_{d=1, T_d}, \{\mathbf{X}_d^{day_sh}\}_{d=1, T_d}, \{\mathbf{X}_w^{week_past}\}_{w=1, T_w}, \{\mathbf{X}_d^{week_sh}\}_{w=1, T_w}) \in \mathbb{R}^{N \times F},$$

где $F = 2T_d + 2T_w + 1$ – число временных рядов.

2.3.2.2 Архитектура нейронной сети

В работе для решения задачи краткосрочного прогнозирования параметров транспортного потока предлагается гибридная модель нейронной сети, которая объединяет операции пространственно-временной свертки на графе со слоями LSTM. Несколько блоков выполняются последовательно, чтобы извлечь большой диапазон пространственно-временных корреляций.

Архитектура нейронной сети представлена на рисунке 27.

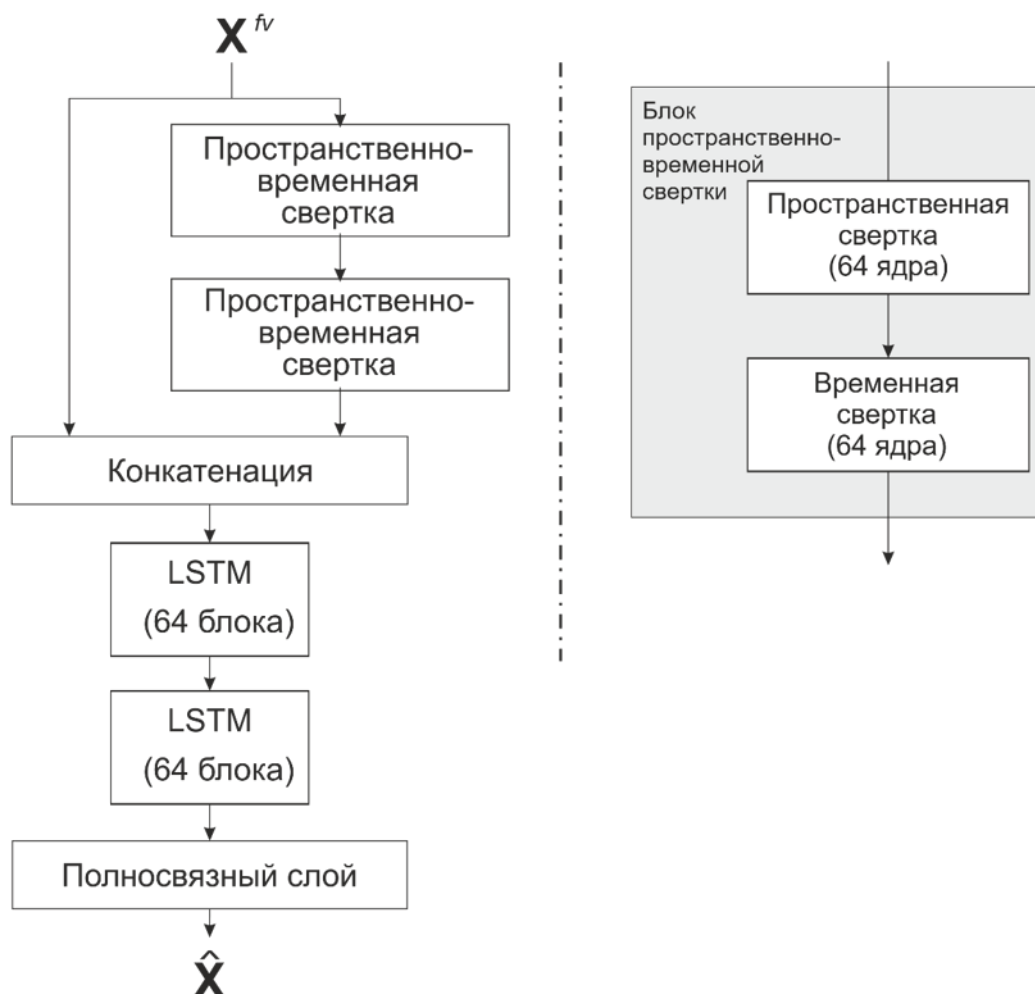


Рисунок 27 – Архитектура нейронной сети

Результат последовательных пространственно-временных сверток объединяется с входным вектором признаков для выполнения обработки LSTM слоем. Выходной слой – это полносвязный слой, который необходим для обеспечения требуемого размера выходных данных.

2.3.3 Экспериментальные исследования

Экспериментальное исследование предложенной модели было проведено на наборе данных, описывающих параметры транспортного потока в области транспортной сети г. Самары, включающей 1760 дорожных сегментов, а также на двух общедоступных наборах данных о дорожном движении PeMSD4 и PeMSD7.

В качестве исходных данных для проведения экспериментальных исследований на дорожной сети г. Самары использовались значения средней скорости движения транспортного потока (в км/ч), полученные за 50 дней, начиная с 1 ноября 2019 г. Средняя

скорость движения агрегирована за 10-минутный интервал времени. Пропуски в данных были заполнены с помощью линейной интерполяции.

Наборы данных PeMSD4 и PeMSD7 собираются системой измерения эффективности Caltrans (PeMS) [326]. Наборы данных агрегируются в 5-минутные интервалы из 30-секундных выборок данных, поэтому каждый детектор содержит 288 записей в день.

Набор данных PeMSD4 был предварительно обработан в [93]. Он содержит данные с 307 датчиков движения за 59 дней. Набор данных PeMSD7 использовался в [91]. Он содержит данные с 228 датчиков за 44 дня. Основные характеристики этих двух наборов данных представлены в таблице 7. Метод нормализации min-max использовался для масштабирования наблюдаемых значений транспортного потока до диапазона [0, 1].

Таблица 7 – Характеристики наборов данных PEMS7 и PEMS4

Набор данных	Скорость, км/ч			Число сенсоров	Число записей
	Мин	Среднее	Макс		
PeMSD7	3,0	58,9	82,6	228	12672
PeMSD4	3,0	63,47	85,2	307	16992

На первом этапе проводилось тестирование модели для области транспортной сети г. Самары. В качестве горизонта прогноза было выбрано $T = 6$ временных интервалов, т.е. прогнозирование транспортного потока проводилось на 1 час. Сравнение графовой нейронной сети было проведено с двумя базовыми моделями:

- 1) линейная регрессия;
- 2) полносвязная нейронная сеть с двумя скрытыми слоями, 24 нейрона в каждом слое (MLP).

Кроме того, было проведено сравнение моделей графовых нейронных сетей (GCNN) для различных значений степеней полиномов Чебышева $K \in \{1,2,3\}$.

В таблице 8 представлено сравнение результатов прогнозирования выбранными моделями при горизонте прогноза в 1 час.

Таблица 8 – Сравнение эффективности моделей

	MAE, км/ч	MAPE, %
Линейная регрессия	2,16	10,46
MLP	1,89	9,04
GCNN, K=1	1,7295	8,352
GCNN, K=2	1,7315	8,336
GCNN, K=3	1,821	8,773

Анализ результатов показывает, что предложенная графовая нейронная сеть дает более качественный результат прогноза, чем базовые модели прогнозирования. Лучший результат по совокупному критерию был показан моделью, использующей полиномы Чебышева второго порядка ($K = 2$). Кроме того, использование графовой нейронной сети позволяет повысить точность прогноза по сравнению с алгоритмами, представленными в разделах 2.1 и 2.2, однако графовая сеть требует большого объема данных для обучения. Кроме того, при изменении топологии дорожной сети может потребоваться переобучение модели.

Далее было проведено сравнение моделей для различных значений горизонта прогноза. Результаты сравнения по критериям средней абсолютной ошибки (MAE) и средней абсолютной ошибки в процентах (MAPE) показаны на рисунках 28 и 29 соответственно.

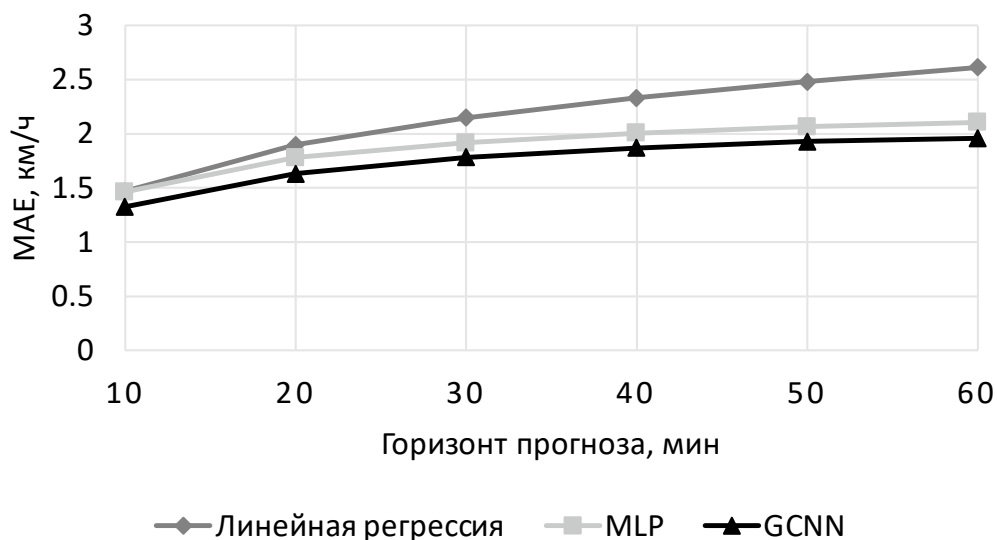


Рисунок 28 – Средняя абсолютная ошибка

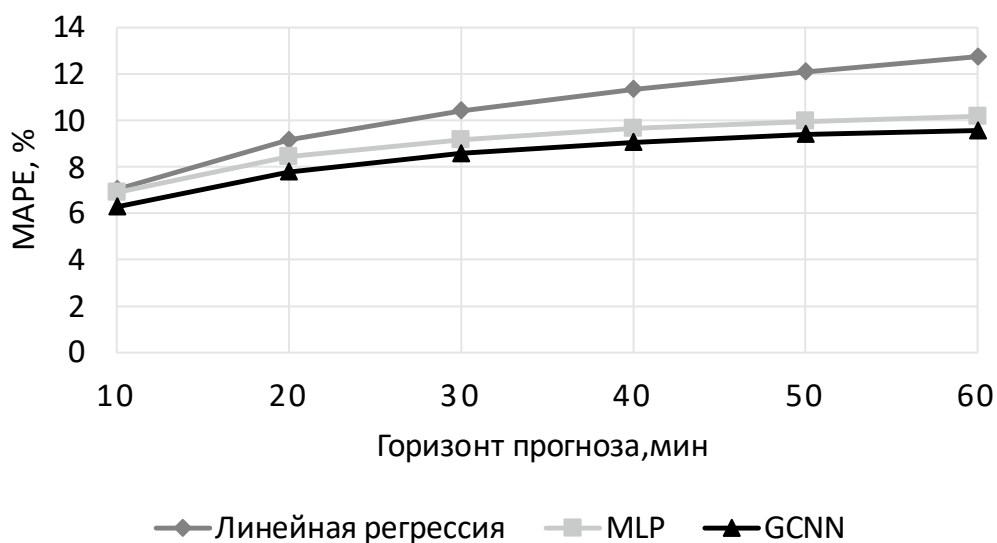


Рисунок 29 – Средняя абсолютная ошибка в процентах

Анализируя полученные результаты, можно сделать вывод, что предложенная модель показывает лучшие результаты для всего горизонта прогноза, при возрастании горизонта прогноза точность прогнозирования ухудшается.

На втором этапе экспериментального исследования была проведена оценка эффективности разработанной модели на наборах данных PeMSD4 и PeMSD7. Предложенная гибридная модель сравнивается со следующими современными (state-of-the-art) методами:

- 1) LSTM [325]: нейронная LSTM-сеть;
- 2) GRU: нейронная GRU-сеть;
- 3) STGCN [91]: нейронная сеть, использующая пространственно-временную свертку на графе с механизмом активации GRU;
- 4) ASTGCN [93]: нейронная сеть, использующая пространственно-временную свертку на графе на основе механизма внимания.

В качестве показателей оценки эффективности используемых моделей сравниваются средняя абсолютная ошибка (MAE) и средняя абсолютная ошибка в процентах (MAPE).

В таблице 9 показаны результаты сравнения эффективности предложенной гибридной модели (Hybrid) и базовых моделей на наборах данных PeMSD7 и PeMSD4 соответственно.

Таблица 9 – Сравнение эффективности моделей на наборах данных PEMS7 и PEMS4

Модель	PEMS7 (15 / 30 / 60 мин)		PEMS4 (15 / 30 / 60 мин)	
	MAE	MAPE	MAE	MAPE
GRU	2,384/3,124/3,868	5,559/7,563/9,67	1,516/1,911/2,336	3,069/4,117/5,293
LSTM	2,329/3,035/3,803	5,446/7,385/9,604	1,536/1,905/2,336	3,084/4,096/5,292
STGCN	2,258/3,065/4,061	5,293/7,483/10,293	1,452/1,923/2,565	2,899/4,108/5,652
ASTGCN	2,469/3,186/4,017	5,793/7,764/10,136	1,744/2,122/2,664	3,634/4,698/6,052
Hybrid	2,261/ 2,917/3,546	5,305/ 7,13/8,938	1,468/ 1,825/2,228	2,95 / 3,915/5,054

Видно, что предложенная модель показывает лучшие результаты по выбранным критериям практически для всех значений горизонта прогноза. С увеличением горизонта прогнозирования ошибка прогнозирования также увеличивается из-за накопления ошибок. При прогнозе на час вперед разница между предложенной моделью и другими базовыми моделями более значительна.

На рисунках 30 и 31 показана зависимость значений критериев для разных методов от горизонта прогноза (интервалов прогнозирования). На рисунке 30 показаны значения MAE на наборе данных PEMS7, на рисунке 31 показаны значения MAPE на наборе данных PEMS4.

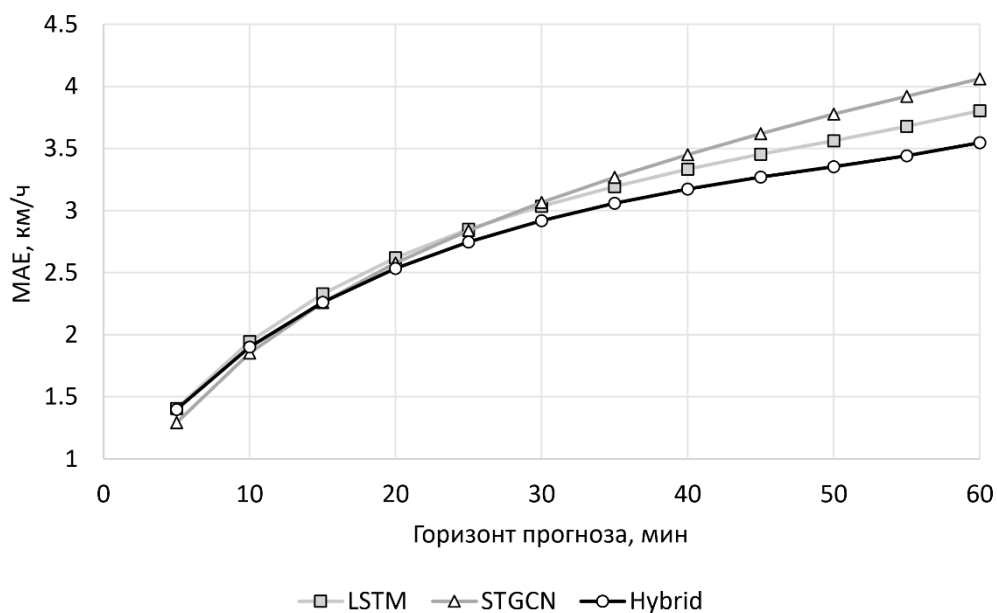


Рисунок 30 – Сравнение точности прогнозов по критериям MAE на наборе данных PEMS7

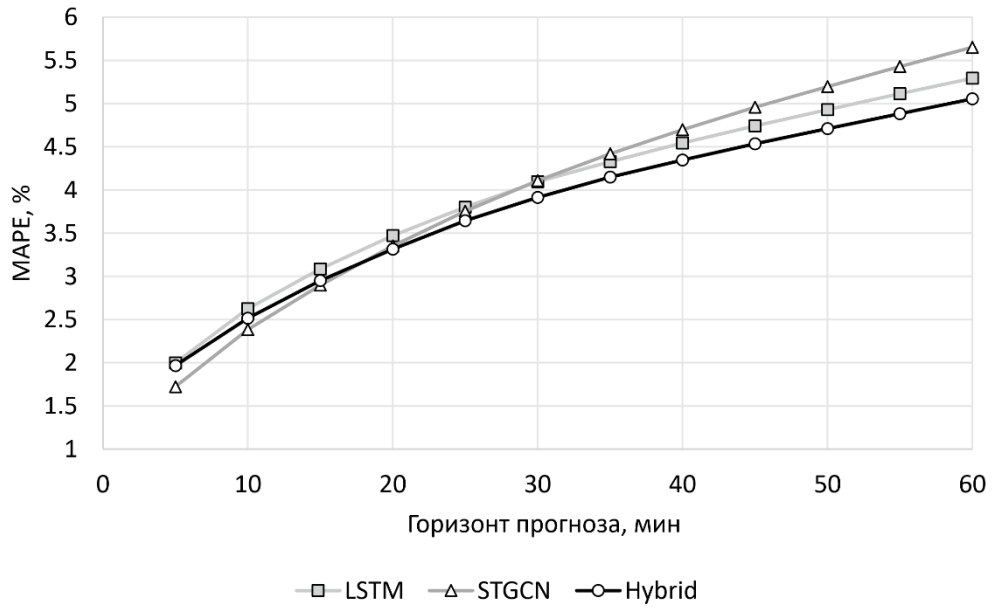


Рисунок 31 – Сравнение точности прогнозов по критериям MAPE на наборе данных PeMSD4

Чтобы оценить результаты модели, также было проведено сравнение результатов работы в утренние и вечерние часы пик для одного датчика на 30 и 60 минут вперед, как показано на рисунках 32 и 33.

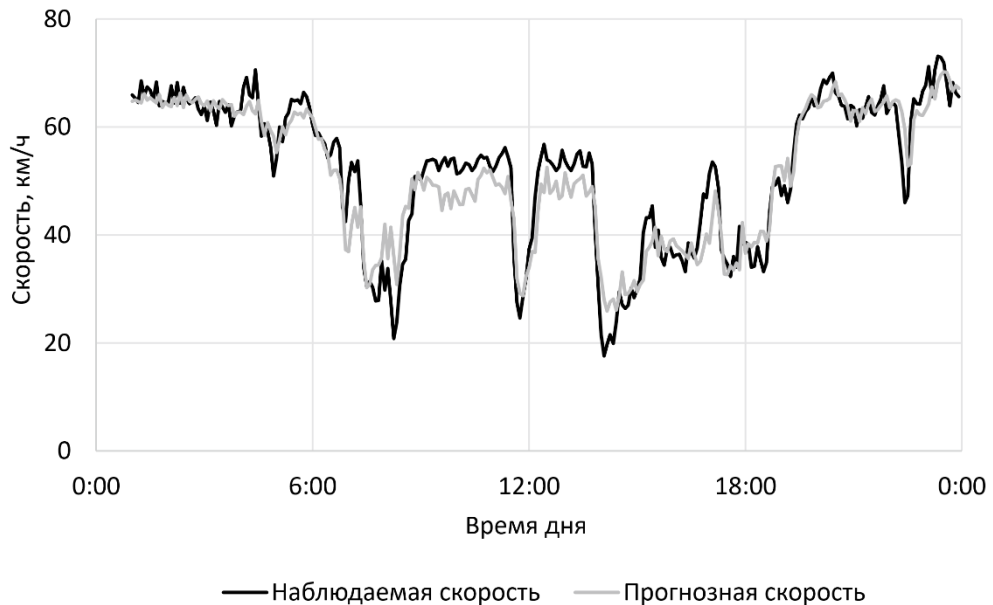


Рисунок 32 – Прогноз скорости в утренние и вечерние часы пик на 30 минут вперед

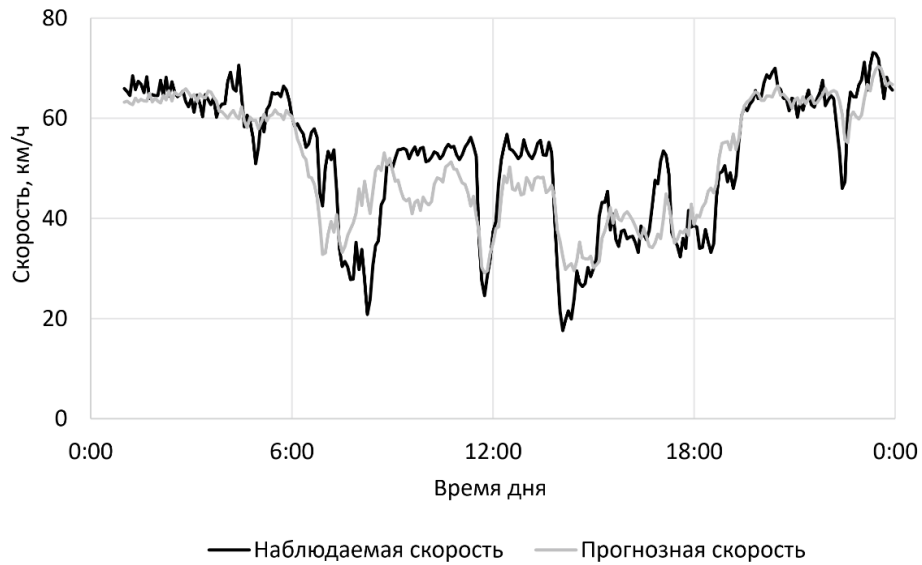


Рисунок 33 – Прогноз скорости в утренние и вечерние часы пик на 60 минут вперед

Учитывая полученные результаты, можно сделать вывод, что предлагаемая модель способна реагировать на динамические изменения наблюдаемых значений параметров потока, но эффективность модели снижается при увеличении горизонта прогнозирования.

На заключительной стадии экспериментального анализа оценивалось время работы моделей. Среднее время обучения за эпоху (в секундах) и время тестирования на одном пакете (прогноз транспортного потока на один час вперед для всех датчиков, в миллисекундах) показаны в таблице 10.

Таблица 10 – Расчетное время обучения и прогнозирования

Модель	PEMSD7		PEMSD4	
	Время обучения, с	Время прогнозирования, мс	Время обучения с	Время прогнозирования мс
LSTM	11,5	5,32	16,2	5,25
STGCN	116,08	33,05	440,59	43,08
ASTGCN	32,11	46,43	120,45	59,98
Hybrid	12,78	13,63	25,7	20,47

Все эксперименты проводились с использованием Linux-кластера: Intel Core i7 9700K, 64 ГБ RAM, GPU NVIDIA GeForce RTX 2080Ti. Следует отметить, хотя все модели были обучены и протестированы с использованием одно и того же Linux-кластера, они были реализованы с использованием разных фреймворков и библиотек. На основании этих экспериментов можно сделать вывод, что все модели, включая предложенную, могут использоваться для прогнозирования транспортного потока в режиме реального времени.

2.4 Прогнозирование параметров гетерогенного транспортного потока

В разделе 2.2 задача краткосрочного прогнозирования транспортного потока решалась с использованием подхода к обработке больших данных, для прогнозирования параметров потока в компактном подграфе графа использовался метод k ближайших соседей. В разделе 2.3 была предложена модифицированная модель на основе графовой нейронной сети для повышения точности прогнозирования. Рассмотренные алгоритмы использовали текущие и архивные статистические значения параметров транспортного потока. В то же время, использование данных от подключенных транспортных средств позволит повысить качество прогнозирования. Как следствие, в данном разделе представлено описание метода краткосрочного прогнозирования смешанного транспортного потока, состоящего из подключенных и управляемых водителями транспортных средств, на основе гетерогенной информации о состоянии транспортного потока и движении отдельных транспортных средств.

2.4.1 Методология решения задачи прогнозирования

2.4.1.1 Постановка задачи

Введем следующие обозначения. Дорожная сеть рассматривается как ориентированный граф $G = (V, E, W)$, с узлами V , представляющими перекрестки дорожной сети, и ребрами E , обозначающими участки дорожной сети. $W \in R^{N \times N}$ обозначает взвешенную матрицу смежности графа G . Каждое ребро $e \in E$ описывается кортежем $(v^{from}, v^{to}, \lambda, L)$, где $v^{from} \in V$ – это начальная вершина, $v^{to} \in V$ – конечная вершина, λ – номер полосы на соответствующем участке дороги, L – длина участка дороги.

Пусть $x_e(t)$ – наблюдаемая плотность транспортного потока на ребре $e \in E$ в момент времени t , усредненная за указанный интервал времени Δ . Набор наблюдаемых значений плотности транспортного потока для каждого ребра обозначим как X .

Обозначим множество подключенных транспортных средств как U . Для каждого транспортного средства $u_k \in U$ считается известной следующая информация:

$$I_k = (o_k, d_k, \tau_k, R_k),$$

где $o_k \in V$ – исходная вершина (вершина отправления);

$d_k \in V$ – конечная вершина (вершина прибытия);

τ_k – время начала движения;

R_k – маршрут транспортного средства;

$$R_k = \{e_0^k, e_1^k, \dots, e_{N^k-1}^k\},$$

где $v_0^{from,k} = o_k$;

$$v_{N-1}^{to,k} = d_k;$$

N^k – число ребер в k -м маршруте.

Задачу прогнозирования параметров транспортного потока можно определить следующим образом: имея граф дорожной сети $G = (V, E, W)$, наблюдаемые значения параметров транспортного потока X , информацию от подключенных транспортных средств I_k для каждого $u_k \in U$, рассчитать оценку (спрогнозировать) параметров транспортного потока \hat{X} за следующие T^{ph} интервалов (горизонт прогноза).

2.4.1.2 Прогнозирование гетерогенного транспортного потока

Предлагаемый метод краткосрочного прогнозирования гетерогенного транспортного потока состоит из двух основных частей: прогнозирование транспортного потока по текущим и архивным статистическим значениям параметров потока (без информации от подключенных транспортных средств) и оценка загрузки транспортной сети подключенными транспортными средствами. В форме псевдокода метод можно сформулировать, как описано далее (Алгоритме 1).

Алгоритм 1: Метод прогнозирования гетерогенного транспортного потока

```

1: Входные данные:  $\tau^{cur}, X^{hist}, \{n_e\}_{e \in E}$ 
2: Выходные данные:  $\hat{X}$ 
3:  $X^{cur} = \{\}$ ;
4: for  $e \in E$  do
5:    $x_e(\tau^{cur}) = ObserveDensity(\tau^{cur})$ ;
6:    $x_e^{cv}(\tau^{cur}) = \frac{n_e(\tau^{cur})}{\lambda_e L_e}$ ;
7:    $x_e^{flow}(\tau^{cur}) = x_e(\tau^{cur}) - x_e^{cv}(\tau^{cur})$ ;
8:    $X^{cur}.append(x_e^{flow}(\tau^{cur}))$ ;
9: end
10:  $X^{FV} = (X^{hist}, X^{cur})$ ;
11:  $\hat{X}^{flow} = PredictFlow(X^{FV})$ ;
12: for  $e \in E$  do
13:   for  $t = 0$  to  $T^{ph}$  do
14:      $\hat{x}_e^{cv}(t) = \frac{n_e(t)}{\lambda_e L_e}$ ;
15:      $\hat{X}.append(\hat{x}_e^{flow}(t) + \hat{x}_e^{cv}(t))$ ;
16:   end
17: end

```

В методе τ^{cur} – текущее время, X^{hist} – архивные значения параметров транспортного потока, $\{n_e\}_{e \in E}$ – накопленное количество подключенных транспортных средств, которые зарезервировали временной интервал на участке дороги $e \in E$ (зарезервированный трафик).

Метод состоит из следующих этапов:

1) строки 4-8: формирование вектора признаков, описывающего транспортную ситуацию на каждом сегменте. В частности, в строке 5 формируется наблюдаемая плотность транспортного потока, усредненная за указанный интервал времени. В строках 4-6 формируется вектор признаков, который содержит статистические значения транспортного потока (без учета данных от транспортных средств типа CV);

2) строка 10: объединение архивных и текущих наблюдений транспортного потока;

3) строка 11: прогнозирование транспортного потока без учета информации от транспортных средств типа CV, используя выбранный алгоритм прогнозирования. В работе используется алгоритм, основанный на графовой нейронной сети, описанный в разделе 2.4.1.3;

4) строки 12-15: формирование результирующего прогноза, который объединяет прогноз транспортного потока без учета данных от ТС типа CV и оценку загрузки транспортной сети ТС типа CV для каждого сегмента и каждого временного интервала.

Чтобы полностью определить метод, необходимо определить используемый алгоритм прогнозирования параметров транспортного потока без учета информации от ТС типа CV *PredictFlow* (алгоритм описан в разделе в разделе 2.4.1.3) и алгоритм оценки загрузки транспортной сети транспортными средствами типа CV, который используется для расчета зарезервированного трафика $\{n_e\}_{e \in E}$ (раздел 2.4.1.4).

2.4.1.3 Прогнозирование транспортного потока без подключенных транспортных средств

Для прогнозирования параметров транспортного потока без учета подключенных транспортных средств предлагается использовать подход, подробнее описанный в разделах 2.2 и 2.3. Кратко опишем шаги предложенного подхода:

1) территория города разделяется на пространственно компактные участки (возможно, перекрывающиеся). Этот шаг необходим для уменьшения размерности векторов признаков, описывающих транспортную ситуацию;

2) определяется текущая транспортная ситуация в пространственно компактном участке дорожной сети с помощью вектора признаков. В работе для описания транспортной ситуации предлагается использовать вектор признаков, который объединяет текущие,

суточно-периодические и еженедельно-периодические временные ряды наблюдаемых значений плотности транспортного потока;

3) прогнозируются параметры распределения транспортного потока с помощью нейронной сети с глубоким обучением;

4) полученные прогнозы транспортных потоков объединяются для заданной пространственной области.

В частности, на третьем этапе алгоритма прогнозирования предлагается использовать гибридную модель глубокого обучения, сочетающую операцию свертки на графах с рекуррентными слоями, предложенную в разделе 2.3. Архитектура гибридной модели глубокого обучения представлена на рисунке 27.

Следует отметить, что нет ограничений на используемый алгоритм прогнозирования, для краткосрочного прогнозирования параметров транспортного потока также могут использоваться и другие алгоритмы.

2.4.1.4 Предварительная маршрутизация транспортных средств

Вторым алгоритмом, используемым в методе краткосрочного прогнозирования гетерогенного транспортного потока, является алгоритм оценки транспортной загрузки, создаваемой подключенными транспортными средствами. В работе используется подход к резервированию маршрутов движения, исследованный в [100] в задаче упреждающей маршрутизации. Этот алгоритм можно описать следующим образом (Алгоритм 2):

Алгоритм 2: Алгоритм оценки транспортной нагрузки CV

- 1: **Входные данные:** $R_k, \hat{X}^{flow}, \{n_e\}_{e \in E}$
 - 2: **Выходные данные:** обновленный $\{n_e\}_{e \in E}$
 - 3: $\tau^{in} = \tau^{cur}$;
 - 4: **for** $e \in R_k$ **do**
 - 5: $\hat{x}_e^{cv}(\tau^{in}) = \frac{n_e(\tau^{in})}{\lambda_e L_e}$;
 - 6: $\hat{x}_e(\tau^{in}) = \hat{x}_e^{flow}(\tau^{in}) + \hat{x}_e^{cv}(\tau^{in})$;
 - 7: $c_e(\tau^{in}) = \frac{L_e}{CalcTravelSpeed(\hat{x}_e(\tau^{in}))}$;
 - 8: $\tau^{out} = \tau^{in} + c_e(\tau^{in})$;
 - 9: **for** $t = \lceil \tau^{in} \rceil$ **to** $\lceil \tau^{in} \rceil + 1$ **do**
 - 10: $n_e(t) = n_e(t) + 1$;
 - 11: **end**
 - 12: $\tau^{in} = \tau^{out}$;
 - 13: **end**
-

В алгоритме $c_e(\tau^{in})$ – это оценка времени движения, τ^{in} и τ^{out} – время въезда и выезда с сегмента дорожной сети соответственно, $[z]$ – целая часть числа z .

Алгоритм состоит из следующих шагов:

- 1) при необходимости подключенное транспортное средство удаляется из зарезервированного трафика (в блоке псевдокода шаг пропускается);
- 2) рассчитывается прогнозируемая плотность транспортного потока на сегменте;
- 3) на основе прогнозируемой плотности рассчитывается скорость движения (и время движения) с использованием выбранного соотношения скорость-плотность;
- 4) обновляется транспортная загрузка для каждого временного шага между временами въезда и выезда на сегмент дорожной сети.

Алгоритм запускается каждый раз, когда подключенное транспортное средство достигает перекрестка.

Чтобы вычислить скорость движения (функция *CalcTravelSpeed*), используется пяти-параметрическая логистическая модель скорости-плотности следующего вида:

$$s^{5PL}(p) = s_b + \frac{s_f - s_b}{(1 + \exp((p - p_b)/\theta_1))^{\theta_2}},$$

где p – плотность транспортного потока;

s_f – средняя скорость прохождения дорожного сегмента в свободном потоке;

s_b – средняя скорость прохождения дорожного сегмента в условиях дорожного затора;

p_b – значение плотности потока, соответствующей переходу из свободного движения к затрудненному движению;

θ_1, θ_2 – параметры, определяющие форму кривой.

Следует также отметить, что в качестве алгоритма оценки загрузки транспортной сети могут использоваться и другие алгоритмы.

В следующем разделе представляется экспериментальное исследование предложенного метода краткосрочного прогнозирования гетерогенного транспортного потока.

2.4.2 Экспериментальные исследования

Чтобы оценить эффективность предложенного метода, использовался пакет микроскопического моделирования транспортного потока с открытым исходным кодом SUMO [327], поскольку для проведения экспериментов необходимо использовать как усредненные наблюдаемые значения плотности транспортного потока, так и траектории

движения транспортных средств типа CV. Эксперименты проводились в нерегулируемой мелкомасштабной сети Sioux Falls [328]. Сеть состоит из 334 сегментов, среднее количество поездов 16570.

В экспериментах прогнозируется плотность транспортного потока на 5 минут вперед (горизонт прогноза $T^{ph} = 5$ минут). Усредненные значения плотности трафика были агрегированы по временным интервалам в 1 минуту. В качестве показателей оценки используется критерий средней абсолютной ошибки (MAE). Общие результаты были усреднены по 20 запускам моделирования.

На рисунке 34 показана зависимость значений MAE для различных значений степени проникновения (доли) подключенных транспортных средств. Степень проникновения обозначает, какая часть транспортных средств в сценарии считается подключенной (с доступной информацией I_k).

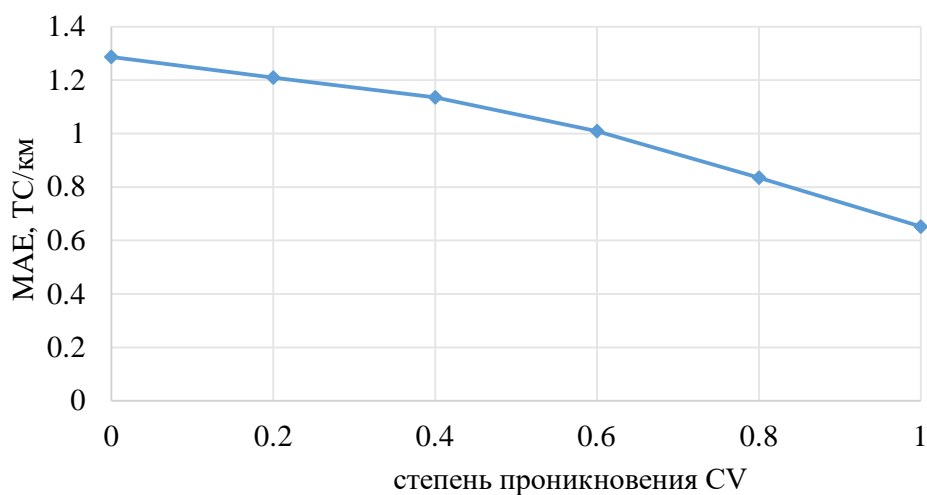


Рисунок 34 – Значение показателя MAE

Как видно, ошибка уменьшается с увеличением доступной информации от подключенных транспортных средств.

Далее рассмотрим более подробно зависимость значений MAE от горизонта прогноза (рисунок 35) для различных значений степени проникновения транспортных средств типа CV.

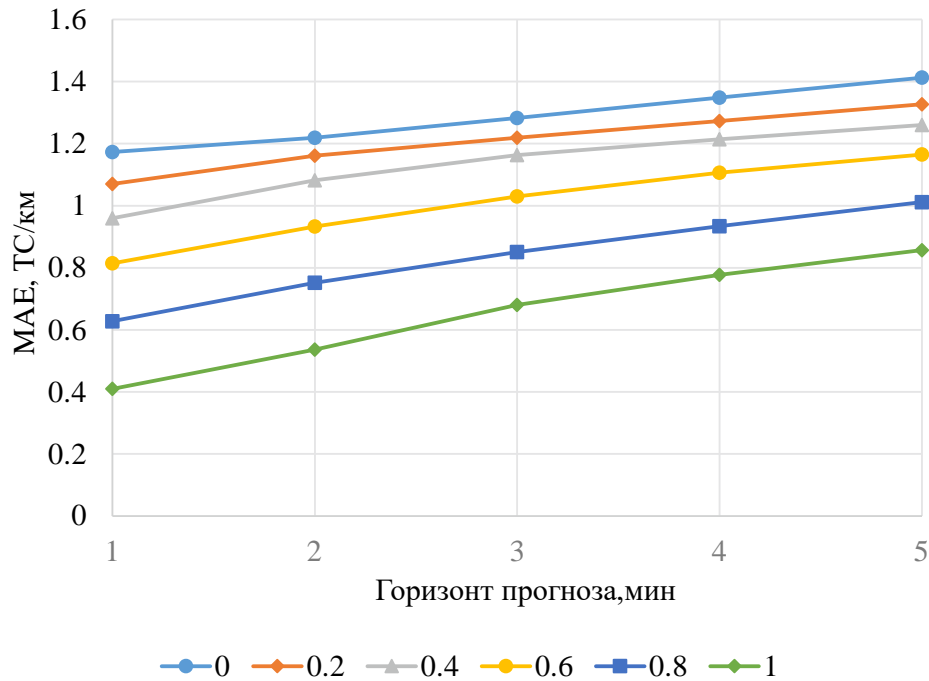


Рисунок 35 – Значение показателя MAE для различных значений степени проникновения CV

Хотя ошибки меньше для больших значений степени проникновения, дельта ошибок уменьшается с увеличением значений горизонта прогнозирования.

2.5 Выводы и результаты второго раздела

В рамках решения задачи анализа и прогнозирования транспортного потока в интеллектуальной транспортной системе были получены следующие результаты:

1) предложен комплекс алгоритмических средств (математический метод и алгоритмы) решения задачи прогнозирования транспортного потока в различных условиях:

а) большой объем данных и возможность использования инфраструктуры обработки больших данных: алгоритм прогнозирования параметров транспортного потока, основанный на непараметрическом регрессионном методе k ближайших соседей и подходе к параллельной обработке «больших» данных MapReduce в распределенном вычислительном кластере (п. 2.2);

б) большой объем данных для обучения нейронных сетей, неизменность топологии УДС: алгоритм прогнозирования параметров транспортного потока, использующий модель сверточной графовой нейронной сети, которая объединяет слои пространственно-временной свертки на графах с операторами LSTM для прогнозирования транспортного потока в компактном подграфе транспортной сети (п. 2.3);

в) гетерогенный транспортный поток: метод прогнозирования гетерогенного транспортного потока, содержащего подключенные и управляемые водителем транспортные средства, объединяющий прогнозирование транспортного потока по текущим и архивным статистическим значениям параметров потока и оценку загрузки транспортной сети подключенными транспортными средствами (п. 2.4);

2) проведено экспериментальное исследование разработанных алгоритмов прогнозирования транспортного потока на созданном и открытых наборах данных, подтвердившее эффективность разработанных алгоритмов (пп. 2.1.4, 2.2.5, 2.3.3). Разработанный алгоритм на основе модели графовой нейронной сети позволил снизить ошибку прогнозирования транспортного потока до 6% по сравнению с современными (state-of-the-art) методами решения задачи прогнозирования на открытых наборах данных;

3) проведено экспериментальное исследование разработанного метода прогнозирования гетерогенного транспортного потока в системе микроскопического моделирования движения транспортных средств, подтвердившее эффективность разработанного метода (п. 2.4.2);

4) предложенный метод и алгоритмы позволяют повысить точность прогнозирования параметров транспортного потока по сравнению со state-of-the-art методами, эффективность решения задачи прогнозирования увеличивается с увеличением доли подключенных транспортных средств в транспортном потоке.

По материалам данного раздела опубликованы работы [63, 64, 71, 96, 329-336].

3 Алгоритмические средства решения задачи директивного управления транспортным потоком в ИТС

Настоящий раздел посвящен решению задачи директивного управления транспортным потоком в ИТС. Задача оптимизация существующей транспортной инфраструктуры с целью повышения эффективности ее использования является ключевой задачей, решаемой в ИТС. В данном разделе рассматриваются методы и алгоритмы управления дорожным движением в транспортных сетях с гетерогенным составом транспортных средств. Управление дорожным движением выполняется посредством светофорного регулирования транспортного потока.

Наибольшие перспективы для решения задачи управления сигналами светофорных объектов имеет использование данных от подключенных и/или автономных транспортных средств. Преимущества внедрения подключенных автономных транспортных средств включают в себя снижение / устранение аварий, сокращение времени движения, повышение эффективности транспортной инфраструктуры и другие. Сосредоточившись на управлении городским движением, можно повысить эффективность решения задачи управления дорожным движением:

- данные от подключенных и автономных подключенных транспортных средств могут использоваться для более точной оценки характеристик транспортного потока и управления транспортным потоком для синхронизации сигналов светофорных объектов;

- системы управления движением могут составить более точные планы управления сигналами светофоров, поскольку время подъезда автомобилей можно спрогнозировать заблаговременно и точнее;

- совместное управление сигналами светофорных объектов и движением подключенных и/или автономных подключенных транспортных средств позволит уменьшить общую загруженность дорожной сети, потребление топлива и выбросы продуктов сгорания, особенно в городских районах.

Для решения задачи управления в разделе 3.1 предварительно вводятся основные определения. В разделе 3.2 задача адаптивного управления сигналами светофорных объектов рассматривается как проблема обучения с подкреплением, в качестве факторов, описывающих пространство состояний, рассматриваются как наблюдаемые, так и прогнозные параметры, описывающие состояние транспортного потока. В качестве алгоритма обучения с подкреплением используется подход двойного Q-обучения, основанный на оптимизации функции полезности. В разделе 3.3 описан разработанный в

рамках данной работы метод адаптивного управления сигналами светофорных объектов, основанный на максимизации взвешенного потока транспортных средств, проходящих через перекресток за выбранный интервал времени (интервал переключения фаз). В частности, в разделе 3.3.1 рассматривается детерминированная модель прогнозирования времени движения транспортных средств типа CV, в разделе 3.3.2 для решения этой подзадачи предложено использовать глубокую нейронную сеть. Предложенный метод обобщается для случая гетерогенного транспортного потока (включающего подключенные и управляемые водителями транспортные средства) в разделе 3.3.3. Совместное управление сигналами светофорных объектов и движением подключенных и/или автономных подключенных транспортных средств рассматривается в разделе 3.4. Экспериментальное исследование разработанных методов и алгоритмов и сравнение эффективности полученных решений с другими методами приводится в последнем подразделе каждого раздела.

3.1 Основные определения

Предварительно будут представлены некоторые определения, часть из которых известна из правил дорожного движения [10].

Под перекрёстком понимается «место пересечения, примыкания или разветвления дорог на одном уровне, ограниченное воображаемыми линиями, соединяющими соответственно противоположные, наиболее удалённые от центра перекрёстка начала закруглений проезжих частей». Под проезжей частью дороги понимается «элемент дороги, предназначенный для движения безрельсовых транспортных средств». Наконец, под полосой движения понимается «любая из продольных полос проезжей части, обозначенная или не обозначенная разметкой и имеющая ширину, достаточную для движения автомобилей в один ряд». На рисунке 36 показан перекрёсток как пересечение двух дорог с разделёнными проезжими частями, каждая из которых имеет по три полосы движения.

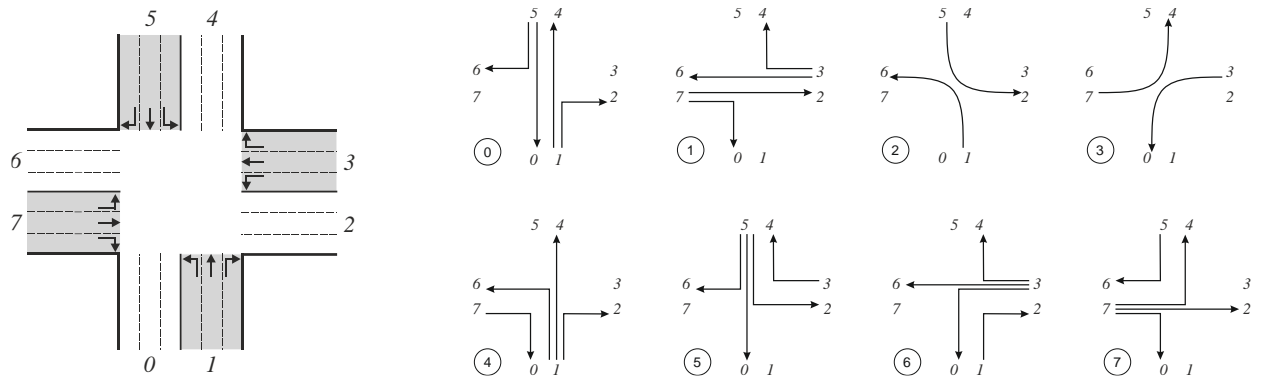


Рисунок 36 – Пример перекрёстка двух дорог: 8 проезжих частей, каждая из которых имеет по 3 полосы движения

Движение транспорта по перекрёстку осуществляется в соответствии с разметкой, которая определяет допустимые направления движения, и сигналами светофорных объектов, которые определяют возможность перемещения по соответствующему направлению в текущий момент времени. Например, обозначение движения парой «16» означает поворот налево с проезжей части, помеченной цифрой «1» на проезжую часть, помеченную цифрой «6». Данное движение, с учётом присутствующей разметки, допустимо только с левой полосы первой проезжей части.

Под фазой светофора будем понимать множество непротиворечивых сигналов светофора на перекрёстке, то есть таких сигналов, которые не могут приводить к аварийной ситуации. Например, непротиворечивыми являются сигналы следующего множества $\{56, 50, 14, 12\}$, а противоречивыми – $\{14, 72\}$. Очевидно, что множество P возможных фаз светофоров на конкретном перекрёстке зависит от числа дорог, проезжих частей, полос и структуры перекрёстка. В частности, для перекрёстка на рисунке 21 можно определить 8 фаз:

$$P = \{ \{56, 50, 14, 12\}, \{34, 36, 71, 72\}, \{16, 52\}, \{30, 74\}, \{12, 14, 16, 70\}, \\ \{56, 50, 52, 34\}, \{34, 36, 30, 12\}, \{70, 72, 74, 56\} \}.$$

Фазы светофорного регулирования могут выбираться как произвольно, так и в рамках некоторой предопределённой последовательности фаз – светофорного цикла. Одной из наиболее простых систем управления – контроллеров – является широко распространённый способ Uniform, подразумевающий переключение фаз светофора через заранее фиксированные промежутки времени из детерминированного светофорного цикла. Если выбор фазы светофорного регулирования происходит на основании данных движения транспортных средств, то говорят о системе адаптивного управления дорожными сигналами. При этом следует различать системы, где светофорный цикл предопределён и

где выбор следующей фазы производится без ограничений. В работе рассматривается адаптивное управление сигналами светофоров, при котором ограничений на выбор следующей фазы светофорного объекта не накладывается.

3.2 Управление сигналами светофоров на основе обучения с подкреплением

В данном разделе задача адаптивного управления сигналами светофорных объектов формулируется и решается как задача обучения с подкреплением.

3.2.1 Постановка задачи

Чтобы определить задачу управления сигналами светофора как проблему обучения с подкреплением, кратко опишем принципы RL-методов в типичной нотации. Задача обучения с подкреплением обычно представляется как Марковский процесс принятия решений, который может быть определен кортежем $\langle S, A, P_a, R_a \rangle$, где:

- 1) S – пространство состояний;
- 2) A – пространство действий;
- 3) $P_a(s, s') = Pr(s_{t+1} = s' | s_t = s, a_t = a)$ – вероятность перехода из состояния s в состояние s' под действием a в момент времени t ;
- 4) $r_t \equiv R_a(s, s')$ – немедленная награда после перехода из состояния s в состояние s' под действием a .

Целевая функция RL-задачи может быть формально определена следующим образом:

$$R^i = \sum_{t=0}^T \gamma^t r_t^i, \quad (35)$$

где T – общее число временных шагов;

$\gamma \in [0,1]$ – коэффициент дисконтирования.

Цель задачи обучения с подкреплением состоит в том, чтобы изучить политику $\pi^i: A \times S \rightarrow [0,1]$, $\pi(a, s) = Pr(a_t = a | s_t = s)$ для каждого агента i , что аппроксимирует распределение вероятностей действия a_t в данный момент времени t и состоянии s_t . Типичное решение этой проблемы состоит в том, чтобы найти Q-функцию, которая определяется как математическое ожидание целевой функции (35):

$$Q^\pi(s, a) = E\{R | s_t = s, a_t = a, \pi\}.$$

Для применения RL-методов для решения задачи адаптивного управления сигналами светофоров необходимо указать используемый RL-метод, структуру глубокой нейронной сети (ГНС), используемую для аппроксимации Q-функции, пространство состояний системы S и награду R . В данном разделе используется подход двойного Q-обучения в качестве RL-метода для решения задачи управления. Этот подход более подробно описан в разделе 3.2.2.2.

В работе рассматривается транспортная сеть, в которой каждый перекресток контролируется независимым RL-агентом, который выбирает действие на основе наблюдаемого состояния на перекрестке.

Предполагается, что каждый агент i на временном шаге t наблюдает текущее состояние системы $s^t \in S$. Для описания дорожной ситуации в зоне перекрестка (т.е. пространства состояний системы S), используются следующие факторы:

- 1) количество движущихся транспортных средств на каждой входящей полосе;
- 2) длина очереди на каждой входящей полосе;
- 3) общее время ожидания (время ожидания транспортного средства – это последовательное время, в течение которого транспортное средство находится без движения);
- 4) суммарная скорость транспортных средств на каждой встречной полосе;
- 5) прогнозируемое количество транспортных средств, которые пересекут перекресток в течение заданного интервала времени Δt .

Факторы (1)–(4), определяющие пространство состояний системы, рассчитываются на основе наблюдаемой информации о состоянии транспортного потока. Для прогнозирования количества транспортных средств, пересекающих перекресток (фактор (5)), предлагается использовать модель ГНС, описанную в разделе 3.2.2.1.

После того, как агент получил текущее состояние системы, он выбирает действие $a_t^i \in A$ для следующего временного интервала Δt . Выбранный набор действий a^t всех агентов отправляется в систему, которая переходит в новое состояние $s_{t+1} \in S$ в соответствии с вероятностью перехода. На финальном этапе определяется награда $R_{a_t}(s_t s_{t+1})$.

Оптимальное управление сигналами светофора направлено на минимизацию общего времени движения, затрачиваемого всеми транспортными средствами в транспортной сети для завершения своей поездки. Однако время движения транспортных средств можно рассчитать только при завершении поездок, поэтому метрику времени движения нельзя использовать для расчета мгновенных вознаграждений после перехода в состояние s_{t+1} . В

данной работе вознаграждение для агента рассчитывается как взвешенная линейная комбинация нескольких факторов, косвенно описывающих состояние транспортного потока [337]:

$$r_t^i = \alpha_0 \sum_{l \in L^i} q_t^l + \alpha_1 \sum_{l \in L^i} v_t^l + \alpha_2 p^i, \quad (36)$$

где $\alpha_j, j = \overline{0,2}$ – весовые коэффициенты;

L^i – множество входящих полос на перекрестке i ;

q_t^l – очередь транспортных средств на полосе l в момент времени t ;

v_t^l – средняя скорость движения всех транспортных средств на полосе l в момент времени t ;

p^i – «давление» [120], то есть разница между количеством въезжающих и выезжающих транспортных средств на перекрестке i .

В следующем разделе описывается методология предлагаемого решения.

3.2.2 Методология решения задачи адаптивного управления на основе обучения с подкреплением

В этом разделе описывается основанный на обучении с подкреплением подход к решению задачи адаптивного управления сигналами светофоров.

3.2.2.1 Прогнозирование динамики движения

Одним из факторов, описывающих рассматриваемое в данной работе пространство состояний системы S , является количество транспортных средств, которые пересекают перекресток за заданный интервал времени Δt .

Для прогнозирования количества транспортных средств предлагается использовать подход на основе ГНС. Входными данными нейронной сети являются распределение транспортных средств по входящей полосе и каждой связанной с ней исходящей полосе. А именно, рассматривается часть дорожной полосы в 200 метров, которая разбивается на отрезки по одному метру. Входной вектор признаков для каждого сегмента дороги содержит следующие данные:

- 1) 1, если транспортное средство находится в сегменте, 0 в противном случае;
- 2) скорость транспортного средства или 0;
- 3) ускорение транспортного средства или 0.

В зависимости от структуры перекрестка и разрешенного движения каждая входящая полоса дороги может быть соединена с четырьмя исходящими полосами

(движение прямо, налево, направо и разворот). В результате были сформированы и обучены четыре модели ГНС. Пример архитектуры ГНС с одной входящей полосой и тремя исходящими полосами представлен на рисунке 37.

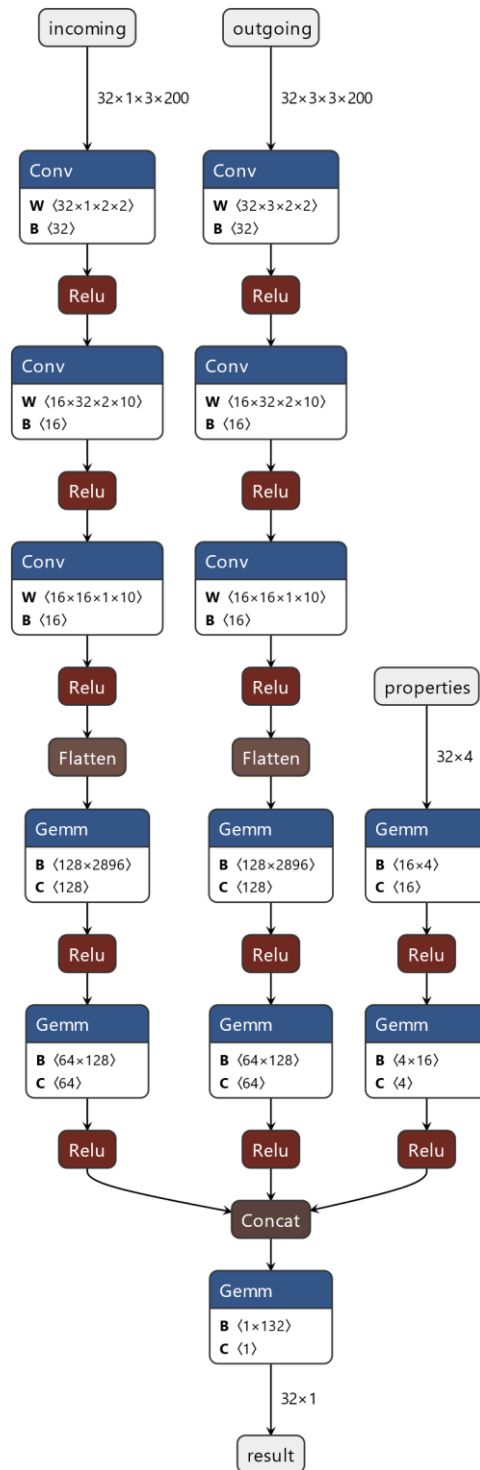


Рисунок 37 – Архитектура модели ГНС для прогнозирования движения транспортных средств

На рисунке 37 «incoming» – это матрица признаков, характеризующая входящую полосу, «outgoing» – набор матриц признаков для каждой исходящей полосы, «properties» – это вектор, содержащий информацию о длине дорожных полосы, N – размер пакета для обучения. Предлагаемая архитектура модели ГНС состоит из нескольких сверточных нейронных слоев (Conv) и полносвязных слоев (Linear).

Модель прогнозирует количество транспортных средств, которые пересекают перекресток в течение заданного интервала времени. Это значение является одним из факторов, описывающих пространство состояний системы.

3.2.2.2 Подход на основе двойного Q-обучения

В качестве алгоритма обучения с подкреплением используется подход двойного Q-обучения, который показывает лучшую производительность в зашумленной среде.

Рассмотрим функцию полезности (функцию «действие-значение», Q-функцию) пары (s, a) в соответствии с политикой π :

$$Q^\pi(s, a) = E\{R|s, a, \pi\}. \quad (37)$$

Одним из возможных решений нахождения оптимальной политики π^* является нахождение оптимальной Q-функции:

$$Q^*(s, a) = \max_{\pi} Q^\pi(s, a). \quad (38)$$

В Q-обучении используется итеративная процедура:

$$Q^{new}(s_t, a_t) \leftarrow (1 - \alpha)Q(s_t, a_t) + \alpha \left(r_t + \gamma \max_a Q(s_t, a) \right), \quad (39)$$

где $\alpha \in (0, 1]$ – скорость обучения.

В подходе двойного Q-обучения [338] две Q-функции Q^A , Q^B используются в качестве двойной оценки. Такой подход позволяет найти оптимальную политику в средах, в которых Q-обучение работает плохо из-за его переобучения.

$$\begin{aligned} Q_{t+1}^A(s_t, a_t) &= (1 - \alpha)Q_t^A(s_t, a_t) + \alpha \left(r_t + \gamma Q_t^B \left(s_{t+1}, \arg \max_a Q_t^A(s_{t+1}, a) \right) \right), \\ Q_{t+1}^B(s_t, a_t) &= (1 - \alpha)Q_t^B(s_t, a_t) + \alpha \left(r_t + \gamma Q_t^A \left(s_{t+1}, \arg \max_a Q_t^B(s_{t+1}, a) \right) \right). \end{aligned} \quad (40)$$

В качестве аппроксиматоров Q-функции используются две нейронные сети с одинаковой архитектурой. Пример архитектуры нейронной сети для перекрестка с десятью полосами движения показан на рисунке 38.

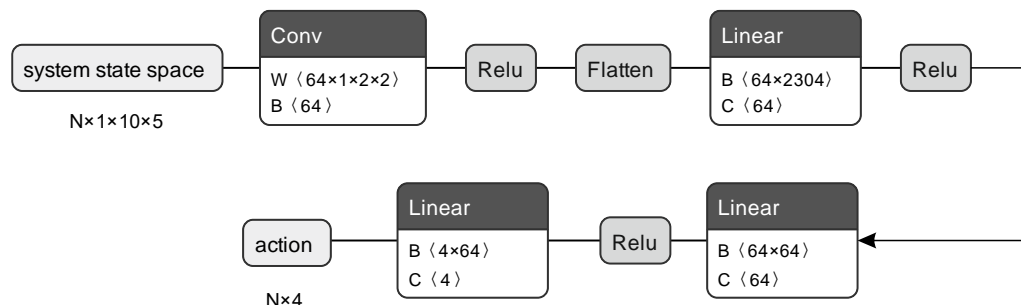


Рисунок 38 – Архитектура нейронной сети с двойным Q-обучением

На рисунке N – размер пакета. Выходным значением сети («action») является вектор действия для перекрестка.

Таким образом, предлагаемый гибридный подход состоит из следующих шагов:

1) каждый RL-агент наблюдает текущее состояние транспортного потока и прогнозирует количество транспортных средств, которые проедут перекресток в течение заданного интервала времени, как описано в разделе 3.2.2.1, формирует вектор признаков, описывающий дорожную ситуацию, используя факторы (1)–(5);

2) каждый RL-агент выбирает действие на следующий временной интервал, используя ГНС с двойным Q-обучением на основе созданного вектора признаков;

3) выбранный набор действий всех агентов отправляется в систему, которая переходит в новое состояние.

Экспериментальное исследование предложенного подхода к адаптивному управлению сигналами светофорных объектов на основе обучения с подкреплением описано в следующем разделе.

3.2.3 Экспериментальное исследование алгоритма адаптивного управления на основе обучения с подкреплением

Экспериментальное исследование алгоритма адаптивного управления на основе обучения с подкреплением было выполнено с использованием пакета микроскопического моделирования движения транспортных средств с открытым исходным кодом SUMO [327]. Эффективность предлагаемого подхода оценивалась как на синтетических, так и на реальных транспортных сценариях [339]:

1) «Grid 4x4», синтетический сценарий движения, дорожная сеть в котором представлена в виде сетки с 16 перекрестками. Каждая дорога состоит из трех полос.

2) «Cologne, congested area», часть хорошо зарекомендовавшего себя сценария SUMO «TAPAS Cologne» [340], который представляет загруженную часть города с восемью светофорами.

Дорожные сети рассматриваемых сценариев представлены на рисунке 39.

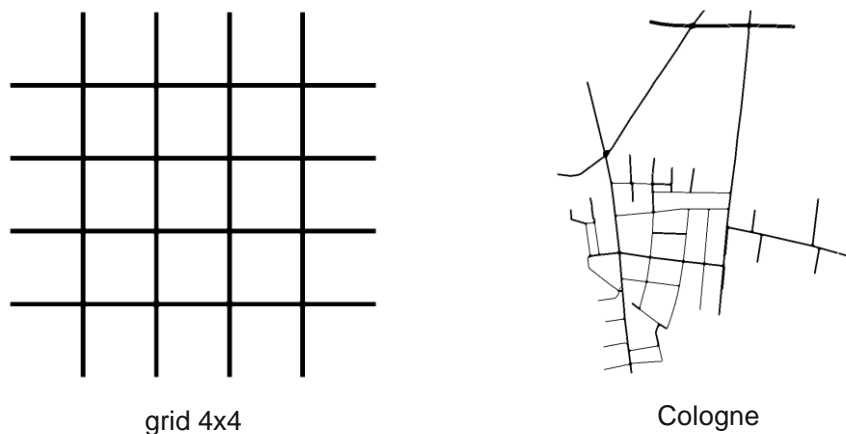


Рисунок 39 – Дорожные сети сценариев моделирования

Каждый сценарий моделирования определяется дорожной сетью и множеством поездок, совершаемых транспортными средствами. Каждая поездка определяется пунктом отправления, пунктом назначения и временем начала поездки. Исходное положение транспортного средства на начальном дорожном сегменте и время начала поездки могут варьироваться для каждого эпизода моделирования. Это позволяет генерировать эпизоды моделирования с различными параметрами транспортного потока.

Модель ГНС для прогнозирования движения транспортных средств (раздел 3.2.2.1) и модель ГНС с двойным Q-обучением (раздел 3.2.2.2) были обучены независимо на 1000 и 500 эпизодах соответственно. В течение одного эпизода выполняется моделирование всех поездок в сценарии.

Чтобы оценить эффективность предложенного гибридного подхода в задаче управления сигналами светофора и сравнить экспериментальные результаты с другими базовыми алгоритмами, обученные модели запускались на нескольких эпизодах. Предложенная модель и базовые алгоритмы были протестированы на идентичных данных (начальные местоположения транспортных средств и время начала поездки были одинаковыми для всех моделей в одном эпизоде).

Для сравнения были выбраны классические алгоритмы и алгоритмы на основе RL в качестве базовых алгоритмов:

1) алгоритм MaxPressure [120];

2) алгоритм глубокого Q-обучения (IDQN) [341]: каждый перекресток контролируется независимым RL-агентом. В качестве пространства наблюдения авторы рассматривали текущую фазу светофора, длину очереди, количество автомобилей и общую скорость движения;

3) MPLight [144], модель на основе ГНС, которая управляет всеми перекрестками.

Эффективность алгоритмов сравнивалась с использованием двух показателей: среднее время ожидания (в секундах) и среднее время движения (в секундах). Критерий среднего времени ожидания показывает среднее время простоя транспортных средств на перекрестках. Критерий среднего времени движения показывает среднее время, затрачиваемое транспортными средствами на совершение поездки.

На первом этапе экспериментального анализа была выполнена оценка сходимости предложенной модели. На рисунке 40 показана кривая обучения (со стандартным отклонением, рассчитанным для пяти эпизодов) для сценария моделирования «Cologne, congested area». Согласно рисунку 40, исследуемый RL-алгоритм был успешно обучен, в результате чего он достиг стабильного значения времени ожидания.

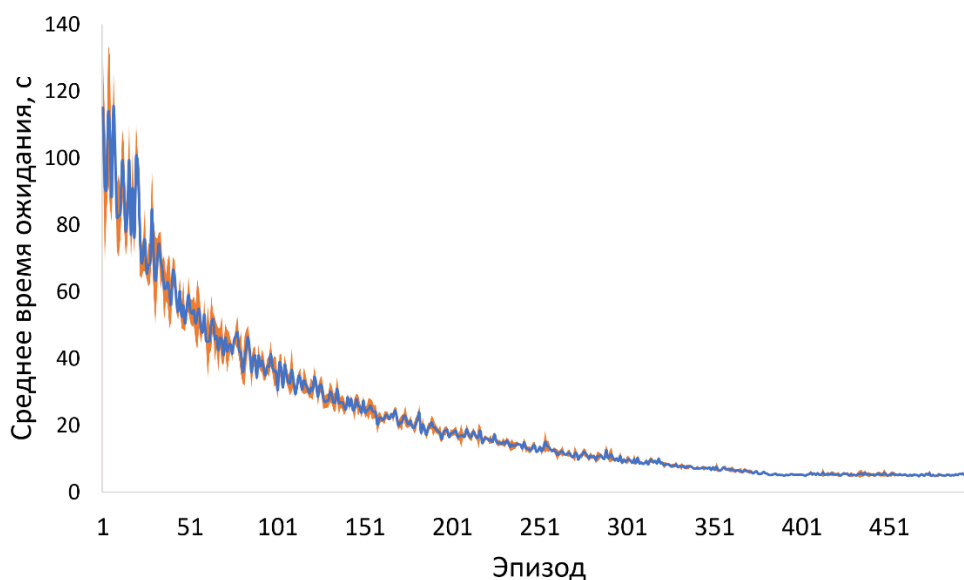


Рисунок 40 – Кривая обучения алгоритма двойного Q-обучения

Далее была выполнена оценка эффективности предложенного подхода и сравнение подхода с базовыми алгоритмами, используя выбранные показатели качества. В таблице 11 представлены результаты сравнения по критерию среднего времени ожидания и среднего

времени движения в описанных сценариях моделирования движения. Результаты сравнения методов и алгоритмов усреднялись по десяти эпизодам.

Предложенный подход (Hybrid в таблице) показал наилучшие результаты по выбранным критериям для каждого сценария моделирования.

Таблица 11 – Сравнение эффективности на тестовых эпизодах

Model	Среднее время ожидания		Среднее время движения	
	Grid 4x4	Cologne	Grid 4x4	Cologne
IDQN	12,3	5,1	145,8	85,49
MPLight	20,5	10,2	158,6	94,9
MaxPressure	23,2	9,1	161	93,8
Hybrid	8,75	4,6	141,9	84,9

На заключительном этапе экспериментального анализа подробно рассматриваются результаты экспериментов на тестовых эпизодах. На рисунке 41 и рисунке 42 показано среднее время ожидания для каждого тестового эпизода в синтетическом сценарии «Grid 4x4» и реальном сценарии «Cologne, congested area» соответственно. Предложенный алгоритм показал наилучшие результаты для каждого сценария и каждого эпизода.

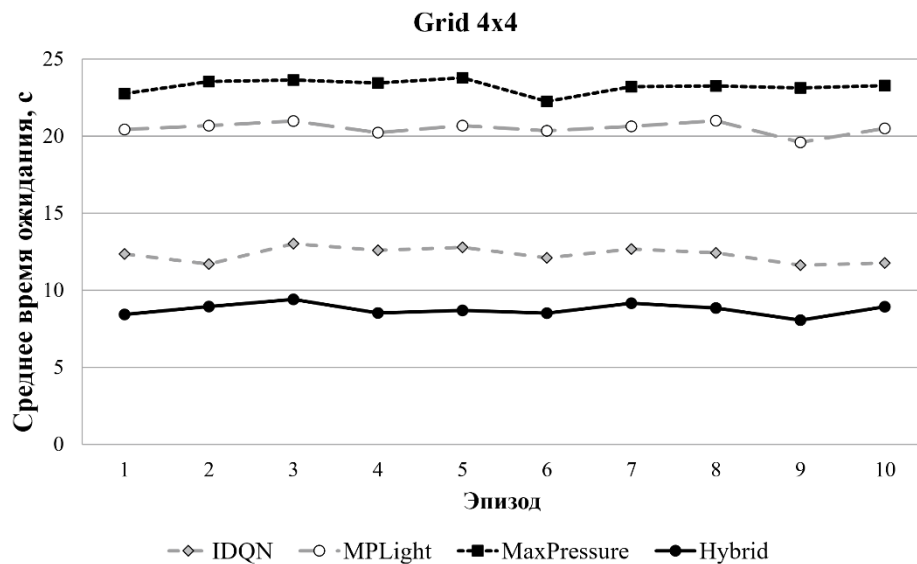


Рисунок 41 – Результаты сравнения по критерию среднего времени ожидания на синтетическом сценарии

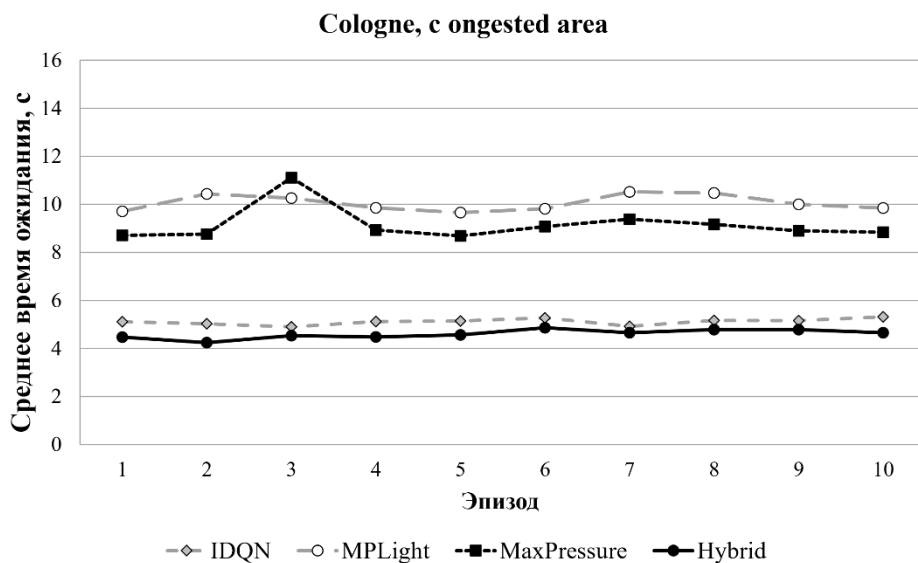


Рисунок 42 – Результаты сравнения по критерию среднего времени ожидания в реальном сценарии

На основании полученных экспериментальных результатов можно сделать вывод, что предлагаемый подход на основе гибридного двойного Q-обучения превосходит классические методы и RL-алгоритмы в рассмотренных сценариях моделирования движения.

3.3 Управление сигналами светофоров на основе метода максимального взвешенного потока

Несмотря на активное развитие RL-методов управления сигналами светофоров, остаются открытыми вопросы обучения моделей и сходимости к глобальному максимуму, чувствительности к выбору гиперпараметров алгоритмов, стабильности их работы на сложных сценариях управления. Кроме того, в случае применения RL-методов при изменении топологии дорожной сети может потребоваться переобучения одного или нескольких агентов, что, в свою очередь, потребует предварительного сбора большого объема данных, описывающих пространство состояний, для обучения.

Для решения указанных проблем в диссертационной работе был разработан метод адаптивного управления, основанный на подходе модельно-предикативного управления [135, 122, 188]. Разработанный метод основан на выборе фазы светофорного объекта, максимизирующей прогнозируемое количество транспортных средств, которые проедут перекресток за выбранный интервал действия фазы. Ключевым моментом предложенного метода является расчет прогнозного потока транспортных средств через перекресток. В

разделе 3.3.1 для прогнозирования движения транспортных средств предложено использовать набор аналитических закономерностей, базирующихся на простейших принципах физики (механики), связывающих возможное перемещение транспортного средства с его возможным ускорением и физическими ограничениями среды, вызванными, например, впереди стоящими ТС. Эти закономерности используют ряд параметров (максимальная скорость, максимальное ускорение, время задержки при начале движения и т.п.), которые на практике могут быть получены на основании оценок параметров движения реальных транспортных средств. В разделе 3.3.2 предложена модификация описанного метода, учитывающая время ожидания (простоя), которое ТС проводит на перекресте, а также использующая более сложную модель ГНС для оценки времени прохождения перекрестка для прогнозирования потока транспортных средств. В разделе 3.3.3 предложенный метод был модифицирован для его применения в гетерогенном потоке транспортных средств, содержащем подключенные и управляемые водителями транспортные средства.

3.3.1 Метод максимального потока на основе детерминированной прогнозной модели

Введем следующие обозначения. Пусть P – множество фаз светофорного объекта, τ_{min} – допустимый минимальный временной интервал переключения фаз, t_{cur} – время активности текущей действующей фазы светофора $p_{cur} \in P$, $p_{out} \in P$ – следующая выбранная фаза светофорного объекта. Задача адаптивного управления заключается в максимизации прогнозируемого потока транспортных средств, проходящих через перекресток за рассматриваемый временной интервал.

В данном разделе предложен метод адаптивного управления сигналами/фазами светофоров, основанный на детерминированной прогнозной модели. Под детерминированной прогнозной моделью понимается набор явных аналитических закономерностей и/или операций, связывающих входные данные, то есть информацию о движении транспортных средств (включая транспортные средства типа CAV и CV) в окрестности конкретного перекрёстка, с данными о прогнозируемом «потоке» (то есть количестве) транспортных средств через перекрёсток за одну конкретную фазу светофорного цикла: $PredictedFlow(phase)$. Собственно, сам предлагаемый метод управления тривиален – он выбирает ту фазу, прогнозируемый поток для которой оказывается максимален (Алгоритм 3):

Алгоритм 3: Метод максимального потока MaxFlow

```

1: Входные данные:  $\tau_{min}, t_{cur}, p_{cur}, P$ 
2: Выходные данные:  $p_{out}$ 
3: if  $t_{cur} < \tau_{min}$  then
4:    $p_{out} = p_{cur}$ 
5:    $t_{cur} = t_{cur} + 1$ 
6: else
7:    $p_{out} = \operatorname{argmax}(\{PredictedFlow(p) \text{ for } p \text{ in } P\})$ 
8:    $t_{cur} = 0$ 
9: end if

```

Очевидно, ключевым моментом предложенного метода является то, как именно рассчитывается поток ТС за интервал τ_{min} через перекрёсток в текущий момент времени, то есть вид функции $PredictedFlow(p)$. Набор используемых в предложенной модели аналитических закономерностей базируется на простейших принципах физики (механики), связывающих возможное перемещение транспортного средства с его возможным ускорением и физическими ограничениями среды, вызванными, например, впереди стоящими ТС. Эти закономерности используют ряд параметров (максимальная скорость, максимальное ускорение, время задержки при начале движения и т.п.), которые на практике могут быть получены на основании оценок параметров движения реальных транспортных средств. В среде моделирования они оказываются известными.

Имея оценку времени $t(c)$, требуемого для достижения перекрёстка ТС-ом c из множества ТС C_l на конкретной l -ой полосе, искомая оценка потока имеет вид:

$$PredictedFlow(p) = \sum_{l \in L_p^{income}} \sum_{c \in C_l} I(t(c) < \tau_{min}),$$

где L_p^{income} – множество входящих полос движения на перекрёстке, для которых допустимо движение при фазе p ;

l – обозначение некоторой полосы движения;

$I(val)$ – индикатор, принимающий значение «1» при $val = True$, и значение «0» – иначе.

Для оценки $t(c)$ используется детерминированная прогнозная модель. Для упрощения изложение далее ведётся для фиксированного ТС, поэтому аргумент c , обозначающий это ТС, в функции $t(c)$ и других опускается. Введем дополнительные обозначения: V_{max} – максимальная разрешённая скорость ТС, a_{max} – максимально

возможное ускорение ТС, v – текущая скорость ТС, S – расстояние от ТС до следующего перекрёстка.

Время, за которое ТС достигнет перекрёстка t , складывается в общем случае из трёх составляющих: t_1 – времени равноускоренного прямолинейного движения ТС (до момента достижения скорости V_{max}), t_2 – времени равномерного прямолинейного движения; d – временной задержки в начале движения:

$$t = t_1 + t_2 + d.$$

В зависимости от текущего значения скорости ТС v способ вычисления слагаемых различается.

Случай 1: если $v = 0$, то предварительная оценка

$$t_1 = \frac{V_{max}}{a_{max}}, \quad (41)$$

то есть это интервал времени, за которое ТС достигнет скорости V_{max} , двигаясь равноускоренно. Поскольку ТС может достигнуть перекрёстка раньше, чем за время t_1 , то это значение корректируется. А именно, пусть S_1 – расстояние, которое преодолет ТС, двигаясь равноускоренно за время t_1 , тогда:

Если $S_1 > S_2$, тогда слагаемое $t_2 = 0$, а

$$t_1 = \sqrt{\frac{2S}{a_{max}}}, \quad (42)$$

$$d = \alpha \cdot \frac{S}{L + L_0},$$

где L – средняя длина ТС;

L_0 – среднее расстояние между соседними ТС на полосе;

α – настраиваемый коэффициент, в экспериментах полагался $\alpha = 1$.

Иначе, то есть при $S_1 \leq S_2$, величина t_1 вычисляется по формуле (41), $t_2 = \frac{S-S_1}{V_{max}}$, а величина d вычисляется по формуле (42).

Случай 2: если $v > 0$, тогда время, за которое ТС достигнет скорости V_{max} , двигаясь равноускоренно:

$$t_1 = \frac{V_{max} - v}{a_{max}},$$

$$S_1 = v \cdot t_1 + \frac{a_{max} \cdot t_1^2}{2},$$
(43)

Если $S_1 > S_2$, тогда слагаемое $t_2 = 0$, при

$$t_1 = \frac{-v + \sqrt{v^2 + 2a_{max} \cdot S}}{a_{max}},$$

$$d = \begin{cases} \frac{S}{L + L_0}, & \text{если } V_{max} - v > \Delta, \\ 0, & \text{иначе} \end{cases}$$
(44)

где Δ – настраиваемый параметр, полагаемый в экспериментах $\Delta = 5$.

Иначе, то есть при $S_1 \leq S_2$, величины t_1 и S вычисляются по (43), d вычисляется по (44), а $t_2 = \frac{S - S_1}{V_{max}}$.

Экспериментальные исследования метода адаптивного управления сигналами светофоров на основе детерминированной прогнозной модели движения транспортных средства представлены в разделе 3.3.4.1

3.3.2 Метод максимального взвешенного потока на основе модели глубокой нейронной сети

Метод максимального потока, предложенный в разделе 3.3.1, обладает рядом недостатков: использует простую модель оценки времени прохождения перекрестка и не учитывает время ожидания (простоя), которое ТС проводит на перекрестке.

Учет времени ожидания ТС необходим для предотвращения ситуаций, при которых время ожидания транспортных средств на второстепенной дороге (с малым транспортным потоком) будет увеличиваться до бесконечности.

Для исправления указанных недостатков было предложено использовать т.н. взвешенный транспортный поток, который учитывает время ожидания ТС следующим образом:

$$PWFlow(p) = \sum_{l \in L_p^{income}} \sum_{c \in C_l} \eta(c, l) I(t^{dnn}(c) < \tau_{min}),$$
(45)

где L_p^{income} – множество входящих полос движения на перекрёстке, для которых допустимо движение при фазе p ;

l – обозначение некоторой полосы движения;

C_l – множество транспортных средств в текущий момент на соответствующей полосе движения l ;

$I(val)$ – индикатор, принимающий значение «1» при $val = True$, и значение «0» – иначе;

$\eta(c, l)$ – весовой коэффициент.

Коэффициент $\eta(c, l)$ необходим для корректировки «веса» ТС c в транспортном потоке и зависит от времени ожидания $delay(c, l)$ (в секундах) этого ТС на полосе l на перекрестке:

$$\eta(c, l) = 1 + \alpha \cdot delay(c, l),$$

где α – эмпирически выбираемый коэффициент. В экспериментах (на основе предварительно проведенного экспериментального анализа) полагается $\alpha = 0,01$.

Для оценки времени прохождения перекрестка ТС c вместо использования детерминированной прогнозной модели $t(c)$ предлагается использовать модель на основе искусственной нейронной сети глубокого обучения $t^{dnn}(c)$. В качестве входных параметров модели используются следующие характеристики, прямо или косвенно описывающие транспортную ситуацию на текущем и смежном дорожных сегментах, а также движение рассматриваемого ТС c :

- положение ТС $S = S(c)$ (расстояние от текущей позиции ТС до перекрестка);
- скорость движения ТС $v = v(c)$;
- ускорение ТС $a = a(c)$;
- максимально разрешенная скорость движения v_{max} ;
- число ТС перед рассматриваемым ТС c до перекрестка $n = n(c)$;
- тип k ожидаемого перестроения на перекрестке (0 – прямо, 1 – направо, 2 – налево, 3 – разворот);
- скорость v_0 и положение S_0 ближайшего ТС на полосе выезда.

Нейронная сеть состоит из 7 полносвязных слоев. Архитектура используемой нейронной сети с указанием количества нейронов каждого слоя показана на рисунке 43.

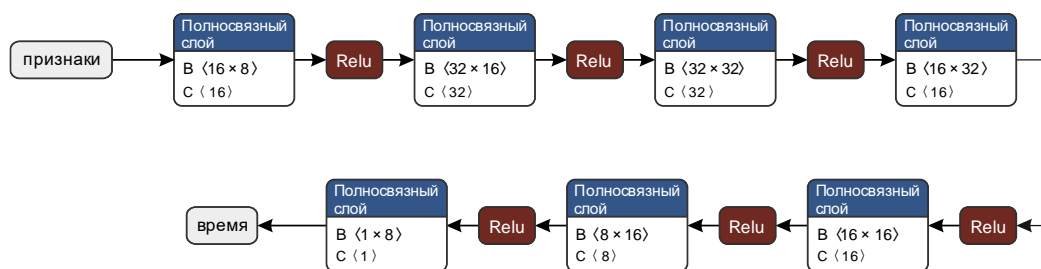


Рисунок 43 – Архитектура нейронной сети для оценки времени прохождения перекрестка

Следующая фаза светофорного цикла выбирается аналогично базовому методу: выбирается фаза, для которой взвешенный транспортный поток максимален, т.е. для выбора фазы в алгоритме 3 используется выражение:

$$p_{out} = \operatorname{argmax}(\{PWFlow(p) \text{ for } p \text{ in } P\}).$$

Таким образом в виде псевдокода метод максимального взвешенного потока, использующий модель ГНС для прогнозирования времени прибытия ТС на перекресток, имеет следующий вид (Алгоритм 4):

Алгоритм 4: Метод максимального взвешенного потока MaxPWFlow

- 1: **Входные данные:** $\tau_{min}, t_{cur}, p_{cur}, P$
 - 2: **Выходные данные:** p_{out}
 - 3: **if** $t_{cur} < \tau_{min}$ **then**
 - 4: $p_{out} = p_{cur}$
 - 5: $t_{cur} = t_{cur} + 1$
 - 6: **else**
 - 7: $p_{out} = \operatorname{argmax}(\{PWFlow(p) \text{ for } p \text{ in } P\})$
 - 8: $t_{cur} = 0$
 - 9: **end if**
-

Экспериментальная оценка эффективности предложенного метода в системе моделирования движения транспортных средств представлена в разделе 3.3.4.2.

3.3.3 Метод максимального взвешенного потока в гетерогенном потоке транспортных средств

В предыдущих разделах при определении метода максимального потока (Алгоритм 3) и метода максимального взвешенного потока (Алгоритм 4) предполагалось полное знание маршрута, положения, скорости и другой информации, собираемой со всех транспортных средств, что в настоящее время на практике неосуществимо. Чтобы применить рассмотренные методы в сценарии движения с гетерогенным транспортным

потоком, введем в рассмотрение три типа транспортных средств, которые можно различить в транспортном потоке [342]:

1) подключенные транспортные средства (CV). Для этого типа транспортного средства предполагается, что маршрут, положение, скорость и другая информация о транспортном средстве известны в реальном времени;

2) наблюдаемые транспортные средства (OV, Observable Vehicles). Это транспортные средства, управляемые людьми, которые движутся в пределах досягаемости датчиков CV (камер, лидаров) впереди или позади транспортных средств типа CV. В эту группу также входят транспортные средства, находящиеся в зоне действия камер видеонаблюдения на перекрестках. Предполагается, что для этого типа транспортного средства известны положение и скорость ТС;

3) ненаблюдаемые транспортные средства (NOV, Non-observable Vehicles). Это транспортные средства, которые не могут быть обнаружены. В результате информация об этих транспортных средствах не используется для оценки прогнозируемого транспортного потока на перекрестках.

Эти типы транспортных средств схематично показаны на рисунке 44.

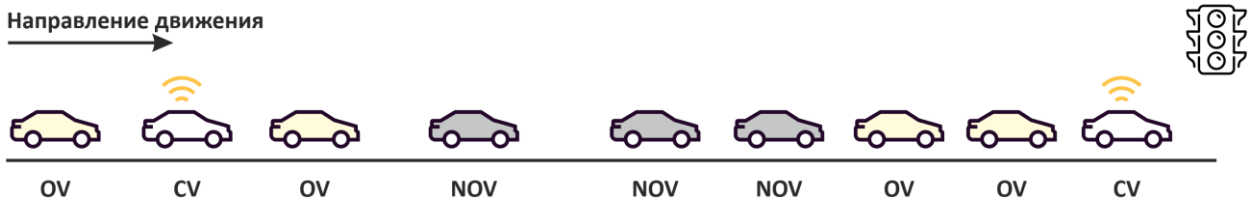


Рисунок 44 – Различные типы транспортных средств

Принимая во внимание эти типы транспортных средств, выражение (31) для расчета потока транспортных средств примет следующий вид:

$$PWFlow(p) = \sum_{l \in L_p^{income}} \sum_{c \in C_l^{obs}} \eta(c, l) I(t^{dnn}(c) < \tau_{min}), \quad (46)$$

где C_l^{obs} – множество наблюдаемых транспортных средств, которое содержит либо только транспортные средства типа CV, либо транспортные средства как CV, так и OV (т.е. подключенные и наблюдаемые ТС).

Экспериментальные исследования метода максимального взвешенного потока в условиях гетерогенного потока транспортных средств представлено в разделе 3.3.4.3.

3.3.4 Экспериментальные исследования метода адаптивного управления на основе максимального взвешенного потока

Экспериментальные исследования разработанного метода на основе максимального потока включают в себя оценку эффективности метода на основе детерминированной модели (раздел 3.3.4.1), метода на основе модели нейронной сети на основе глубокого обучения (раздел 3.3.4.2), а также применение метода в сценарии с гетерогенным потоком транспортных средств (раздел 3.3.4.3).

3.3.4.1 Метод максимального потока на основе детерминированной прогнозной модели

На первом этапе экспериментальных исследований метода адаптивного управления выполнялось сравнение эффективности предложенного метода MaxFlow (Алгоритм 3) и двух наиболее эффективных методов адаптивного управления сигналами светофора: эмпирического MaxPressure и RL-метода на базе современного алгоритма A2C. В качестве данных для описания состояния в RL-методе на базе A2C использовался следующий набор:

- 1) текущий сигнал светофора, представленный в виде бинарного вектора;
- 2) нормированный вектор, состоящий из количества ТС на каждой из регулируемой светофором полос движения;
- 3) нормированный вектор, состоящий из количества стоящих на месте ТС, для которых $v(a) = 0$, на каждой из регулируемой светофором полос движения.

Таким образом, для восьми статусного светофора, изображённого на рисунке 36, состояние перекрёстка описывается в виде вектора размерности 32 ($=8+12+12$).

Функция награды для RL-метода на базе A2C в экспериментах имела следующий вид:

$$R = w_{t-1} - w_t,$$

где w_t – сумма длительностей ожиданий ТС на регулируемых светофором полосах в момент времени t .

Для возможности сопоставления результатов с потенциально наихудшим значением параллельно оценивалась эффективность метода Uniform, интервал переключения фаз τ_{\min} во всех экспериментах и для всех методов полагался равным 10 сек. Экспериментальные исследования были проведены в среде моделирования SUMO. Движение транспортных средств в SUMO может быть описано двумя способами. При использовании первого способа задаётся маршрут и время начала поездки для каждого ТС отдельно. При втором

способе движение транспортных средств задаётся с использованием потоков: указывается маршрут поездки/потока и вероятность появления ТС в потоке (период появления ТС в потоке не превышает 1 секунды). В ходе проведения всех экспериментальных исследований использовался второй способ описания движения.

Эксперимент 1: экспериментальные исследования проводились при шаге симуляции, равном 1 секунде и при общем времени симуляции 300 секунд. Для проведения экспериментальных исследований было разработано три сценария (таблица 12).

Сценарий «1x1»: транспортная сеть представляет собой один регулируемый светофором перекрёсток, аналогичный показанному на рисунке 36. Каждая проезжая часть длиной 500 метров состоит из трёх полос движения. Для каждой проезжей части, ведущей к перекрёстку, заданы три потока транспортных средств, при этом вероятности использования центральных полос (проезд «напрямую») выше, чем для остальных. Выбор проезжих частей равновероятен.

Сценарий «2x2»: транспортная сеть имеет вид решётки и включает в себя 4 перекрёстка, регулируемых светофорами. Каждая проезжая часть длиной 150 метров состоит из трёх полос. Движение транспортных средств задано в виде четырёх потоков: два горизонтальных с разными пунктами отправления и единым пунктом назначения, два вертикальных непересекающихся потока. Все светофоры используют один и тот же метод.

Сценарий «5x5»: транспортная сеть имеет вид решётки и включает в себя 25 перекрёстков, регулируемых светофорами с помощью одного и того же метода. Каждая проезжая часть длиной 200 метров состоит из трёх полос. Для каждой проезжей части, смежной с центральными и крайними перекрёстками на границе транспортной сети, формируются два потока: первый с конечным пунктом, расположенным на противоположной стороне решётки; конечный пункт второго – соседний для конечного пункта первого, лежащий на противоположной стороне (центральный или крайний).

Для каждого из методов эксперименты с моделированием/обучением проводились по 10 раз, для каждого эксперимента оценивался критерий эффективности управления светофором: среднее время движения всех транспортных средств в сети. Статистические показатели этого критерия (среднее, минимум, максимум и СКО) для всех сравниваемых методов и типов сетей представлены в таблице 12.

Таблица 12 – Критерий эффективности управления дорожным движением с использованием сигналов светофоров

Тип сети	Критерий	Uniform	MaxPressure	A2C	Предложенный метод
1x1	Среднее	16,450	9,706	14,604	10,180
	СКО	1,570	0,869	1,911	0,803
	Мин	14,288	8,350	11,972	8,614
	Макс	18,756	11,358	18,850	11,569
2x2	Среднее	54,229	17,945	10,815	6,413
	СКО	5,390	3,055	1,880	0,912
	Мин	42,891	13,392	8,430	5,163
	Макс	63,819	22,681	14,486	7,675
5x5	Среднее	117,752	93,694	66,549	63,422
	СКО	0,726	2,636	1,109	1,235
	Мин	116,121	87,577	65,272	61,121
	Макс	118,961	97,936	69,088	64,812

Как видно из представленных в таблице результатов, по среднему, минимальному и максимальному среднему времени корреспонденций в сети предложенный метод MaxFlow превосходит передовой RL-метод на базе A2C во всех случаях, а метод MaxPressure – в двух из трёх случаев, проигрывая только в тривиальном случае с одним перекрёстком.

Эксперимент 2: отдельно исследовалась зависимость суммарного и среднего времени ожидания ТС на светофорах в зависимости от числа ТС в сети (рисунок 45).

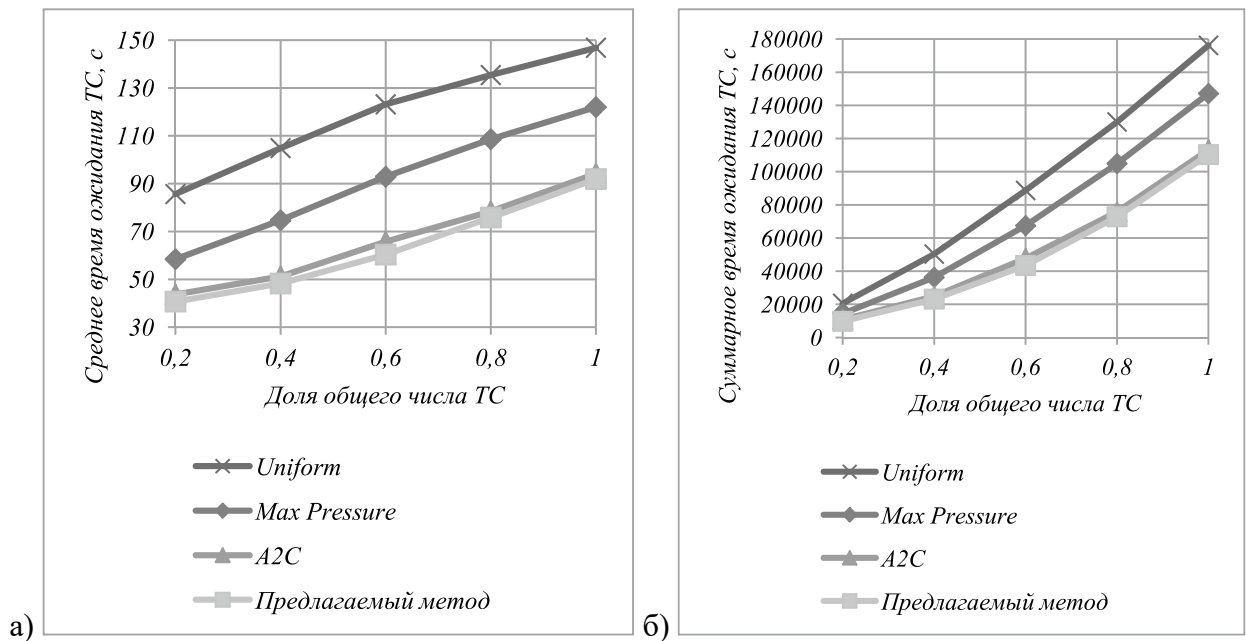


Рисунок 45 – Время ожидания ТС на светофорах в зависимости от числа ТС в сети: а) среднее время ожидания, б) суммарное время ожидания

В представленных на рисунке 45 результатах исследования проводились для сценария «5x5», единица по оси абсцисс соответствует 1200 ТС.

Как видно из представленных результатов, предложенный метод везде превосходит все методы-конкуренты, включая RL-алгоритм на базе A2C.

3.3.4.2 Метод максимального взвешенного потока на основе модели глубокой нейронной сети

На втором этапе экспериментального анализа проводилось исследование метода максимального взвешенного потока на основе модели глубокой нейронной сети, нацеленное на сравнение эффективности предложенного метода MaxPWFlow (Алгоритм 4) с базовым методом MaxFlow (Алгоритм 3), классическим алгоритмом MaxPressure [120] и современными методами на основе обучения с подкреплением. Экспериментальные исследования разработанных алгоритмов проводились в системе моделирования движения транспортных средств SUMO [327].

Для оценки эффективности алгоритмов использовались сценарии имитационного моделирования, основанные на сценарии движения SUMO «TAPAS Cologne» [340]:

1) Cologne-8 – сценарий, выполняющий моделирование движения транспортных средств в области транспортной сети малого размера [339].

2) Cologne-316 – разработанный крупномасштабный сценарий моделирования движения транспортных средств.

Дорожная сеть сценария Cologne-316 показана на рисунке 46. Параметры сценариев представлены в таблице 13.



Рисунок 46 – Дорожная сеть сценария моделирования движения Cologne-316

Таблица 13 – Параметры сценариев движения

Сценарии	Светофоры	Перекрестки	Сегменты	Поездки в сети
Cologne-8	8	78	149	1740
Cologne-316	316	2928	5808	13570

Каждый сценарий определяется дорожной сетью и поездками, совершаемыми в сети (созданы на основе исходных данных о загруженности дорог). RL-модели обучались для каждого сценария на данных из нескольких эпизодов. Эпизоды различаются начальным расположением транспортных средств на сегментах сети, временем начала движения и маршрутами движения. В течение одного эпизода выполняется моделирование всех поездок в сценарии. Сравнение всех моделей проводилось на одинаковых данных, на выборке из десяти эпизодов (начальные положения транспортных средств на сегментах сети, время начала движения и маршруты были одинаковыми для всех моделей в одном конкретном эпизоде).

В качестве базовых алгоритмов для оценки эффективности предложенного метода использовались классические алгоритмы и методы на основе обучения с подкреплением. Для сравнения на основе анализа современного состояния исследований были выбраны методы различных классов, показавшие хорошие результаты в решении задачи адаптивного управления сигналами светофоров:

- 1) метод максимального потока MaxFlow [122];
- 2) MaxPressure [120];
- 3) IDQN – независимый алгоритм глубокого Q-обучения, основанный на сверточной нейронной сети [339]. Состояние каждого агента характеризуют следующие величины: текущая фаза светофора, длина очереди для каждой входящей полосы, количество приближающихся транспортных средств и сумма скоростей приближающихся транспортных средств для каждой входящей полосы. В качестве функции вознаграждения использовалось общее время ожидания с отрицательным знаком;

- 4) IPPO – алгоритм обучения с подкреплением для оптимизации политики [341]. Пространство наблюдения и функция вознаграждения использовались такие же, как и в методе IDQN;

- 5) A2C [343] – алгоритм класса «actor-critic».

RL-алгоритмы были реализованы на языке программирования Python с использованием библиотеки PyTorch и библиотеки для глубокого обучения с подкреплением PFRL [344].

Далее представлены результаты экспериментов. На первом этапе исследований оценивалось качество прогнозирования времени прохождения перекрестков с использованием детерминированной прогнозной модели $t(c)$ и с использованием модели искусственной нейронной сети (ИНС) $t^{dnn}(c)$. Для обучения модели ИНС использовались данные о движении транспортных средств на выборке из 70 эпизодов, для валидации и тестирования модели использовались данные, полученные при моделировании 15 эпизодов движения. График зависимости средней абсолютной ошибки от эпохи обучения показан на рисунке 47. Полученная кривая обучения подтверждает сходимость модели.

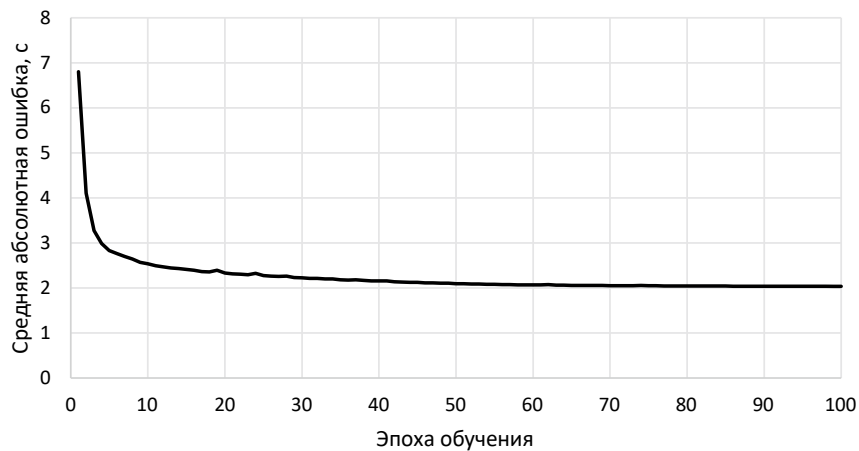


Рисунок 47 – Сходимость модели ИНС для прогнозирования времени движения

График зависимости средней абсолютной ошибки прогноза от времени прохождения перекрестка (т.н. горизонта прогноза) для детерминированной прогнозной модели и модели ИНС показан на рисунке 48. Из представленного графика видно, что предложенный алгоритм прогнозирования позволяет значительно (в 2-3 раза) снизить ошибку прогноза.

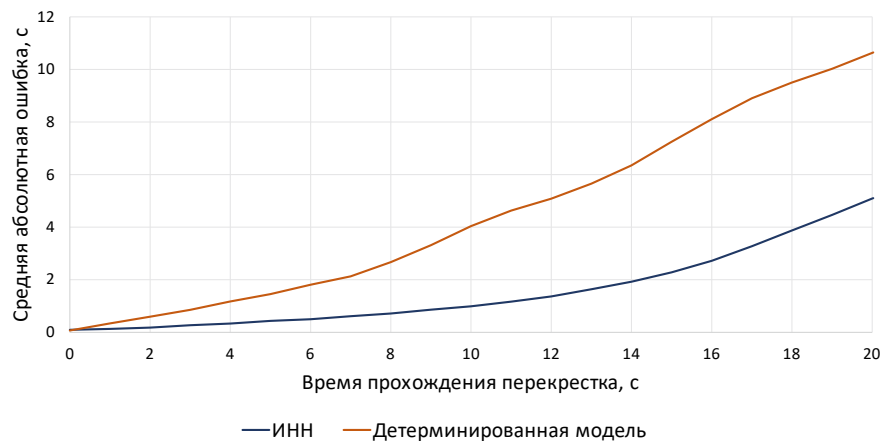


Рисунок 48 – Средняя абсолютная ошибка прогнозирования времени прохождения перекрестка

На втором этапе экспериментального исследования оценивалась сходимость RL-алгоритмов. Кривые обучения RL-алгоритмов на сценариях Cologne-8 (рисунок 49, а) и Cologne-316 (рисунок 49, б) подтверждают сходимость алгоритмов. Алгоритмы MaxPFlow, MaxFlow и MaxPressure – необучаемые алгоритмы с точки зрения правила принятия решения о выборе фазы, поэтому для них подобные графики не представлены.

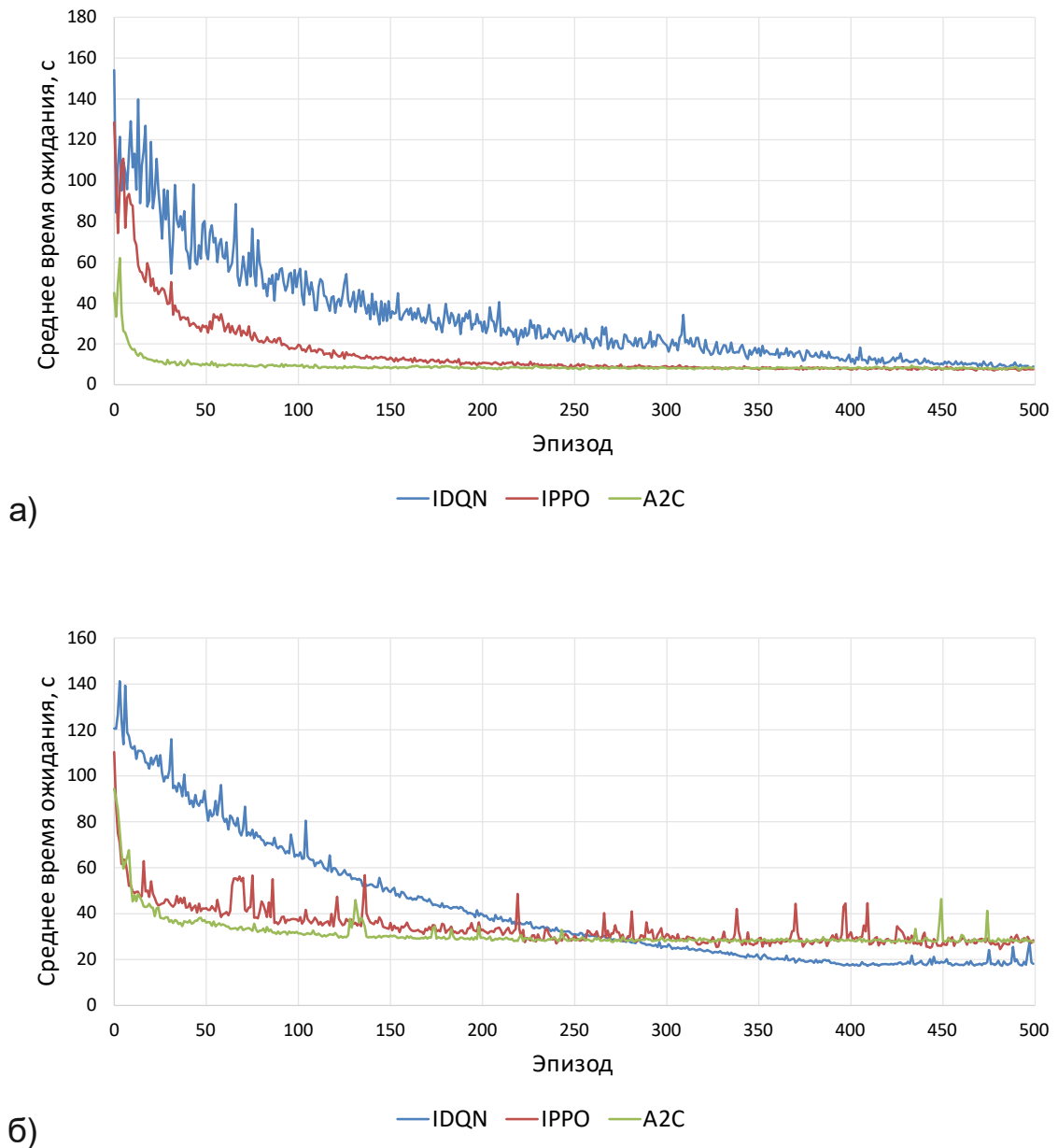


Рисунок 49 – Сходимость алгоритмов на сценариях: а – Cologne-8, б – Cologne-316

Далее проводилось сравнение эффективности алгоритмов на тестовой выборке из десяти эпизодов по двум критериям: среднее время ожидания и среднее время движения. Критерий среднего времени ожидания (в секундах) показывает среднее время, которое транспортное средство провело без движения на перекрестке при движении по маршруту,

среднее время движения (в секундах) – это среднее время, затрачиваемое транспортными средствами на совершение поездки из начального пункта их маршрута к месту назначения. Результаты сравнения эффективности алгоритмов представлены в таблице 14.

Таблица 14 – Сравнение эффективности управления транспортными потоками

Модель	Среднее время ожидания		Среднее время движения	
	Cologne-8	Cologne-316	Cologne-8	Cologne-316
IDQN	4,14	15,3	89,62	329,6
IPPO	4,69	31,93	90,83	347,72
A2C	7,24	26,32	94,91	347,81
MaxFlow	3,46	17,58	88,86	334,99
MaxPressure	9,16	20,63	93,82	337,91
MaxPWFlow	3,2	14,43	88,03	328,09

Результаты экспериментальных исследований показывают, что предложенный алгоритм MaxPWFlow превосходит детерминированный метод MaxPressure и современные RL-алгоритмы по критериям среднего времени ожидания и среднего времени движения на рассматриваемых сценариях. Кроме того, правило принятия решения не требует обучения, что также является преимуществом предложенного алгоритма.

В заключительной части исследований оценивалось время ожидания и время движения отдельно, по каждому тестовому эпизоду. Результат сравнения по критерию времени ожидания на сценарии Cologne-316 показан на рисунке 50.

Как видно из графика, предложенный метод превосходит остальные алгоритмы в каждом из эпизодов.

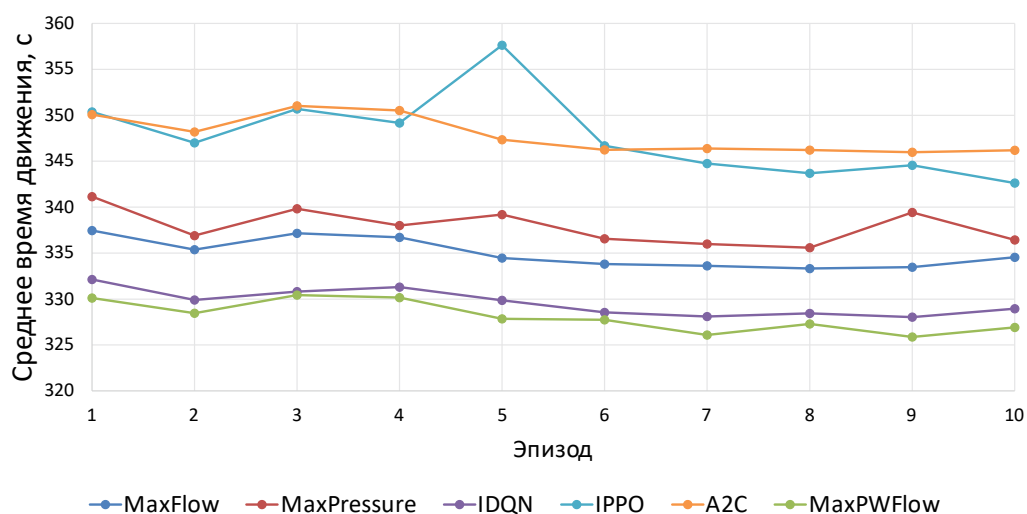


Рисунок 50 – Среднее время движения в каждом эпизоде сценария Cologne-316

3.3.4.3 Метод максимального взвешенного потока в гетерогенном потоке

транспортных средств

На третьем этапе экспериментального исследования метода максимального взвешенного потока оценивается эффективность метода в частично подключенной среде с гетерогенным потоком транспортных средств. Сравнивается алгоритм с различной доступной информацией, используемой для оценки прогнозируемого взвешенного потока в функции $PWFlow(p)$ в выражении (45):

- 1) «CV only»: используется только информация в режиме реального времени от подключенных транспортных средств;
- 2) «CV + observed»: используется информация из транспортных средств типа CV и OV.

Экспериментальные исследования алгоритма проводились в системе моделирования SUMO [327] в сценарии «Cologne-316» (рисунок 46). Моделирование проводилось в утренние часы пик с 7:00 до 9:00.

Рассмотренный подход протестирован для доли подключенных транспортных средств типа CV в транспортном потоке от 0% до 100% с шагом 10%. Моделирование повторялось в десяти эпизодах, каждый эпизод отличался начальным положением ТС и временем начала движения. Каждый эпизод моделировался десять раз с различным выбором подмножества транспортных средств, которые считались подключенными в рамках сценария. Выбор подмножества транспортных средств производился случайным образом.

Для оценки эффективности подхода алгоритмы оценивались по следующим критериям:

- 1) среднее время движения (в секундах), т.е. время движения, необходимое транспортным средствам для завершения поездки от исходного положения до пункта назначения, усредненное по транспортным средствам и эпизодам;
- 2) среднее время ожидания на перекрестке (в секундах), которое показывает время простоя транспортного средства на перекрестке, усредненное по транспортным средствам и эпизодам.

Зависимость среднего времени движения и среднего времени ожидания от доли подключенных ТС типа CV представлена на рисунках 51 и 52 соответственно.

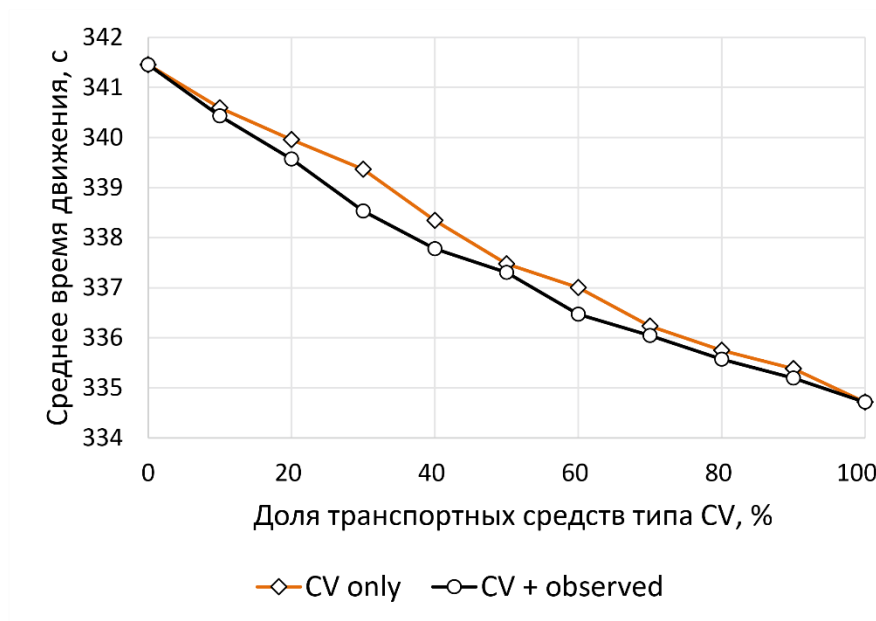


Рисунок 51 – Зависимость среднего времени движения от доли CV

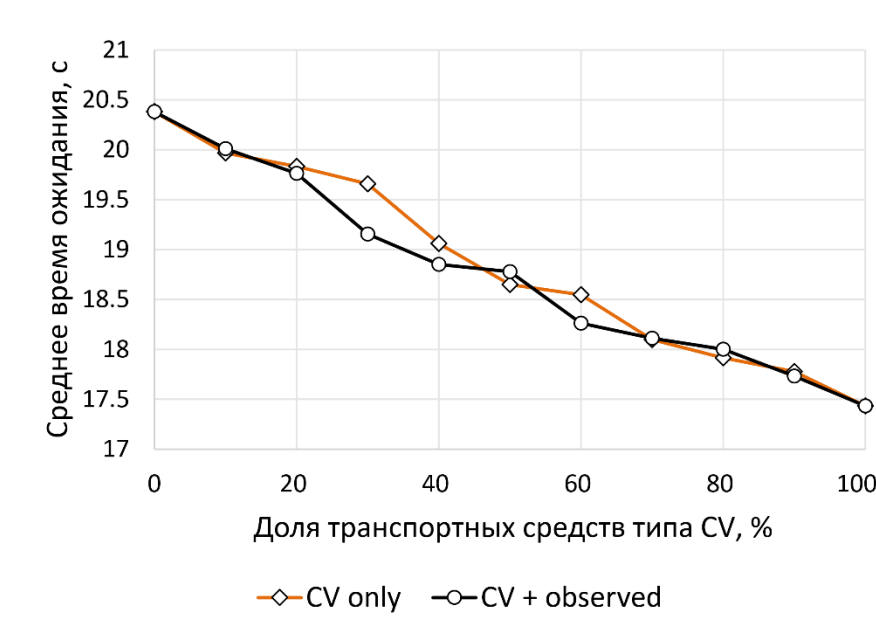


Рисунок 52 – Зависимость среднего времени ожидания от от доли CV

Как видно из представленных графиков, как среднее время движения, так и среднее время ожидания уменьшаются с увеличением количества подключенных транспортных средств в сценарии. Среднее время в пути сократилось до 1,9%, а среднее время ожидания – до 14,6%. Использование дополнительной информации о движении наблюдаемых транспортных средств также снижает значения рассматриваемых критериев.

3.4 Совместное управление сигналами светофоров и траекториями движения транспортных средств

Разработанные в разделах 3.2 и 3.3 методы адаптивного управления сигналами светофорных объектов используют информацию от подключенных транспортных средств для эффективного контроля транспортного потока путем светофорного регулирования. В то же время, возможность управлять траекторией движения транспортного средства на перекрестке позволяет рассматривать задачу управления транспортным потоком как задачу совместного управления сигналами светофоров и траекториями движения транспортных средств на перекрестке.

В разделе 3.4.1 рассматривается метод управления траекториями движения транспортных средств на перекрестке, использующий информацию о прогнозной фазе светофора, рассчитанной с использованием метода адаптивного управления на основе максимального взвешенного потока (раздел 3.3.2). В разделе 3.4.2 предложена модификация метода, заключающаяся в совместном применении метода управления траекториями движения транспортных средств и метода адаптивного управления сигналами светофоров.

3.4.1 Метод управления траекториями движения с учетом адаптивного управления сигналами светофоров

Основная цель управления траекториями транспортных средств с учетом сигналов светофоров – оптимизация транспортного потока и повышение эффективности транспортной системы. Для достижения этих целей обычно учитываются следующие показатели: временные затраты на совершение транспортных корреспонденций и расход топлива. Ключевой задачей является минимизация времени движения транспортных средств, в т.ч. за счет сокращения времени ожидания на перекрестках, поскольку это уменьшает дорожные заторы и повышает мобильность водителей. Вторым важным критерием является снижение расхода топлива, так как это может способствовать сокращению выбросов и повышению энергоэффективности транспортной системы.

С учетом сказанного, задача, решаемая разработанным методом управления транспортным потоком на регулируемом перекрестке, формулируется следующим образом:

$$\lambda \cdot R_T \left(TravelT_{\Sigma}(A_{TS}, A_{Tr}) \right) + \eta \cdot R_T \left(WaitingT_{\Sigma}(A_{TS}, A_{Tr}) \right) + \delta \cdot \\ \cdot R_F \left(FuelC_{\Sigma}(A_{TS}, A_{Tr}) \right) \rightarrow \min_{A_{TS}, A_{Tr}},$$

где $TravelT_{\Sigma}(A_{TS}, A_{Tr})$ – общее время (длительность) совершения всех корреспонденций ТС на перекрестке;

$WaitingT_{\Sigma}(A_{TS}, A_{Tr})$ – общее время (длительность) ожидания всех ТС на перекрёстке;

$FuelC_{\Sigma}(A_{TS}, A_{Tr})$ – объем общего потребления топлива всеми ТС при движении по перекрёстку;

R_T – скалярная функция, возвращающая вещественное (безразмерное) число, равное в секундах задаваемой аргументом длительности времени;

R_F – скалярная функция, возвращающая вещественное (безразмерное) число, равное в миллилитрах задаваемому аргументом объему потребленного топлива;

A_{TS}, A_{Tr} – используемые алгоритмы управления сигналами светофоров и построения траектории движения (включая их параметры);

λ, η, δ – вещественные безразмерные весовые коэффициенты, совместно характеризующие относительную важность того или иного фактора.

Для описания метода управления траекториями движения транспортных средств с учетом адаптивного управления сигналами светофоров последовательно рассмотрим метод адаптивного управления сигналами светофоров (раздел 3.4.1.1), алгоритм построения траекторий движения транспортных средств (раздел 3.4.1.2), а также метод их согласованного применения.

3.4.1.1 Метод адаптивного управления сигналами светофоров

Метод адаптивного управления сигналами светофоров A_{TS} является одним из методов, использующихся в составе метода совместного управления сигналами светофоров и траекториями движения транспортных средств.

Для решения задачи адаптивного управления сигналами светофоров в работе используется метод максимального взвешенного потока, описанный в разделе 3.3.2 (Алгоритм 4). Метод управления основывается на выборе фазы светофорного цикла, максимизирующей прогнозируемый поток транспортных средств, проходящих через перекресток за рассматриваемый временной интервал.

Для удобства изложения продублируем краткое формальное описание используемого метода. Введем следующие обозначения. Пусть P – множество фаз, τ_{min} –

допустимый минимальный интервал переключения фаз, t_{cur} – длительность текущей активной фазы светофорного объекта $p_{cur} \in P$, $p_{out} \in P$ – следующая выбранная фаза.

Тогда в виде псевдокода алгоритм может быть представлен следующим образом:

Алгоритм 4: Метод максимального взвешенного потока MaxPWFlow

```

1: Входные данные:  $\tau_{min}, t_{cur}, p_{cur}, P$ 
2: Выходные данные:  $p_{out}$ 
3: if  $t_{cur} < \tau_{min}$  then
4:    $p_{out} = p_{cur}$ 
5:    $t_{cur} = t_{cur} + 1$ 
6: else
7:    $p_{out} = \operatorname{argmax}(\{PWFlow(p) \text{ for } p \text{ in } P\})$ 
8:    $t_{cur} = 0$ 
9: end if

```

Основным шагом метода является расчет прогнозируемого «взвешенного» потока транспортных средств через перекресток для заданной фазы $p \in P$, выполняемый функцией $PWFlow(p)$. В работе используется следующий вид функции:

$$PWFlow(p) = \sum_{l \in L_p^{income}} \sum_{c \in C_l} \eta(c, l) I(t(c) < \tau_{min}). \quad (47)$$

где $t(c)$ – оценка времени, необходимого транспортному средству $c \in C_l$ для достижения перекрестка;

C_l – множество транспортных средств, движущихся по полосе l из множества полос L_p^{income} , движение по которым разрешено при активной фазе $p \in P$ светофорного объекта;

$I(val)$ – индикатор, который возвращает значение 1, если $val = True$, и значение 0 в противном случае;

$\eta(c, l)$ – весовой коэффициент, учитывающий время ожидания транспортного средства c на полосе l .

В следующем подразделе приведено описание алгоритма построения траектории движения транспортного средства.

3.4.1.2 Алгоритм построения траектории

Применение алгоритма построения траектории A_{Tr} является вторым этапом, используемым в составе метода совместного управления сигналами светофоров и траекториями движения транспортных средств.

Пусть траектория – это квадратичная дифференцируемая функция $tr(t)$, для которой производная первого порядка (скорость) $tr'(t)$ является абсолютно непрерывной функцией и правая производная второго порядка (ускорение) $tr''(t)$ является интегрируемой.

Пусть контролируемое ТС въезжает на смежную с перекрестком полосу движения в момент времени t_0 со скоростью v_0 в позицию l_0 ; L – длина полосы движения; v_{max} – максимальная допустимая скорость движения на рассматриваемой полосе движения; \bar{a} – максимальное ускорение, с которым может двигаться рассматриваемое ТС; \underline{a} – максимальное торможение, достижимое рассматриваемым ТС. Пусть перекресток управляется светофорным объектом с фиксированным временем переключения фаз. Требуется построить физически достижимую и безопасную траекторию движения ТС до перекрестка tr с учетом следующих ограничений:

$$\begin{aligned} tr(t_0) &= l_0, \\ 0 \leq tr'(t) &\leq v_{max}, \quad \underline{a} \leq tr''(t) \leq \bar{a}, \quad \forall t \in (-\infty, +\infty), \\ G(T(tr, L)) &= T(tr, L), \end{aligned}$$

где $T(tr, L)$ – время, при котором ТС достигнет позиции L (т.е. въедет на перекресток), двигаясь по траектории tr ;

$G(t)$ – функция, которая возвращает ближайшее к t время зеленой фазы светофора ($G(t) = t$, если в момент времени t горит зеленый сигнал светофора, $G(t) > t$, если в момент времени t горит красный).

Функция $G(t)$ определяется следующим образом:

$$G(t) = \min\{t' : t' > t \wedge t' \in [T_0 + nT_c, T_0 + nT_c + T_g], n \in \mathbb{Z}^+\} \quad (48)$$

где \mathbb{Z}^+ – множество целых положительных чисел;

T_0 – время начала светофорного цикла,

$T_c = T_g + T_y + T_r$ – длительности полного рабочего цикла светофора, которое включает в себя длительности зеленой фазы T_g , желтой фазы T_y и красной фазы T_r .

Для управления движением ТС на перекрестке в работе используется модифицированный вариант Shooting Heuristic алгоритма [177]. В рамках оригинального Shooting Heuristic алгоритма для каждого ТС, находящегося на смежной с контролируемым перекрестком дорожной полосе, строится траектория движения, учитывающая траекторию впереди идущего ТС (при наличии такового). В рамках разработанного метода траектория формируется только для ближайшего к перекрестку ТС на дорожной полосе. Остальные ТС

двигаются согласно используемой модели следования за контролируемым ТС. В данной работе в качестве такой модели следования применяется модель следования за лидером [345].

На первом этапе Shooting Heuristic алгоритма под названием Forward Shooting Process (FSP) (рисунок 53), последовательно формируются два сегмента траектории. Первый сегмент соответствует равноускоренному движению с ускорением $\bar{a}^f \in (0, \bar{a}]$ из начального состояния (t_0, v_0, l_0) до достижения крейсерской скорости $v_{cruise} \in (0, v_{max}]$. Движение по этому сегменту траектории занимает $(v_{cruise} - v_0)/\bar{a}^f$ секунд. Второй сегмент траектории соответствует равномерному прямолинейному движению со скоростью v_{cruise} с момента времени $t_0 + (v_{cruise} - v_0)/\bar{a}^f$ до момента въезда непосредственно на перекресток. В результате ТС, двигаясь по рассматриваемому сегменту, въедет на перекресток в момент времени \hat{t}^+ , определяемый следующим образом:

$$\hat{t}^+(v_{cruise}, \bar{a}^f) = t_0 + \begin{cases} \frac{-v_0 - \sqrt{(v_0)^2 + 2\bar{a}^f L}}{\bar{a}^f}, & \text{если } L \leq \frac{v_{cruise}^2 - (v_0)^2}{2\bar{a}^f}, \\ \frac{L}{v_{cruise}} + \frac{(v_{cruise} - v_0)^2}{2\bar{a}^f v_{cruise}}, & \text{иначе.} \end{cases}$$

Полученная траектория tr^f , состоящая из двух сегментов, является результирующей, если в момент времени \hat{t}^+ для полосы движения, по которой движется рассматриваемое ТС, горит зеленый сигнал светофора. В противном случае при $G(T(tr^f, L)) \neq T(tr^f, L)$ начинается второй этап алгоритма – Backward Shooting Process (BSP).

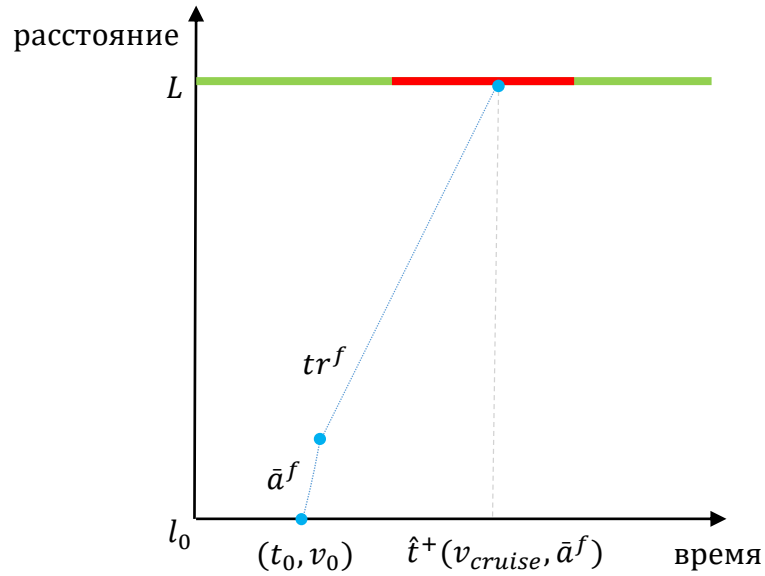


Рисунок 53 – Первый этап алгоритма построения траектории – Forward Shooting Process

В начале второго этапа алгоритма последний сегмент траектории tr^f сдвигается вправо по оси времени до начала зеленой фазы светофора $G(\hat{t}^+)$. Полученный в результате такого сдвига сегмент является исходным для этапа BSP. Затем из этого исходного сегмента формируются сегменты траектории tr^b с ускорением $\bar{a}^b \in (0, \bar{a}]$ и торможением $\underline{a}^b \in [\underline{a}, 0)$, благодаря которым траектории tr^b и tr^f плавно соединяются в результирующую траекторию tr (рисунок 54).

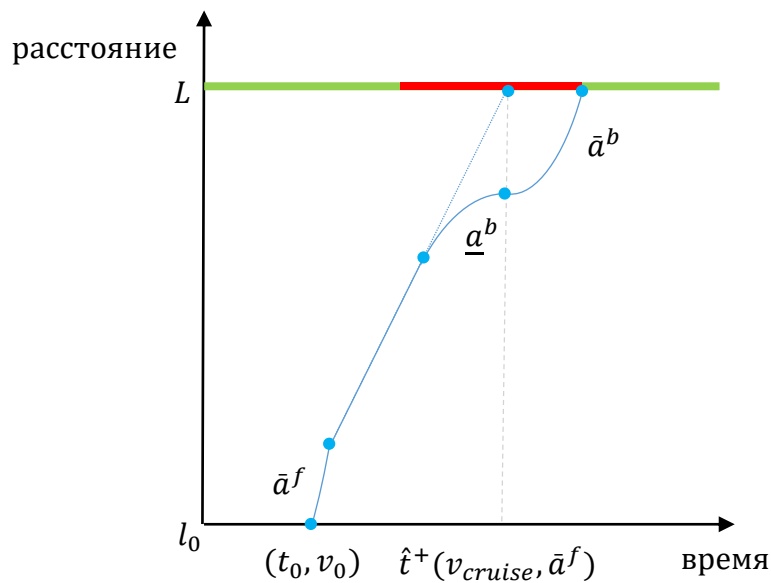


Рисунок 54 – Второй этап алгоритма построения траектории – Backward Shooting Process

Таким образом, используемая в работе модификация Shooting Heuristic алгоритма последовательно формирует траектории для ТС, расположенных наиболее близко к перекрестку на каждой входящей дорожной полосе. Набор параметров $(\bar{a}^f, \bar{a}^b, \underline{a}^b, v_{cruise})$, используемый для построения траектории, определяет ее «гладкость». Следует отметить, что при заданном наборе входных параметров траектории $(\bar{a}^f, \bar{a}^b, \underline{a}^b, v_{cruise})$ и исходных данных положения ТС (t_0, v_0, l_0) не всегда представляется возможным построить траекторию с помощью алгоритма Shooting Heuristic.

Пример построения траекторий с помощью Shooting Heuristic алгоритма представлен на рисунке 55. На данном рисунке показаны две пространственно-временные диаграммы, содержащие траектории движения транспортных средств на регулируемом перекрестке, контролируемых вручную (а), и транспортных средств, управляемых с помощью Shooting Heuristic алгоритма (б).

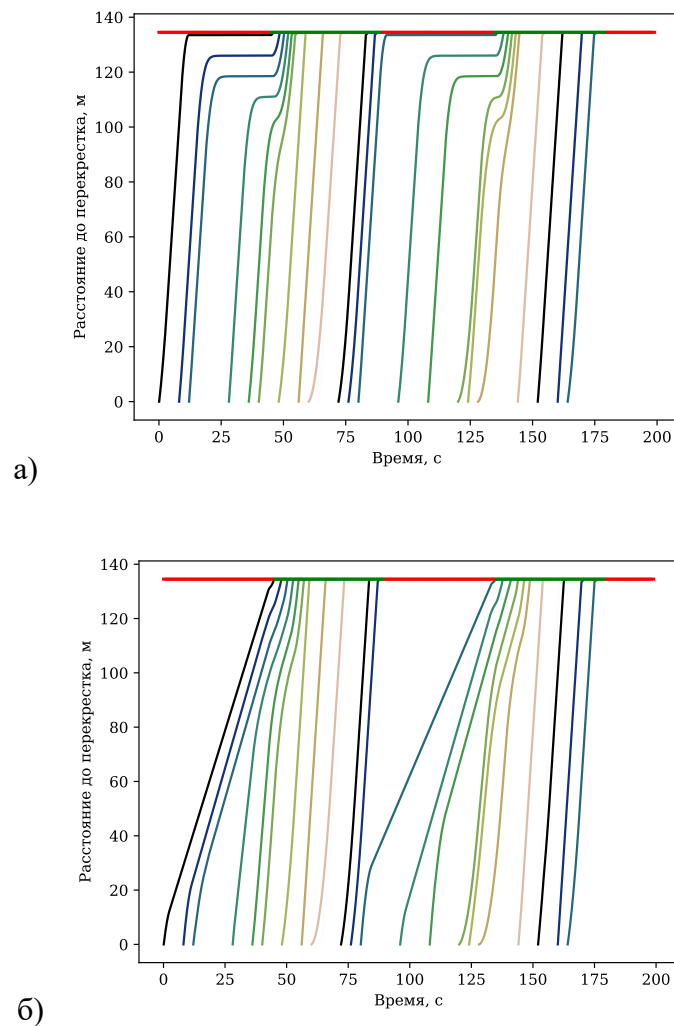


Рисунок 55 – Пространственно-временные диаграммы движения управляемых вручную транспортных средств (а) и с помощью Shooting Heuristic алгоритма (б)

Полученную с помощью Shooting Heuristic алгоритма траекторию можно оценить по различным показателям эффективности дорожного движения. В рамках предложенного в данной работе метода была использована взвешенная сумма нескольких критериев оценки траектории:

$$M(tr) = \lambda \cdot R_T(TravelT(tr)) + \eta \cdot R_T(WaitingT(tr)) + \delta \cdot R_F(FuelC(tr)),$$

где $TravelT(tr)$ – время (длительность) совершения корреспонденции ТС при движении по траектории tr ;

$WaitingT(tr)$ – время (длительность) ожидания/простоя ТС при движении по траектории tr ;

$FuelC(tr)$ – объем потребления топлива ТС при движении по траектории tr .

Скалярные функции R_T и R_F определены ранее, также как и весовые коэффициенты λ, η, δ . Значения последних были подобраны экспериментально: $\lambda = 1,0; \eta = 2,0; \delta = 1,0$. Поскольку в качестве единиц измерения времени движения ТС и времени ожидания ТС использовались секунды, в качестве единиц потребления топлива – миллилитры, чем меньше значение оценки $M(tr)$ для траектории tr , тем «лучше» траектория.

Для расчета потребления топлива была использована модель расчета потребления топлива, основанная на данных из базы данных The Handbook Emission Factors for Road Transport (HBEFA) версии 3.1 [346]. Данная модель используется по умолчанию в системе моделирования SUMO для расчета выбросов в атмосферу и потребления топлива транспортными средствами. Согласно модели потребление топлива транспортным средством стандарта Евро-4 с бензиновым двигателем, двигающимся со скоростью v (м/с) и ускорением a (м/с²) в момент времени $t(c)$ рассчитывается по следующей формуле [327]:

$$FuelC(tr(t)) = \begin{cases} 0, & \text{если } tr''(t) < 0, \\ \frac{3014 + tr'(t)(299,3tr''(t) - 149 + 9,014tr'(t))}{2671,2}, & \text{иначе.} \end{cases}$$

Для подбора оптимальных параметров траектории ($\bar{a}^f, \bar{a}^b, \underline{a}^b, v_{cruise}$) использовался субградиентный метод [12]. Задача оптимизации формулируется следующим образом:

$$M\left(\overline{a}^f, \overline{a}^b, \underline{a}^b, v_{cruise}\right) \rightarrow \min_{\overline{a}^f, \overline{a}^b, \underline{a}^b, v_{cruise}},$$

$$0 < \overline{a}^f \leq \overline{a}, 0 < \overline{a}^b \leq \overline{a},$$

$$\underline{a} \leq \underline{a}^b < 0, 0 \leq v_{cruise} \leq v_{max}$$

На первом этапе данного метода выбирается такой набор параметров $(\overline{a}^f, \overline{a}^b, \underline{a}^b, v_{cruise})$, при котором с помощью алгоритма Shooting Heuristic можно построить траекторию. Выбор начального набора параметров осуществляется путем перебора. Затем рассчитывается субградиент путем небольшого изменения текущих входных параметров используемого алгоритма построения траектории. Далее рассчитывается оценка траектории при изменении параметров в субградиентном направлении. Движение в направлении градиента выполняется до тех пор, пока не будет найдена точка решения, которая приводит к значительному уменьшению значения функции оценки M . Если такая точка найдена, вектор входных параметров изменяется путем перемещения в эту новую точку. В противном случае алгоритм может застрять в локальном оптимуме, поэтому в этом случае вектор входных параметров просто перемещается в ближайшую точку, даже если значение функции оценки не уменьшается. Этот процесс поиска вектора входных параметров повторяется до тех пор, пока не будут удовлетворены определенные конечные критерии (максимальное количество шагов при движении по градиенту). Во время этого процесса всякий раз, когда вычисляется оценка траектории в новой точке, оптимальное решение (в виде набора параметров $(\overline{a}^f, \overline{a}^b, \underline{a}^b, v_{cruise})$) обновляется в том случае, если оценка траектории в этой точке, меньшее, чем все предыдущие оценки.

Общая схема используемого в данной работе алгоритм построения оптимальной траектории движения ТС представлена на рисунке 5б.

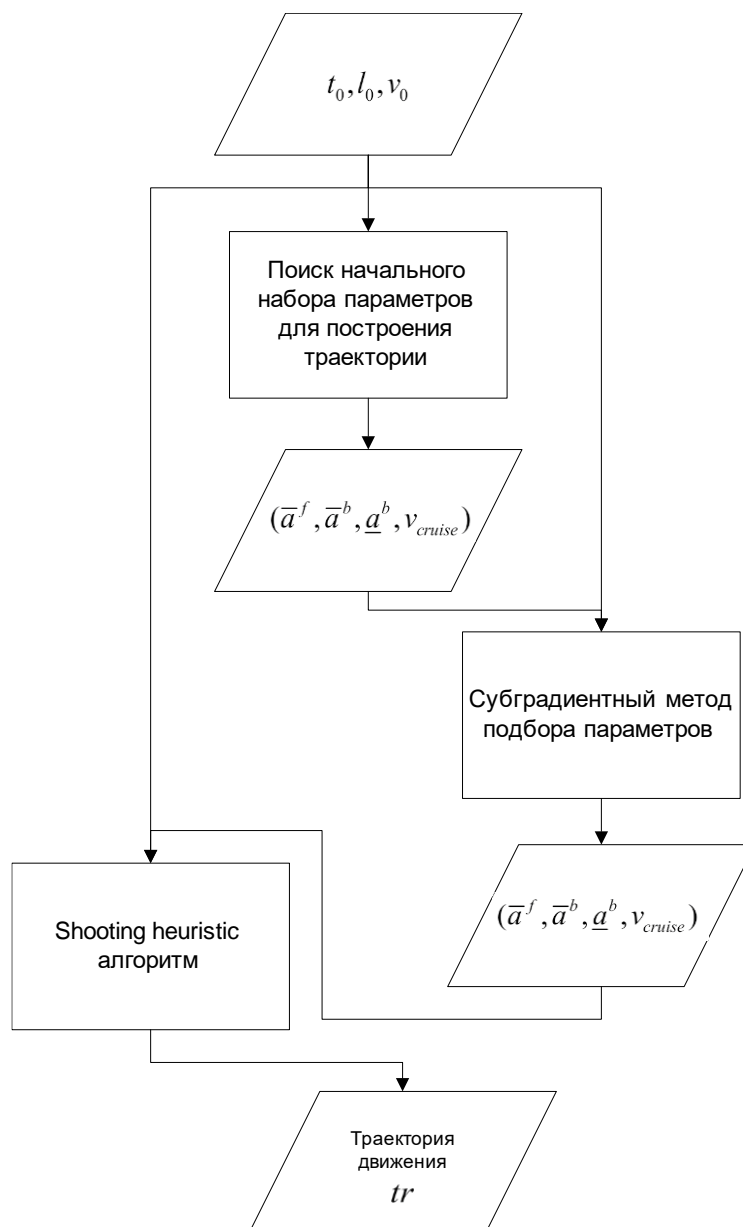


Рисунок 56 – Общая схема алгоритма построения оптимальной траектории движения ТС

3.4.1.3 Метод согласованного управления траекториями движения с учетом адаптивного управления сигналами светофоров

В рамках предлагаемого метода согласованного управления на перекрестке управление светофорным объектом осуществляется описанным в разделе 3.4.1.1 методом адаптивного управления. Траектории движения транспортных средств на входящих к перекрестку дорожных полосах формируются с помощью алгоритма, описанного в разделе 3.4.1.2 Однако, для согласованной работы вышеназванного метода и алгоритма, алгоритм построения траектории должен быть адаптирован для работы со светофорным объектом, контролируемым адаптивным методом.

Для возможности применения алгоритма 3.4.1.2 на перекрестке, управляемым светофорным объектом, работающем без фиксированного расписания, была изменена функция $G(t)$. Как было отмечено выше, эта функция для заданного момента времени t возвращает следующее ближайшее к нему время зеленой фазы светофора. Она используется в описанном ранее алгоритме Shooting Heuristic на этапе BSP. Так как длительность полного цикла работы светофора при использовании метода, описанного в разделе 3.4.1.1, непостоянно, использовать определение функции (48) не представляется возможным.

Пусть p – текущая активная фаза светофора для входящей в перекресток полосы движения; p_{next} – следующая прогнозируемая фаза светофора; τ_{min} – минимальный интервал переключения фаз; T_o^p – время начала текущей фазы, $T_o^{p_{next}} = T_o^p + \tau_{min}$ – время переключения фазы; $c(p, l) \in \{green, yellow, red\}$ – сигнал светофоры фазы p для полосы движения l . Тогда, по аналогии с $G(t)$, определим функцию $G(t, p, l)$ следующим образом:

$$G(t, p, l) = \begin{cases} t, & \text{если } c(p, l) = green \wedge (c(p_{next}, l) = green \vee t < T_o^{p_{next}}), \\ \max\{t, T_o^{p_{next}}\}, & \text{если } c(p, l) \neq green \wedge c(p_{next}, l) = green, \\ \max\{t, T_o^{p_{next}} + \tau_{min}\}, & \text{иначе.} \end{cases}$$

Согласно используемому методу адаптивного управления сигналами светофорного объекта, выбор и установка следующей фазы осуществляется через фиксированный временной интервал τ_{min} по формуле:

$$p_{next} = \operatorname{argmax}(\{PWFlow(p) \text{ for } p \text{ in } P\}). \quad (49)$$

Значение фазы p_{next} является «прогнозируемым», так как построение траектории может осуществляется в те моменты времени, когда установка следующей фазы светофора не выполняется.

Для согласованного управления транспортным потоком на перекрестке предлагается обновлять сформированные траектории движения ТС, при изменении результата прогноза следующей фазы светофора. В результате построения траектории сохраняется не только непосредственно построенная траектория для ТС, но и результат прогноза следующей фазы светофора, при котором эта траектория была получена. Обновление траекторий движения ТС осуществляется с заданной периодичностью $\tau_{upd} < \tau_{min}$.

Обновление траекторий осуществляется следующим образом. Сначала по формуле (49) вычисляется прогнозируемая фаза светофорного объекта p_{next} для момента

следующего переключения его фазы. Затем для построенной ранее траектории ТС выполняется проверка, изменился ли прогноз следующей фазы. Тогда, если прогноз для ТС на дорожной полосе l поменялся ($c(p, l) \neq c(p_{next}, l)$), то траектория перестраивается (с использованием алгоритма построения траектории, описанного в разделе 3.4.1.2 – функция $build_traj()$) с учетом обновленного прогноза для следующей фазы. В противном случае, траектория движения ТС остается неизменной. С учетом сказанного выше псевдокод процедуры обновления траектории ТС выглядит следующим образом:

Алгоритм 5: Процедура обновления траектории

```

1: Входные данные:  $l, p, tr, P$ 
2: Выходные данные:  $tr, p_{next}$ 
3:  $p_{next} = \text{argmax}(\{PWFlow(p) \text{ for } p \text{ in } P\})$ 
4: if  $c(p, l) \neq c(p_{next}, l)$  then
5:    $tr = build\_traj(p_{next})$ 
6: end if

```

Экспериментальные исследования метода согласованного управления траекториями движения транспортных средств на перекрестке, использующего информацию о прогнозной фазе светофора, рассчитанной с использованием метода адаптивного управления на основе максимального взвешенного потока, представлены в разделе 3.4.3.1.

В разделе 3.4.1 рассматривался метод управления траекториями движения транспортных средств на перекрестке, использующий информацию о прогнозной фазе светофора, рассчитанной с использованием метода адаптивного управления на основе максимального взвешенного потока (раздел 3.3.2). В разделе 3.4.2 представлена модификация метода, заключающаяся в совместном применении метода управления траекториями движения транспортных средств и метода адаптивного управления сигналами светофоров.

3.4.2 Метод совместного управления траекториями движения транспортных средств и адаптивного управления сигналами светофоров.

Предложенный в предыдущем разделе метод построения траектории движения транспортных средств, согласованной с адаптивным управлением сигналов светофора, использует информацию о прогнозной фазе. При этом траектория движения не учитывается при оценке транспортного потока в методе адаптивного управления. В данном разделе предложена модификация метода, заключающаяся в совместном применении метода управления траекториями движения транспортных средств и метода адаптивного управления сигналами светофоров.

Пусть $tr_c(t)$ – траектория движения транспортного средства c , L – длина полосы движения, $T(tr_c, L)$ – время, при котором ТС c достигнет позиции L (т.е. въедет на перекресток), двигаясь по траектории tr_c .

Модифицируем функцию расчета взвешенного транспортного потока (45), используемую в методе адаптивного управления (Алгоритм 4), следующим образом:

$$PWFlow(p) = \sum_{l \in L_p^{income}} \sum_{c \in C_l} \eta(c, l) I(T_o^{pnext} \leq t(c) < T_o^{pnext} + \tau_{min}), \quad (50)$$

где L_p^{income} – множество входящих полос движения на перекрёстке, для которых допустимо движение при фазе p ;

C_l – множество транспортных средств на полосе l ;

T_o^{pnext} – время переключения фаз;

$t(c)$ – время, когда транспортное средство $c \in C_l$ будет пересекать перекресток;

$\eta(c, l)$ коэффициент, учитывающий время ожидания транспортного средства c на полосе l ;

$I(val)$ – функция индикации:

$$I(val) = \begin{cases} 1, & \text{если } val = True, \\ 0, & \text{иначе.} \end{cases} \quad (51)$$

Коэффициент $\eta(c, l)$ определяется следующим образом:

$$\eta(c, l) = 1 + \alpha \cdot delay(c, l), \quad (52)$$

где $delay(c, l)$ – время ожидания транспортного средства c на полосе l , α – эмпирически выбираемый коэффициент.

Функция $t(c)$ определяется следующим образом:

$$t(c) = \begin{cases} T(tr_c, L), & \text{если } tr_c \text{ определена,} \\ t^{dnn}(c), & \text{иначе.} \end{cases} \quad (53)$$

Таким образом, для оценки времени прохождения перекрестка $t(c)$ используются два подхода:

1) для транспортных средств с известной (построенной) траекторией движения время прохождения перекрестка рассчитывается именно по траектории движения, так как она определяет скорость транспортного средства в каждый момент времени;

2) для остальных транспортных средств время прохождения перекрестка оценивается с использованием модели прогнозирования, основанной на модели глубокой нейронной сети, описанной в разделе 3.3.2.

В работе выполняется построение траектории только для ведущих транспортных средств на полосах движения, ближайших к регулируемым перекресткам. Другие транспортные средства управляются с помощью модели следования за автомобилем. Во-первых, такой подход позволяет значительно сократить вычислительные ресурсы. Во-вторых, учет различных типов транспортных средств позволяет применять предложенный метод в условиях гетерогенного транспортного потока с участием автономных транспортных средств и транспортных средств, управляемых человеком.

Таким образом, предлагаемый метод совместного управления сочетает в себе адаптивный метод управления сигналами светофора, основанный на максимизации взвешенного транспортного потока (описанный в разделе 3.3.2) и алгоритм построения траектории движения подключенных транспортных средств (описанный в разделе 3.4.1).

Обозначим $c(p, l) \in \{green, yellow, red\}$ – сигнал светофора фазы p для полосы движения l . Метод совместного управления можно описать следующим образом:

1) построить траектории для всех ведущих транспортных средств на каждой полосе l , предполагая, что для всех полос $c(p_{next}, l) = green$ (т.е. для всех транспортных средств движение разрешено);

2) рассчитать оценку времени прохождения перекрестка $t(c)$: для транспортных средств с известной траекторией движения $t(c)$ рассчитывается на основе построенной траектории, для остальных транспортных средств $t(c)$ рассчитывается с использованием алгоритма прогнозирования на основе глубокой нейронной сети, описанного в разделе 3.3.2;

3) выбрать следующую фазу p_{next} , используя алгоритм адаптивного управления светофором MaxPWFlow (Алгоритм 4): рассчитать взвешенный транспортный поток, используя формулу (50), и выбрать следующую фазу, максимизирующую рассчитанный поток;

4) учитывая прогнозную следующую фазу p_{next} , перестроить траектории для всех транспортных средств, для которых предположение $c(p_{next}, l) = green$ не выполняется.

В следующем разделе представлены экспериментальные исследования разработанных методов согласованного и совместного управления траекториями движения транспортных средств и адаптивного управления сигналами светофоров.

3.4.3 Экспериментальные исследования методов совместного управления

Экспериментальные исследования разработанных методов включают в себя оценку эффективности метода согласованного управления траекториями движения транспортных средств с учетом адаптивного управления сигналами светофоров (раздел 3.4.3.1) и метода совместного управления траекториями движения транспортных средств и адаптивного управления сигналами светофоров (раздел 3.4.3.2).

3.4.3.1 Метод управления траекториями движения с учетом адаптивного управления сигналами светофоров

Экспериментальные исследования эффективности разработанного метода согласованного управления траекториями движения транспортных средств с учетом адаптивного управления сигналами светофоров проводились с использованием системы моделирования SUMO (Simulation of Urban Mobility) [327].

В экспериментах проводилось сравнение метода согласованного управления сигналами траекториями движения транспортных средств с методом адаптивного управления сигналами светофоров MaxPWFlow (без управления движением транспортных средств, Алгоритм 4). Разработка алгоритмов и методов совместного управления является областью активных исследований. Существующие работы обладают существенными ограничениями: используют predetermined цикл фаз [186] или не рассматривают перестроение между полосами в процессе движения [187], а также не предоставляют исходный код и данные для сравнения. Эти особенности делают невозможным сравнение разработанного метода с существующими алгоритмами и методами того же класса.

Экспериментальные исследования проводились при шаге симуляции, равном 0,1 секунде, и при общем времени симуляции 3600 секунд. Для оценки эффективности разработанного метода использовались три сценария моделирования движения транспортных средств:

- 1) изолированный перекресток, являющийся пересечением равнозначных однополосных дорог длиной 150 метров. Количество транспортных средств – 900;
- 2) синтетическая сеть «4x4», состоящая из 16 перекрестков одинаковой структуры: пересечение равнозначных дорог длиной 300 метров с 3 полосами движения. Количество транспортных средств – 1472;
- 3) городская область, являющаяся частью крупномасштабного сценария моделирования «TAPAS Cologne» [339]. Сценарий содержит 8 регулируемых перекрестков различной топологии, количество транспортных средств – 2046.

Дорожные сети рассматриваемых сценариев представлены на рисунке 57.

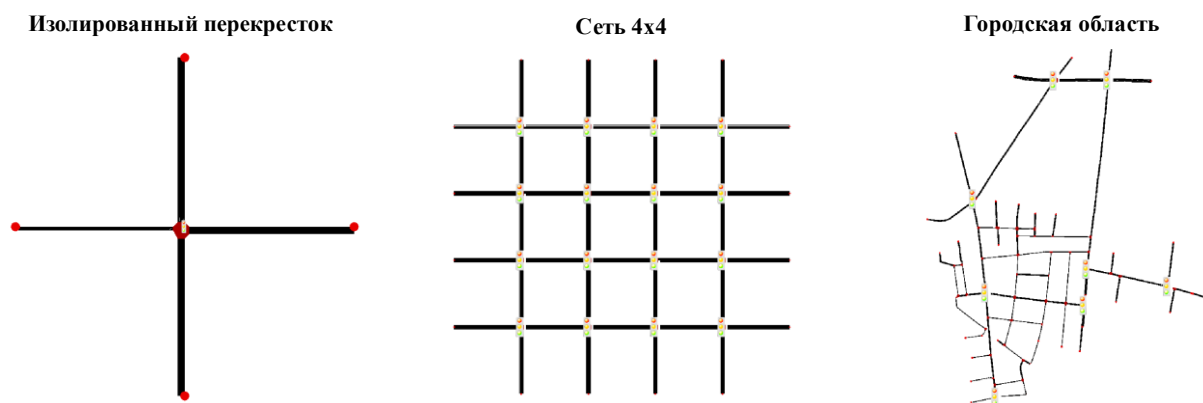


Рисунок 57 – Сценарии моделирования движения ТС

В ходе проведения экспериментальных исследований все транспортные средства имели одинаковые характеристики, бензиновый двигатель и соответствовали одному экологическому стандарту – Евро-4.

В экспериментах использовались следующие значения параметров: минимальный интервал переключения фаз $\tau_{min} = 10$ (секунд), интервал обновления траектории движения транспортного средства $\tau_{upd} = 5$ (секунд).

Сравнение методов проводилось по трем критериям:

- 1) средний расход топлива на одно транспортное средство (основной критерий) (в миллилитрах), позволяющий оценить потребление топлива всеми транспортными средствами в сценарии;
- 2) среднее время движения одного транспортного средства (в секундах), затрачиваемое транспортными средствами на совершение поездок в сети;
- 3) среднее время ожидания одного транспортного средства (в секундах), т.е. время, которое транспортное средство провело без движения на перекрестке.

Для оценки показателей качества методов для каждого сценария было проведено 10 запусков моделирования движения транспортных средств (10 эпизодов), полученные результаты были усреднены. Начальные положения транспортных средств, время начала движения и маршруты были различными в различных эпизодах одного сценария. Результаты сравнения методов по критерию «расход топлива» представлены в таблице 15, по критерию «среднее время движения» – в таблице 16, по критерию «среднее время ожидания» – в таблице 17.

Таблица 15 – Сравнение методов по критерию «средний расход топлива», мл

	Изолированный перекресток	Сеть 4x4	Городская область
MaxPWFlow	35,45	115,47	87,92
Согласованное управление	30,56	111,98	86,47

Таблица 16 – Сравнение методов по критерию «среднее время движения», с

	Изолированный перекресток	Сеть 4x4	Городская область
MaxPWFlow	27,07	104,81	88,84
Согласованное управление	25,56	103,4	87,23

Таблица 17 – Сравнение методов по критерию «среднее время ожидания», с

	Изолированный перекресток	Сеть 4x4	Городская область
MaxPWFlow	2,01	3,61	3,57
Согласованное управление	0,04	1,16	1,44

Проведенные эксперименты подтверждают эффективность предложенного метода совместного управления сигналами светофоров и траекториями движения транспортных средств по всем рассмотренным критериям.

На рисунке 58 представлен расход топлива для каждого эпизода моделирования в сценарии «городская область».

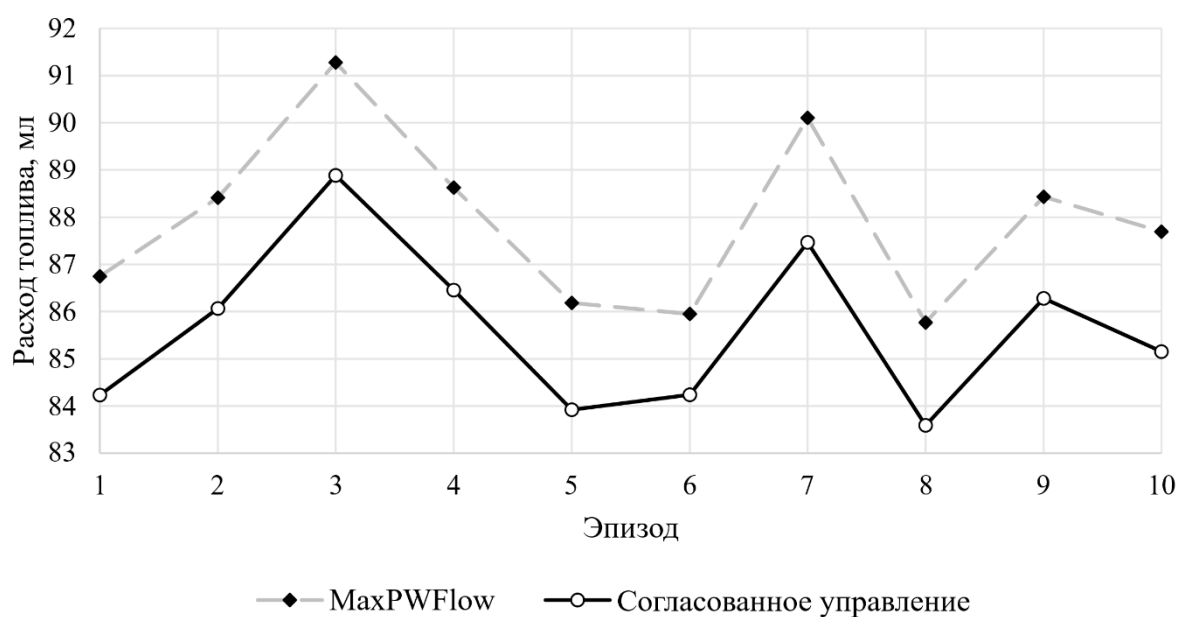


Рисунок 58 – Расход топлива в каждом эпизоде моделирования движения

Как видно из графика, предложенный метод позволил снизить потребление топлива в каждом из эпизодов рассматриваемого сценария.

Основываясь на результатах проведенных экспериментов, можно сделать вывод, что предложенный в работе метод согласованного управления позволяет повысить эффективность использования транспортной инфраструктуры, снизить расход топлива и время движения.

3.4.3.2 Метод совместного управления траекториями движения транспортных средств и адаптивного управления сигналами светофоров

На втором этапе экспериментального анализа исследовался модифицированный метод совместного управления траекториями движения транспортных средств и адаптивного управления сигналами светофоров. Экспериментальные исследования метода проводились в трех сценариях моделирования, созданных на основе сценария SUMO «TAPAS Cologne» [340]:

- 1) сценарий моделирования движения на автомагистрали «Cologne-3» [347];
- 2) сценарий моделирования в области транспортной сети малого размера «Cologne-8» [347];
- 3) крупномасштабный сценарий моделирования движения транспортных средств «Cologne-316» [135].

Основные параметры сценариев представлены в таблице 18.

Таблица 18 – Параметры сценариев моделирования

Сценарий	Количество светофоров	Количество перекрестков	Количество сегментов	Количество маршрутов
«Cologne-3»	3	29	48	2830
«Cologne-8»	8	78	149	1740
«Cologne-316»	316	2928	5808	13530

На рисунке 59 изображены дорожные сети описанных сценариев.

Как и в предыдущих экспериментальных исследованиях, для оценки показателей качества методов для каждого сценария было проведено 10 запусков моделирования движения транспортных средств (10 эпизодов), полученные результаты были усреднены. Начальные положения транспортных средств, время начала движения и маршруты были различными в различных эпизодах одного сценария.



Рисунок 59 – Дорожные сети рассматриваемых сценариев моделирования:
 а) «Cologne-316»; б) «Cologne-8»; в) «Cologne-3»

Сравнение методов проводилось по трем критериям: средний расход топлива, среднее время движения и среднее время ожидания, подробнее описанных в разделе 3.4.3.1.

В экспериментальном исследовании сравниваются предлагаемый метод совместного управления с несовместными стратегиями адаптивного управления светофором и

согласованными подходами к построению траектории движения транспортных средств, учитывающими адаптивную фазу светофоров при построении траектории. А именно, сравниваются следующие методы и алгоритмы:

1) IDQN: независимый алгоритм глубокого Q-обучения, основанный на сверточной нейронной сети, каждый перекресток контролируется одним агентом RL [347];

2) IPPO: алгоритм обучения с подкреплением для оптимизации политики [347];

3) A2C: алгоритм Advantage Actor-Critic [344];

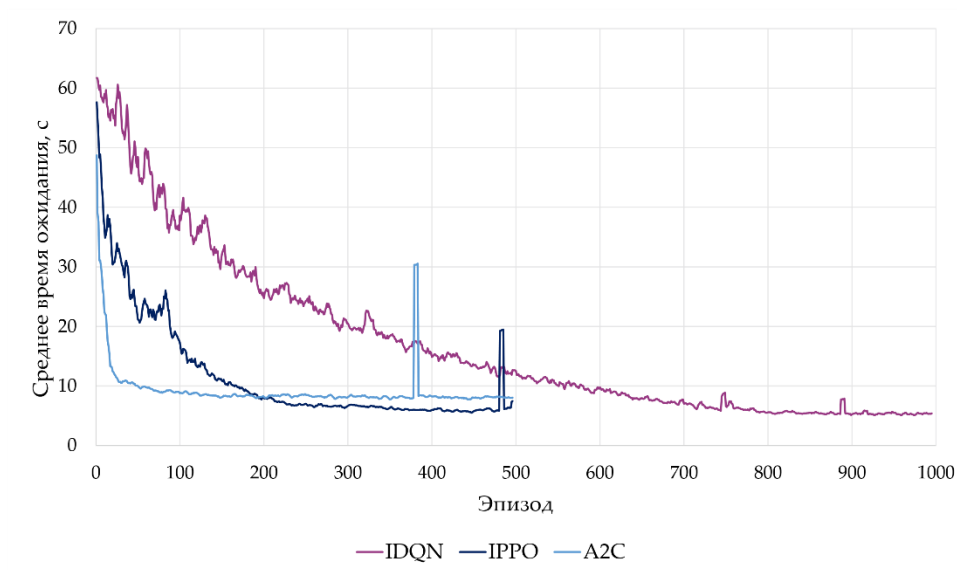
4) MaxPWFlow: адаптивный метод управления, основанный на максимизации взвешенного транспортного потока (Алгоритм 4);

5) Trajectory Control: согласованный алгоритм управления траекториями транспортных средств с учетом адаптивного управления сигналами светофоров методом MaxPWFlow;

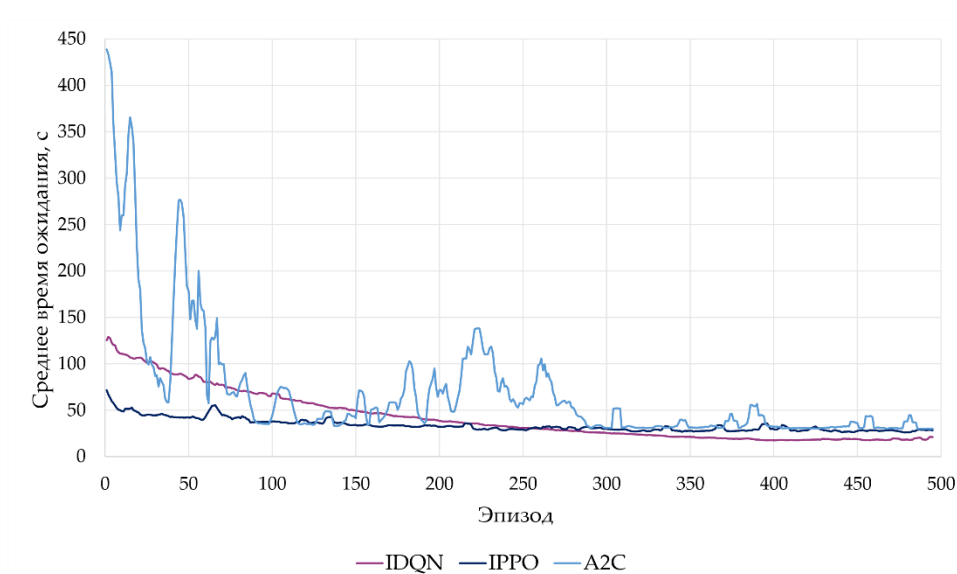
6) Trajectory Control + RL: согласованный алгоритм управления траекториями транспортных средств с учетом адаптивного управления сигналами светофоров алгоритмом IDQN;

7) Cooperative Control: разработанный метод совместного управления траекториями движения транспортных средств и адаптивного управления сигналами светофоров.

На первом этапе экспериментального исследования проводилась оценка сходимости процесса обучения RL-алгоритмов путем сравнения среднего времени ожидания в каждом эпизоде моделирования. На рисунке 60 показаны кривые обучения для каждого алгоритма в сценариях «Cologne-8» и «Cologne-316». Согласно представленным на рисунке 60 графикам, алгоритму IDQN требуется больше эпизодов для достижения стабильного значения среднего времени ожидания.



а)



б)

Рисунок 60 – Кривые обучения алгоритмов RL: (а) сценарий "Cologne-8"; (б) "Cologne-316" scenario

Далее исследовалась эффективность алгоритмов по выбранным критериям. В таблице 19 сравнивается средний расход топлива для каждого сценария. Таблица показывает среднее значение и стандартное отклонение, усредненное по десяти эпизодам моделирования движения транспортных средств.

Таблица 19 – Средний расход топлива (мл) для различных сценариев моделирования

Модель \ Сценарий	«Cologne-3»	«Cologne-8»	«Cologne-316»
IDQN	64,24 ± 0,84	88,51 ± 1,76	334,74 ± 3,37
IPPO	64,42 ± 0,88	88,52 ± 1,74	416,93 ± 8,85
A2C	66 ± 0,6	93,68 ± 1,75	355,62 ± 9,36
MaxPWFlow	62,15 ± 0,4	86,48 ± 1,77	328,62 ± 1,81
Trajectory Control	60,55 ± 0,46	84,42 ± 1,58	331,76 ± 1,75
Trajectory Control + RL	61,76 ± 0,43	86,52 ± 1,6	333,82 ± 1,7
Cooperative Control	59,57 ± 0,43	83,41 ± 1,5	325,25 ± 1,71

Как показано в таблице, предложенный метод совместного управления позволяет снизить расход топлива по сравнению с базовыми методами в каждом сценарии. Среднее снижение расхода топлива составило от 1% для сценария «Cologne-316» до 4,2% для сценария «Cologne-3» по сравнению с адаптивным управлением сигналами светофоров.

На следующем этапе оценивалось среднее время движения и среднее время ожидания. Результаты сравнения методов по указанным критериям представлены в таблицах 20 и 21 соответственно.

Таблица 20 – Среднее время движения (с) для различных сценариев моделирования

Модель \ Сценарий	«Cologne-3»	«Cologne-8»	«Cologne-316»
IDQN	57,9 ± 1,08	89,89 ± 2,07	332,15 ± 3,19
IPPO	58,25 ± 1,01	89,51 ± 1,98	406,94 ± 7,92
A2C	60,57 ± 0,9	95,15 ± 2,09	350,28 ± 8,32
MaxPWFlow	55,01 ± 0,57	87,69 ± 2,03	327,08 ± 1,85
Trajectory Control	53,48 ± 0,7	85,57 ± 1,88	328,68 ± 1,72
Trajectory Control + RL	54,5 ± 0,52	88,11 ± 1,83	331,83 ± 1,85
Cooperative Control	52,12 ± 0,44	84,32 ± 1,89	323,96 ± 1,95

Таблица 21 – Среднее время ожидания (с) для различных сценариев моделирования

Модель \ Сценарий	«Cologne-3»	«Cologne-8»	«Cologne-316»
IDQN	7,49 ± 0,82	4,46 ± 0,28	18,2 ± 4,52
IPPO	7,57 ± 0,81	4,01 ± 0,14	101,45 ± 9,24
A2C	8,9 ± 0,67	7,44 ± 0,31	29,12 ± 9,41
MaxPWFlow	6,08 ± 0,36	3,18 ± 0,15	14,79 ± 0,75
TrajectoryControl	3,65 ± 0,4	0,71 ± 0,18	11,15 ± 0,67
Trajectory Control + RL	3,27 ± 0,34	0,86 ± 0,1	12,52 ± 1,46
Cooperative Control	3,38 ± 0,37	0,62 ± 0,07	10,76 ± 0,87

Как видно из представленных результатов, метод совместного управления показывает наилучшие результаты в восьми из девяти случаев. Согласованный подход «Trajectory Control + RL» позволяет незначительно уменьшить средние задержки остановок, но на 0,1 секунды и только для сценария «Cologne-3».

Оценка эффективности методов по критерию среднего времени движения показывает, что предложенный метод позволяет сократить время от 1% для сценария «Cologne-316» до 5,3% для сценария «Cologne-3» по сравнению с алгоритмом MaxPWFloW. Оценка среднего времени ожидания показывает, что метод совместного управления позволяет сократить время ожидания до 27% по сравнению с алгоритмом MaxPWFloW в крупномасштабном сценарии «Cologne-316».

Полученные результаты позволяют сделать вывод, что наилучший результат среди алгоритмов на основе обучения с подкреплением показал алгоритм IDQN на основе оптимизации функции полезности, однако для обучения этого алгоритма требуется больше эпизодов, как видно из графиков сходимости на рисунке 60. Кроме того, RL-алгоритмы на основе обучения с подкреплением имеют более высокую дисперсию, что косвенно указывает на то, что эти алгоритмы обеспечивают менее надежные результаты, чем разработанный алгоритм адаптивного управления на основе максимизации взвешенного транспортного потока.

На финальном этапе экспериментального исследования более подробно анализируются показатели эффективности. На рисунке 61 представлены результаты экспериментальных исследований разработанного метода отдельно для каждого эпизода моделирования для сценария «Cologne-316» по критериям среднего расхода топлива и среднего времени движения транспортных средств в сценарии. Из графиков видно, что предложенный метод работает лучше в каждом эпизоде рассматриваемого сценария.

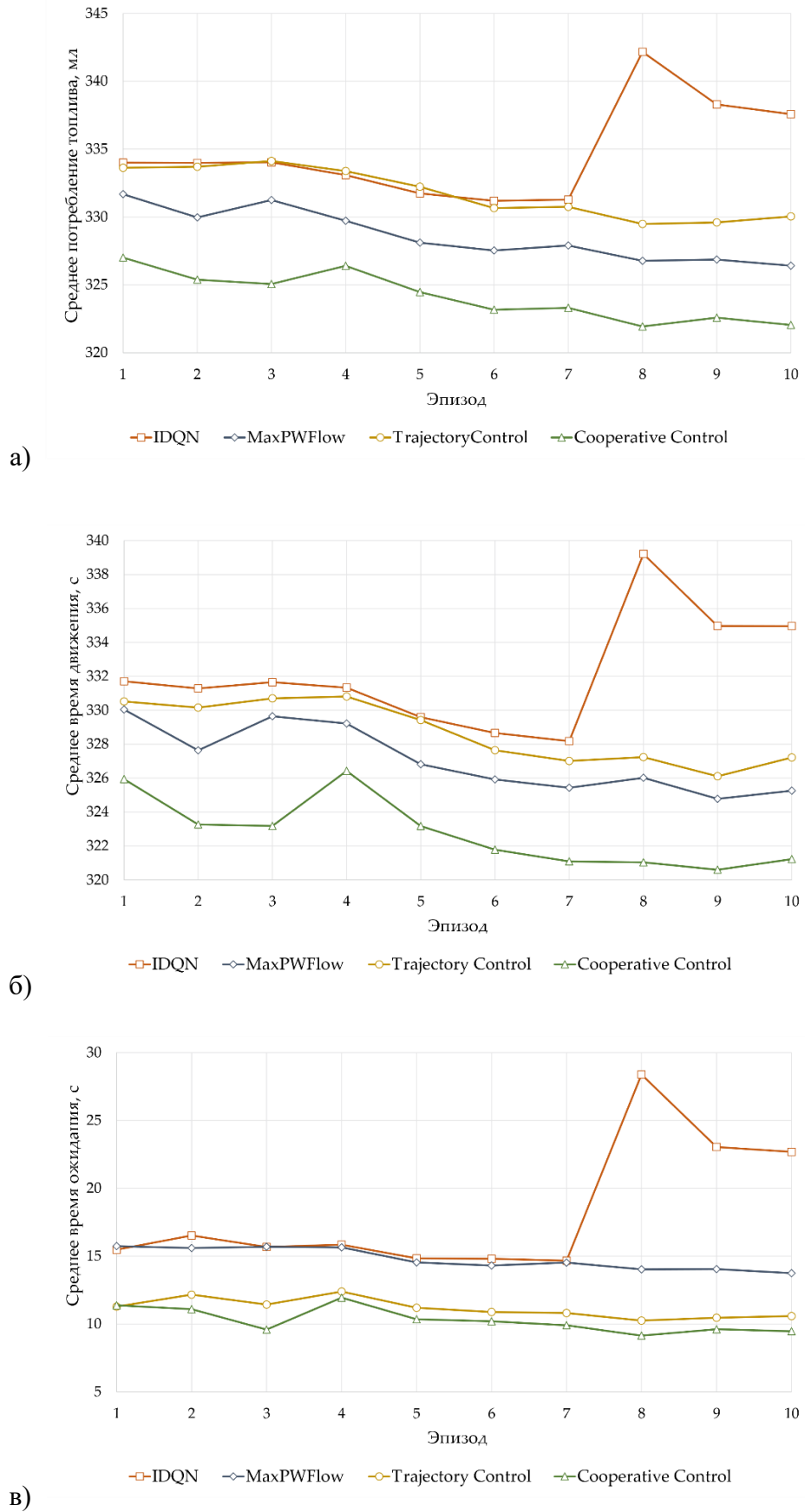


Рисунок 61 – Сравнение эффективности методов для сценария «Cologne-316»: (а) средний расход топлива; (б) среднее время движения; (в) среднее время ожидания

3.5 Выводы и результаты третьего раздела

В рамках решения задачи директивного управления транспортным потоком в интеллектуальной транспортной системе получены следующие результаты:

1) проведена формализация задачи директивного управления транспортным потоком с гетерогенным составом транспортных средств, определены информационные потоки и состав гетерогенной информации, используемой для решения задачи управления.

2) предложен комплекс алгоритмических средств (математические методы и алгоритмы) решения задачи директивного управления транспортным потоком в различных условиях:

а) большой объем данных для обучения искусственной нейронной сети: алгоритм адаптивного управления сигналами светофорных объектов на основе метода обучения с подкреплением, оптимизирующего функцию полезности, учитывающий как наблюдаемое, так и прогнозное состояние транспортного потока для описания пространства состояний (п. 3.2);

б) информация о транспортном потоке на перекрестке: метод адаптивного управления сигналами светофорных объектов на основе максимизации взвешенного потока транспортных средств, проходящих через перекресток за выбранную фазу светофорного цикла (п.3.3), алгоритмы оценки прогнозируемого транспортного потока на основе детерминированной модели прогнозирования движения транспортных средств (не требующей настройки/обучения) (п. 3.3.1) и на основе обучаемой модели глубокой нейронной сети регрессионного вида (п. 3.3.2);

в) переход к потоку, включающему подключенные автономные транспортные средства: метод совместного управления траекториями движения подключенных транспортных средств и адаптивного управления сигналами светофорных объектов, применимый, в т.ч., в среде с гетерогенным транспортным потоком, включающим подключенные и управляемые водителями транспортные средства (п. 3.4.2);

3) проведено экспериментальное исследование разработанных методов и алгоритмов в системе микроскопического моделирования движения транспортных средств в различных синтетических и реальных сценариях моделирования;

4) экспериментальные исследования в среде моделирования показывают, что предложенные методы позволяют снизить средний расход топлива (до 4,2%), среднее время движения (до 5.3%) и среднее время ожидания (до 27%) по сравнению с современными методами решения задачи адаптивного управления светофорными объектами;

5) эффективность управления повышается как с увеличением доли подключенных транспортных средств в рассматриваемых сценариях моделирования, так и с применением последовательно методов адаптивного управления только сигналами светофорных объектов и метода совместного управления траекториями движения подключенных транспортных средств и адаптивного управления сигналами светофорных объектов.

По материалам данного раздела опубликованы работы [122, 135, 161, 188, 337, 348-361].

4 Алгоритмические средства решения задач косвенного управления транспортным потоком и информирования в ИТС

Внедрение подключенных и автономных транспортных средств позволяет повысить точность решения задач прогнозирования и управления транспортным потоком и повысить эффективность использования транспортной инфраструктуры путем внедрения интеллектуальных транспортных систем. Данные задачи были рассмотрены в разделах 1) и 3) диссертационной работы. В то же время, информация о прогнозных значениях параметров транспортного потока и способе управления потоком на регулируемом перекрестке может использоваться для решения сопутствующих задач косвенного управления и информирования, востребованных и понятных для конечного потребителя – участника дорожного движения. Среди таких задач можно выделить следующие:

- прогнозирование времени движения и времени прибытия отдельных транспортных средств с учетом информации о маршрутах движения транспортных средств, их положениях и скорости движения;

- построение маршрута движения отдельного транспортного средства (решение навигационной задачи) с учетом информации о текущем и прогнозном распределении транспортного потока в сети;

- перераспределение транспортных потоков путем маршрутизации всех транспортных средств или их отдельной части для повышения эффективности использования транспортной инфраструктуры, снижения загруженности, потребления топлива и т.д.

Отдельное внимание требует решение задач, посвященных обеспечению информационной безопасности коммуникаций в кооперативных ИТС.

Эти задачи последовательно решаются в данном разделе.

В разделе 4.1 рассматривается задача прогнозирования движения отдельных транспортных средств в контексте ИТС. В частности, в разделе 4.1.1 данная задача рассматривается в постановке прогнозирования времени прибытия общественных транспортных средств на остановочные пункты. Решение этой задачи необходимо как для управления движением и внесения своевременных корректировок диспетчерскими службами, так и для оповещения пассажиров о времени прибытия общественных транспортных средств на остановочные пункты. В разделе 4.1.2 рассматривается задача прогнозирования времени движения подключенных транспортных средств по маршруту с использованием информации о загрузке сегментов транспортной сети и учетом

прогнозного времени ожидания на регулируемых перекрестках, управляемых методом адаптивного светофорного регулирования.

Раздел 4.2 посвящен решению навигационной задачи, т.е. задачи построения маршрута движения отдельного транспортного средства с учетом информации о текущем и прогнозном распределении транспортного потока в сети. В работе рассматривается задача нахождения надежного пути в стохастической транспортной сети. Надежный путь учитывает дисперсию времени прохождения сегментов дорожной сети, что делает его более применимым для решения задач маршрутизации в транспортных сетях по сравнению со стандартными алгоритмами поиска кратчайшего пути, учитывающими только среднее время прохождения дорожных сегментов. В данной работе задача нахождения надежного пути рассматривается в следующей постановке: определить оптимальную стратегию навигации, максимизирующую вероятность прибытия в пункт назначения за выбранный бюджет поездки (в течение заранее определенного интервала времени). В работе предлагается использовать устойчивое распределение вероятностей для описания времени прохождения дорожных сегментов, что позволит заменить операцию вычисления свертки на пересчет параметров плотности распределения и значительно сократит время работы алгоритма. В разделе 4.2.1 представлена постановка задачи нахождения надежного пути в стохастической зависящей от времени транспортной сети и описание базового алгоритма решения задач. В разделе 4.2.2 предложен модифицированный алгоритм, учитывающий актуальную и прогнозную информацию для решения задачи навигации. Развитие этого алгоритма путем применения устойчивого распределения Леви для описания времени прохождения сегментов дорожной сети представлено в разделе 4.2.3. Показано, что использование устойчивых распределений позволяет перейти от операции вычисления свертки для определения надежности пути к пересчету параметров плотности распределения, что значительно сокращает время исполнения алгоритма.

В разделе 4.3 рассматривается задача перераспределения транспортных потоков путем маршрутизации всех транспортных средств или их отдельной части (для гетерогенного транспортного потока) с целью повышения эффективности использования транспортной инфраструктуры. В разделе рассматриваются методы и алгоритмы маршрутизации транспортных средств в детерминированных и стохастических зависящих от времени транспортных сетях на основе итеративной процедуры достижения транспортного равновесия в сети. В частности, были разработаны:

- алгоритм централизованного управления транспортными средствами, минимизирующая общее время поездок в транспортной сети;

- алгоритмы маршрутизации транспортных средств в детерминированной (раздел 4.3.1) или стохастической зависящей от времени транспортной сети (раздел 4.3.2) на основе численного метода резервирования маршрутов, позволяющие достичь транспортного равновесия в сети;

- алгоритмы маршрутизации транспортных средств в гетерогенном транспортном потоке с учетом адаптивного светофорного регулирования (раздел 4.2.3).

Наконец, в разделе 4.4 рассматривается задача обеспечения информационной безопасности коммуникаций в кооперативной ИТС. Для обеспечения информационной безопасности необходимо обеспечивать защиту как от внешних угроз, путем аутентификации участников коммуникаций, так и от внутренних угроз. При защите от внутренних угроз предполагается, что злоумышленником является аутентифицированное и авторизованное транспортное средство, поэтому одним из основных подходов в этом случае является обнаружение аномального поведения участников движения. В разделе 4.4.1 рассматривается задача аутентификации транспортных средств. Обнаружение аномального поведения транспортных средств в транспортной сети рассматривается в разделе 4.4.2. Частный случай движения транспортных средств в колонне рассматривается в разделе 4.4.3.

4.1 Прогнозирование движения отдельных транспортных средств

Задача прогнозирования времени движения отдельных транспортных средств является одной из наиболее востребованных задач для участников дорожного движения. Хотя задачи прогнозирования времени движения подключенных транспортных средств и общественных транспортных средств (автобусов, троллейбусов, трамваев) похожи, характер движения этих транспортных средств отличается (например, общественные транспортные средства часто двигаются со скоростью, меньшей, чем скорость транспортного потока, могут использовать выделенные полосы движения, останавливаются на остановочных пунктах и т.д.). Кроме того, для решения этих задач может использоваться различная гетерогенная информация о транспортной ситуации, прямо или косвенно влияющую на прогнозируемое время движения. В разделе 4.1.1 задача прогнозирования движения общественных транспортных средств рассматривается в постановке прогнозирования времени прибытия на остановочные пункты. Задача прогнозирования движения подключенных транспортных средств по маршруту рассматривается в разделе 4.1.2.

4.1.1 Прогнозирование движения общественного транспорта

Время прибытия общественного транспорта на остановочные пункты является ключевым аспектом системы информирования пассажиров. Предоставление точной информации о прибытии транспортных средств имеет важное значение для планирования поездок, так как способствует сокращению времени ожидания общественного транспорта и позволяет выбирать оптимальные альтернативные маршруты движения. В последнее время такая информация используется в том числе и в навигационных мобильных приложениях. В данном разделе для прогнозирования времени прибытия общественного транспорта на остановочные пункты предлагается использовать модель рекуррентной нейронной сети LSTM. Данная модель учитывает разнородную информацию о транспортной ситуации, прямо или косвенно влияющую на прогнозируемое время движения, включая статистические данные и данные реального времени о движении транспортных средств, а также прогнозируемую информацию о распределении транспортного потока в сети.

4.1.1.1 Модель на основе рекуррентной нейронной сети

Для решения задачи прогнозирования времени прибытия общественного транспорта на остановочные пункты предлагается использовать сеть глубокого обучения с долгой краткосрочной памятью (LSTM). Предлагаемая модель учитывает разнородную информацию, влияющую на величину прогноза, а не только данные о прохождении сегментов транспортной сети в реальном времени.

Постановка задачи

Введем следующие обозначения. Пусть S – множество остановок, R – множество автобусных маршрутов. Задача прогнозирования времени прибытия автобуса может быть сформулирована следующим образом:

$$t_j^{arr} = t_i^{dep} + T_{ij}^{travel} \quad (54)$$

где t_j^{arr} – время прибытия на остановку $j \in S$;

t_i^{dep} – время отправления с остановки $i \in S$;

T_{ij}^{travel} – время прохождения сегмента транспортной сети между остановками i и j .

Тогда прогноз времени прибытия автобуса t_j^{arr} на заданную остановку j эквивалентен прогнозу времени прохождения T_{ij}^{travel} автобусом сегмента транспортной сети между остановками i и j .

Факторы прогнозирования

Пусть рассматриваемое транспортное средство движется по маршруту $r \in R$.

Для оценки времени прохождения сегмента T_{ij}^{travel} между остановками i и j могут быть использованы различные факторы, которые прямо или косвенно влияют на целевое значение оценки прогноза. В данном разделе предлагается использовать следующую разнородную информацию, описывающую транспортную ситуацию:

- 1) день недели day . Этот фактор используется для разделения рабочих и выходных дней;
- 2) время суток $time$ в минутах с полуночи;
- 3) скорость прохождения $v_{i-1,i}$ предыдущего сегмента. Это значение может показывать степень загруженности дорожных сегментов на данном маршруте;
- 4) интервал времени h^r между предыдущим автобусом того же маршрута r и рассматриваемым автобусом;
- 5) время прохождения сегмента $T_{ij}^{m,r}$ предыдущим автобусом m того же маршрута r ;
- б) взвешенное время прохождения сегмента \hat{T}_{ij}^r предыдущими автобусами того же маршрута r :

$$\hat{T}_{ij}^r = \frac{\sum_{k \in N_r} \omega(t - t_i^{dep,k}) T_{ij}^{travel,k}}{\sum_{k \in N_r} \omega(t - t_i^{dep,k})},$$

где t – текущее время;

$t_i^{dep,k}$ – время отправления автобуса k от остановки i ;

$T_{ij}^{travel,k}$ – время прохождения сегмента между остановками i и j автобусом k ;

N_r – количество автобусов с маршрутом r ;

$\omega(t)$ – функция ядра следующего вида:

$$\omega(t) = \begin{cases} e^{-\alpha t}, & t \leq \Delta_{\max}; \\ 0, & t > \Delta_{\max}, \end{cases}$$

где Δ_{\max} – максимальный рассматриваемый временной интервал;

7) интервал времени h^{any} между предыдущим автобусом любого маршрута и рассматриваемым автобусом;

8) время прохождения сегмента $T_{ij}^{m,any}$ предыдущим автобусом m любого маршрута;

9) взвешенное время прохождения сегмента \hat{T}_{ij}^{any} предыдущими автобусами любого маршрута:

$$\hat{T}_{ij}^{any} = \frac{\sum_{k \in N} \omega(t - t_i^{dep,k}) T_{ij}^{travel,k}}{\sum_{k \in N} \omega(t - t_i^{dep,k})},$$

где N – количество автобусов с любым маршрутом;

10) среднее время прохождения дорожного сегмента $T_{ij}^{hist}(t)$ автобусами любых маршрутов по статистическим данным в момент времени t ;

11) среднее время прохождения дорожного сегмента $T_{ij}^{flow}(t)$ транспортным потоком по статистическим данным в момент времени t ;

12) количество автобусов c_{ij} на заданном дорожном сегменте. Большое количество транспортных средств на сегменте может привести к дополнительной задержке на ближайшей остановке.

Вектор признаков, описывающий транспортную ситуацию на сегменте между остановками i и j для автобуса с маршрутом r , имеет следующий вид:

$$s_{i,j} = (day, time, v_{i-1,i}, h^r, T_{ij}^{m,r}, \hat{T}_{ij}^r, h^{any}, T_{ij}^{m,any}, \hat{T}_{ij}^{any}, T_{ij}^{hist}, T_{ij}^{flow}, c_{ij}) \quad (55)$$

Нормализация данных

Входной вектор признаков и результирующая оценка времени прохождения сегментов транспортной сети должны быть нормализованы, чтобы ускорить обучение нейронной сети, улучшить сходимость и снизить вероятность нахождения локального оптимума. Данные нормализуются следующим образом:

$$y = \frac{(y_{max} - y_{min})(x - x_{min})}{(x_{max} - x_{min})} + y_{min}$$

где x – исходные данные;

y – нормализованные данные в диапазоне $[0,1]$.

Сеть долгой краткосрочной памяти (LSTM)

Для прогноза времени прохождения общественным транспортным средством каждого сегмента транспортной сети от текущего положения транспортного средства до конца маршрута в работе предлагается использовать нейронную сеть с долгой краткосрочной памятью (LSTM), которая представляет собой особый тип рекуррентной нейронной сети (RNN).

Сеть LSTM была предложена в [325] и, в отличие от RNN, способна работать с долговременными зависимостями. Это достигается за счет ее способности передавать состояние ячейки с предыдущего временного шага на следующий, а также управлять информационным потоком в ячейке LSTM с помощью трех фильтров (gates): входного фильтра, фильтр забывания и выходного фильтра.

На рисунке 26 показана структура ячейки LSTM. Пусть на временном шаге t входное значение равно x_t , состояние входного фильтра – i_t , состояние фильтра забывания – f_t , состояние выходного фильтра – o_t , выходное состояние ячейки – C_t , входное состояние ячейки – \tilde{C}_t и значение выхода модуля – h_t . Значение выхода h_t вычисляется следующим образом:

$$\begin{aligned} i_t &= \sigma(W_{ix}x_t + W_{ih}h_{t-1} + b_i), \\ f_t &= \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f), \\ o_t &= \sigma(W_{ox}x_t + W_{oh}h_{t-1} + b_o), \\ \tilde{C}_t &= \tanh(W_{cx}x_t + W_{ch}h_{t-1} + b_c), \\ C_t &= i_t * \tilde{C}_t + f_t * C_{t-1}, \\ h_t &= o_t * \tanh(C_t), \end{aligned}$$

где $W_{ix}, W_{fx}, W_{ox}, W_{cx}$ – весовые коэффициенты, связывающие x_t с тремя фильтрами и \tilde{C}_t ;

$W_{ih}, W_{fh}, W_{oh}, W_{ch}$ – весовые коэффициенты, связывающие h_{t-1} с тремя фильтрами и \tilde{C}_t ;

b_i, b_f, b_o, b_c – значения смещения трех фильтров и \tilde{C}_t ;

$$\sigma(x) = \frac{1}{1 + \exp(-x)};$$

$$\tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)}.$$

Полученные значения C_t и h_t используются в качестве входных данных для следующего модуля LSTM на следующем шаге работы сети.

Для решения задачи прогнозирования времени прибытия транспорта в работе используется нейронная сеть на основе LSTM, структурная схема которой представлена на рисунке 62.

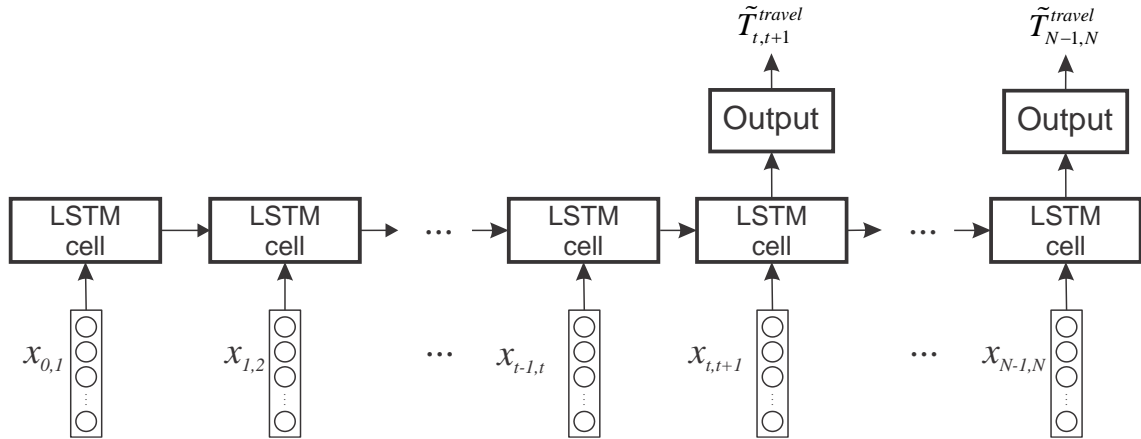


Рисунок 62 – Структурная схема нейронной сети LSTM

Пусть анализируемый маршрут общественного транспорта состоит из N остановок. Тогда выбранный маршрут включает в себя $N - 1$ сегментов транспортной сети, и для каждого из них строится отдельная сеть LSTM, так как размер входных и выходных данных различен для каждой транспортной сегмента. В качестве одного временного шага рассматривается поездка между парой соседних остановок. Каждая сеть LSTM прогнозирует время прохождения каждого из оставшихся на маршруте транспортных сегментов, используя информацию, полученную во время движения транспортного средства по своему маршруту. На временном шаге t ($0 < t < N - 1$) в качестве входных данных для соответствующей сети LSTM может использоваться от 1 до t предыдущих шагов. При использовании максимального количества предыдущих шагов входные данные имеют следующий вид: $(x_{0,1}, x_{1,2}, \dots, x_{t-1,t})$, где $x_{i,i+1} = (T_{i,i+1}^{travel} s_{0,1}, s_{1,2}, \dots, s_{N-2,N-1})$. Количество используемых шагов s является параметром предложенной модели. Тогда результатом работы сети LSTM являются прогнозируемые данные о времени прохождения оставшихся сегментов $(\tilde{T}_{t+1,t+2}^{travel}, \tilde{T}_{t+2,t+3}^{travel}, \dots, \tilde{T}_{N-1,N}^{travel})$. Вектор $x_{i,i+1}$ содержит значение времени прохождения сегмента между остановками i и $i + 1$ и описание транспортной ситуации (55) на всем маршруте в момент времени, когда автобус прибывает на остановку $i + 1$. Использование такого сложного описания обусловлено тем фактом, что сеть LSTM позволяет учитывать влияние транспортной ситуации на удаленных транспортных сегментах в долгосрочном прогнозе.

Разработанная модель прогнозирования времени прибытия транспортных средств с использованием LSTM-сетей были реализована с использованием Keras [362] – библиотеки для обучения нейронных сетей с открытым исходным кодом. Каждая LSTM-сеть состоит из одного слоя LSTM с 64 блоками (unit). Выход последнего LSTM модуля связан с выходным слоем, состоящим из определенного числа нейронов. Это число отличается для каждой из сети, предназначенной для прогнозирования конкретного маршрута, и соответствует количеству оставшихся транспортных сегментов от текущего положения транспортного средства до конечной остановки. В качестве метода оптимизации использовался метод Adam [363]; в качестве функции потерь использовалась средняя абсолютная ошибка.

4.1.1.2 Экспериментальные исследования

Экспериментальные исследования проводились на основе данных о движении транспортных средств пассажирских автобусных маршрутов в городе Самара, Россия, полученных за сентябрь 2018 года. Для экспериментального анализа было выбрано 5 произвольных автобусных маршрутов, содержащих от 22 до 39 остановок. Средняя протяженность выбранных маршрутов составляет 16 км. Траектории движения автобусов, полученные с помощью технологии GPS, были обработаны и преобразованы во время прохождения сегментов транспортной сети. Используемый набор данных был сгруппирован по рейсам, и разделен на три части: обучающая выборка (80% рейсов), контрольная выборка (10% рейсов) и валидационная выборка (10% рейсов).

Было проведено сравнение предложенной модели на основе LSTM-сетей с базовой моделью искусственной нейронной сети (ИНС, один скрытый слой, состоящий из 32 нейронов) и моделью линейной регрессии.

Для оценки качества прогноза нейронной сети LSTM и базовых моделей прогноза были использованы две стандартные метрики: средняя абсолютная ошибка (MAE) и средняя абсолютная ошибка в процентах (MAPE):

$$\text{MAE} = \frac{1}{n} \sum_{t=1}^n |T^{travel} - \hat{T}^{travel}|,$$

$$\text{MAPE} = \frac{1}{n} \sum_{t=1}^n \frac{|T^{travel} - \hat{T}^{travel}|}{T^{travel}} \times 100\%,$$

где T^{travel} – истинное время прохождения сегмента;

\hat{T}^{travel} – прогнозное время прохождения сегмента;

n – размер выборки.

На первом шаге экспериментальных исследований было проведено сравнение предлагаемой модели прогнозирования и базовых моделей при прогнозировании времени прибытия автобуса на каждую из оставшихся остановок на маршруте. Это равносильно прогнозированию и аккумулярованию времени прохождения каждого оставшегося транспортного сегмента на маршруте. Графики зависимости MAE и MAPE от реального времени движения показаны на рисунках 63 и 64.

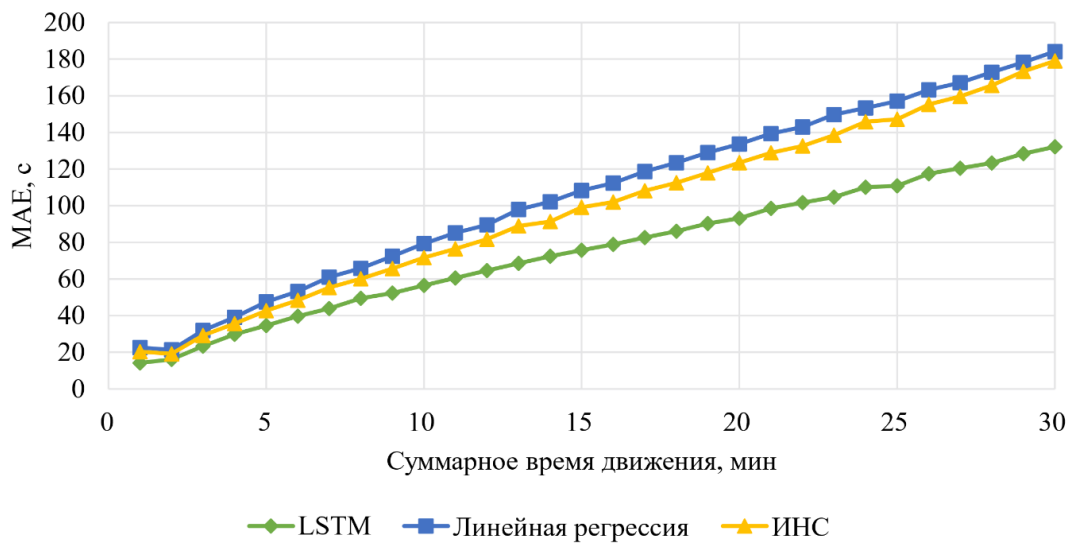


Рисунок 63 – Сравнение прогнозируемого и измеренного суммарного времени движения по критерию MAE

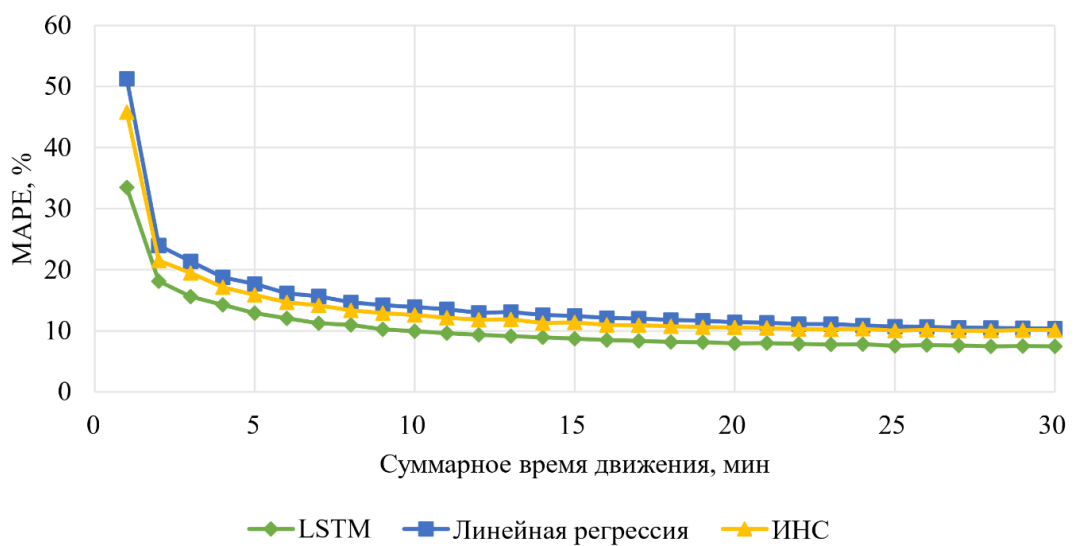


Рисунок 64 – Сравнение прогнозируемого и измеренного суммарного времени движения по критерию MAPE

Эксперименты показывают, что предлагаемый алгоритм имеет более высокую точность прогнозирования, чем базовые модели. На рисунках 63 и 64 представлены усредненные результаты для пяти маршрутов. Отдельные результаты для каждого из используемых маршрутов, усредненные за 30 минут, представлены на рисунках 65 и 66 по критериям средней абсолютной и средней относительной ошибок соответственно.

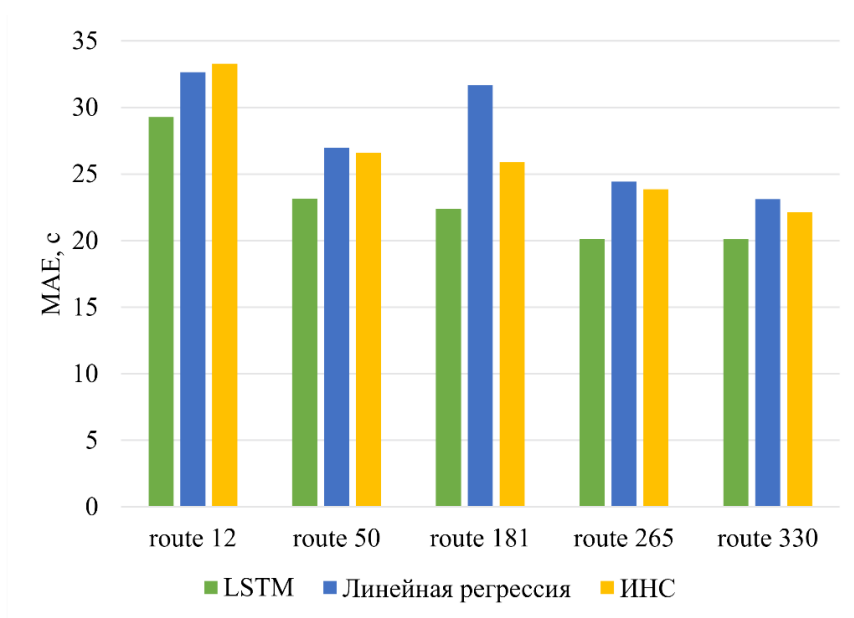


Рисунок 65 – Сравнение точности прогноза по критерию MAE для каждого маршрута

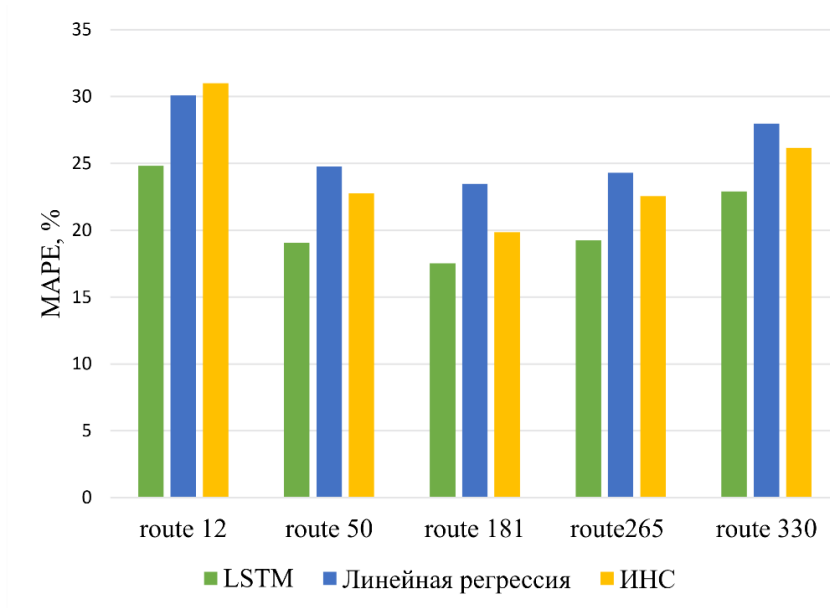


Рисунок 66 – Сравнение точности прогноза по критерию MAPE для каждого маршрута

На каждом из пяти маршрутов лучший результат был показан при использовании предложенной модели, худший результат в большинстве случаев был показан при использовании модели линейной регрессии.

На рисунке 67 показана зависимость прогнозного времени прибытия транспортных средств от реального времени прибытия для одного из используемых маршрутов для предложенной модели. По графику видно, что предложенная модель обеспечивает высокую точность прогноза.

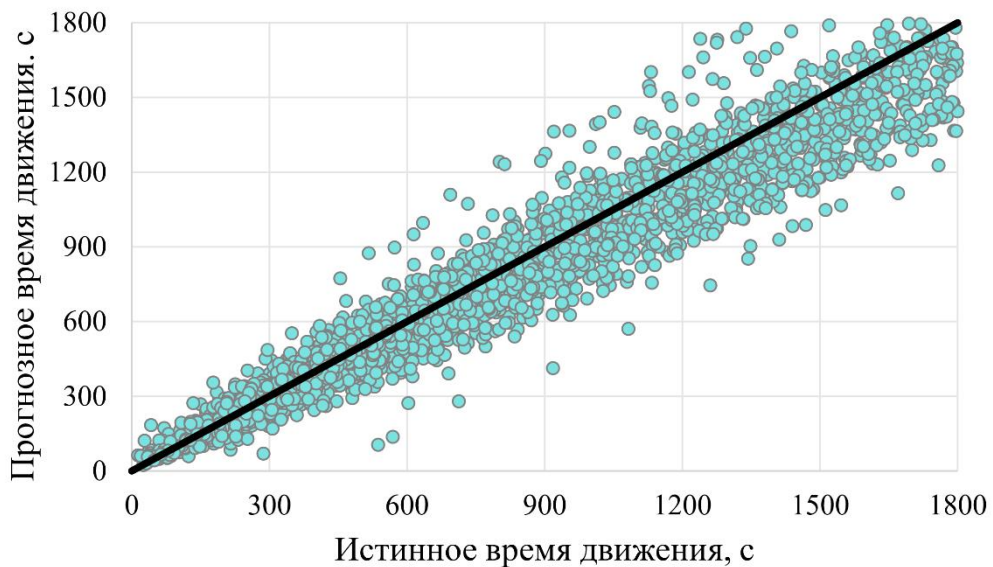


Рисунок 67 – Сравнение прогнозного и реального времени прибытия автобусов

В ходе проведения экспериментальных исследований была также выполнена оценка времени, затрачиваемого на выполнение прогноза. Поскольку прогноз необходимо делать каждый раз, когда транспортное средство отправляет свои GPS-координаты на сервер, время вычислений является решающим фактором. Зависимость среднего времени вычислений от размера пакета (количества выборок в наборе данных) с использованием стандартного персонального компьютера (Intel Core i5-3470 3,20 ГГц, 8 ГБ ОЗУ) и графического процессора nVidia GeForce GTX1080 Ti показана на рисунке 68. Время вычисления на GPU отдельно показано на рисунке 69.

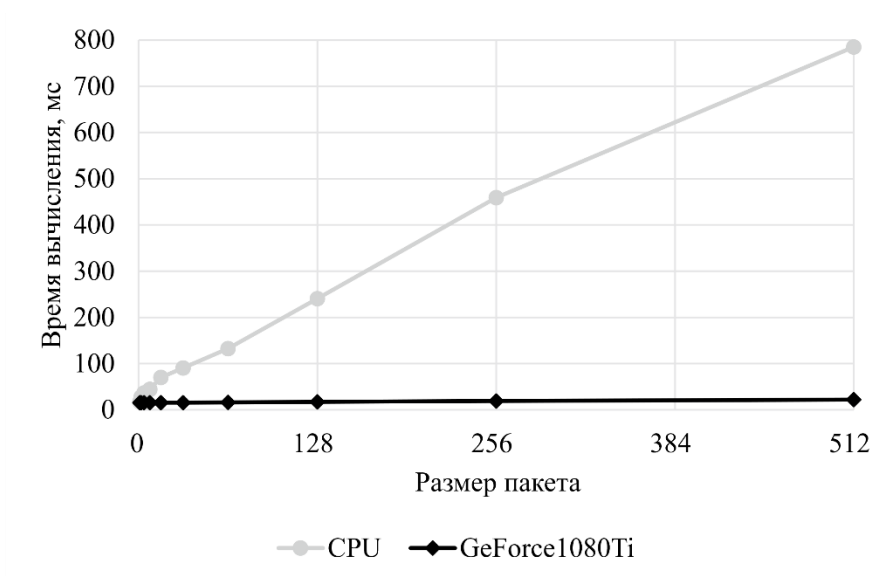


Рисунок 68 – Сравнение времени вычислений на CPU и GPU

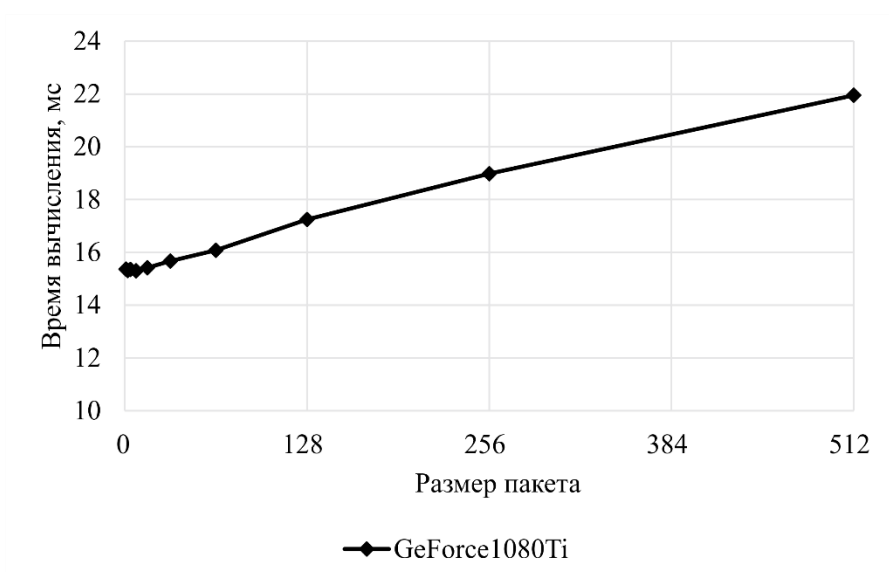


Рисунок 69 – Время вычисления на GPU

Полученные результаты показывают, что модель LSTM может использоваться для прогнозирования времени прибытия в реальном времени при работе в крупномасштабной транспортной сети.

4.1.2 Прогнозирование движения подключенных транспортных средств

В данном разделе рассматривается задача прогнозирования времени движения подключенных транспортных средств по маршруту в сценарии с адаптивным светофорным регулированием.

4.1.2.1 Модель на основе глубокой нейронной сети

Прогнозируемое время движения транспортного средства по маршруту рассчитывается как сумма прогнозируемого времени движения по сегментам дорожной сети, входящих в маршрут движение, и прогнозируемого времени ожидания на регулируемом перекрестке.

Для оценки времени движения по сегментам дорожной сети предлагается использовать нейросетевую модель. Для оценки времени ожидания выполняется прогноз фазы светофора в момент прибытия транспортного средства на перекресток. Время ожидания рассчитывается на основе времени прибытия транспортного средства и времени длительности фазы светофорного цикла. Далее каждая часть предлагаемого решения рассматривается более подробно.

Прогнозирование времени движения

Для оценки времени движения $t(c)$ транспортного средства c по сегменту дорожной сети предлагается использовать модель на основе искусственной нейронной сети. Вектор признаков, поступающий на вход нейронной сети, содержит следующие параметры:

- 1) расстояние от текущего положения транспортного средства до перекрестка;
- 2) скорость транспортного средства;
- 3) ускорение транспортного средства;
- 4) максимально допустимая скорость на сегменте дорожной сети;
- 5) количество транспортных средств перед выбранным ТС сегменте дорожной сети;
- 6) тип ожидаемого перестроения на перекрестке;
- 7) скорость и положение ближайшего ТС на полосе выезда.

Архитектура используемой модели ИНС показана на рисунке 70. Предлагаемая архитектура состоит из семи полносвязных слоев. Выходным значением модели ИНС является время движения по сегменту сети.

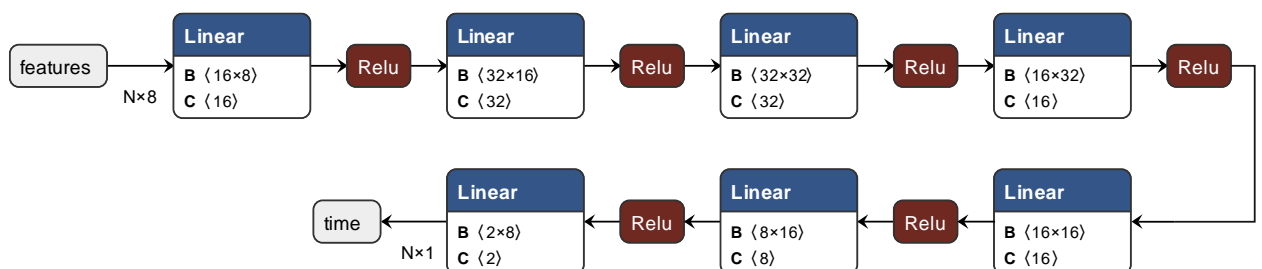


Рисунок 70 – Архитектура модели ИНС

В работе исследуются два подхода к расчету времени движения транспортного средства по маршруту. В первом базовом подходе обучаются две ИНС с одинаковой архитектурой: для дорожных сегментов, примыкающих к регулируемым и нерегулируемым перекресткам. Модель ИНС, обученная на данных наблюдаемых параметров транспортного потока с регулируемых перекрестков прогнозирует время движения, которое неявно содержит среднее время ожидания на перекрестке (согласно обучающему набору данных).

Во втором подходе предлагается явно оценивать время ожидания на перекрестке по прогнозируемой фазе светофорного объекта и продолжительности фазы. Для этого модель ИНС обучается на данных о движении транспортных средств, которые движутся по сегменту дорожной сети, когда для этого сегмента разрешено движение (т.е. горит зеленый сигнал светофора). Другими словами, данные от транспортных средств со временем ожидания на перекрестках не включаются в набор обучающих данных.

Для оценки времени ожидания на перекрестке необходимо спрогнозировать фазу светофорного объекта в момент прибытия транспортного средства на перекресток. В данной работе предполагается, что светофор управляется методом адаптивного управления сигналами светофора, основанным на максимизации взвешенного потока транспортных средств, пересекающих перекресток за выбранную фазу светофорного объекта. Метод подробно описан в разделе 3.3 данной работы.

Прогнозирование времени ожидания

Оценка времени ожидания транспортного средства на регулируемом перекрестке рассчитывается исходя из продолжительности прогнозируемой фазы светофора в момент прибытия транспортного средства на перекресток.

Прогнозируемая фаза выбирается на основе следующих шагов:

1) для каждого момента времени выбора фазы формируется оценка количества транспортных средств, прошедших перекресток в следующем интервале времени для каждой фазы. Оценка основана на прогнозируемом времени движения от положения транспортного средства до ближайшего перекрестка, полученного моделью из раздела 3.3.2;

2) выбирается фаза светофора, для которой оценка количества транспортных средств максимальна.

Время ожидания на перекрестке для выбранного транспортного средства равно нулю, если движение по соответствующему дорожному сегменту в прогнозируемую фазу

светофора разрешено (горит зеленый сигнал светофора). В противном случае время ожидания равно оставшейся продолжительности фазы светофора.

В следующем разделе представлены экспериментальные исследования предлагаемого подхода.

4.1.2.2 Экспериментальные исследования

Для проведения экспериментального исследования предлагаемого подхода был использован пакет моделирования SUMO [327]. В исследовании использовались два сценария моделирования:

1) «Cologne-8» – сценарий моделирования в области транспортной сети малого размера;

2) «Cologne-316» – крупномасштабный сценарий моделирования движения транспортных средств, который содержит 2928 перекрестков различной топологии, в том числе 316 регулируемых перекрестков.

Моделирование движения проводилось с 7:00 до 9:00 в утренние часы пик.

В экспериментальном исследовании проводилось сравнение трех алгоритмов:

1) алгоритм прогнозирования движения на основе детерминированной модели, предсказывающей время движения на основе аналитических закономерностей движения транспортных средств (подробно описана в разделе 3.3.1);

2) алгоритм прогнозирования движения на основе двух моделей ИНС, обученных на данных о движении транспортных средств по сегментам дорожной сети, примыкающих к регулируемым и нерегулируемым перекресткам соответственно (обозначается далее как ИНС);

3) алгоритм прогнозирования движения на основе модели ИНС с учетом времени ожидания, подробно описанной в предыдущем разделе (обозначается далее как ИНС + время ожидания).

Сравнение алгоритмов проводилось по критерию средней абсолютной ошибки (в секундах), который можно сформулировать следующим образом:

$$MAE = \frac{1}{n} \sum_{t=1}^n |T^{travel} - \tilde{T}^{travel}| \quad (56)$$

где T^{travel} – истинное время движения по сегментам дорожной сети;

\tilde{T}^{travel} – прогнозируемое время движения по сегментам дорожной сети;

n – общее количество тестовых образцов.

Результаты сравнения оцениваемых моделей представлены в таблице 22.

Таблица 22 – Сравнение моделей по критерию средней абсолютной ошибки (в секундах)

Модель	Сценарии	
	Cologne-8	Cologne-316
Детерминированная модель	13.88	65.73
ИНС	7.92	26.25
ИНС + время ожидания	7.62	23.92

Как видно из таблицы, предложенный подход с явным расчетом времени ожидания показывает наилучший результат по критерию MAE для обоих сценариев.

На следующем шаге сравнивается общее время движения по всем сегментам дорожной сети, входящих в маршрут движения транспортного средства. На рисунке 71 представлена зависимость средней абсолютной ошибки от аккумулярованного истинного времени движения.

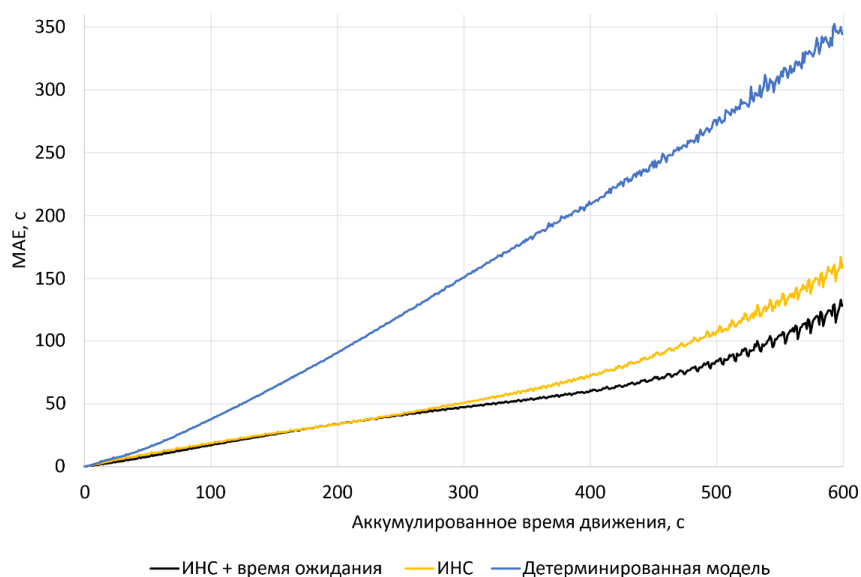


Рисунок 71 – Сравнение прогнозируемого и наблюдаемого аккумулярованного времени движения

Результаты экспериментальных исследований позволяют сделать вывод, что предлагаемый подход имеет более высокую точность прогноза, чем базовые модели.

4.2 Решение навигационных задач в ИТС

Повсеместный рост загрузки транспортных сетей, вызванный как увеличением числа транспортных средств в крупных мегаполисах, так и априорной неготовностью

созданной в прошлом дорожной инфраструктуры к современным потокам, делает всё более актуальным решение навигационных задач, связанных с задачами выбора оптимального маршрута. Многие существующие системы позволяют вычислительно эффективно находить кратчайший путь до места назначения и предоставляют информацию об альтернативных вариантах маршрута на основе информации о дорожных заторах. Однако, в основном они работают с детерминированными сетями, т.е. полагают время прохождения сегментов сети постоянным. В то же время реальная ситуация совершенно иная, то есть время/скорость прохождения участка транспортной сети конкретным транспортным средством существенно зависит от многих факторов, включая время года, день недели, текущее время, предшествующую ситуацию на данном участке и т.п. Учитывая такую значительную изменчивость только основного параметра движения, и задача выбора оптимального пути, и само понятие «оптимальный путь» оказываются не такими простыми. Одним из целесообразных способов формализации такой задачи является задача нахождения надёжного пути.

Определение надёжного пути в зависящих от времени стохастических сетях с использованием информации реального времени – это проблема как теоретического, так и практического характера. За последние десять лет этой проблеме было посвящено много исследований, подробнее описанных в разделе 1.4.2. Несмотря на это, многие существующие решения используют значительные упрощения. Например, не учитывают пространственную и/или временную корреляцию сегментов дорожной сети, либо учитывают её для пары последовательных сегментов/ребер, и др. К тому же, большинство работ носят «тестовый» характер, т.е. предлагаемые решения не работают для реальных крупных транспортных сетей.

В данном разделе предлагается разработка алгоритма определения надёжного пути в зависящей от времени стохастической (транспортной) сети, учитывающего информацию о пространственной и временной корреляции сегментов дорожной сети, текущую и прогнозную информацию о состоянии сети. В отличие от многих существующих решений, определяющих решение в виде пути на стохастическом графе, предлагаемый алгоритм даёт решение в виде правила принятия решения о следующем используемом сегменте сети (из текущей вершины) так, чтобы максимизировать вероятность достижения вершины прибытия в заданный интервал времени. Таким образом, предлагаемый в решении маршрут заранее не фиксирован, а адаптивно меняется в зависимости от изменения реального и прогнозного состояний сети.

В разделе 4.2.1 представлена постановка задачи нахождения надежного пути в стохастической зависящей от времени транспортной сети и описание базового алгоритма решения задач. В разделе 4.2.2 предложен модифицированный алгоритм, учитывающий актуальную и прогнозную информацию для решения задачи навигации. Развитие этого алгоритма путем применения устойчивого распределения Леви для описания времени прохождения сегментов дорожной сети представлено в разделе 4.2.3. Показано, что использование устойчивых распределений позволяет перейти от операции вычисления свертки для определения надежности пути к пересчету параметров плотности распределения, что значительно сокращает время исполнения алгоритма.

4.2.1 Постановка задачи нахождения надежного пути

В задаче зависящая от времени стохастическая улично-дорожная сеть рассматривается в виде ориентированного графа $G = (N, A, P)$, где N – множество вершин графа, $|N|$ – количество вершин, A – множество ребер, $|A|$ – количество ребер, P – вероятностное описание времени прохождения ребер графа (т.е. сегментов улично-дорожной сети).

В зависящих от времени стохастических сетях вес каждого сегмента $(i, j) \in A$ обычно представляется как случайная величина $T_{ij}(\tau)$ с зависящей от времени плотностью вероятности $p_{ij}^\tau(t)$.

Обозначим конечную вершину (пункт назначения) как $d \in N$, интервал времени, допустимый для достижения конечной вершины (т.е. бюджет поездки) как T . Оптимальная стратегия навигации определяется как стратегия максимизации вероятности прибытия в конечную вершину $d \in N$ из начальной (текущей) вершины $o \in N$ при наличии временного бюджета T .

Чтобы формально определить стратегию навигации, введем дополнительную величину $u_i(t)$ – вероятность прибытия в конечную вершину d из вершины i за время, не превышающее t [217, 221]. Тогда оптимальная стратегия навигации может быть сформулирована следующим образом:

$$u_i^\tau(t) = \max_{j \in N \wedge (i,j) \in A} \int_0^t p_{ij}^\tau(\theta) u_j^{\tau+\theta}(t - \theta) d\theta, \quad (57)$$

$$\forall i \in N \setminus \{d\}, t \in [0, T], \tau \geq 0,$$

$$u_d^\tau(t) = 1, t \in [0, T], \tau \geq 0.$$

Определим вероятность прохождения пути J за время t при условии, что поездка была начата в момент τ , как $u_j^\tau(t)$. Будем считать, что граф удовлетворяет условию стохастического FIFO (First-In-First-Out), если выполняются следующие неравенства [217]:

$$u_j^{t_1}(T) \geq u_j^{t_2}(T - (t_2 - t_1)), \\ \forall J, \forall T, t_1, t_2: 0 \leq t_1 \leq t_2, t_2 - t_1 \leq T.$$

Согласно этому определению, незамедлительное начало движения по выбранному пути увеличивает вероятность прибытия в пункт назначения за выбранный бюджет поездки по сравнению с отложенным началом движения.

Более формально модель поведения в графах, удовлетворяющих условию стохастического FIFO, характеризуется следующим положением: ожидание начала движения в нетерминальной вершине не удовлетворяет оптимальной стратегии навигации (57).

Решение проблемы (57) на практике производится с использованием численного алгоритма, заменяющего операции свёртки в (57) реализацией с использованием быстрого алгоритма Фурье. В виде псевдокода, дискретный алгоритм решения проблемы (57), представленный в работе [217], выглядит следующим образом (Алгоритм б).

Алгоритм б: Дискретный алгоритм решения SOTA

Шаг 0: Инициализация

- 1: $k = 0$;
- 2: $u_i^k(x) = 0, \quad \forall i \in N, i \neq d, x \in \mathbb{N}, 0 \leq x \leq \frac{T}{\Delta T}$;
- 3: $u_d^k(x) = 1, \quad x \in \mathbb{N}, 0 \leq x \leq \frac{T}{\Delta T}$;

Шаг 1: Обновление

- 4: **for** ($k = 1, 2, \dots, L$) **do**
 - 5: $\tau^k = k\delta$;
 - 6: $u_d^k(x) = 1, \quad x \in \mathbb{N}, 0 \leq x \leq \frac{T}{\Delta T}$;
 - 7: $u_i^k(x) = u_i^{k-1}(x), \quad \forall i \in N, i \neq d, (i, j) \in A, x \in \mathbb{N}, 0 \leq x \leq \frac{\tau^k - \delta}{\Delta t}$;
 - 8: $u_i^k(x) = \max_j \sum_{h=0}^x p_{ij}(h) u_j^{k-1}(x-h), \quad \forall i \in N, i \neq d, (i, j) \in A, x \in \mathbb{N}, \frac{\tau^k - \delta}{\Delta t} \leq x \leq \frac{\tau^k}{\Delta t}$;
 - 9: **end for**
-

В алгоритме Δt – интервал дискретизации, δ – минимальное время прохождения дорожного сегмента в сети.

Тогда выбор следующей вершины j в графе дорожной сети (и, соответственно, следующего дорожного сегмента) с учетом оставшегося бюджета поездки t и вычисленного массива вероятностей прибытия $u_i(x)$ производится следующим образом:

$$j = \arg \max_{i \in N} u_i(t). \quad (58)$$

Предложенный в данном разделе алгоритм решения задачи определения надёжного пути в стохастической зависящей от времени транспортной сети учитывает только текущую информацию о загрузке транспортной сети для определения веса каждого сегмента $(i, j) \in A$ и не учитывает прогнозную информацию о распределении транспортного потока в сети. В разделе 4.2.2 предложен модифицированный алгоритм, учитывающий актуальную и прогнозную информацию для решения задачи навигации. Развитие этого алгоритма путем применения устойчивого распределения Леви для описания времени прохождения сегментов дорожной сети представлено в разделе 4.2.3.

4.2.2 Алгоритм нахождения надежного пути на основе прямого вычисления сверток

4.2.2.1 Модифицированная постановка задачи и предлагаемый алгоритм

Пусть $I(\tau_0)$ – актуальная информация о транспортной сети в текущий момент времени τ_0 . Здесь под актуальной информацией понимается:

- 1) собственно текущий момент времени τ_0 ;
- 2) текущие и исторические характеристики транспортных потоков для некоторого наперёд заданного исторического интервала $\Delta\tau$, в данной задаче – время прохождения конкретных участков сети: $t_{ij}(\tau)$, $\tau_0 - \Delta\tau \leq \tau \leq \tau_0$.

Определим $p_{ij}^\tau(t|I(\tau_0))$ ($\tau \geq \tau_0$) как (прогнозное, на практике – оценённое) распределение случайной величины $T_{ij}(\tau)$ для будущего момента τ , определяемое для известной в текущий момент времени τ_0 актуальной информации $I(\tau_0)$. Очевидно, чем меньше временной «интервал» $(\tau - \tau_0)$, тем меньшая дисперсия у величины $T_{ij}(\tau)$. Совместные распределения для набора сегментов $(i, j), (j, k), (k, l) \in A$ могут быть заданы аналогичным образом: $p_{ijkl}^\tau(t_{ij}, t_{jk}, t_{kl}|I(\tau_0))$. Для упрощения текста изложения далее описан случай, соответствующий модели с независимыми сегментами: $p_{ij}^\tau(t|I(\tau_0))$.

Актуальная информация $I(\tau_0)$, таким образом, непосредственно используется для определения (на практике – для оценки) прогнозного распределения вероятностей $p_{ij}^\tau(t|I(\tau_0))$ ($\tau \geq \tau_0$) времени прохождения сегмента в определённый будущий момент времени τ

Пусть в момент времени отправления τ_0 транспортное средство находится в вершине $s \in N$ (вершина отправления), $d \in N$ – вершина прибытия. Пусть, как и ранее, $T > 0$ – допустимый бюджет по времени, тогда прибыть в вершину прибытия d необходимо не позднее момента $\tau_0 + T$. Обозначим $u_i^\tau(t|I(\tau_0))$ ($\tau \geq \tau_0$) – вероятность (конструируемую оценку) достижения вершины прибытия $d \in N$ за время не более чем t при условии старта из вершины $i \in N$ в момент времени τ . Соответствующая вероятность рассчитывается в текущий момент времени τ_0 , в который известна актуальная информация о параметрах дорожной сети $I(\tau_0)$. В рамках введённых обозначений оптимальная стратегия навигации (57), обозначаемая ниже как задача $Z(\tau_0, T)$, может быть переписана следующим образом:

$$\begin{aligned} u_i^\tau(t|I(\tau_0)) &= \max_{j \in N \wedge (i,j) \in A} \int_0^t p_{ij}^\tau(\theta|I(\tau_0)) u_j^{\tau+\theta}(t-\theta|I(\tau_0)) d\theta, \\ &\forall i \in N \setminus \{d\}, t \in [0, T], \tau \geq \tau_0, \\ u_d^\tau(t|I(\tau_0)) &= 1, t \in [0, T], \tau \geq \tau_0. \end{aligned} \quad (59)$$

Из приведённых соотношений очевидно, что чем больший «интервал» $(\tau - \tau_0)$ участвует в расчётах, тем с большей «дисперсией» времени участвуют плотности этой величины $p_{ij}^\tau(t|I(\tau_0))$ ($\tau \geq \tau_0$) в расчёте интеграла в (59). Учитывая неотрицательность самой величины времени, рост её дисперсии однозначно приводит к снижению значения итогового интеграла, и, следовательно, к ошибкам в прогнозах на больших временных интервалах.

Предлагаемая модификация алгоритма и стратегии состоит в использовании при расчётах актуальной (на момент выбора дальнейшего пути) информации. В предельном (гипотетическом) случае стратегия (59) модифицируется следующим образом:

$$\begin{aligned} u_i^\tau(t|I(\tau)) &= \max_{j \in N \wedge (i,j) \in A} \int_0^t p_{ij}^\tau(\theta|I(\tau)) u_j^{\tau+\theta}(t-\theta|I(\tau+\theta)) d\theta, \\ &\forall i \in N \setminus \{d\}, t \in [0, T], \tau \geq \tau_0, \\ u_d^\tau(t|I(\tau)) &= 1, t \in [0, T], \tau \geq \tau_0. \end{aligned} \quad (60)$$

Для стратегии (60) могут быть легко доказаны утверждения, аналогичные приведённым в разделе 4.2.1: Если граф удовлетворяет условию стохастического FIFO, то ожидание (не незамедлительное начало движения) в нетерминальной вершине не удовлетворяет оптимальной стратегия навигации (57).

Граф называется удовлетворяющим условию динамического стохастического FIFO, если выполняются неравенства:

$$u_j^{t_1}(t|I(t_1)) \geq u_j^{t_2}(T - (t_2 - t_1)|I(t_2)),$$

$$\forall J, \forall T, t_1, t_2: 0 \leq t_1 \leq t_2, t_2 - t_1 \leq T. \quad (61)$$

Модель поведения в таких графах характеризуется следующим положением: если граф удовлетворяет условию динамического стохастического FIFO, то ожидание (не незамедлительное начало движения) в нетерминальной вершине не удовлетворяет оптимальной стратегии навигации (61).

Очевидна нереализуемость вычислений в рамках данной стратегии по причине неизвестности в момент начала движения τ_0 будущей актуальной информации о транспортной сети: $I(\tau)$, $\tau > \tau_0$.

Предлагаемым приближённым алгоритмом решения задачи (61) является алгоритм решения набора задач (60) для последовательных моментов времени $\tau_0, \tau_k (k = \overline{1, K-1})$, $\tau_K = \tau_0 + T$. Определение моментов τ_k решения соответствующих задач $Z(\tau_k, \tau_K - \tau_k)$ может быть выполнено либо через наперёд заданный временной интервал, либо по моментам прохождения заданного числа вершин в графе транспортной сети. Независимо от способа задания моментов времени, предлагаемый оптимальный алгоритм навигации в виде псевдокода имеет следующий вид (Алгоритм 7).

Алгоритм 7: Алгоритм навигации на основе актуальной информации

```

1: /* блок псевдокода выполняется при достижении оче –
редной вершины  $r$  графа сети в некоторый момент  $\tau \in [\tau_0, \tau_0 + T]$  */
2: vertexes.add(d);
3: if ( $r == d$ ) then stop // навигация завершена успешно
4: if ( $\tau \geq \tau_K$ ) then stop // навигация завершена неуспешно (временной бюджет исчерпан)
5: if ( $\tau \in [\tau_k, \tau_{k+1}]$ ) then begin
6:   получение подмножества  $\Xi$  вершин по стратегии  $Z(\tau_k, \tau_K - \tau_k)$ ,
7:   обеспечивающих одинаковую (максимальную) надёжность;
8:   if ( $|\Xi| > 1$ ) then
9:     выбор следующей вершины из подмножества  $\Xi$  по критерию
10:    минимума среднего времени достижения вершины назначения;
11:   else //  $|\Xi| = 1$ 
12:     следующая вершина – единственная вершина из  $\Xi$ ;
13: end;

```

Численное решение каждой из задач $Z(\tau_k, \tau_K - \tau_k)$ может производиться с использованием дискретного алгоритма SOTA, приведённого ранее. Определение вершины из подмножества по критерию минимума среднего времени достижения вершины выполняется известным алгоритмом Дейкстры поиска кратчайших путей в

детерминированном графе. Данная модификация введена для устранения проблемы циклов, возникающих в базовом алгоритме.

Примечание 1: учитывая возросшую вычислительную сложность предложенного алгоритма по сравнению с дискретным алгоритмом SOTA для ускорения решения набора задач $Z(\tau_k, \tau_K - \tau_k)$ целесообразно использовать разбиение графа сети на подсети меньшего размера, содержащие текущую r и конечную d вершины, и использовать при решении одну из таких подсетей.

Примечание 2: при использовании способа ускорения, предложенного в примечании 1, можно добиться требуемой (по отношению к базовому дискретному алгоритму) вычислительной сложности. Например, допустив, что размеры выбираемой подсети оказываются пропорциональными площади квадратной области со стороной (или диагональю), определяемой расстоянием между текущей и конечной вершиной, а также допустив, что моменты τ_k выбираются так, что расстояние сокращается пропорционально, легко получить следующую относительную характеристику роста вычислительной сложности решения предлагаемой задачи:

$$c = \frac{1}{K^2} \sum_{k=0}^K k^2 = \frac{(K+1) \cdot (2K+1)}{6K} \quad (62)$$

Значения для некоторых K этой суммы приведены в таблице 23. Из этой таблицы видно, что усложнение в два раза достигается при наличии временных пяти интервалов, в четыре – при одиннадцати. Учитывая, что на практике задача навигации в мегаполисах решается в краткосрочной перспективе (длительность поездки обычно не превышает часа), получаемое усложнение вычислений (длительность интервалов оказывается в диапазоне 5-10 минут) представляется незначительным.

Таблица 23 – Значение показателя роста вычислительной сложности предлагаемого алгоритма по сравнению с дискретным алгоритмом SOTA

K	2	3	4	5	6
Величина (7)	1,25	1,556	1,875	2,2	2,528
K	7	8	9	10	11
Величина (7)	2,857	3,188	3,519	3,85	4,182

Примечание 3: основными данными для решения задач $Z(\tau_k, \tau_K - \tau_k)$ являются плотности вероятностей $p_{ij}^{\tau}(t|I(\tau_0))$ ($\tau \geq \tau_0$). На практике указанные плотности неизвестны и подлежат оценке. В данной работе предложен алгоритм оценки, изложенный ниже.

В качестве плотности вероятности $p_{ij}^{\tau}(t|I(\tau_0))$ ($\tau \geq \tau_0$) распределения времени движения на конкретном сегменте (i, j) предлагается использовать двухпараметрические семейства плотностей. Естественным ограничением является неотрицательность области определения выбираемых функций плотности. В рамках данной работы в качестве конкретного претендента рассмотрено логнормальное распределение. Определение распределения, включая выражения для математического ожидания и дисперсии, приведено ниже в таблице 24.

Параметрическая оценка плотностей в текущие моменты времени, а именно $p_{ij}^{\tau}(t|I(\tau))$ ($\tau = \tau_0$), выполняется с использованием известного метода моментов: выполняется оценка среднего и дисперсии и расчёт параметров распределений посредством соотношений, приведённых в таблице 24.

Таблица 24 – Логнормальное распределение

Вид распределения	$f_X(x) = \frac{1}{x\sigma\sqrt{2\pi}} \exp\left(\frac{-(\ln x - \mu)^2}{2\sigma^2}\right),$ $x \geq 0$
Моменты через параметры	$MX = \exp\left(\mu + \frac{\sigma^2}{2}\right),$ $DX = (e^{\sigma^2} - 1) e^{2\mu + \sigma^2}$
Параметры через моменты	$\sigma^2 = 1 + \ln \frac{DX}{(MX)^2},$ $\mu = \ln(MX) - \frac{1}{2} \left(1 + \ln \frac{DX}{(MX)^2}\right)$

Прогноз функции плотностей для будущих моментов времени, то есть получение выражений $p_{ij}^{\tau}(t|I(\tau_0))$ ($\tau > \tau_0$), осуществляется через прогноз моментов распределений (математического ожидания и корня из дисперсии – среднеквадратического отклонения) на основании исторических данных. Поскольку суть данной задачи – это прогноз параметров транспортного потока, для её решения используется описанный в разделе 1) подход. Основные этапы подхода следующие:

1) граф уличной дорожной сети (УДС) разбивается на пересекающиеся подграфы (обозначаемы далее $\{A_l\}_{l=0}^{L-1}$) по территориальному признаку;

2) в каждом подграфе УДС A_l в текущий момент времени τ формируется вектор признаков $\bar{v}_l(\tau)$, состоящий из параметров участков сети (в данном случае моментов – среднего времени прохождения сегмента и среднеквадратического отклонения времени) в данный момент и на протяжении некоторого временного интервала;

3) прогноз параметров для участков сети конкретного подграфа производится одним из алгоритмов, описанных разделе 1).

Дополнительным этапом к представленным, необходимым в данном случае для получения прогнозной оценки плотности $p_{ij}^{\tau}(t|I(\tau_0))$ ($\tau > \tau_0$), является этап перехода от спрогнозированных величин моментов распределений к их параметрам. Выполняется этот этап тривиально на основании представленных в таблице 24 соотношений.

4.2.2.2 Повышение эффективности вычисления надежного пути с использованием GPU

Для снижения времени работы алгоритма поиска надежного пути предлагается выполнить его реализацию на графическом акселераторе с использованием CUDA.

CUDA (Compute Unified Device Architecture) – программно-аппаратная архитектура параллельных вычислений, разработанная компанией Nvidia, позволяющая использовать графические процессоры для вычислений общего назначения.

Модель вычислений представлена на рисунке 72. Хост-компьютер передает данные в память устройства и вызывает специальную функцию, называемую ядром. При вызове функции-ядра задаются два параметра: число блоков (blocks) и число потоков в блоке (threads). Каждый поток выполняет один и тот же набор команд, но с разными элементами данных. Потоки в рамках одного блока могут обмениваться результатами вычислений с использованием механизма общей памяти.

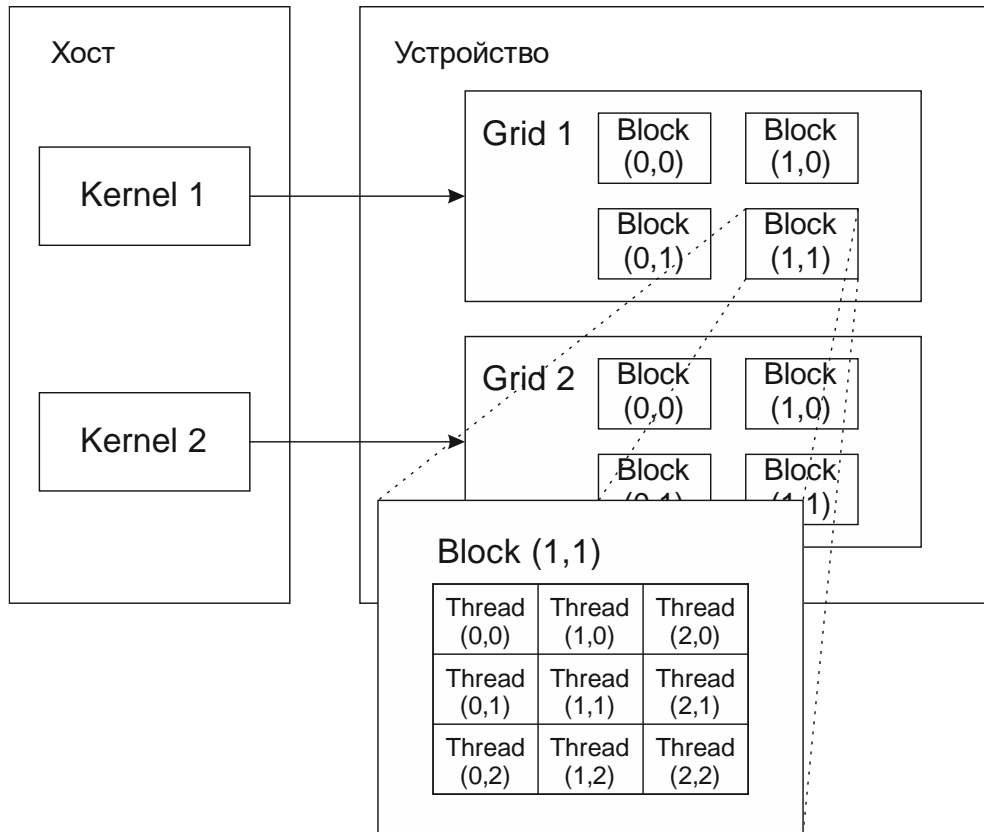


Рисунок 72 – Модель вычислений

Введем обозначение:

$$u_{ij}(x) = \sum_{h=0}^x p_{ij}(h) u_j(x-h), \quad (63)$$

где $u_{ij}(x)$ – вероятность достижения конечной вершины d из вершины i за время x при движении по ребру (i, j) .

Алгоритм 6 должен выполняться последовательно по времени (цикл по переменной k), однако расчет вероятностей прибытия $u_i^k(x)$ может быть выполнен параллельно. Реализация этого процесса в CUDA состоит из вызова функции-ядра (Алгоритм 8), рассчитывающей вероятности достижения конечной вершины из каждой вершины графа на заданном временном шаге.

В алгоритме 8 $threadIdx$ – индекс потока, $blockDim$ – размерность блока, $blockIdx$ – индекс блока обработки.

Алгоритм 8: Параллельный алгоритм расчета вероятности прибытия $u_i^k(x)$

```

3: for  $k = 1, 2, \dots, L$  do
3:   process « $N_c, T_c$ »( $k$ ) do
3: end for

4: function process( $k$ ):
6:   /*рассчитать индекс обрабатываемой вершины  $x^*$ */
8:    $x = \text{threadIdx}.x + \text{blockDim}.x * \text{blockIdx}.x$ ;
9:   загрузить  $p(1), \dots, p(t)$  и соответствующие им  $u$  из памяти;
9:   рассчитать  $u_{ix}^k \forall (i, x) \in A$ ;
9:   рассчитать  $u_x^k = \max_i u_{ix}^k$ ;
13: return

```

Таким образом, расчет вероятностей достижения вершины назначения за заданный интервал времени будет выполняться параллельно для каждой вершины графа в отдельном потоке графического акселератора.

4.2.3 Алгоритм нахождения надежного пути с использованием параметрически заданных устойчивых распределений вероятностей

Наиболее вычислительно сложным этапом работы алгоритма нахождения надежного пути является вычисление свертки $\sum_{h=0}^x p_{ij}(h)u_j^{k-1}(x-h)$ в алгоритме 6.

В разделе 4.2.2 в качестве плотности вероятности $p_{ij}(t)$ распределения времени движения на дорожном сегменте использовалось логнормальное распределение. В данном разделе в качестве описания веса сегмента предлагается использовать устойчивое распределение Леви, что позволит заменить операцию свертки на пересчет параметров функции распределения.

В данном разделе предложен алгоритм решения задачи определения надежного пути (57), максимизирующего вероятность прибытия в пункт назначения в течение заранее определенного интервала времени (бюджета поездки). Предлагаемое решение ограничивается рассмотрением не зависящих от времени стохастических сетей. В частности, в следующих подразделах:

- 1) представлено устойчивое распределение Леви для описания времени прохождения сегментов дорожной сети;
- 2) показано, что операция свертки в алгоритме 6 может быть заменена пересчетом параметров функции распределения Леви;

- 3) предложен алгоритм аппроксимации функции $\max_j(\dots)$ в алгоритме б с помощью функции распределения Леви;
- 4) предложен новый алгоритм решения задачи определения надежного пути;
- 5) предложен способ ускорения работы алгоритма путем предварительного вычисления аппроксимаций.

4.2.3.1 Устойчивое распределение Леви

Распределение называется устойчивым, если линейная комбинация двух независимых случайных величин с этим распределением имеет то же распределение с точностью до коэффициента сдвига и масштаба.

Закон распределения случайной величины X называется устойчивым, если для любых идентично распределенных и независимых случайных величин X_1, X_2 и для любых действительных чисел $a > 0, b > 0$ найдутся числа $c > 0, b$ такие, что случайные величины $(aX_1 + bX_2)$ и $cX + d$ распределены одинаково.

В работе для задания веса каждого сегмента $(i, j) \in A$ используется устойчивое распределение Леви. Плотность вероятности распределения Леви для области определения $x \geq \mu$ имеет вид:

$$f(x; \mu, c) = \sqrt{\frac{c}{2\pi}} \frac{e^{-c/2(x-\mu)}}{(x-\mu)^{3/2}}, \quad (64)$$

где μ – коэффициент сдвига;

c – коэффициент масштаба.

Функция распределения имеет вид:

$$F(x; \mu, c) = \operatorname{erfc}\left(\sqrt{c/(2(x-\mu))}\right), \quad (65)$$

где $\operatorname{erfc}(z)$ – функция ошибок.

Пример плотности вероятности и функции распределения для распределения Леви с параметрами $\mu = 0, c = 1$ показан на рисунке 73.

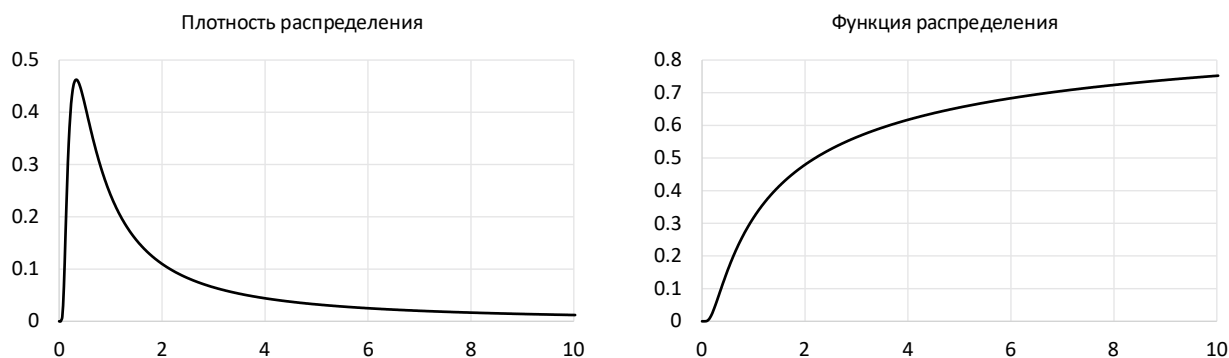


Рисунок 73 – Плотность вероятности и функция распределения для распределения Леви

Если $X_1 \sim \text{Levy}(\mu_1, c_1)$, $X_2 \sim \text{Levy}(\mu_2, c_2)$, то $X_1 + X_2 \sim \text{Levy}(\mu, c)$, где

$$\begin{aligned} \mu &= \mu_1 + \mu_2, \\ |c| &= (\sqrt{c_1} + \sqrt{c_2})^2. \end{aligned} \quad (66)$$

Применительно к распределению времени прохождения сегментов дорожной сети коэффициент сдвига μ определяет минимальное время, за которое можно пройти сегмент, коэффициент масштаба определяет моду: $mode = \frac{c}{3}$.

4.2.3.2 Вычисление свертки и использованием распределений Леви

Покажем, что если время прохождения дорожных сегментов описывается устойчивым распределением Леви, операция свертки в алгоритме б может быть заменена на пересчет коэффициентов сдвига и масштаба распределения Леви.

Рассмотрим подробнее операцию свертки в алгоритме б. Введем обозначение

$$u_{ij}(x) = \sum_{h=0}^x p_{ij}(h) u_j(x-h), \quad (67)$$

где $u_{ij}(x)$ – вероятность достижения конечной вершины d из вершины i за время x при движении по ребру (i, j) .

Тогда выражение (57) можно записать в виде:

$$u_i(x) = \max_j \sum_{h=0}^x p_{ij}(h) u_j(x-h) = \max_j u_{ij}(x). \quad (68)$$

Рассмотрим сначала выражение $u_{ij}(x)$. Учитывая, что $u_d(x) = 1$, для ребер графа, входящих в конечную вершину d , мы можем получить:

$$u_{md}^k(x) = \sum_{h=0}^x p_{md}(h) u_d^{k-1}(x-h) = \sum_{h=0}^x p_{md}(h) = P_{md}(x), \quad (69)$$

$$\forall m \in N: \exists (m, d) \in A,$$

где $P_{md}(x)$ – функция распределения.

То есть для вершин графа $m \in N: (m, d) \in A$, связанных с конечной вершиной d , вероятность достижения конечной вершины можно описать распределением Леви.

Далее, для предыдущих вершин графа $i: (i, m) \in A$ получим:

$$\begin{aligned} u_{im}(x) &= \sum_{h=0}^x p_{im}(h) u_m(x-h) = \sum_{h=0}^x p_{im}(h) \sum_{s=0}^{x-h} p_{md}(s) = \\ &= p_{im}(0) \sum_{s=0}^x p_{md}(s) + p_{im}(1) \sum_{s=0}^{x-1} p_{md}(s) + \dots + p_{im}(x) p_{md}(0) = \\ &= \sum_{l=0}^x p_{im}(l) p_{md}(x-l) + \sum_{l=0}^{x-1} p_{im}(l) p_{md}(x-1-l) + \dots \\ &+ \sum_{l=0}^{x-x} p_{im}(l) p_{md}(x-x-l) = \sum_{n=0}^x \sum_{l=0}^{x-n} p_{im}(l) p_{md}(x-n-l) = \\ &= \sum_{n=0}^x p_{im+md}(x-n) = P_{im+md}(x), \end{aligned}$$

где $p_{im+md}(t)$ – плотность вероятности суммы случайных величин.

Аналогично могут быть посчитаны значения $u_{ij}(x) \forall i, j \in N: (i, j) \in A$, что позволяет заменить операцию свертки в алгоритме б на вычисление значения функции распределения. Значение коэффициентов масштаба и сдвига рассчитываются по формуле (66).

4.2.3.3 Аппроксимация функции максимума

Для определения надежного пути на следующем шаге необходимо получить оценку функции $u_i(x) = \max_j u_{ij}(x)$. Будем аппроксимировать значение $u_i(x)$ функцией распределения Леви. Обозначим аппроксимируемую функцию как $F^*(x)$, искомую функцию как $F(x; \hat{\mu}, \hat{c})$. Тогда задача аппроксимации, в результате решения которой оцениваются искомые параметры $\hat{\mu}$ и \hat{c} , заключается в минимизации ошибки вида:

$$J = \epsilon^2 = \frac{1}{2} \sum_j \left(F^*(x_j) - F(x_j; \hat{\mu}, \hat{c}) \right)^2 \rightarrow \min_{\hat{\mu}, \hat{c}}. \quad (70)$$

Примеры аппроксимации вероятности прибытия $u_i(x)$ представлены на рисунке 74. На рисунке 74 слева показан пример случая для двух возможных маршрутов движения в зависимости от времени прибытия в вершину, справа — для трех возможных маршрутов.

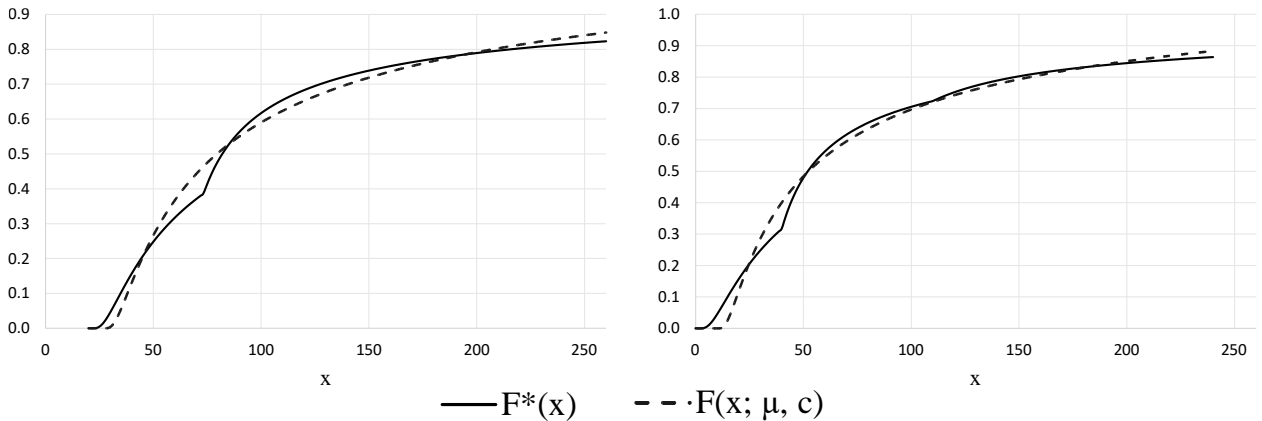


Рисунок 74 – Аппроксимация вероятности прибытия для случая двух и трех связанных вершин

Для минимизации ошибки (70) используется метод градиентного спуска. Для реализации градиентного спуска необходимо вычислить значения частных производных $\frac{\partial}{\partial \mu} F(x; \mu, c)$, $\frac{\partial}{\partial c} F(x; \mu, c)$.

Запишем частные производные в аналитическом виде.

$$\begin{aligned} \frac{\partial}{\partial \mu} F &= \frac{\partial}{\partial \mu} \operatorname{erfc} \left(\sqrt{\frac{c}{2(x-\mu)}} \right) = - \frac{2 \cdot c \cdot e^{\frac{c}{2(x-\mu)}}}{\sqrt{\pi}(2x-2\mu)^2 \sqrt{\frac{c}{2(x-\mu)}}}, \\ \frac{\partial}{\partial c} F &= \frac{\partial}{\partial c} \operatorname{erfc} \left(\sqrt{\frac{c}{2(x-\mu)}} \right) = \frac{e^{\frac{c}{2(x-\mu)}}}{\sqrt{\pi}(2x-2\mu) \sqrt{\frac{c}{2(x-\mu)}}}. \end{aligned} \quad (71)$$

Введем дополнительные обозначения для упрощения вида формул (71). Пусть

$$\beta(x, \mu, c) \triangleq \frac{c}{2(x-\mu)}, \quad \gamma(x, \mu, c) \triangleq \frac{\beta \cdot e^{-\beta}}{c\sqrt{\pi\beta}}. \quad (72)$$

Тогда частные производные (71) примут вид:

$$\begin{aligned}\frac{\partial}{\partial \mu} F &= -\frac{2\beta^2 \cdot e^{-\beta}}{c\sqrt{\pi\beta}} = -2\beta\gamma, \\ \frac{\partial}{\partial c} F &= -\frac{\beta \cdot e^{-\beta}}{c\sqrt{\pi\beta}} = -\gamma.\end{aligned}\tag{73}$$

Таким образом, поиск искомой функции $F(x; \hat{\mu}, \hat{c})$ осуществляется путем минимизации ошибки (70) методом градиентного спуска с использованием соотношений (72), (73). Алгоритм градиентного спуска применительно к рассмотренной задаче (70) выглядит следующим образом (Алгоритм 9).

Алгоритм 9: Метод градиентного спуска

```

1: Входные данные:  $\theta_{init}, \mu_{init}, c_{init}, \varepsilon^{min}, I_0, I_1, M$ 
2: Выходные данные:  $\hat{\mu}, \hat{c}$ 
3:  $\theta = \theta_{init}, \mu = \mu_{init}, c = c_{init}$ ;
4:  $\varepsilon = \sum_j (F^*(x_j) - F(x_j; \mu, c))^2$ ;
5: for ( $i_0 = 1, 2, \dots, I_0$ ) do
6:   for ( $i_1 = 1, 2, \dots, I_1$ ) do
7:     for ( $j = 1, 2, \dots, M$ ) do
8:        $\Delta\mu = \Delta\mu - 2(F(x_j; \mu, c) - F^*(x_j))\beta(x_j; \mu, c)\gamma(x_j; \mu, c)$ ;
9:        $\Delta c = \Delta c - (F(x_j; \mu, c) - F^*(x_j))\gamma(x_j; \mu, c)$ ;
10:    end for
11:     $\mu = \mu - \theta\Delta\mu$ ;
12:     $c = c - \theta\Delta c$ ;
13:  end for
14:   $\varepsilon^{next} = \sum_j (F^*(x_j) - F(x_j; \mu, c))^2$ ;
15:  if  $\varepsilon^{next} > \varepsilon$  do
16:     $\theta = \frac{\theta}{2}$ ;
17:  end if
18:  if  $\varepsilon^{next} < \varepsilon^{min}$  do
19:    break;
20:  end if
21:   $\varepsilon = \varepsilon^{next}$ ;
22: end for
23:  $\hat{\mu} = \mu$ ;
24:  $\hat{c} = c$ ;

```

В алгоритме 9 используются следующие параметры: θ_{init} – начальный шаг, μ_{init}, c_{init} – начальное приближение, ε^{min} – точность расчета, I_0, I_1 – число итераций, M – число используемых значений функции.

С учетом предложенного алгоритма аппроксимации вероятность достижения конечной вершины $u_i(t)$ из вершины i также в итоге описывается функцией распределения Леви.

4.2.3.4 Алгоритм нахождения надежного пути

Для описания алгоритма нахождения надежного пути введем следующие обозначения.

Пусть время прохождения $p_{ij}(t)$ дорожного сегмента $(i, j) \in A, i \in N, j \in N$ описывается функцией распределения Леви с параметрами $P_{ij} = (\mu_{ij}^p, c_{ij}^p)$, вероятность достижения конечной вершины $u_{ij}(t)$ из вершины i за время t при движении по ребру (i, j) описывается функцией распределения Леви с параметрами $U_{ij} = (\mu_{ij}^u, c_{ij}^u)$, вероятность достижения конечной вершины $u_i(t)$ из вершины i за время t описывается функцией распределения Леви с параметрами $U_i = (\mu_i^u, c_i^u)$.

Для определения надежного пути, максимизирующего вероятность прибытия в пункт назначения в течение заранее определенного интервала времени (бюджета поездки), предлагается алгоритм навигации, состоящий из следующих шагов:

1) для каждой помеченной вершины графа $i \in N$ выбираются все исходящие из нее ребра $(i, j) \in A$;

2) считая известными параметры функции распределения U_j достижения конечной вершины d из вершины j и параметры функции распределения P_{ij} времени прохождения сегмента (i, j) , рассчитываются параметры функции распределения U_{ij} достижения конечной вершины d из вершины i при движении по ребру (i, j) пересчетом коэффициентов масштаба и сдвига по формуле (66);

3) по рассчитанным параметрам функций распределения $U_{ij}, i \in N, j \in N, (i, j) \in A$ вычисляются параметры функции распределения U_i путем аппроксимации распределением Леви, как описано в предыдущем подразделе;

4) если рассчитанные параметры функции распределения U_i изменились, вершины $k \in N: (k, i) \in A$, связанные с вершиной i , помечаются для просмотра на следующей итерации алгоритма;

5) если помеченные вершины отсутствуют — алгоритм завершает работу, иначе выполняется следующая итерация алгоритма (переход на шаг 1).

В виде псевдокода предложенный алгоритм может быть записан следующим образом:

Алгоритм 10: Алгоритм решения SOTA с использованием распределений Леви

```

1: vertexes = {}, vertexesNextStep = {};
2: vertexes.add(d);
3: while (!vertexes.isEmpty()) do
4:   for (vertex: vertexes) do
5:     params = {};
6:     for (edge: vertex.outgoingEdges()) do
7:       params.add(convolution(vertex.U, edge.P));
8:     end for
9:     newU = approximate(params);
10:    if (vertex.U != newU) do
11:      vertex.U = newU;
12:      vertex.params = params;
13:      vertexesNextStep.addAll(incomingVertexes(vertex));
14:    end if
15:  end for
16:  vertexes = vertexesNextStep;
17: end while

```

В алгоритме 10 функция *convolution* пересчитывает коэффициенты масштаба и сдвига (шаг 2 алгоритма 10), функция *approximate* выполняет аппроксимацию функции *max* распределением Леви (шаг 3 алгоритма 10).

Следует отметить, что после завершения работы алгоритма каждая вершина i хранит список параметров функций распределения U_{ij} . Это необходимо для адаптивного выбора следующей вершины пути при движении по маршруту в зависимости от оставшегося бюджета поездки.

4.2.3.5 Предварительное вычисление аппроксимаций

Наиболее вычислительно сложным этапом работы алгоритма 10 является аппроксимация распределением Леви на шаге 3. Учитывая, что распределение Леви имеет стандартную форму, обладающую следующим свойством:

$$f(x; \mu, c)dx = f(y; 0, 1)dy,$$

где y определяется как $y = \frac{x-\mu}{c}$, в работе предлагается провести предварительное вычислений аппроксимирующих функций Леви для различных значений коэффициентов сдвига и масштаба.

Пусть Δ – шаг, с которым изменялись коэффициенты сдвига в диапазоне $[0,1]$ и коэффициенты масштаба в диапазоне $[\Delta, 1]$.

Выбор требуемых параметров аппроксимирующей функции состоит из следующих шагов:

1) определение для вершины $i \in N$ набора параметров распределения Леви $\{U_{ij}\}$ для всех связанных вершин $j \in N: (i, j) \in A$ (шаг 3 алгоритма 10);

2) определение минимального и максимального значения аргумента функции распределения:

$$\begin{aligned} x_{min} &= \min_j \mu_{ij}^u, \\ x_{max} &= \max_j \mu_{ij}^u + \alpha c_{ij}^u, \end{aligned} \quad (74)$$

где α – параметр, определяющий рассматриваемый диапазон функции распределения;

3) расчет параметра масштабирования $scale = x_{max} - x_{min}$ и масштабирование параметров распределений Леви:

$$\hat{U}_{ij} = \left(\frac{\mu_{ij}^u - x_{min}}{scale}, \frac{c_{ij}}{scale} \right); \quad (75)$$

4) выбор параметров ближайшей аппроксимирующей функции $\hat{U}^{approx} = (\hat{\mu}^{approx}, \hat{c}^{approx})$ по набору рассчитанных параметров \hat{U}_{ij} ;

5) обратное масштабирование:

$$U^{approx} = (\hat{\mu}^{approx} \cdot scale + x_{min}, \hat{c}^{approx} \cdot scale). \quad (76)$$

Результатом аппроксимации будут являться параметры U^{approx} .

В разделе 4.2.5 представлены экспериментальные исследования алгоритма нахождения надежного пути в стохастической транспортной сети на основе прямого вычисления сверток и с использованием параметрически заданных устойчивых распределений вероятностей.

4.2.4 Алгоритм нахождения оптимального пути на общественном транспорте

Помимо нахождения оптимального маршрута движения на индивидуальном транспорте, интерес для участников дорожного движения представляет задача расчета оптимального пути на общественном транспорте в стохастической сети. Предложенный в

данном разделе алгоритм учитывает время ожидания общественного транспорта на основе актуальной и прогнозной информации о движении транспортных средств.

Транспортную сеть будем рассматривать как ориентированный граф $G = (N, E)$, вершины $n_i \in N$ которого соответствуют остановкам, ребра $e_{ij} \in E, i \in V, j \in V$ – сегментам транспортной сети с длиной $|e_{ij}|$, причем каждая остановка представляется как минимум тремя вершинами графа.

Введем следующие обозначения:

- S_k – остановка из множества S ;
- R_i – маршрут пассажирского транспорта из множества R .

Часть графа, представляющего два маршрута R_1 и R_2 , содержащих общую остановку S_k , показана на рисунке 75.

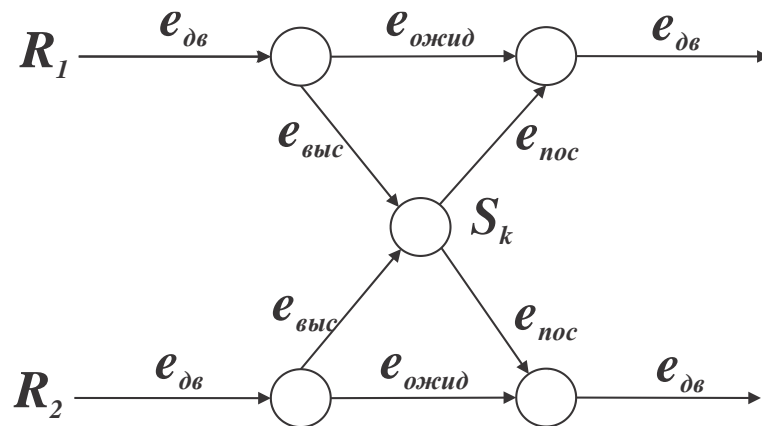


Рисунок 75 – Структура графа маршрутной сети

Граф содержит ребра следующих типов:

- $e_{дв}$ – представляет участок маршрута между остановками, характеризуется средним временем μ_{ij} и дисперсией прохождения σ_{ij} ;
- $e_{ожид}$ – ожидание посадки / высадки пассажиров на остановке, в работе в качестве веса ребра используется константное время ожидания;
- $e_{выс}$ – ребро высадки пассажиров, в работе в качестве веса ребра используется константное время высадки;
- $e_{пос}$ – ребро посадки (ожидания транспортного средства), определяется временем прибытия транспортного средства соответствующего маршрута на остановку;
- $e_{пеш}$ (не представлено на рисунке) – переход пешком от остановок до пунктов отправления / прибытия, вес ребра зависит от расстояния перехода.

Задача нахождения маршрута движения сводится к задаче нахождения кратчайшего пути в графе. Для нахождения кратчайшего пути в зависящей от времени стохастической транспортной сети используется модифицированный алгоритм Дейкстры. Предложенный алгоритм позволяет находить оптимальный путь в стохастической сети с учетом прогнозной информации о времени прибытия транспортных средств на остановочные пункты. Стохастические свойства транспортной сети учитываются благодаря использованию не только среднего времени движения по сегменту сети, но и дисперсии времени движения. В виде псевдокода алгоритм представлен в блоке Алгоритм 11.

В алгоритме 11 в качестве метки вершины l_i используется пара $(n_i, cost_i)$, содержащая вершину n_i и многомерную цену маршрута $cost_i$, которую необходимо затратить для достижения вершины n_i из вершины отправления n_s . Цена маршрута представляется в виде пары $cost_i = (c_i, t_i)$, где c_i – обобщенная цена, t_i – время поездки.

Очередь с приоритетом pq определяет следующую вершину для просмотра на основе обобщенной цены c_i . Ассоциативные массивы $predMaps$ и $costsMap$ используются для хранения предшествующей вершины для заданной и цены поездки до заданной вершины соответственно.

В алгоритме 11 функции $getCost()$ и $getTime()$ – получение обобщенной цены c_i и времени t_i из цены достижения вершины $cost_i$ соответственно.

Алгоритм 11: Алгоритм нахождения кратчайшего пути

1: **Входные данные:** вершина отправления n_s , вершина прибытия n_d , время отправления t

2: **Выходные данные:** кратчайший путь

// Инициализация

3: $PriorityQueue pq = \emptyset$;

4: $Map predMap = \emptyset$;

5: $Map costsMap = \emptyset$;

6: $Label l_s = Label(n_s, cost_s)$;

7: $pq.insert(l_s)$;

8: **while** ($!pq = \emptyset$) **do**

9: $Label l_i = pq.pop()$;

10: $n_i = l_i.getNode()$;

11: **for** (*each* $e_{ij} \in E$ *do*) **do**

12: $cost_j = calculateCost(e_{ij}, t + l_i.getCost().getTime())$;

13: **if** $cost_j.getCost() > costsMap.get(n_i).getCost()$ **then**

11: $continue$;

14: **end if**

15: $costsMap.put(n_j, cost_j)$;

16: $predMap.put(n_j, n_i)$;

```

17:   Label  $l_j = Label(n_j, cost_j)$ ;
18:    $pq.insert(l_j)$ ;
19: end for
20: end while

```

Основной интерес представляет функция получение оценки цены прохождения ребра графа $calculateCost()$. Предлагаемый способ определения цены ребра в зависимости от его типа описан в алгоритме 12. В алгоритме 12 используются следующие обозначения:

- $t_{wait} = 30$ (секунд) – время высадки / посадки;
- $t_{out} = 10$ (секунд) – время высадки;
- t_{arr} – прогнозируемое время прибытия транспортного средства на остановку;
- $s_{walk} = 1$ (м/с) – скорость движения пешком;
- $\alpha_0, \alpha_1, \alpha_2$ – параметры учета индивидуальных предпочтений, влияющие на выбор

надежного пути, числа пересадок и расстояния пешком.

Алгоритм 12: Расчет функции $calculateCost()$

```

1: Входные данные: ребро графа  $e_{ij}$ , время прибытия  $t$  в вершину  $i$ 
2: Выходные данные: цена  $cost_j = (c_j, t_j)$ 
3: if ( $e_{ij}$  is  $e_{дв}$ ) then
4:   // сегмент маршрута между остановками
5:   return ( $\mu_{ij} + \alpha_0 \sqrt{\sigma_{ij}}, \mu_{ij}$ );
6: end if
7: else if ( $e_{ij}$  is  $e_{ожид}$ ) then
8:   // высадка / посадка
9:   return ( $t_{wait}, t_{wait}$ );
10: end if
11: else if ( $e_{ij}$  is  $e_{выс}$ ) then
12:   // высадка
13:   return ( $t_{out}, t_{out}$ );
14: end if
15: else if ( $e_{ij}$  is  $e_{пос}$ ) then
16:   // ожидание транспорта
17:   return ( $\alpha_1(t_{arr} - t), t_{arr} - t$ );
18: end if
19: else if ( $e_{ij}$  is  $e_{пеш}$ ) then
20:   // переход пешком
21:   return ( $\alpha_2 |e_{ij}| / s_{walk}, |e_{ij}| / s_{walk}$ );
22: end if

```

4.2.5 Экспериментальные исследования алгоритмов нахождения надежного пути

В данном разделе последовательно проводятся экспериментальные исследования алгоритма нахождения надежного пути в стохастической сети на основе прямого вычисления сверток, в т.ч., с использованием графического акселератора (раздел 4.2.5.1), а также алгоритма, основанного на использовании параметрически заданных устойчивых распределений вероятностей Леви (раздел 4.2.5.2).

4.2.5.1 Экспериментальный анализ алгоритма на основе прямого вычисления сверток

Целью проводимых экспериментальных исследований является не только демонстрация работоспособности предложенного алгоритма, но и доказательство его превосходства над известным алгоритмом-прототипом, представленном в [217] и кратко описанном в разделе 4.2.1.

Экспериментальные исследования разработанного алгоритма проводились для части улично-дорожной сети г. Самары. Рассматриваемая дорожная сеть состоит из 9721 вершин и 26018 сегментов. В качестве веса дорожного сегмента использовались данные о времени прохождения сегмента, усреднённые за десятиминутный интервал.

Для экспериментальных исследований производилось разбиение графа дорожной сети на подграфы по территории размером 1 км². Каждый подграф содержал в среднем 100 ребер. Число используемых в векторе признаков архивных значений параметров транспортных потоков для каждого сегмента сети $M = 6$, значение временного интервала $\Delta = 10$ минут, т.е. вектор признаков содержит архивные данные за последний час.

Для сравнения алгоритмов были выбраны 6 пар различных вершин отправления-прибытия на графе сети, после чего для каждой пары решалась задача навигации, варьируя время отправления и временной ресурс. Каждая из таких задач решалась предлагаемым алгоритмом и известным алгоритмом. Всего было проведено 165 экспериментов. Прогноз плотности вероятности по времени для сравниваемых алгоритмов выполнялся одинаковым способом с использованием логнормального распределения. Перестроение маршрута проводилось каждые 5 или каждые 10 минут.

Гистограмма распределения разности времени движения, полученного предложенным алгоритмом (для различных временных интервалов) и алгоритмом-прототипом, показана на рисунке 76. Положительная разность соответствует выигрышу предложенного алгоритма, отрицательная - проигрышу.

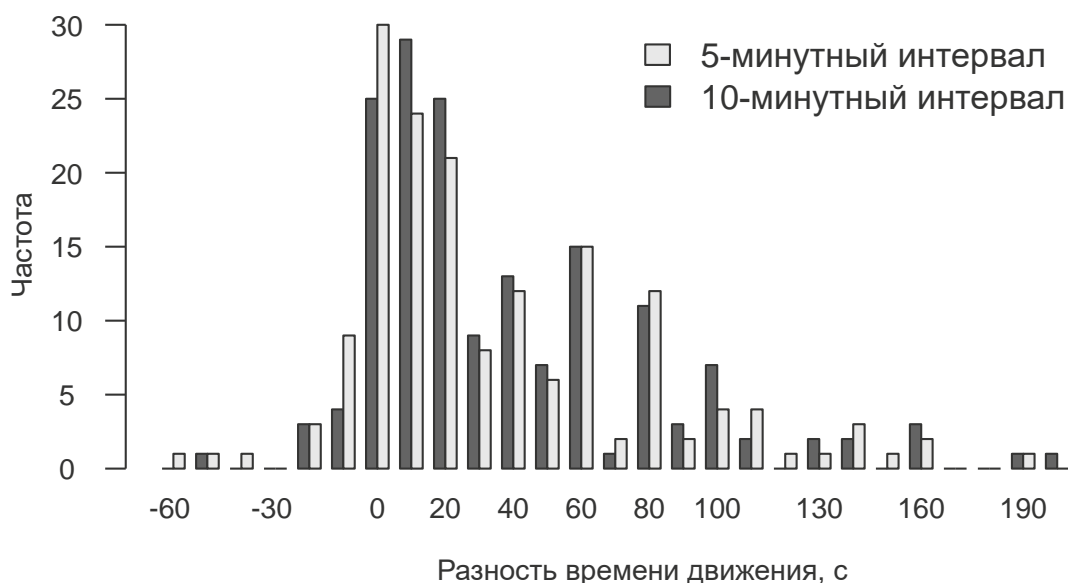


Рисунок 76 – Сравнение алгоритмов по фактически затраченному ресурсу времени на навигацию транспортного средства

Для наглядного сравнения алгоритмов по надёжности определяемого кратчайшего пути частоты выигрыша (положительные значения разности), проигрыша (отрицательные значения) и совпадения (нулевая разность по времени) сведены в таблицу 25.

Таблица 25 – Результаты сравнения алгоритмов по надёжности

–	5-минутный интервал	10-минутный интервал
Проигрыш предложенного	0,091	0,048
Выигрыш предложенного	0,727	0,8
Совпадают	0,182	0,152

Как следует из представленных результатов предложенный алгоритм существенно чаще (в 8-17 раз) выполняет навигацию транспортного средства с меньшими временными затратами.

На следующем этапе экспериментального анализа было проведено исследование эффективности вычисления надежного пути с использованием GPU. Для сравнения времени работы базовой и параллельной реализаций алгоритма нахождения надежного пути были выбраны 6 пар различных вершин отправления-прибытия на графе дорожной сети, после чего задача навигации решалась для каждой пары вершин и различных дней недели, времени начала движения и бюджета поездки. Вершины были выбраны таким образом, чтобы среднее время поездки составляло от 15 до 60 минут. Всего было проведено 6300 экспериментов.

Характеристики используемой ПЭВМ: процессор Intel Core i7-9700K 3.60 GHz, оперативная память 64 ГБ, графический акселератор GeForce RTX 2080 Ti. Среднее время работы алгоритмов представлено в таблице 26. Реализация алгоритма на CUDA позволяет сократить время вычислений в среднем в 5 раз.

Таблица 26 – Среднее время работы алгоритма

–	Базовый алгоритм	Реализация с использованием CUDA
Время работы, с	20,58	3,88

Время работы алгоритмов зависит от выбранного бюджета поездки, т.е. количества временных шагов алгоритма. График зависимости времени работы от бюджета поездки показан на рисунке 77.

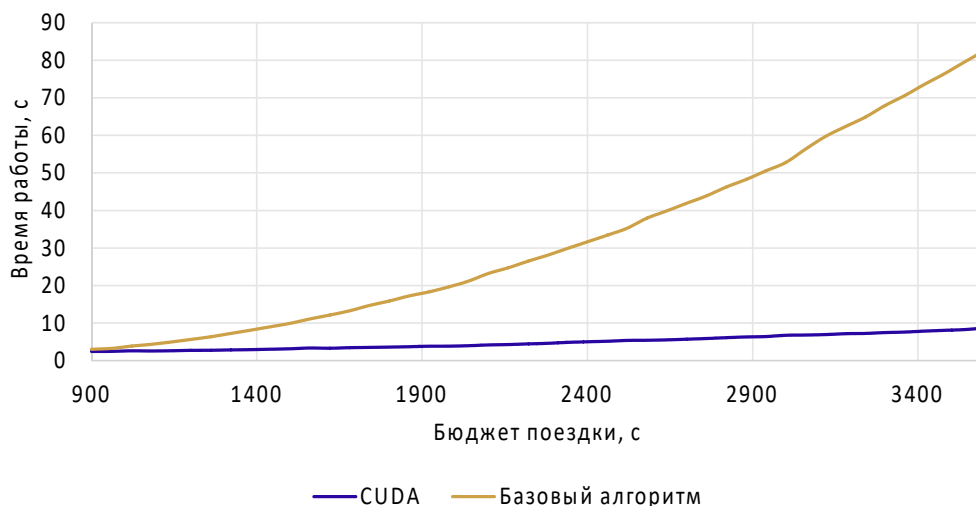


Рисунок 77 – Зависимость времени работы алгоритма от бюджета поездки

При увеличении бюджета поездки выигрыш параллельного алгоритма по времени работы увеличивается.

Подробный анализ времени работы параллельного алгоритма позволяет сделать вывод, что большую часть времени работы алгоритма занимает обмен данными хост-компьютера с устройством.

4.2.5.2 Экспериментальные исследования алгоритма на основе параметрически заданных устойчивых распределений вероятностей

На данном этапе экспериментального анализа исследовался алгоритм нахождения надежного пути на основе параметрически заданных устойчивых распределений

вероятностей. Целью экспериментальных исследований является сравнение результатов работы алгоритмов маршрутизации 6 (базовый алгоритм с использованием прямого вычисления сверток) и 10 (предложенный алгоритм), вычисляющих надежность пути через точное вычисление сверток и с помощью пересчета параметров функции Леви. Для сравнения алгоритмов необходимо оценить время работы процедуры маршрутизации построения надежного пути, а также оценить полученные маршруты движения.

На первом этапе исследований была проведена оценка ошибки аппроксимации целевой функции распределением Леви (рисунок 74) по критерию среднеквадратического отклонения:

$$\text{RMSE} = \frac{1}{m} \sum_{i=1}^m \sqrt{\frac{1}{n} \sum_{j=1}^n \left(F_i^*(x_j) - F_i(x_j; \hat{\mu}_i, \hat{c}_i) \right)^2}, \quad (77)$$

где n – количество используемых отсчетов;

m – количество аппроксимаций.

Ошибки аппроксимации считались отдельно для случаев, когда ребро графа (i, j) связано с двумя соседними ребрами (т.е. существует два возможных маршрута движения из вершины j , исключая движение в обратном направлении) и тремя соседними ребрами. Результаты приведены в таблице 27.

Таблица 27 – Среднеквадратическое отклонение

–	RMSE	Количество аппроксимаций
2 соседних ребра	0,0425	1694315
3 соседних ребра	0,0418	968040

Далее оценивались результаты работы алгоритмов маршрутизации.

Для проведения экспериментальных исследований разработанного алгоритма использовалась крупномасштабная улично-дорожная сеть г. Самара, состоящая из 47274 дорожных сегментов и 18582 вершин. Часть улично-дорожной сети г. Самара показана на рисунке 78.



Рисунок 78 – Часть улично-дорожной сети г. Самара

Для определения параметров распределений времени прохождения сегментов дорожной сети использовались усредненные за десятиминутный интервал данные о скорости прохождения дорожных сегментов за два месяца. Для оценки прогнозного времени движения использовалось среднее время прохождения сегментов за два месяца, для оценки актуального времени прохождения использовались данные за конкретный день.

Для сравнения разработанного и базового алгоритмов по качеству решения задачи маршрутизации были выбраны 6 пар различных вершин отправления-прибытия на графе дорожной сети, после чего задача навигации решалась для каждой пары вершин и различных дней недели, времени начала движения и бюджета поездки. Вершины были выбраны таким образом, чтобы среднее время поездки составляло от 15 до 50 минут. Для каждого набора параметров задача решалась предлагаемым алгоритмом 10 и базовым алгоритмом 6. Всего было проведено 6300 экспериментов.

Гистограммы распределения разности времени движения, полученного путем решения задачи навигации предложенным алгоритмом и базовым алгоритмом на основе операции вычисления свертки, показаны на рисунке 79. Гистограммы приведены для прогнозного и актуального времени движения. Положительная разность соответствует проигрышу предложенного алгоритма (т.е. время движения по маршруту, рассчитанному предложенным алгоритмом, больше, чем время движения по маршруту, вычисленному базовым алгоритмом), отрицательная – выигрышу. В данном эксперименте рассматривались ситуации, когда время движения по маршрутам, найденным базовым и предложенным алгоритмами, не превышает бюджет поездки.

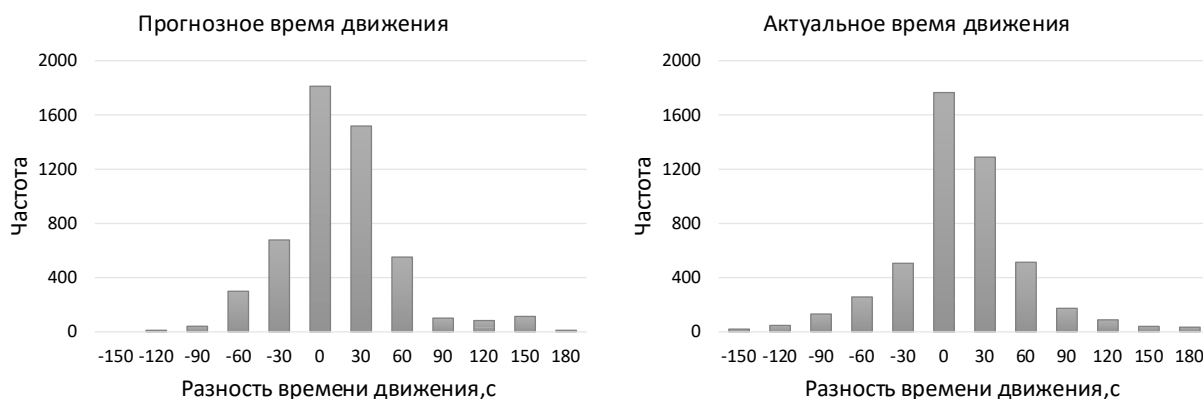


Рисунок 79 – Сравнение алгоритмов по фактическому времени движения транспортных средств

Как видно из представленных гистограмм, в большинстве случаев предложенный алгоритм показывает тот же результат, что и алгоритм на основе вычисления свертки, либо предлагает маршрут движения с большими временными затратами. Среднее время задержки на маршруте, найденном предложенным алгоритмом, приведено в таблице 28.

Таблица 28 – Среднее время задержки

–	Прогнозное время движения	Актуальное время движения
Среднее время задержки, с	17,5	13,9

Далее оценивалось, насколько предложенный маршрут движения укладывался в требуемый бюджет поездки. Гистограмма распределения количества поездок в рамках временного бюджета (прибытие в указанный интервал) и вне его (прибытие вне интервала) показана в таблице 29 и на рисунке 80. Всего было проведено 6300 экспериментов. Из полученных результатов можно сделать вывод, что предложенный алгоритм ведет к нахождению маршрута движения вне выбранного бюджета чаще базового алгоритма примерно на 9%.

Таблица 29 – Сравнение алгоритмов по затраченному временному ресурсу относительно бюджета поездки

–	Пересчет параметров	Вычисление свертки
Вне интервала	0,154	0,064
В рамках интервала	0,846	0,936

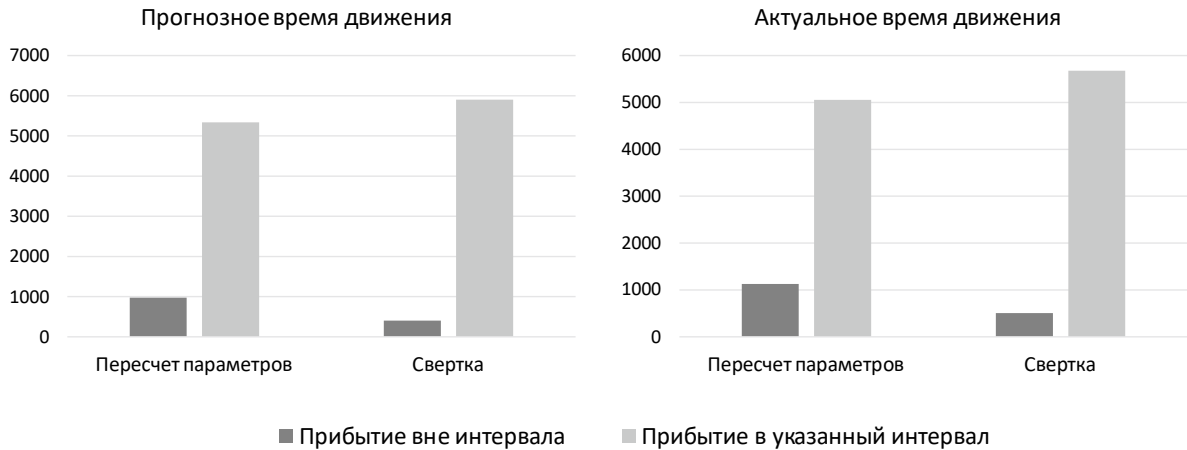


Рисунок 80 – Сравнение алгоритмов по затраченному временному ресурсу относительно бюджета поездки

На заключительном этапе экспериментального анализа проводилось измерение времени работы алгоритмов. Следует отметить, что время работы алгоритма-прототипа зависит от бюджета поездки, который определяет количество итераций. Число итераций разработанного алгоритма не зависит от заданного бюджета поездки. Среднее время работы алгоритмов представлено в таблице 30. Характеристики используемой ПЭВМ: процессор Intel Core i5-3740 3.20 GHz, оперативная память 16 ГБ, ОС – Windows 8.1.

Таблица 30 – Сравнение времени работы алгоритмов

–	Пересчет параметров	Вычисление свертки
Время работы, мс	606	23625

Подробнее среднее время работы базового алгоритма в зависимости от используемого бюджета поездки показано на рисунке 81.

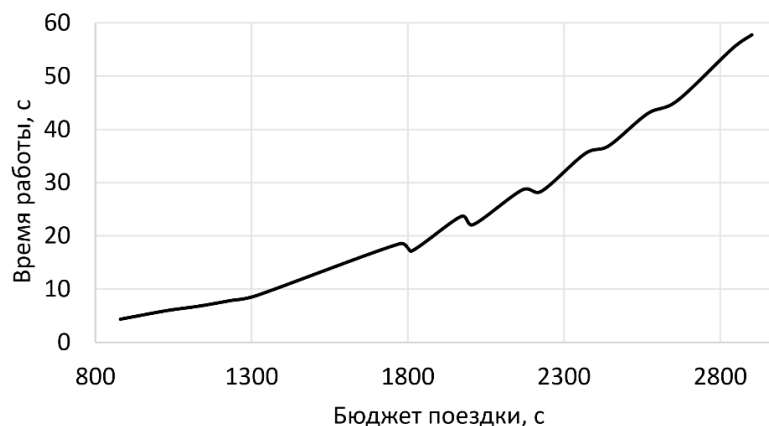


Рисунок 81 – Время работы базового алгоритма

Время работы разработанного алгоритма примерно в 40 раз меньше времени работы базового алгоритма и составляет в среднем 606 миллисекунд, что позволяет использовать алгоритм для решения задачи нахождения надежного пути в стохастической сети в режиме реального времени.

4.3 Маршрутизация подключенных транспортных средств в сети

Существующие алгоритмы поиска пути, реализованные в ИТС, картографических сервисах или бортовых навигационных системах, в основном позволяют находить кратчайшие маршруты на основе текущей и прогнозной информации о распределении транспортных потоков (обзор алгоритмов подробнее описан в разделах 1.4.2 и 1.4.3). Одним из примеров постановки навигационной задачи является задача нахождения надежного пути, рассмотренная в разделе 4.2. Однако следует отметить, что предоставление схожих маршрутов движения и информации о состоянии транспортных потоков разным водителям может приводить к образованию новых заторов, т.к. большинство водителей будут выбирать менее загруженные маршруты движения. Такое поведение приводит к колебаниям состояния сети и ухудшает транспортную ситуацию в целом. Постепенное развитие автономных и подключенных транспортных средств позволяет решать задачу минимизации времени движения с точки зрения эффективного распределения транспортных средств в сети. Такое распределение позволит уменьшить уровень дорожных заторов и сократить общее время поездок в сети.

В разделе 4.3.1 предложен алгоритм маршрутизации подключенных транспортных средств в транспортной сети в контексте интеллектуальной транспортной системы на основе численного метода резервирования маршрутов, позволяющий достичь транспортного равновесия в сети. В разделе 4.3.2 предложена модификация алгоритма резервирования для его применения в стохастической транспортной сети. Данный алгоритм для оценки времени прохождения сегмента учитывает не только предполагаемую загрузку дорожного сегмента, но и стохастические свойства сети, т.е. рассматривает не только среднее время прохождения сегмента, но и дисперсию времени. В разделе 4.3.3 предложен алгоритм маршрутизации транспортных средств в гетерогенном потоке с адаптивным регулированием, учитывающий прогнозные значения параметров транспортного потока.

4.3.1 Алгоритм резервирования маршрутов движения транспортных средств

В данном разделе рассматривается задача маршрутизации подключенных транспортных средств в контексте интеллектуальной транспортной системы.

Предполагается, что каждое транспортное средство взаимодействует с единой системой построения маршрутов. Исследуется алгоритм маршрутизации, в котором предполагается, что скорость прохождения дорожных сегментов зависит от количества транспортных средств, зарезервировавших слоты на выбранном сегменте в выбранный момент времени. Предложенная схема предполагает возможность перестроения маршрута в процессе движения.

4.3.1.1 Формулировка проблемы

Улично-дорожную сеть будем рассматривать как ориентированный граф $G = (V, E)$, в котором вершины V , $N_V = |V|$ соответствуют перекрёсткам дорожной сети, ребра E , $N_E = |E|$ соответствуют сегментам дорожной сети между перекрёстками. Каждый дорожный сегмент $(i, j) \in E, i \in V, j \in V$ описывается следующими параметрами: длина дорожного сегмента λ_{ij} , число полос N_{ij} , количество транспортных средств на сегменте r_{ij} , максимальное количество транспортных средств r_{ij}^{\max} , соответствующее критической плотности потока на сегменте.

Пусть U – множество транспортных средств. Для каждого транспортного средства $u_k \in U$ считаются известными вершины отправления O_k и назначения D_k , а также время начала движения τ_k .

Задача нахождения наименьшего ожидаемого времени прибытия

Рассмотрим отдельное транспортное средство $u_k \in U$ с известными вершинами отправления-прибытия O, D и временем начала движения τ .

Пусть p_h обозначает h -й путь из вершины отправления O к вершине назначения D , $p_h = (v_0^h, v_1^h), (v_1^h, v_2^h), \dots, (v_{L_h-1}^h, v_{L_h}^h)$, где $v_j^h \in V$ – j -я посещённая вершина в h -м пути, $v_0^h = O, v_{L_h}^h = D, L_h$ – количество вершин в h -м пути.

Время прохождения дорожного сегмента $(v_i, v_j) \in E$ между вершинами $v_i \in V$ и $v_j \in V$ в момент времени t обозначим как $c_{v_i, v_j}(t)$.

Обозначим $d_{v_j}^h$ как время прибытия в вершину v_j при движении по пути p_h . Тогда время прибытия в каждую вершину может быть записано в следующем виде:

(78)

$$d_D^h = d_{v_{L_h-1}^h}^h + c_{v_{L_h-1}^h, D} (d_{v_{L_h-1}^h}^h).$$

Тогда задача маршрутизации заключается в нахождении пути с наименьшим ожидаемым временем прибытия в вершину назначения и может быть записана как:

$$d_D^* = \min_{p_h} d_D^h. \quad (79)$$

4.3.1.2 Алгоритм резервирования маршрутов

По сути, задача маршрутизации одного транспортного средства заключается в нахождении кратчайшего пути в зависящем от времени графе. Однако в рассматриваемой в разделе задаче необходимо осуществлять маршрутизацию с учётом маршрутов движения других транспортных средств в транспортной сети.

Для решения этой задачи в работе предлагается использовать алгоритм резервирования маршрутов в рамках интеллектуальной транспортной системы. Каждый сегмент дорожной сети дискретизируется во временные интервалы с шагом дискретизации T_{discr} . Для каждого временного интервала хранится оценка количества транспортных средств, которые будут находиться на дорожном сегменте в выбранный интервал времени при движении по заданному маршруту.

Общая схема процедуры маршрутизации, таким образом, состоит из следующих шагов:

1) когда транспортное средство планирует начать поездку, оно отправляет в ИТС координаты точки отправления (текущего положения транспортного средства), координаты точки назначения и время начала поездки, чтобы получить маршрут движения;

2) учитывая текущее состояние резервирования дорожных сегментов, ИТС решает задачу нахождения кратчайшего пути в зависящей от времени транспортной сети и возвращает путь движения транспортному средству;

3) одновременно ИТС обновляет состояние резервирования каждого дорожного сегмента, входящего в кратчайший путь, для временных интервалов, в которых, как ожидается, транспортное средство будет находиться на выбранном сегменте, если будет следовать указанному маршруту с указанной скоростью движения. Скорость движения на дорожном сегменте рассчитывается исходя из текущего состояния резервирования.

Очевидно, что предположение о том, что все транспортные средства будут двигаться с указанной скоростью, является невыполнимым, поэтому на практике ожидаются значительные отклонения наблюдаемой дорожной ситуации от прогнозируемой.

Для уменьшения отклонения предлагается использовать процедуру перестроения маршрута в процессе движения. Такой подход применяется в моделях распределения транспортных потоков [364].

В следующем разделе алгоритмы маршрутизации будут описаны более формально.

4.3.1.3 Алгоритм резервирования маршрутов

Обозначим $n_{ij}(t)$ общее количество транспортных средств, зарезервировавших временной слот t на дорожном сегменте $(i, j) \in E$. Тогда переменная $p_{ij}(t) = n_{ij}(t) / (\lambda_{ij} N_{ij})$ обозначает мгновенную плотность транспортного потока на дорожном сегменте (i, j) в момент времени t .

Время прохождения дорожного сегмента $c_{v_i, v_j}(t)$ (и, соответственно, скорость) напрямую зависит от плотности транспортного потока на сегменте. В [365] приведён обзор основных детерминированных соотношений скорости и плотности транспортного потока. В данном разделе для оценки скорости и времени прохождения дорожного сегмента использовалась линейная модель Гриншилда (Grinshield) c_{ij}^{Gr} (80), модель Андервуда (Underwood) c_{ij}^{Und} (81) и BPR-соотношение c_{ij}^{BPR} (82), являющееся стандартным соотношением в моделях распределения транспортных потоков [366].

$$c_{ij}^{Gr}(t) = t_{ij}^f / \left(1 - \frac{p_{ij}(t)}{p_{ij}^{jam}} \right), \quad (80)$$

$$c_{ij}^{Und}(t) = t_{ij}^f / \exp \left(- \frac{p_{ij}(t)}{p_{ij}^{jam}} \right), \quad (81)$$

$$c_{ij}^{BPR}(t) = t_{ij}^f \left(1 + \alpha \left(\frac{p_{ij}(t)}{p_{ij}^{jam}} \right)^\beta \right), \quad (82)$$

где t_{ij}^f – время прохождения дорожного сегмента в свободном потоке;

p_{ij}^{jam} – плотность потока на дорожном сегменте (i, j) , соответствующая дорожному затору.

Введем дополнительное обозначение $T_{reroute}$ – интервал перестроения маршрута транспортного средства. Процедура маршрутизации состоит из двух частей: отправка

координат текущего положения транспортного средства каждые $T_{reroute}$ секунд для построения / перестроения маршрута движения, и непосредственно расчёт пути в ИТС.

Алгоритм резервирования маршрутов (RRA) состоит из следующих шагов (Алгоритм 13):

1) если для выбранного транспортного средства маршрут движения был рассчитан – удалить транспортное средство из зарезервированного трафика на каждом сегменте, входящего в маршрут движения;

2) рассчитать кратчайший путь между вершинами отправления и назначения, используя алгоритм A^* . Время прохождения дорожных сегментов рассчитывается исходя из аккумулярованного трафика на сегменте с помощью выбранного соотношения скорости и плотности транспортного потока;

3) обновить зарезервированный трафик для дорожных сегментов, входящих в новый маршрут движения.

В алгоритме используются следующие обозначения: τ_{in} и τ_{out} – время въезда на дорожный сегмент и время выезда с дорожного сегмента соответственно, $[z]$ – целая часть числа z , $A^*(O, D, n)$ – расчёт кратчайшего пути алгоритмом A^* из вершины O в вершину D с учётом состояния загрузки транспортной сети n .

Алгоритм 13: Алгоритм резервирования маршрутов (RRA)

```

1: Входные данные:  $O, D, \tau, k$ 
2: if  $p_k = \emptyset$  then // Очистить зарезервированный трафик
3:   for  $(v_i, v_j) \in p_k$  do
4:      $\tau_{in} = \left\lceil \frac{d(v_i)}{T_{discr}} \right\rceil$ ;
5:      $\tau_{out} = \left\lceil \frac{d(v_j)}{T_{discr}} \right\rceil$ ;
6:     for  $t = \tau_{in}, \tau_{out}$  do
7:        $n_{v_i, v_j}(t) = n_{v_i, v_j}(t) - 1$ ;
8:     end for
9:   end for
10: end if
11:  $p_k = A^*(O, D, n)$ ; // Обновить маршрут
12: for  $(v_i, v_j) \in p_k$  do // Обновить зарезервированный трафик
13:    $\tau_{in} = \left\lceil \frac{d(v_i)}{T_{discr}} \right\rceil$ ;
14:    $\tau_{out} = \left\lceil \frac{d(v_j)}{T_{discr}} \right\rceil$ ;
15:   for  $t = \tau_{in}, \tau_{out}$  do
16:      $n_{v_i, v_j}(t) = n_{v_i, v_j}(t) + 1$ ;
17:   end for
18: end for

```

В следующем разделе предложена модификация данного алгоритма маршрутизации транспортных средств на основе численного метода резервирования маршрутов, позволяющие достичь транспортного равновесия в сети. Модифицированный алгоритм учитывает стохастические свойства транспортной сети.

4.3.2 Алгоритм резервирования маршрутов движения транспортных средств в стохастической транспортной сети

В данном разделе предлагается модификации алгоритма резервирования маршрутов в многоагентной децентрализованной системе маршрутизации, в которой каждое транспортное средство представляется агентом, взаимодействующим с дорожно-транспортной инфраструктурой. Для оценки времени прохождения сегмента в разделе предлагается учитывать стохастические свойства сети, т.е. рассматривать не только среднее время прохождения сегмента, но и дисперсию времени. Разрабатывается стохастическая модель соотношений скорости и плотности транспортного потока в предположении, что характеристики сегмента зависят от количества зарезервированных слотов на сегменте. Для уменьшения отклонения между прогнозным и наблюдаемым временем движения и достижения транспортного равновесия в сети предлагается использовать итеративную процедуру.

Для учета стохастических свойств транспортной сети в разделе 4.3.2.1 предложен алгоритм оценки скорости движения по плотности потока, алгоритм оценки дисперсии скорости движения описан в разделе 4.3.2.2, в разделе 4.3.2.3 предложен гибридный алгоритм оценки скорости движения транспортных средств.

4.3.2.1 Оценка скорости движения по плотности потока

Скорость прохождения дорожного сегмента $V_{ij}(t)$ напрямую зависит от плотности транспортного потока на сегменте. В разделе 4.3.1 при решении задачи маршрутизации с использованием метода резервирования маршрутов лучший результат для оценки скорости был показан с использованием модели Андервуда (Underwood) (согласно экспериментальным исследованиям, раздел 4.3.4.1):

$$V^{Und}(p) = V_f \exp\left(-\frac{p}{p^{jam}}\right), \quad (83)$$

где V_f – скорость прохождения дорожного сегмента в свободном потоке;

p^{jam} – плотность потока, соответствующая дорожному затору.

В [102] для оценки скорости движения используется 5-параметрическая логистическая модель (5PL) следующего вида:

$$V^{5PL}(p) = V_b + \frac{V_f - V_b}{(1 + \exp((p - p_b)/\theta_1))^{\theta_2}}, \quad (84)$$

где V_b – скорость прохождения дорожного сегмента в условиях дорожного затора;

p_b – значение плотности потока, соответствующей переходу из свободного движения к затрудненному движению;

θ_1, θ_2 – параметры, определяющие форму кривой.

Пример исходных данных, полученных в результате моделирования движения в нерегулируемой тестовой сети малого размера [328], и моделей оценки скорости движения показан на рисунке 82.

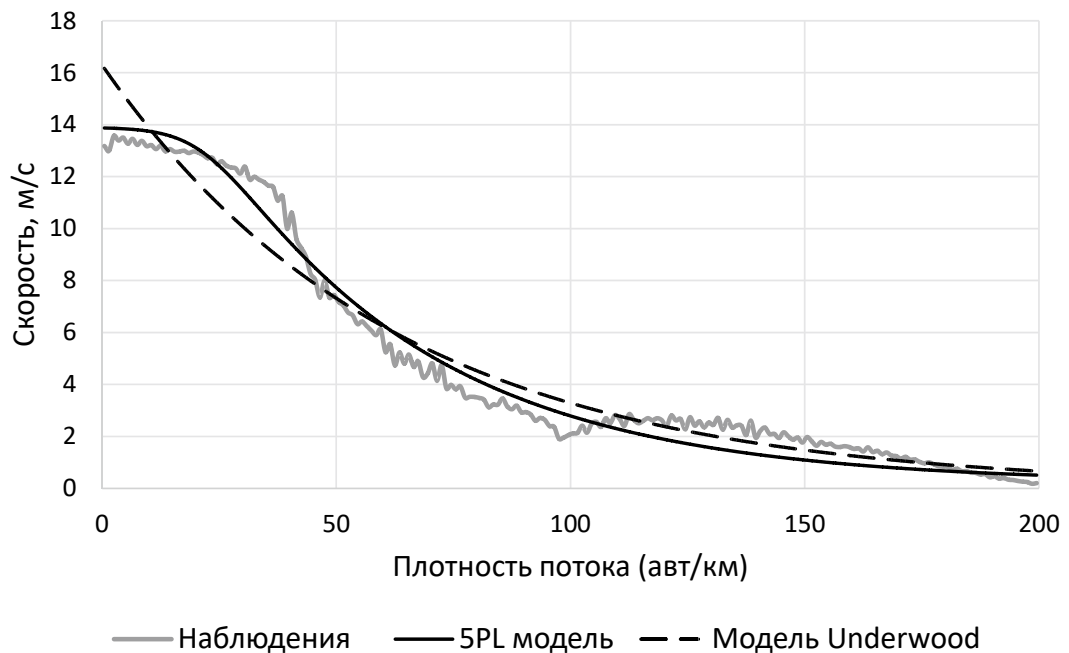


Рисунок 82 – Оценка скорости движения

Для оценки параметров соотношений (83) и (84) использовался итеративный алгоритм Левенберга-Марквардта. В качестве исходных данных о скорости/плотности транспортного потока использовались данные микроскопического моделирования движения транспортных средств в пакете моделирования SUMO [367].

4.3.2.2 Оценка дисперсии скорости движения

Задача оценивания дисперсии скорости движения в зависимости от значения плотности транспортного потока рассматривалась в [250]. Была предложена зависимость следующего вида:

$$\sigma^{5PL}(p) = \delta^2 \left(1 + \tau V^{5PL}(p) (V_f - V^{5PL}(p)) \right), \quad (85)$$

где δ^2, τ – параметры модели.

В данном разделе предлагается использовать логнормальное соотношение (Log-Normal) следующего вида:

$$\sigma^{LN}(p) = \frac{\vartheta_1}{p\delta} \exp\left(-\frac{(\vartheta_2 \ln(p) - \mu)}{2\delta^2}\right), \quad (86)$$

где $\vartheta_1, \vartheta_2, \delta^2, \mu$ – параметры модели.

Для оценки параметров соотношений (85) и (86) также использовался итеративный алгоритм Левенберга-Марквардта. Пример исходных данных зависимости дисперсии скорости от плотности транспортного потока и моделей оценки дисперсии показан на рисунке 83.

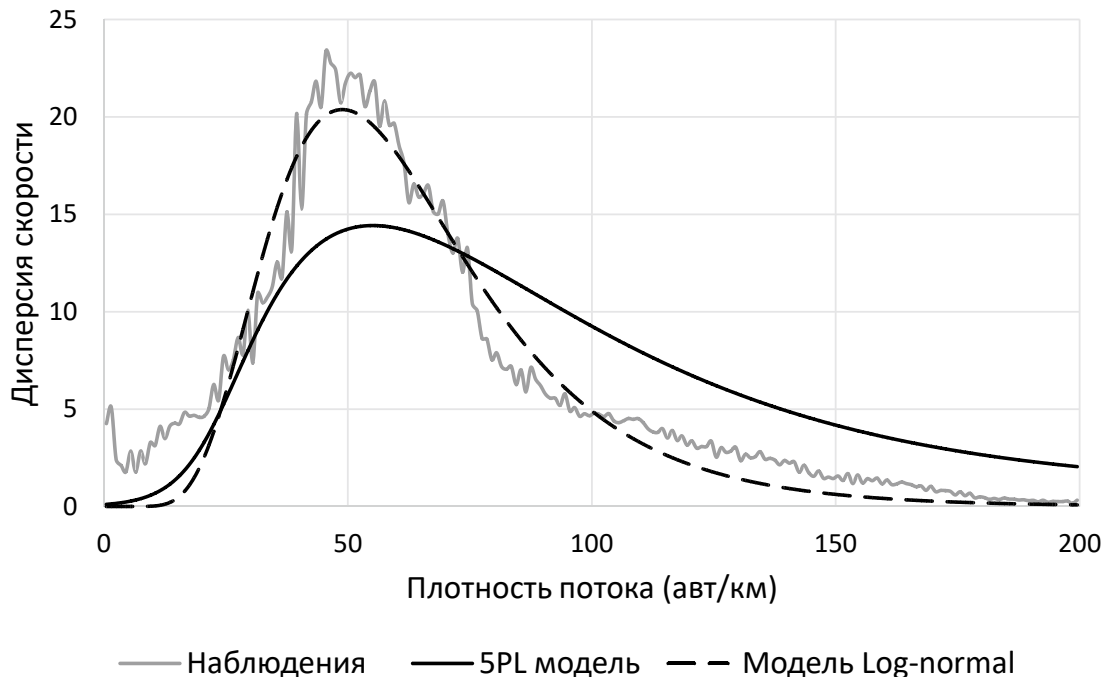


Рисунок 83 – Оценка дисперсии скорости движения

4.3.2.3 Гибридный алгоритм оценки скорости движения с учетом дисперсии

На практике для моделирования движения транспортных средств необходимо учитывать стохастические свойства транспортной сети. В частности, необходимо учитывать отклонения как в прогнозируемой загрузке сегментов дорожной сети, так и в оценке скорости движения, исходя из прогнозируемой плотности транспортного потока на сегменте.

Для оценки скорости движения предлагается учитывать следующие величины: t – момент времени, для которого требуется получить оценку скорости, t^{cur} – текущий момент времени, p_{ij}^{cur} – наблюдаемая плотность транспортного потока на сегменте (i, j) в текущий момент времени, V_{ij}^{cur} – наблюдаемая скорость прохождения сегмента (i, j) в текущий момент времени.

В работе предлагается следующее соотношение оценки скорости:

$$V_{ij}^{hyb} = \begin{cases} V_{ij}^{cur}, & |p_{ij}^{cur} - p_{ij}| < \Delta_p^{max} \wedge |\sigma(p_{ij}^{cur}) - \sigma(p_{ij})| < \Delta_\sigma^{max}; \\ V(p_{ij}), & \sigma(p_{ij}) < \Delta_\sigma^{stable}; \\ \left(1 - \frac{t - t^{cur}}{\Delta_t^{max}}\right) V^{cur} + \frac{t - t^{cur}}{\Delta_t^{max}} V(p_{ij}), & \text{иначе.} \end{cases} \quad (87)$$

где Δ_p^{max} , Δ_σ^{max} , Δ_σ^{stable} – пороговые параметры;

Δ_t^{max} – максимальный интервал времени, для которого оценивается скорость движения.

Согласно (87), для оценки скорости учитываются несколько состояний сети:

1) если оценки наблюдаемой и прогнозной плотностей транспортного потока и их дисперсии различаются на величины, меньшие пороговых значений Δ_p^{max} и Δ_σ^{max} , то текущая и прогнозная дорожная ситуации схожие, и в качестве оценки скорости используется наблюдаемая скорость V_{ij}^{cur} ;

2) если оценка дисперсии прогнозной плотности меньше порогового значения Δ_σ^{stable} , то дорожная ситуация в момент времени t является стабильной (свободное движение или движение в условиях дорожного затора), и в качестве оценки скорости используется прогнозная скорость $V(p_{ij})$;

3) иначе используется взвешенное среднее наблюдаемой и прогнозной оценок скорости.

В качестве прогнозной оценки скорости $V(p_{ij})$ используется 5PL модель (84), в качестве оценки дисперсии скорости $\sigma(p_{ij})$ используется логнормальное соотношение (86).

Предложенная оценка скорости движения используется в алгоритме резервирования маршрутов движения транспортных средств (Алгоритм 13).

4.3.3 Алгоритм резервирования маршрутов движения транспортных средств в гетерогенном транспортном потоке

Предложенные в предыдущих разделах 4.3.1 и 4.3.2 алгоритмы маршрутизации транспортных средств предполагают полное знание о маршрутах движения всех транспортных средств в сети. В данном разделе предполагается движение в гетерогенном потоке, включающем подключенные и управляемые водителями транспортные средства.

В данном разделе предложен подход, который комбинирует как метод прогнозирования гетерогенного транспортного потока, описанный в разделе 2.4, так и адаптивный метод управления сигналами светофорных объектов, описанный в разделе 3.3. Таким образом, разработанный алгоритм маршрутизации транспортных средств учитывает загрузку транспортной сети и адаптивное регулирование светофорных объектов для построения маршрутов движения транспортных средств.

Разработанный алгоритм построения маршрутов состоит из четырех периодически выполняемых этапов:

- сбор и предварительная обработка данных о состоянии потоков и движении транспортных средств;
- обнаружение загруженных участков дорожной сети;
- выбор транспортных средств для перестроения маршрутов исходя из текущей загрузки транспортной сети;
- построение нового маршрута движения для выбранных транспортных средств.

На первом этапе осуществляется сбор и предварительная обработка информации о текущем состоянии транспортной сети. Эта информация используется для оценки и прогнозирования параметров транспортного потока с использованием метода, описанного в разделе 2.4. Параметры транспортного потока в свою очередь используются как веса графа УДС для расчета кратчайшего пути.

На втором этапе выбираются сегменты транспортной сети с высоким уровнем загруженности. В предложенном алгоритме предлагается использовать следующую оценку уровня загрузки сегмента сети:

$$cong_i = 1 - \frac{V_i}{V_i^f},$$

где V_i – средняя скорость движения транспортных средств на сегменте i на предыдущем временном шаге;

V_i^f – средняя скорость движения свободного транспортного потока.

Сегмент транспортной сети помечается как перегруженный, если $cong_i > \delta$, где $\delta \in [0, 1]$ представляет собой пороговое значение.

Далее осуществляется выбор транспортных средств для перестроения маршрутов при наличии признаков заторов на определенном участке дороги исходя из текущей загрузки транспортной сети. В частности, для перестроения маршрутов выбираются транспортные средства, расположенные на участках въезда на рассматриваемый перегруженный сегмент сети. Параметр расстояния L указывает максимальное расстояние в сегментах от перегруженного сегмента, где может находиться самое дальнее транспортное средство, для которого будет выполняться перестроение маршрута.

Влияние перегруженного сегмента сети на время движения транспортного средства зависит от оставшегося расстояния до пункта назначения транспортного средства. Понятно, что заторы по-разному влияют на водителей, которые находятся близко к месту назначения, и на водителей, которые находятся далеко от места назначения. Рассматриваемый алгоритм использует оценку срочности для ранжирования транспортных средств, выбранных для изменения маршрута. Срочность определяется как разница между оставшимся временем в пути в текущей дорожной ситуации и оставшимся временем в пути при движении с максимально допустимой скоростью. Ранжирование транспортных средств, выбранных для перестроения маршрута, является важным этапом, поскольку алгоритм построения маршрута, используемый на завершающем этапе рассматриваемого алгоритма, учитывает информацию о маршруте движения всех транспортных средств, чтобы исключить возможность возникновения новых дорожных заторов. Таким образом, маршрут транспортных средств с более высокой срочностью перестраиваются в первую очередь.

Для учета времени задержки на регулируемых перекрестках, управляемых светофорными объектами с адаптивным регулированием, предлагается следующий алгоритм:

1) исходный граф дорожной сети преобразуется в граф, ребра которого соответствуют дорожным полосам, а вершины соответствуют соединениям между полосами. В этом случае весовые коэффициенты участков ребер исходного графа УДС переносятся на соответствующие полосы сегментов в новом графе. Если участок дороги состоит из нескольких полос, то каждой такой полосе присваивается весовой коэффициент

исходного участка дороги. Соединения между полосами на перекрестке (внутренние соединения) изначально имеют нулевой вес;

2) для каждого транспортного средства осуществляется выбор ближайших регулируемых перекрестков, для которых будут обновляться весовые коэффициенты. Весовые коэффициенты устанавливаются только на ограниченном количестве близлежащих перекрестков в связи с потерей актуальности информации о состоянии светофоров с течением времени. Выбираются регулируемые перекрестки, удаленные от рассматриваемого транспортного средства на время движения, не превышающее период времени перестроения маршрута $\tau_{reroute}$.

В рамках предложенного алгоритма предполагается, что управление сигналами светофоров выполняется с использованием метода, описанного в разделе 3.3. Пусть период перестроения маршрутов $\tau_{reroute}$ равен минимальному интервалу переключения фаз светофора τ_{min} ; τ_{yellow} – желтый сигнал светофора; p_{in}^{out} – текущий активный сигнал светофора для соединения между входящей полосой in и исходящей полосой out , \tilde{p}_{in}^{out} – следующий сигнал светофор для соединения между входящей полосой in и исходящей полосой out . Тогда вес внутреннего соединения регулируемого перекрестка w_{in}^{out} рассчитывается следующим образом:

$$w_{in}^{out} = \begin{cases} 0, & p_{in}^{out} = \text{green} \wedge \tilde{p}_{in}^{out} = \text{green}, \\ \tau_{min}, & p_{in}^{out} = \text{green} \wedge \tilde{p}_{in}^{out} = \text{red}, \\ \tau_{min} + \tau_{yellow}, & p_{in}^{out} = \text{red} \wedge \tilde{p}_{in}^{out} = \text{red}, \\ \tau_{yellow}, & p_{in}^{out} = \text{red} \wedge \tilde{p}_{in}^{out} = \text{green}. \end{cases}$$

Для построения маршрута транспортного средства в [368] авторы представили модифицированную версию известного алгоритма поиска A^* , названную A^* с отталкиванием (AR^*). Этот алгоритм использует информацию о текущих маршрутах транспортных средств в транспортной сети при расчете кратчайшего пути, чтобы предотвратить образование новых дорожных заторов. Для этого в алгоритме используется т.н. «взвешенный след» сегмента транспортной сети, который в исходном алгоритме определяется следующим образом:

$$f_i = n_i \frac{len_{avg} V_{f_{avg}}}{len_i V_{f_i}},$$

где n_i – это общее количество транспортных средств, маршруты которых включают полосу i ;

len_{avg} – средняя длина полосы в транспортной сети;

len_i – длина полосы i ;

$V_{f_{avg}}$ – средняя скорость свободного потока в транспортной сети;

V_{f_i} – скорость свободного потока дорожной полосы i .

Таким образом, чем выше значение f_i , тем менее предпочтителен рассматриваемый сегмент транспортной сети. При этом следует отметить, что чем дальше расположен сегмент сети от начального положения транспортного средства, тем меньше он должен оказывать влияние на процесс перестроения маршрута, поскольку изменение маршрута может повторяться по мере движения транспортного средства к месту назначения. Поэтому в предложенном алгоритме предлагается использовать следующее определение коэффициента n_i :

$$n_i = \sum_{c \in C} \sum_{j=0}^{len_{r_c}-1} \alpha^j I(r_{c_j} = i),$$

где C – это множество подключенных транспортных средств в транспортной сети;

len_{r_c} – длина (количество сегментов) маршрута r_c транспортного средства $c \in C$;

r_{c_j} – j -й сегмент маршрута r_c ;

α – коэффициент воздействия.

Эвристическая функция оценки стоимости маршрута $F(x)$ алгоритма AR* была модифицирована по сравнению с исходным алгоритмом A*. Оценка отталкивания $R(x)$ вершины x графа G определяется как сумма взвешенных следов от начальной вершины до вершины x . Таким образом, функция стоимости маршрута $F(x)$ определяется следующим образом:

$$F(x) = (1 - \beta) \times (T(x) + H(x)) + \beta \times R(x),$$

где $T(x)$ – время движения от текущей вершины до вершины x ;

$H(x)$ – эвристическая оценка времени движения, которая рассчитывается как евклидово расстояние между текущей вершиной и вершиной x , деленное на максимально допустимую скорость в транспортной сети;

β – весовой коэффициент.

На основании вышеизложенного одна итерация предлагаемого алгоритма перестроения маршрута в виде псевдокода может быть представлена следующим образом (Алгоритм 14):

Алгоритм 14: Метод перестроения маршрутов

```

1: Входные данные:  $G, \delta, L, \tau_{reroute}$ 
2:  $G = \text{updateEdgeWeights}(G)$  // Обновить веса графа
3: congestedSegments = getCongestedSegments( $G, \delta$ )
4: if len(congestedSegments) = 0 then
5:   return;
6: end if
7: vehsToReroute = [];
8: for segment in congestedSegments do
9:   vehsToReroute.append(getVehiclesToReroute(segment,  $L$ ))
10: end for
11: vehsToReroute = sortByUrgency(vehsToReroute) // Ранжирование по срочности
11:  $G_{lane} = \text{getLaneBasedGraph}(G)$ 
12: for veh in vehsToReroute do
13:   fp = getEdgesFootprint(); // Получение «взвешенного следа»
14:   current, dest = getVehicleRoute(veh);
15:    $G_{lane} = \text{updateVehWeights}G_{lane}(veh, \tau_{reroute})$ 
16:   newRoute = AstarRepulsion( $G_{lane}$ , current, dest, fp) // Перестроение маршрута
17:   setRoute(veh, newRoute) // Обновление маршрута ТС
18: end for

```

4.3.4 Экспериментальные исследования алгоритмов маршрутизации подключенных транспортных средств в сети

В данном разделе представлена постановка и результаты экспериментальных исследований алгоритма маршрутизации подключенных транспортных средств на основе подхода к резервированию маршрутов (раздел 4.3.4.1), а также его модификации для учета стохастических свойств транспортной сети (раздел 4.3.4.2). В разделе 4.3.4.3 представлены экспериментальные исследования алгоритма резервирования маршрутов для гетерогенного транспортного потока.

4.3.4.1 Экспериментальные исследования алгоритма резервирования маршрутов движения транспортных средств

Для проведения экспериментальных исследований алгоритма резервирования маршрутов движения транспортных средств был выбран тестовый сценарий в нерегулируемой дорожной сети малого размера [328].

Для моделирования движения транспортных средств используется пакет микроскопического моделирования с открытым исходным кодом SUMO [367], предназначенный для моделирования интермодальных сценариев движения в крупномасштабных транспортных сетях. Для моделирования движения транспортных средств использовалась модель следования за лидером Краусса [369]. В работе использовались стандартные параметры модели: длина транспортного средства (ТС) 5 метров, максимальная скорость 15 м/с, ускорение 2.5 м/с², торможение 4.5 м/с², минимальное расстояние между ТС 2.5 м.

Сравнение предлагаемого алгоритма с разными соотношениями оценки скорости прохождения дорожных сегментов (80) – (82) проводилось по критериям среднего времени движения и среднего времени задержки отправления. Задержка возникает, если при моделировании в указанное время отправления на сегменте недостаточно места для движения нового ТС, тогда это ТС добавляется в очередь, и попытка отправления будет выполнена на следующих итерациях моделирования.

На первом этапе экспериментов фиксировался интервал времени перестроения маршрутов $T_{reroute} = 15$ секунд, и исследовалась зависимость от интервала дискретизации для разных соотношений скорости и плотности потока. Моделирование проводилось в режиме плотного движения с заторами для 84110 транспортных средств за один день. Результаты экспериментов показаны на рисунках 84 и 85.

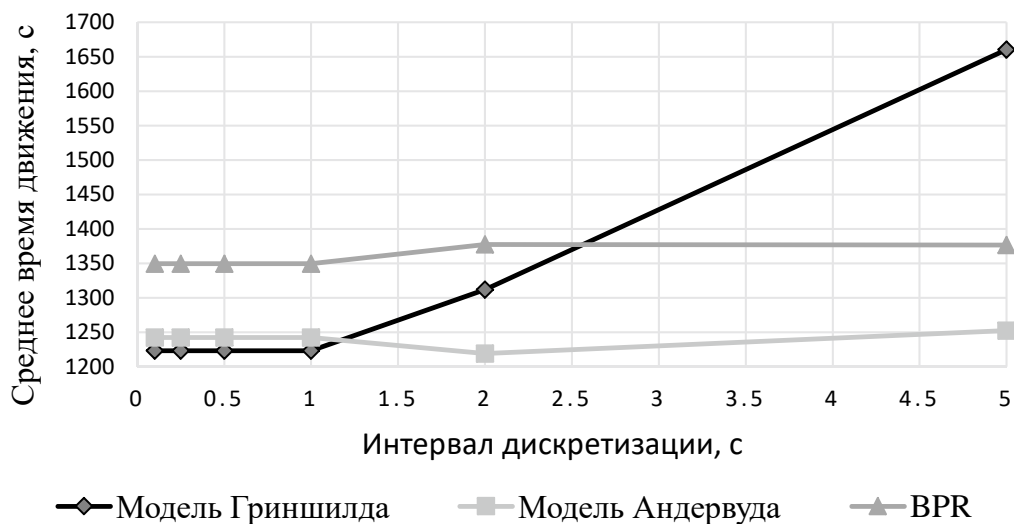


Рисунок 84 – Среднее время движения для различных соотношений скорости-плотности потока

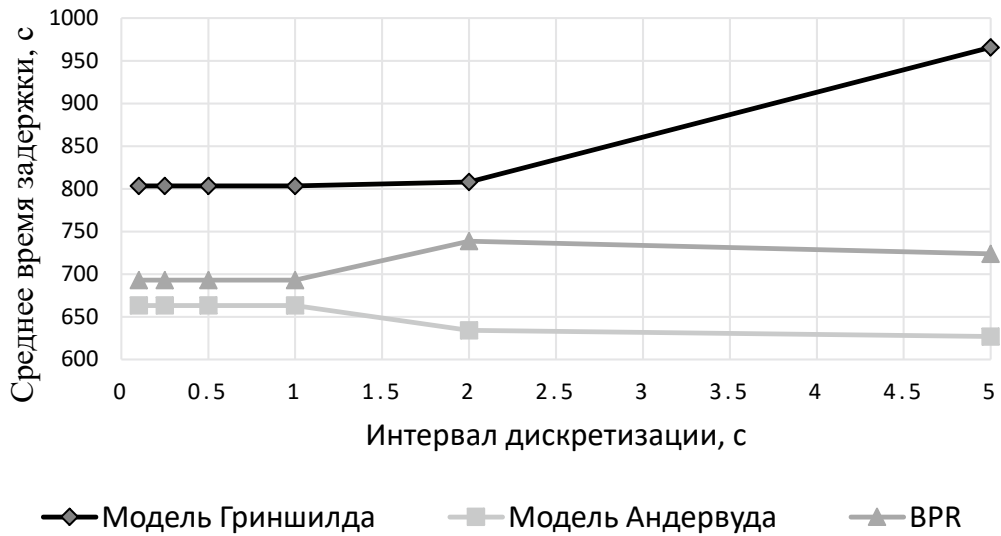


Рисунок 85 – Среднее время задержки отправления для различных соотношений скорости-плотности потока

По критерию среднего времени движения модели Андервуда и Гриншилда показали схожие результаты для интервала дискретизации меньше одной секунды, однако по критерию средней задержки времени отправления модель Гриншилда показала худший результат. Учитывая оба критерия в совокупности, результаты, полученные с использованием соотношения Андервуда скорости-плотности потока, превосходят результаты, полученные с использованием BPR-соотношения и модели Гриншилда.

Следующим этапом экспериментального анализа было исследование зависимости среднего времени поездки и задержки отправления для модели Андервуда в зависимости от интервала дискретизации для разных значений интервала времени перестроения маршрутов $T_{reroute}$ (в секундах). Среднее время поездки и средняя задержка отправления показаны на рисунках 86 и 87 соответственно.

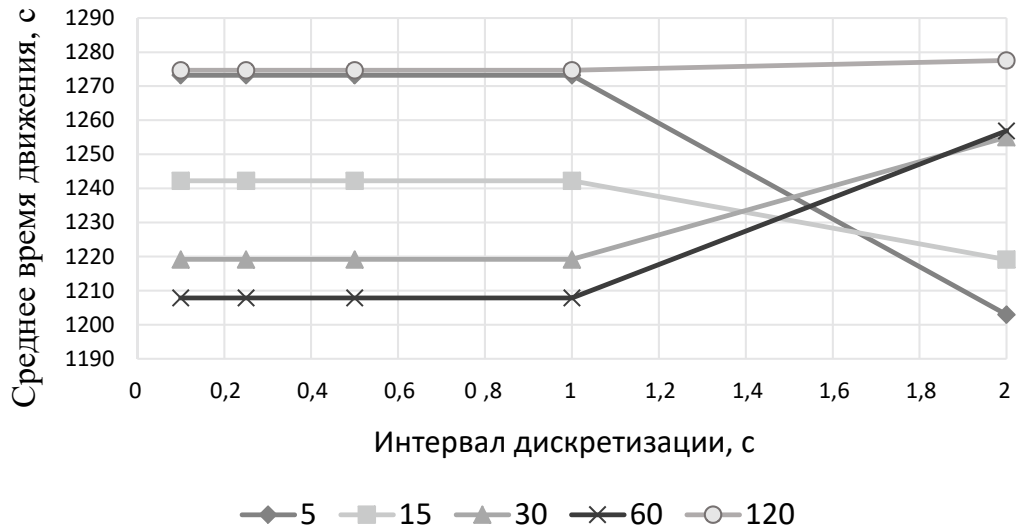


Рисунок 86 – Среднее время поездки для различных интервалов перестроения маршрута

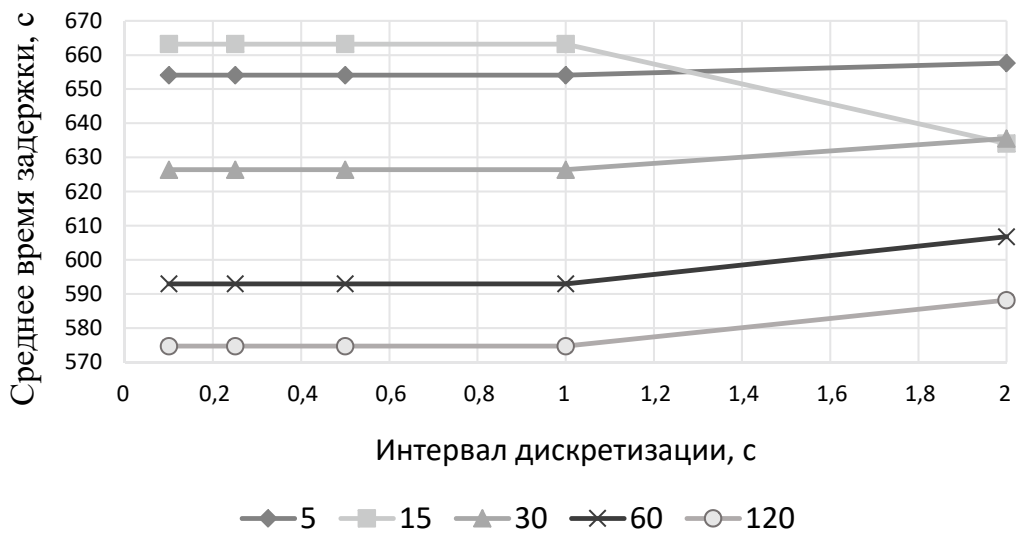


Рисунок 87 – Среднее время задержки отправления для различных интервалов перестроения маршрута

Учитывая оба критерия в совокупности, лучшие результаты были показаны моделью с интервалом перестроения маршрута $T_{reroute} = 60$ секунд.

Далее, было проведено сравнение результатов, полученных с помощью алгоритма резервирования маршрутов (RRA) с неконтролируемой динамической маршрутизацией, основанной на использовании текущей информации о состоянии транспортных потоков в сети. Таблица 31 показывает среднее время движения (в секундах) для различных интервалов перестроения маршрутов с фиксированным интервалом дискретизации $T_{discr} = 1$ секунда.

Таблица 31 – Среднее время поездки в тестовой сети

$T_{reroute}$	5	15	30	60	120
RRA	1273,2	1242,2	1219,2	1207,9	1274,7
Динамическая маршрутизация	1388,3	1374,7	1416,0	1439,8	1410,4

Следующим этапом проведения экспериментов было сравнение быстродействия алгоритмов. Рисунок 88 показывает среднее время моделирования сценария. Можно отметить, что время моделирования уменьшается при увеличении интервала дискретизации T_{discr} и увеличении интервала перестроения маршрута $T_{reroute}$.

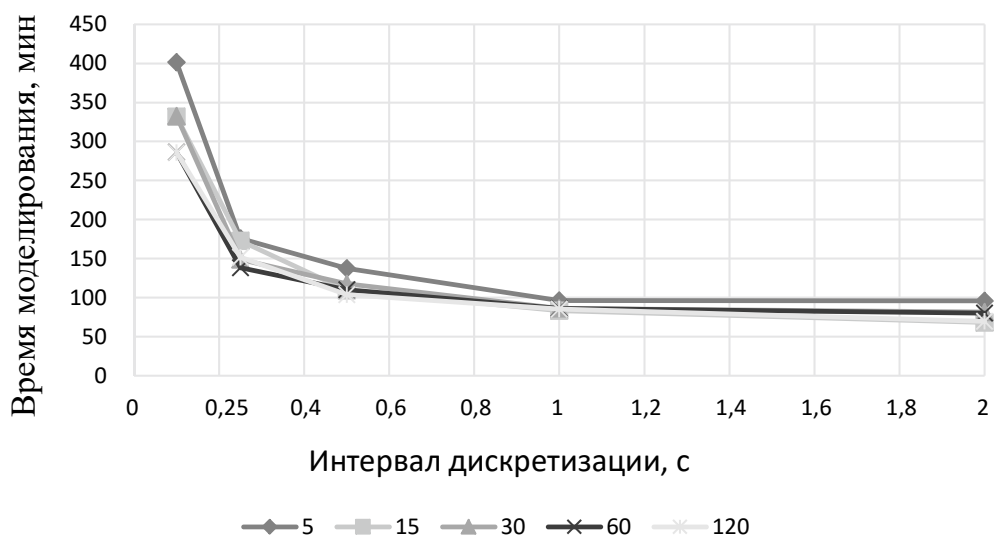


Рисунок 88 – Среднее время моделирования для различных интервалов перестроения маршрута

На заключительном этапе экспериментального анализа было проведено сравнение алгоритма резервирования маршрутов ($T_{reroute} = 60$, $T_{discr} = 1$) с динамической маршрутизацией с использованием текущей информации в крупномасштабной транспортной сети. Для тестирования был выбран общедоступный сценарий «TAPAS Cologne» [340]. Для моделирования использовалась транспортная сеть г. Кёльн, содержащая 71368 дорожных сегментов. Рассматривался период моделирования в утренний час пик, с 6 до 9 утра. Среднее время поездки (в секундах) в зависимости от количества моделируемых транспортных средств (загрузки сети) показано в таблице 32.

Среднее время задержки отправления (в секундах) в зависимости от количества моделируемых транспортных средств показано в таблице 33.

Таблица 32 – Среднее время поездки в крупномасштабной сети

–	50547	75820	101093
RRA	589,2	626,3	677,1
Динамическая маршрутизация	618,1	641,5	678,1

Таблица 33 – Среднее время задержки отправления в крупномасштабной сети

–	50547	75820	101093
RRA	1,1	6,6	13,1
Динамическая маршрутизация	1,7	8,8	23,9

Предложенный алгоритм резервирования маршрутов показал лучшие результаты по критериям среднего времени поездки и средней задержки отправления по сравнению с динамической маршрутизацией транспортных средств.

4.3.4.2 Экспериментальные исследования алгоритма резервирования маршрутов движения транспортных средств в стохастической сети

Для проведения экспериментальных исследований алгоритма резервирования маршрутов движения транспортных средств с учетом стохастических свойств транспортной сети использовались модели следования за лидером Краусса и IDM [369].

Проводилось сравнение алгоритма резервирования маршрутов с использованием оценки скорости соотношением Андервуда (83), оценки скорости гибридным алгоритмом (87), а также алгоритма неконтролируемой динамической маршрутизацией, основанной на использовании текущей информации о скорости движения в сети (SUMO). Сравнение проводилось по критериям среднего времени движения, среднего времени задержки отправления и средней потери времени из-за движения ниже идеальной скорости. Интервал времени перестроения маршрутов фиксировался и составлял $T_{reroute} = 60$ секунд.

На первом этапе проведения экспериментальных исследований был выбран тестовый сценарий в нерегулируемой дорожной сети малого размера [328]. Результаты сравнения алгоритмов для модели следования за лидером Краусса показаны в таблице 34, для модели IDM – в таблице 35.

Таблица 34 – Результаты маршрутизации для модели следования за лидером Краусса

–	SUMO	Соотношение Андервуда	Гибридный алгоритм
Среднее время движения, с	1439,7	1254,5	1170,4
Среднее время задержки отправления, с	1249,5	520,2	836,0
Средняя потеря времени, с	895,0	813,3	704,7

Таблица 35 – Результаты маршрутизации для модели следования за лидером IDM

–	SUMO	Соотношение Андервуда	Гибридный алгоритм
Среднее время движения, с	1734,0	2725,0	1706,8
Среднее время задержки отправления, с	1701,5	1490,1	1257,1
Средняя потеря времени, с	1208,5	2248,1	1229,3

По критерию среднего времени движения предложенная модель показала лучшие результаты в обоих рассмотренных сценариях движения. Модель с использованием соотношения Андервуда показала лучший результат по критерию среднего времени задержки отправления при использовании модели Краусса, однако значительно уступила при использовании модели IDM.

Основной интерес представляет выполнение сценария маршрутизации в крупномасштабной транспортной сети. Для тестирования был выбран общедоступный сценарий «TAPAS Cologne» [340]. Для моделирования использовалась транспортная сеть г. Кёльна, содержащая 71368 дорожных сегментов. Рассматривался период моделирования в утренний час пик, с 6 до 9 утра. Среднее время поездки (в секундах) в зависимости от масштаба спроса (относительного количества моделируемых транспортных средств) показано на рисунке 89. Среднее время задержки отправления (в секундах) в зависимости от масштаба спроса показано на рисунке 90.

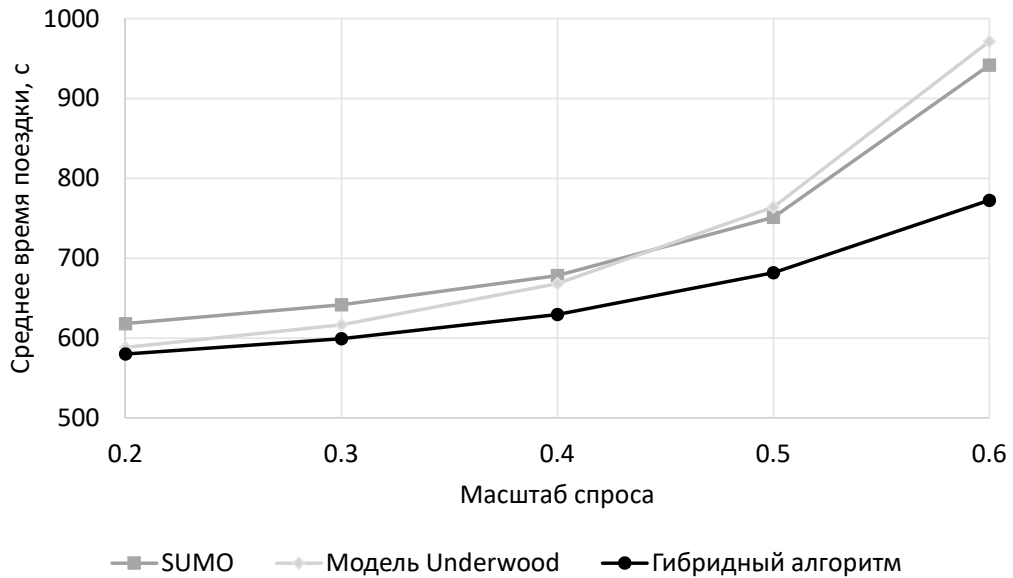


Рисунок 89 – Среднее время движения

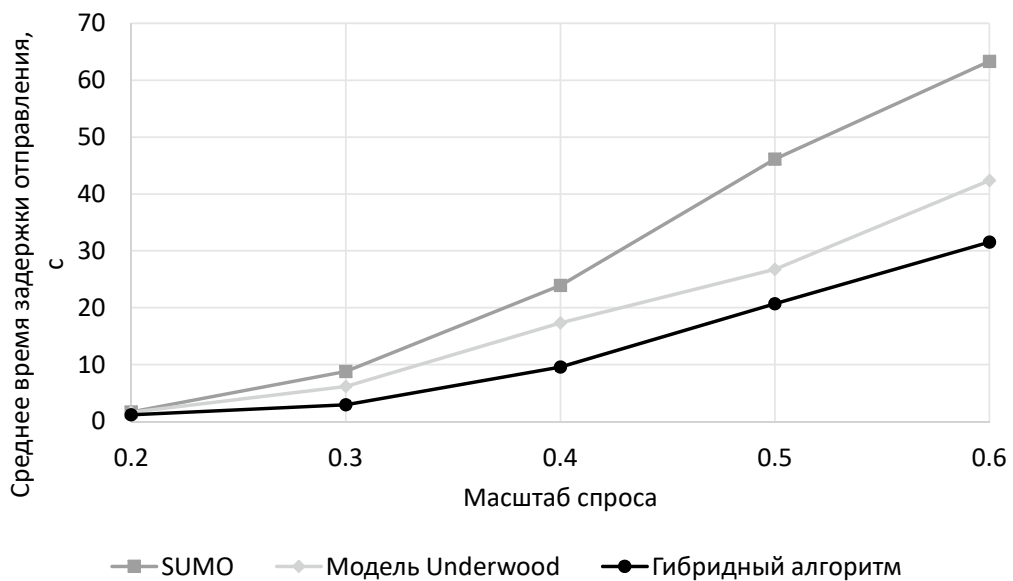


Рисунок 90 – Среднее время задержки отправления

Предложенный алгоритм оценки скорости показал лучший результат по всем рассматриваемым критериям.

4.3.4.3 Экспериментальные исследования алгоритма резервирования маршрутов движения транспортных средств в гетерогенном транспортном потоке

Экспериментальные исследования алгоритма построения маршрутов движения подключенных транспортных средств в гетерогенном транспортном потоке проводились для двух сценариев моделирования в системе SUMO:

- сценарий «grid4x4» – синтетическая сеть, состоящая из 16 перекрестков одинаковой структуры. Общее количество транспортных средств в сценарии – 2270.

- сценарий «Cologne8» – транспортная сеть малого размера, содержащая 8 регулируемых перекрестков различной структуры. Общее количество транспортных средств в сценарии – 1740.

В экспериментах проводилось сравнение предложенного алгоритма построения маршрутов движения с алгоритмом, представленным в [368], и неуправляемой маршрутизацией ТС. Сравнение алгоритмов перестроения маршрутов движения проводилось по двум критериям:

- среднее время ожидания транспортных средств;
- среднее время движения транспортных средств.

Результаты сравнения алгоритмов по указанным критериям представлены в таблицах 36 и 37 соответственно.

Таблица 36 – Сравнение алгоритмов по критерию среднего времени ожидания, с

Сценарий	Предложенный алгоритм	Базовый алгоритм	Неуправляемая маршрутизация
«grid4x4»	38,59	42,97	192,40
«Cologne8»	3,49	3,89	3,92

Таблица 37 – Сравнение алгоритмов по критерию среднего времени движения, с

Сценарий	Предложенный алгоритм	Базовый алгоритм	Неуправляемая маршрутизация
«grid4x4»	188,73	202,07	397,90
«Cologne8»	84,31	89,21	89,28

Проведенные эксперименты подтверждают эффективность предложенного алгоритма построения маршрутов движения подключенных транспортных средств в гетерогенном транспортном потоке во взаимодействии с адаптивным алгоритмом управления сигналами светофоров по всем рассмотренным критериям.

4.4 Обеспечение информационной безопасности в контексте ИТС

При рассмотрении кооперативной ИТС с возможностью обмена данными между подключенными транспортными средствами и/или объектами инфраструктуры посредством автомобильных самоорганизующихся сетей (VANET) отдельное внимание заслуживают вопросы, связанные с обеспечением информационной безопасности

коммуникаций, обеспечению конфиденциальности, целостности и доступности передаваемых данных.

Для обеспечения информационной безопасности необходимо обеспечивать защиту как от внешних угроз, путем аутентификации участников коммуникаций, так и от внутренних угроз. При защите от внутренних угроз предполагается, что злоумышленником является аутентифицированное и авторизованное транспортное средство, поэтому одним из основных подходов в этом случае является обнаружение аномального поведения участников движения.

В разделе 4.4.1 рассматривается задача аутентификации транспортных средств. Обнаружение аномального поведения транспортных средств в транспортной сети рассматривается в разделе 4.4.2. Частный случай движения транспортных средств в колонне рассматривается в разделе 4.4.3.

4.4.1 Аутентификация транспортных средств

4.4.1.1 Схема аутентификации

Для описания разработанной схемы аутентификации, введем следующие обозначения.

Пусть $\{V_i\}_{i=1, \overline{N}}$ – множество транспортных средств, N – общее количество транспортных средств, $\{R_j\}_{j=1, \overline{M}}$ – множество придорожных блоков (англ.: Road-Side Unit, RSU), M – общее количество RSU.

Обозначим публичный и приватный ключи ТС V_i как PuK_i^V и $Pr K_i^V$, публичный и приватный ключи RSU R_j как PuK_j^R и $Pr K_j^R$. Оператор конкатенации обозначим как $||$.

В данном разделе при выполнении процедуры аутентификации предполагается, что каждое ТС хранит список публичных ключей и идентификаторов всех RSU, полученный из центра сертификации. Рассматриваемая схема аутентификации на основе инфраструктуры придорожных блоков состоит из трех этапов, представленных на рисунке 91 [262].



Рисунок 91 – Этапы алгоритма аутентификации

Далее подробнее рассмотрен каждый из этапов, а также граничный случай аутентификации сообщений между транспортными средствами, находящимися в радиусах действия разных RSU, выполнение которого полностью покрывает процесс аутентификации с помощью представленной схемы.

4.4.1.2 Этап рукопожатия

Когда ТС V_i обнаруживает в своем радиусе связи блок R_j , оно инициирует процедуру обмена общего секретного ключа между ТС и RSU. Для выполнения этого шага используется протокол Диффи-Хеллмана на эллиптических кривых. Обозначим публичные значения ТС и RSU, используемые для генерации общего секретного ключа $P^{c_{ij}}$ на этапе рукопожатия как $PuVal_i^V$ и $PuVal_j^R$, сертификат транспортного средства как $Cer(V_i)$, код аутентификации сообщения, вычисленный с помощью общего секретного ключа $P^{c_{ij}}$, как $HMAC(\cdot)_{P^{c_{ij}}}$ и идентификатор для этапа рукопожатия как $hsId$. Алгоритм состоит из следующих шагов:

- 1) ТС V_i отправляет блоку R_j сообщение вида:

$$hsId \parallel \left(SIG_{PrK_i^V} [PuVal_i^V] \parallel (PuVal_i^V) \right)_{PuK_j^R} \parallel Cer(V_i);$$

- 2) после получения сообщения блок R_j дешифрует его зашифрованную часть, используя свой приватный ключ PrK_j^R и, верифицируя цифровую подпись $SIG_{PrK_i^V} [PuVal_i^V]$ с помощью сертификата узла, получает $PuVal_i^V$. Далее, используя

$PuVal_i^V$, R_j вычисляет значение общего с V_i секретного ключа P^c_{ij} . На данном этапе только R_j вычислил общий секретный ключ, поэтому он отправляет V_i следующее сообщение:

$$hsId \parallel SID_i \parallel RID_j \parallel PuVal_j^R \parallel HMAC(SID_i \parallel RID_j \parallel PuVal_j^R)_{P^c_{ij}},$$

где SID_i – сгенерированный R_j псевдо-идентификатор для V_i ;

RID_j – идентификатор R_j ;

3) после получения сообщения V_i с помощью значения $PuVal_j^R$ вычисляет общий секретный ключ P^c_{ij} . Далее V_i находит $HMAC(SID_i \parallel RID_j \parallel PuVal_j^R)_{P^c_{ij}}$ и сравнивает его с полученным значением. Если они совпадают, то общий секретный ключ P^c_{ij} распределен и этап рукопожатия завершен. Более того, сессионный идентификатор SID_i присваивается ТС V_i для сохранения конфиденциальности.

Блок RSU хранит таблицу (таблица 38), которая пополняется каждый раз при успешной процедуре рукопожатия с новым автомобилем. Таблица содержит четыре столбца:

- 1) псевдо-идентификатор – сгенерированный блоком RSU псевдо-идентификатор для V_i , который используется для сохранения анонимности реального идентификатора автомобиля;
- 2) идентификатор – реальный идентификатор автомобиля V_i ;
- 3) сертификат – сертификат автомобиля V_i ;
- 4) время жизни ключа – время жизни общего секретного ключа P^c_{ij} .

Таблица 38 – Таблица сессионных идентификаторов

Псевдо-идентификатор	Идентификатор	Сертификат	Время жизни ключа
SID_1	ID_1	$Cert(V_1)$	T_1
SID_2	ID_1	$Cert(V_2)$	T_2
...

Предполагается, что в регионе, в котором на данный момент передвигается автомобиль, находится M других блоков дорожной инфраструктуры. Поэтому после генерации общего секретного ключа P^c_{ij} с первым попавшим в радиус взаимодействия R_j должно быть сгенерировано еще M сессионных ключей: $K_{i,1}, K_{i,1}, \dots, K_{i,m}$ для каждого из M RSU в регионе соответственно. Генерация ключей происходит на основе ключа P^c_{ij} с

помощью односторонней хеш-функции F , для того, чтобы другие RSU не смогли получить значение ключа P^c_{ij} . Так образуется цепочка из M хеш-ключей:

$$K_{i,M} = F(ID_i \parallel P^c_{ij}), K_{i,M-1} = F(ID_i \parallel K_{i,M}), \dots, K_{i,1} = F(ID_i \parallel K_{i,2}),$$

где ID_i – идентификатор узла V_i .

Далее блок R_j в зашифрованном виде пересылает каждому из M RSU в регионе их сессионные ключи $K_{i,m}$. На основе этих полученных данных у каждого из M RSU пополняется таблица сессионных ключей (таблица 39). В ней содержатся следующие столбцы:

- 1) псевдо-идентификатор - псевдо-идентификатор автомобиля V_i ;
- 2) блок-информатор – идентификатор RSU, который передал данному блоку RSU сессионный ключ $K_{i,m}$;
- 3) сессионный ключ – сессионный ключ $K_{i,m}$ с автомобилем V_i ;
- 4) время жизни ключа – время жизни сессионного ключа $K_{i,m}$.

Таблица 39 – Таблица сессионных ключей

Псевдо-идентификатор	Блок-информатор	Сессионный ключ	Время жизни ключа
SID_5	RID_2	$K_{5,1}$	$T_{5,1}$
SID_i	RID_2	$K_{i,1}$	$T_{i,1}$
SID_1	RID_m	$K_{1,1}$	$T_{1,1}$
...

Помимо этого, каждое ТС ведет таблицу идентификаторов придорожных блоков (таблица 40), содержащую столбцы:

- 1) RID – идентификатор R_j ;
- 2) сессионный ключ – общий с блоком R_j сессионный ключ $K_{i,j}$;
- 3) время жизни ключа – время жизни сессионного ключа $K_{i,j}$.

Таблица 40 – Таблица идентификаторов блоков

RID	Сессионный ключ	Время жизни ключа
RID_1	$K_{i,1}$	$T_{i,1}$
RID_2	$K_{i,2}$	$T_{i,2}$
...
RID_m	$K_{i,m}$	$T_{i,m}$

Данные таблицы используются для аутентификации сообщений ТС, находящихся в областях действия разных RSU (подробнее описано в разделе 4.4.1.5).

4.4.1.3 Этап подписи сообщения

Этап подписи сообщения может начаться только после успешного этапа рукопожатия. Когда ТС V_i отправляет информационное сообщение M_i , оно использует общий симметричный ключ P_{ij}^c для вычисления кода аутентификации сообщения $HMAC_{P_{ij}}(ID_i||M_i)$ и рассылает широковещательное сообщение следующего вида $ID_i||M_i||HMAC_{P_{ij}}(ID_i||M_i)$.

$$V_i \rightarrow *: SID_i||M_i||HMAC_{P_{ij}}(SID_i||M_i)$$

Верификация сообщения выполняется RSU R_j . Когда RSU R_j получает сообщение, оно верифицирует его с использованием общего ключа P_{ij}^c и вычисляет результат аутентификации сообщения $H(SID_i||M_i)$. RSU агрегирует сообщения от транспортных средств в течение определенного интервала времени, объединяет результаты аутентификации сообщений, подписывает их с использованием приватного ключа RSU PrK_j^R и широковещательно рассылает результирующее сообщение. Обозначим $H(ID_1 || M_2) || H(ID_1 || M_2) || \dots || H(ID_n || M_n)$ как Ha_g . Далее RSU_j подписывает Ha_g с помощью своего закрытого ключа PrK_j^R и производит широковещательную рассылку следующего сообщения:

$$R_j \rightarrow *: Ha_g || H(Ha_g)_{PrK_j^R}$$

4.4.1.4 Этап верификации сообщения

Когда ТС V_k получает информационное сообщение от ТС V_i , оно сохраняет его в буфер. Далее, когда ТС получает сообщение с результатом аутентификации сообщения $H(ID_i||M_i)$ от RSU R_j , оно проверяет его цифровую подпись с использованием публичного ключа RSU PuK_j^R и получает результат аутентификации. Если проверка аутентификации сообщения была успешно выполнена, сообщение от ТС V_i принимается; иначе оно отклоняется.

4.4.1.5 Аутентификация сообщений между ТС в радиусах действия разных блоков

В ходе работы схемы может возникнуть ситуация, изображенная на рисунке 92.

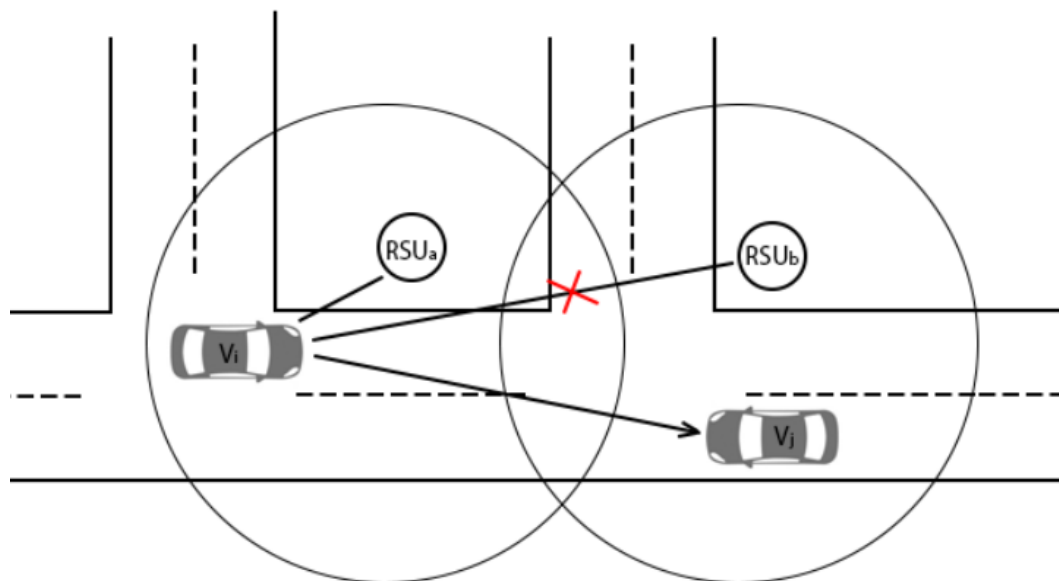


Рисунок 92 – Расположение ТС в радиусах действия разных RSU

Когда автомобиль V_i посылает сообщение автомобилю V_j , R_b не может аутентифицировать это сообщение, так как не имеет общего секретного ключа между V_i и R_a . Для решение этой проблемы используется раннее описанная таблица 40.

После получения сообщение от V_i , V_j , который находится в области действия R_b проверяет, находится ли отправитель в области действия того же R_b . В случае успеха, оба автомобиля находится в области R_b , который и будет аутентифицировать сообщение. В противном случае V_j посылает ответный запрос к V_i . Запрос представляет собой следующее сообщение:

$$RID_b \parallel SID_i,$$

где RID_b – номер RSU, через который следует аутентифицировать сообщение;

SID_i – псевдо-идентификатор ТС, которому направлен запрос.

Далее V_i , используя таблицу 40, находит необходимый сессионный ключ для работы с RID_b – $K_{i,b}$ и снова отправляет сообщение, подтвержденное уже этим ключом. После этого RID_b может провести аутентификацию сообщения и сообщить о результате V_j .

В следующем разделе описано экспериментальное исследование описанного алгоритма.

4.4.1.6 Экспериментальные исследования схемы аутентификации

Для проведения экспериментальных исследований описанная схема аутентификации была реализована в Veins, среде с открытым исходным кодом для моделирования автомобильных сетей [370]. Veins основан на использовании двух систем моделирования: OMNeT ++, симуляторе сети, основанном на событиях, и SUMO, симуляторе дорожного движения. Для реализации связи V2V и V2I Veins полагается на полностью детализированные модели сетевых уровней IEEE 802.11p и IEEE 1609.4 DSRC / WAVE.

В экспериментальных исследованиях проводилось сравнение традиционной схемы аутентификации на основе PKI, реализованной с помощью подписи на основе эллиптических кривых, и описанной схемы на основе RSU. Согласно сценарию, N ТС выдвигаются из точки отправления и двигаются вместе по указанному маршруту. Через 50 секунд после начала движения, ведущее ТС останавливается, тем самым моделируя дорожный инцидент. После того, как остановка продолжается более 10 секунд, данное ТС начинает широковещательную рассылку, в которой содержится идентификатор дороги, на которой произошел инцидент.

При получении такого сообщения другие ТС принимают решение о смене маршрута, а также делают три широковещательные рассылки полученной об инциденте информации.

Основой реализованной схемы PKI является цифровая RSA-подпись. Характеристики используемых криптографических примитивов приведены в таблице 41.

Таблица 41 – Количественные характеристики криптографических примитивов

Размер приватного RSA ключа для V_i	256 байт
Размер публичного RSA ключа для V_i	256 байт
Размер сертификата для V_i	1281 байт
Размер цифровой подписи	256 байт

Экспериментальные исследования в моделируемой сети для схемы PKI показали результаты, приведенные в таблице 42. Общие накладные расходы превышают сумму размеров компонентов информационного сообщения, так как результирующее сообщение кодируется стандартом Base64 из-за особенностей реализации схемы на языке C++. Под накладными расходами подразумевается объем трафика необходимы при передаче сообщения. Исследования проводились на рабочей станции с процессором Intel Core i5-2,3 ГГц с 2048 Мб RAM.

Таблица 42 – Время выполнения этапов PKI-схемы

Время необходимое для создания RSA-подписи информационного сообщения с ключом размером 256 байт	1,517 мс
Время необходимое для верификации RSA-подписи с ключом размером 256 байт	0,561 мс
Время необходимое на передачу и прием с последующей обработкой одного сообщения	3,79 мс
Общие накладные расходы на одно сообщение	1825 байт

При реализации схемы с использованием инфраструктуры придорожных блоков используются сертификаты, цифровая RSA-подпись, RSA-шифрование, криптография на основе эллиптических кривых, а также хеш-функция с ключом (HMAC). Основные характеристики используемых криптографических примитивов приведены в таблице 43.

Таблица 43 – Криптографические характеристики RSU-схемы

Размер приватного RSA ключа для V_i	256 байт
Размер публичного RSA ключа для V_i	256 байт
Размер приватного RSA ключа для RSU_j	512 байт
Размер публичного RSA ключа для RSU_j	512 байт
Размер сертификата для V_i	1281 байт
Размер сертификата для RSU_j	1281 байт
Размер цифровой подписи для V_i	256 байт
Размер цифровой подписи для RSU_j	512 байт
Размер $PuVal_i^V$	178 байт
Размер $PuVal_j^R$	178 байт
Размер кода аутентификации HMAC	32 байта
Идентификатор эллиптической кривой для протокола EDCH	prime256v1

Далее было подсчитано время вычислений и накладные расходы на передачу, необходимые для завершения процесса аутентификации в предлагаемой схеме. Экспериментальные исследования в моделируемой сети для схемы на основе инфраструктуры придорожных блоков показали следующие результаты (таблица 44):

Таблица 44 – Время выполнения этапов RSU-схемы

Время, необходимое для создания RSA-подписи информационного сообщения с ключом 256 байт	1,517 мс
Время, необходимое для верификации RSA-подписи с ключом 256 байт	0,561 мс
Время, необходимое для RSA-шифрования с ключом 512 байт	0,146 мс
Время, необходимое для RSA-дешифрования с ключом 512 байт	7,736 мс
Время, необходимое для нахождения кода аутентификации (HMAC)	0,0072 мс
Время, необходимое на совершение этапа рукопожатия	12,08 мс
Время, необходимое на совершение этапа подписи сообщения	2,134 мс
Время, необходимое на совершение этапа верификации сообщения	0,225 мс
Общие накладные расходы на этап рукопожатия	2461 байт
Общие накладные расходы на передачу одного информационного сообщения	46 байт
Общие накладные расходы на передачу верификационного сообщения	$64 \times K + 512$ байт, где K – количество полученных RSU сообщений от автомобилей за установленный период времени T

Исходя из полученных экспериментальных данных можно сделать вывод о том, что в схеме на основе инфраструктуры придорожных блоков самым затратным с точки зрения времени выполнения и количества накладных расходов является этап рукопожатия. Так, накладные расходы на совершение рукопожатия с автомобилем составляют 2461 байт и превышают затраты в 1825 байт на отправление сообщения в схеме PKI. То же самое можно сказать и о времени выполнения: 3,79 мс на полный цикл отправки и получения сообщения в PKI против 12,08 мс на проведения этапа рукопожатия в разработанной схеме.

Однако стоит отметить, что схема PKI была бы эффективнее только в случае, если каждое ТС отправляло бы только одно сообщение во время проведения моделирования. В реальных условиях одно ТС может отправлять произвольное количество сообщений за рассматриваемый интервал времени. В рамках проводимого моделирования каждое транспортное средство после получения информации об инциденте делает широковещательную рассылку еще трех сообщений о полученной информации.

На рисунке 93 представлен график, построенный на основе экспериментальных данных, полученных с помощью проведения моделирования. Он отображает зависимость времени необходимого для рассылки всех сообщений в сети после получения информации

об инциденте от количества участвующих в моделировании ТС. Исходя из графика, видно, что уже при участии 5 ТС, время, необходимое на обработку всех сообщений, у разработанной схемы меньше – 13,363 мс против 19,657 мс у схемы РКІ. При этом скорость роста времени у разработанной схемы намного меньше, чем у РКІ.

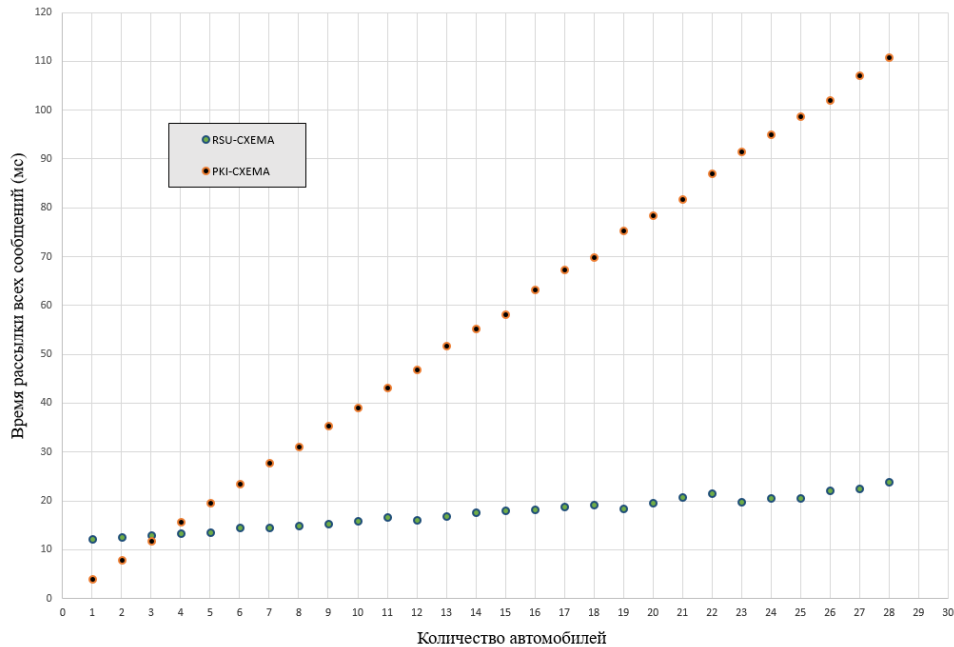


Рисунок 93 – График зависимости времени рассылки всех сообщений от количества ТС в моделируемом сценарии

Далее проведена оценка зависимость накладных расходов на передачу всех сообщений в сети от количества участвующих в моделировании автомобилей. Предположим, что имеется N транспортных средств в диапазоне текущего RSU, и каждое из них будет передавать K сообщений в течение периода T . Накладные расходы в зависимости от количества транспортных средств с количеством сообщений K для схем на основе РКІ и RSU показаны на рисунке 94 для различного количества сообщений.

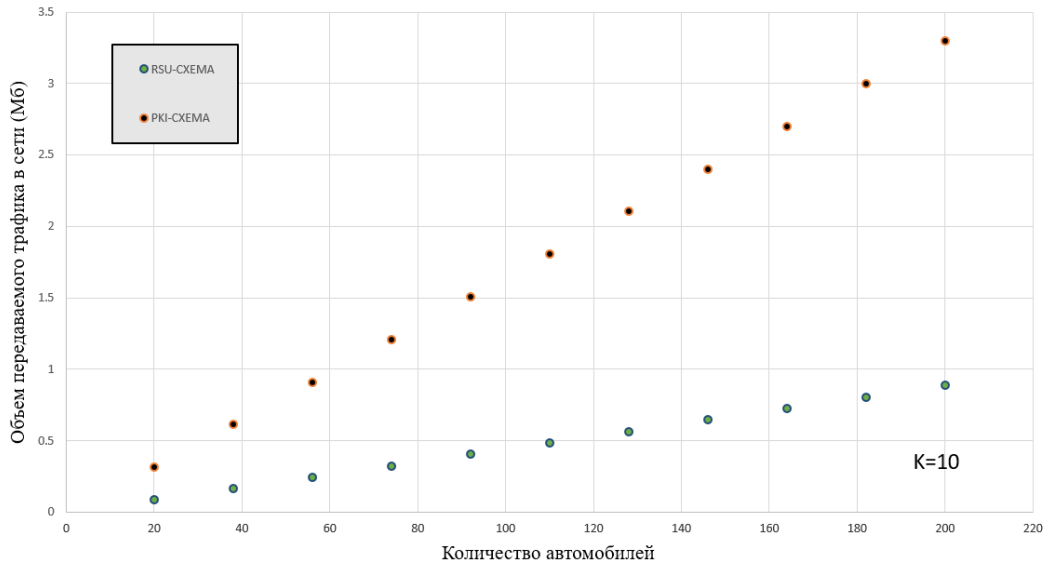
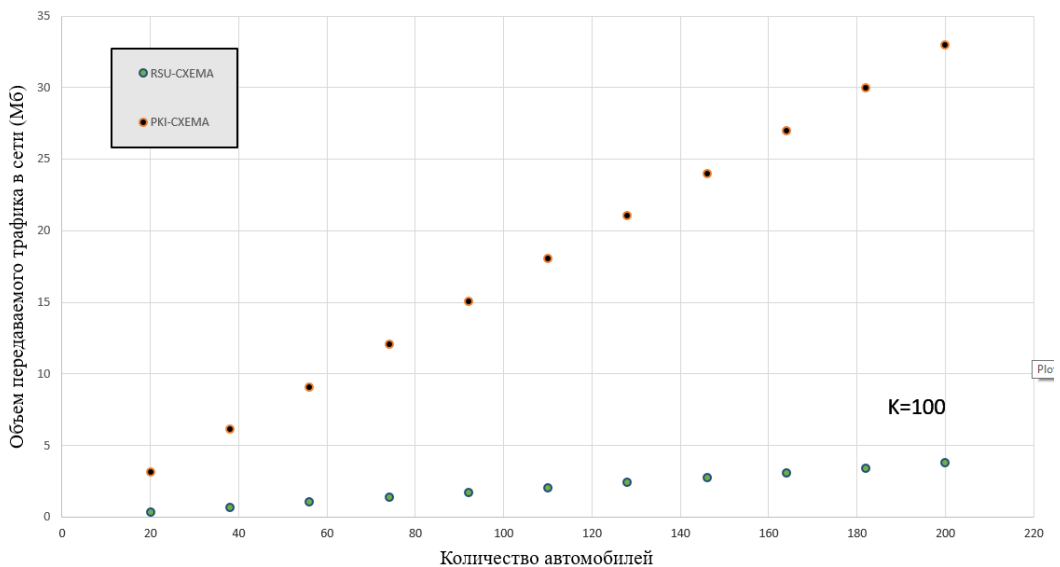
а) $K = 10$ б) $K = 100$

Рисунок 94 – Зависимость объема трафика в сети от количества ТС в сценарии

Как видно из рисунков, для $K = 100$ и количестве ТС равном 200 накладные расходы на передачу для схемы на основе RSU почти в 7 раз меньше, чем для схемы на основе PKI.

4.4.2 Обнаружение аномального поведения транспортных средств

В данном разделе рассматривается задача обеспечения защиты от внутренних угроз. При защите от внутренних угроз предполагается, что злоумышленником является аутентифицированное и авторизованное транспортное средство, поэтому одним из основных подходов в этом случае является обнаружение аномального поведения участников движения.

4.4.2.1 Децентрализованный алгоритм обнаружения аномального поведения

Децентрализованный алгоритм обнаружения аномального поведения предполагает, что вычисления локального и глобального доверия будут проходить в головных устройствах (ГУ) самих транспортных средств. Данная концепция потенциально более подвержена различным видам атак, так как отсутствуют заведомо доверенные объекты дорожной инфраструктуры, которые будут гарантировать корректность вычисленных значений глобального доверия в сети. Далее будет рассмотрен сам алгоритм, и способы противодействия описанной выше проблемы.

За основу модели был взят алгоритм EigenTrust [371]. Изначально он разрабатывался для пиринговых файлообменных сетей и использовался для расчета локального и глобального доверия между ее пирами. Основные положения EigenTrust:

- саморегулируемая система. Общая этика участников обеспечивается самими участниками, а не доверенным органом;

- низкий начальный уровень доверия у новичков. Уровень доверия должен быть получен в результате последовательного положительного взаимодействия участниками. То есть нарушителям должно быть не выгодно изменять идентификаторы, чтобы стать новичком в сети;

- отсутствие объектов инфраструктуры. Участники самостоятельно осуществляют вычисление, хранение и передачу данных, которые позволяют давать оценки уровня локального и глобального доверия в сети;

- устойчивость к союзам нарушителей, которые знают друг друга и совершают злонамеренные действия, т.е. противодействие к атаке в сговоре.

Все эти положения делают данный алгоритм потенциально устойчивым к различным видам атак, следовательно, некоторая модификация данного алгоритма может быть использована для обнаружения аномального поведения в автомобильных сетях VANET.

Далее будет рассмотрен алгоритм на основе распределенного алгоритма EigenTrust, с внесением некоторых модификаций, которые позволят использовать его в автомобильных сетях в качестве модели обнаружения аномального поведения транспортных средств. В EigenTrust ведется подсчет верных и ложных сообщений, т.е. каждое транспортное средство хранит информацию о взаимодействиях с другими транспортными средствами. Эти данные используются для базового расчета уровня локального доверия:

$$s_{ij} = sat(i, j) - unsat(i, j), \quad (88)$$

где $sat(i, j)$ – количество верных сообщений;

$unsat(i, j)$ – количество ложных сообщений, которые получило транспортное средство v_i посредством взаимодействия с v_j .

Данный базовый расчет не предполагает защиты от злоумышленников, которые могут присвоить высокие значения локального доверия своим союзникам и низкие добросовестным транспортным средствам. Для решения данной проблемы происходит нормализация и агрегирование значений локального доверия.

Нормализованный вид значения локального доверия c_{ij} выглядит следующим образом:

$$c_{ij} = \frac{\max(s_{ij}, 0)}{\sum_j \max(s_{ij}, 0)}. \quad (89)$$

У данной нормализации есть недостаток: нормализованные значения не предполагают различия между транспортными средствами, с которыми отсутствует какое-либо взаимодействие, и транспортными средствами, с которыми было превалирующее количество негативных взаимодействий. Несмотря на это, данный алгоритм предполагает противодействие различным видам атак, которые направлены непосредственно на саму модель обнаружения аномального поведения.

Агрегирование нормализованных значений локального доверия происходит по следующей логике: транспортное средство v_i при взаимодействии с v_j получает данные о локальных довериях v_j по отношению к другим транспортным средствам.

Объединение своих и чужих наблюдений позволит осуществить расчет значения глобального доверия:

$$t_{ik} = \sum_j c_{ij} c_{jk}, \quad (90)$$

где t_{ik} – значение глобального доверия, которое транспортное средство v_i оказывает к v_k на основе опроса других транспортных средств.

Данная запись представима и в матричном виде: обозначим матрицу локального доверия как C , \vec{c}_i – вектор локального доверия ТС i , \vec{t}_i – вектор глобального доверия ТС i . Тогда $\vec{t}_i = C^T \vec{c}_i$.

Подобный подход позволит расширить представление v_i о сети в целом не только на основе своих наблюдений, но и на основе наблюдений других транспортных средств.

Для противодействия вредоносным союзам транспортных средств вводится коэффициент a относительной важности мнения авторитетных и обычных транспортных средств. В итерационном виде формула расчета примет следующий вид:

$$\vec{t}^{(k+1)} = (1 - a)C^T \vec{t}^{(k)} + a\vec{p}, \quad (91)$$

где $a \in [0,1]$;

\vec{p} – начальный вектор доверия к транспортным средствам в сети.

Вектор \vec{p} рассчитывается следующим образом:

$$p_i = \begin{cases} 1/|P|, & \text{если } v_i \in P \\ 0, & \text{иначе} \end{cases}, \quad (92)$$

где P – множество авторитетных транспортных средств.

Использование данного алгоритма в контексте автомобильных сетей VANET накладывает некоторые ограничения. Кратковременные связи между транспортными средствами накладывают ограничение на возможность масштабирования представления транспортного средства о других транспортных средствах, так как взаимодействие между ними происходит только в момент дистанционной доступности того или иного транспортного средства. Существует потенциальная возможность предоставления заведомо ложных данных об уровне локального доверия транспортным средством-нарушителем, которые могут быть использованы для расчета вектора глобального доверия добросовестными транспортными средствами.

Ввиду этих и других ограничений и недостатков были введены и рассмотрены дополнительные коэффициенты и положения, необходимые для возможности качественного расчета уровня локального и глобального доверия в рассматриваемой модели:

- при расчете значения локального доверия вводится коэффициент относительной важности ложных сообщений, позволяющий повысить важность негативных взаимодействий. Данный коэффициент не будет влиять на работу сети в случае проведения атаки в сговоре, так как после нормализации данных нижний предел уровня доверия все еще будет равен нулю. Следовательно, нарушители при совершении атаки, оставляя исключительно негативные отзывы о взаимодействии с добросовестными транспортными средствами, не смогут влиять на сеть. В то же время, оценки добросовестных транспортных средств о негативном взаимодействии с транспортными средствами-нарушителями будут значительно понижать уровень доверия сети к нарушителям в целом;

- при первом вычислении значений глобального доверия происходит расчет на основе двух типов представления транспортного средства о сети: вектор значений локального доверия личного представления о сети и матрица значений локальных доверий C , которая содержит свои и чужие представления о сети. Расчеты при дальнейших итерациях алгоритма, будут проводится на основе рассчитанного вектора значений глобального доверия и матрицы C . Использование подобного подхода позволит учитывать и накапливать информацию об исторических взаимодействиях участников сети. Следовательно, данный подход позволит противодействовать атаке on-off, так как негативное взаимодействие с нарушителями будет учтено при дальнейших расчетах вектора значений глобального доверия даже в случае, если нарушители начнут имитировать поведение добросовестных транспортных средств;

- введение авторитетных транспортных средств позволит задать начальные высокие значения доверия некоторым ее заведомо доверенным участникам. Данный подход позволит избавиться от возможности нарушить качественную работу модели на первых итерациях работы алгоритма расчета глобального доверия;

- коэффициент относительной важности мнения авторитетных транспортных средств a позволит учесть случай, когда в ряды заведомо доверенных транспортных средств может попасть нарушитель. Дополнительно, данный коэффициент учитывается и для важности мнения не авторитетных транспортных средств, что позволит производить более точную настройку при различных сценариях взаимодействия.

Использование децентрализованного алгоритма в контексте автомобильных сетей состоит из следующих шагов:

- 1) сбор данных, необходимых для расчета локального доверия: верные и ложные сообщения;
- 2) расчет значений локального доверия на основе формулы (88);
- 3) нормализация значений локального доверия по формуле (89);
- 4) сбор данных, необходимых для расчета глобального доверия: значения локального доверия и матрица локальных значений;
- 5) вычисление значений глобального доверия: при первом расчете на основе формулы (90), при последующих на основе формулы (91);
- 6) фильтрация значений доверия с учетом выбранного порога.

4.4.2.2 Экспериментальные исследования алгоритма обнаружения аномального

поведения

Экспериментальные исследования разработанного алгоритма обнаружения аномального поведения проводятся для подтверждения работоспособности алгоритмов в различных сценариях. Сравнение эффективности разработанного алгоритма проводилось с централизованным алгоритмом AATMS [276] в сценарии нормального движения (без атак) и в сценариях с атаками. Злоумышленниками считаются транспортные средства, нарушающие скоростной режим. Промежуточные значения глобальных доверий рассматриваются на 1, 10, 30 и 50 итерациях работы алгоритма и предварительно проходят нормализацию.

Движение в сценарии без атак

Экспериментальные исследования в сценариях без атак проводились со следующими значениями заданных параметров алгоритмов:

Сценарий 1: сценарий взаимодействия транспортных средств при равном количестве нарушителей и доверенных транспортных средств. Параметры сценария: общее количество транспортных средств – 100; количество транспортных средств нарушителей – 20; количество доверенных транспортных средств – 20.

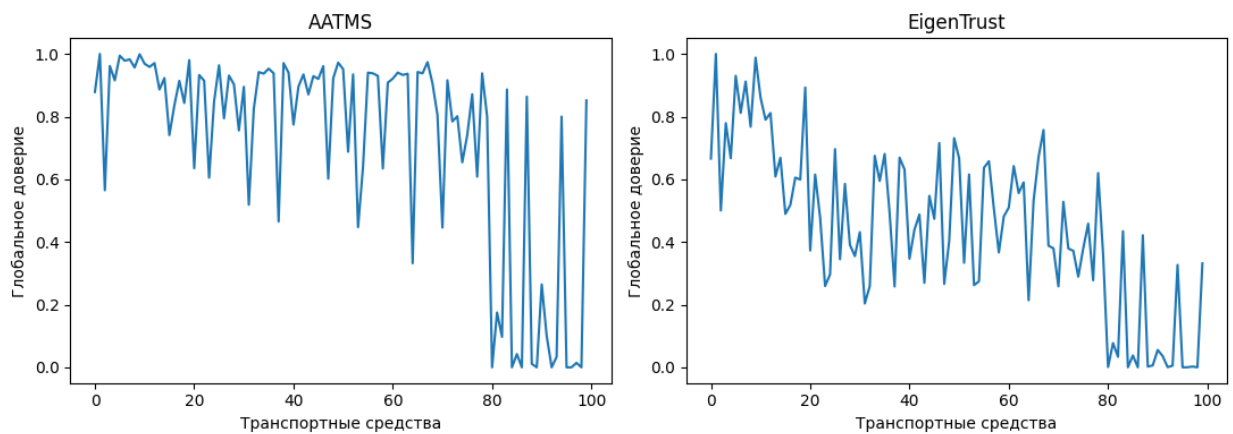


Рисунок 95 – Сценарий 1. Значения глобального доверия на 1 итерации работы алгоритмов

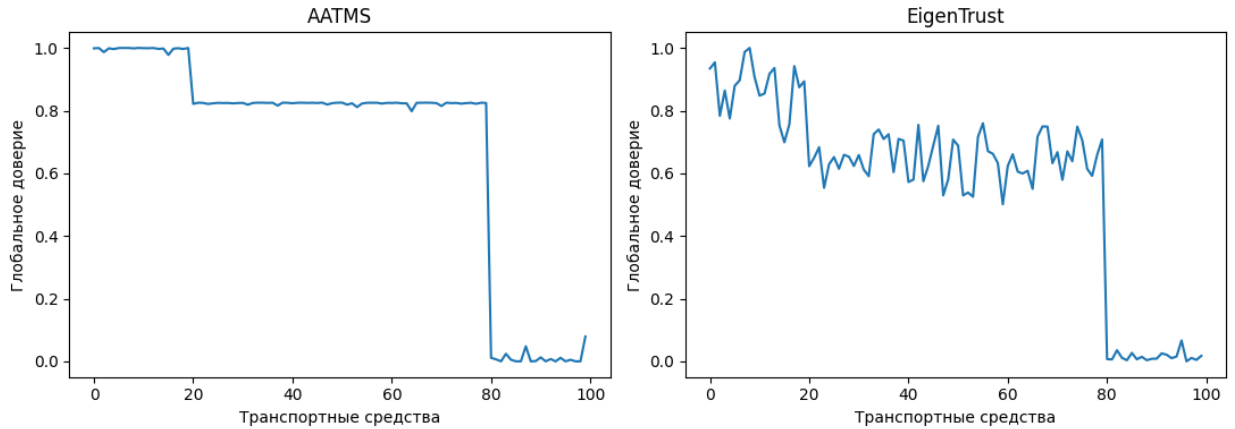


Рисунок 96 – Сценарий 1. Значения глобального доверия на 10 итерации работы алгоритмов

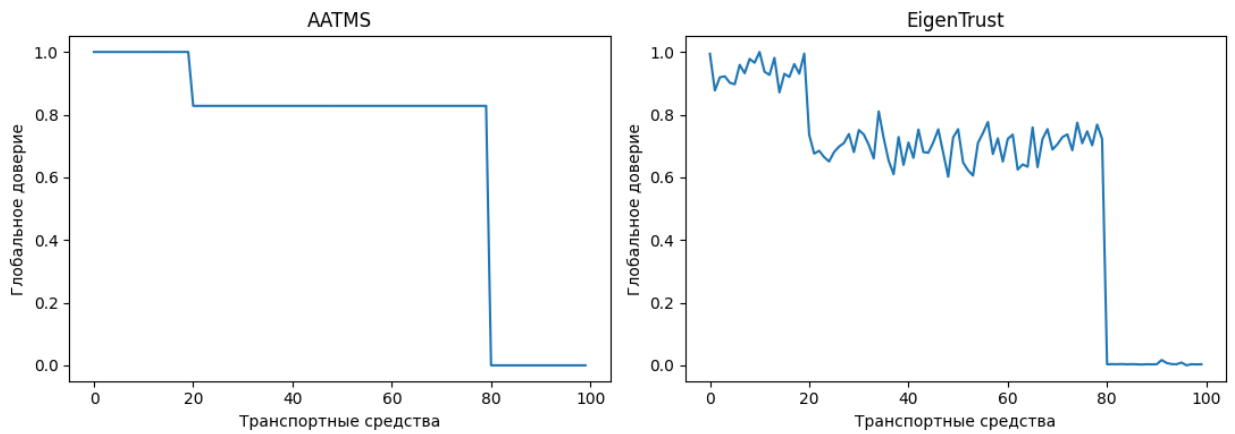


Рисунок 97 – Сценарий 1. Значения глобального доверия на 30 итерации работы алгоритмов

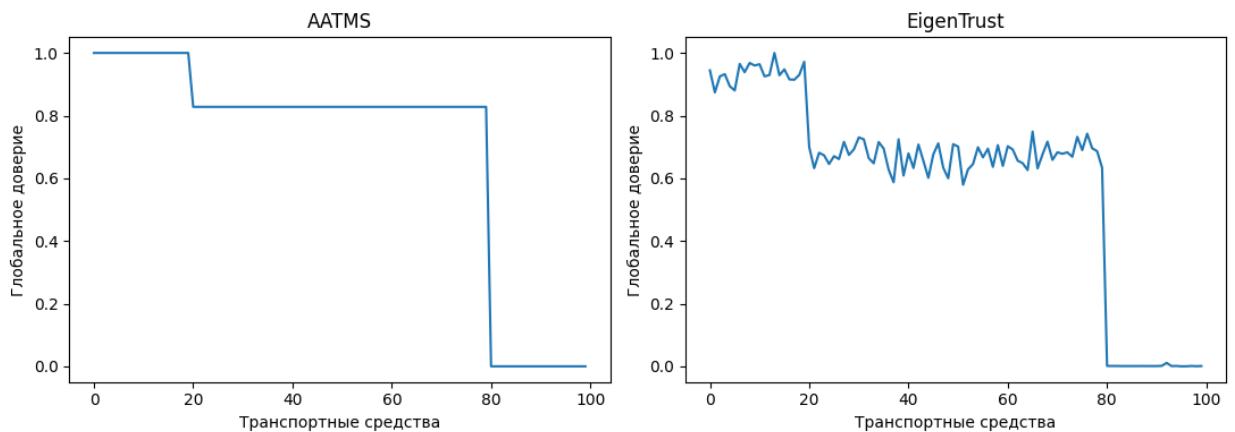


Рисунок 98 – Сценарий 1. Значения глобального доверия на 50 итерации работы алгоритмов

В сценарии 1 видно, что уже на 10 итерации работы оба алгоритма успешно справились с обнаружением аномального поведения ТС-злоумышленников. Алгоритм AATMS к 30 итерации явно и однозначно выделил 3 типа транспортных средств: авторитетные, обычные и нарушители. Алгоритм на основе EigenTrust дает более разнообразные оценки транспортным средствам в пределах его типа.

Сценарий 2: сценарий взаимодействия транспортных средств при преобладающем количестве нарушителей над авторитетными транспортными средствами. Параметры сценария: общее количество транспортных средств – 100; количество транспортных средств нарушителей – 50; количество доверенных транспортных средств – 20.

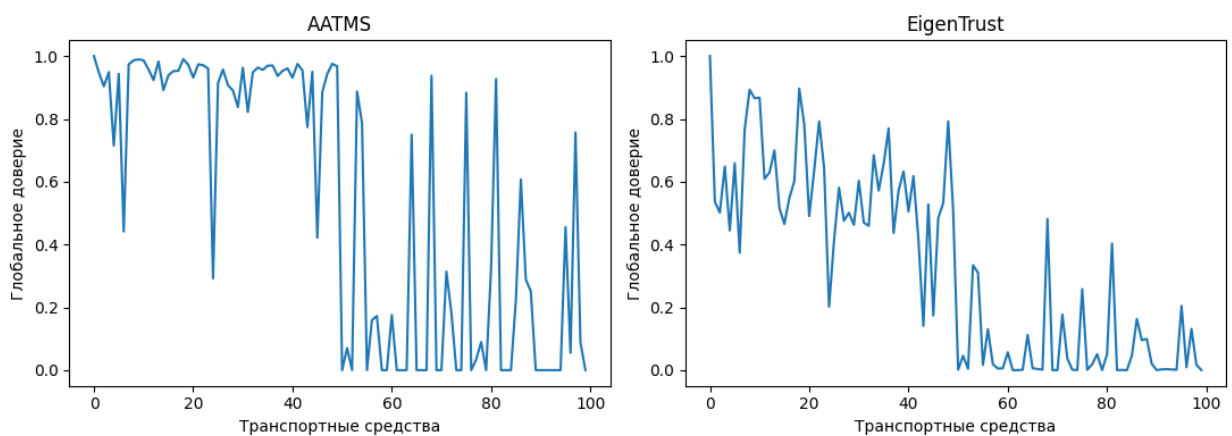


Рисунок 99 – Сценарий 2. Значения глобального доверия на 1 итерации работы алгоритмов

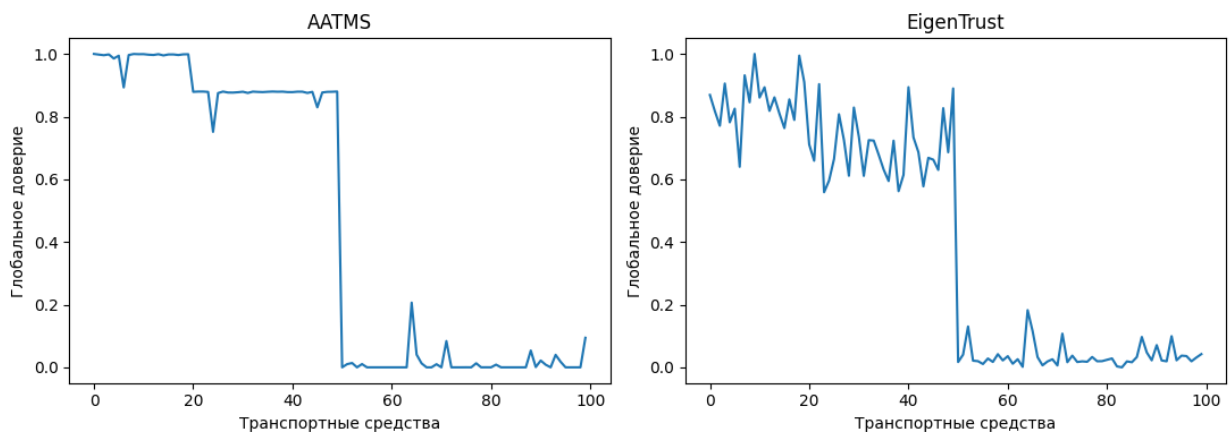


Рисунок 100 – Сценарий 2. Значения глобального доверия на 10 итерации работы алгоритмов

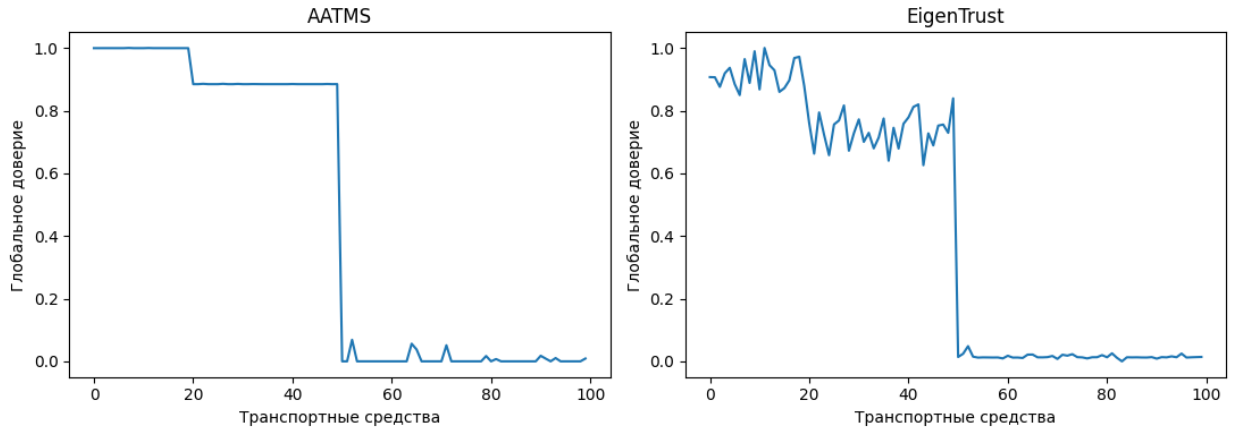


Рисунок 101 – Сценарий 2. Значения глобального доверия на 30 итерации работы алгоритмов

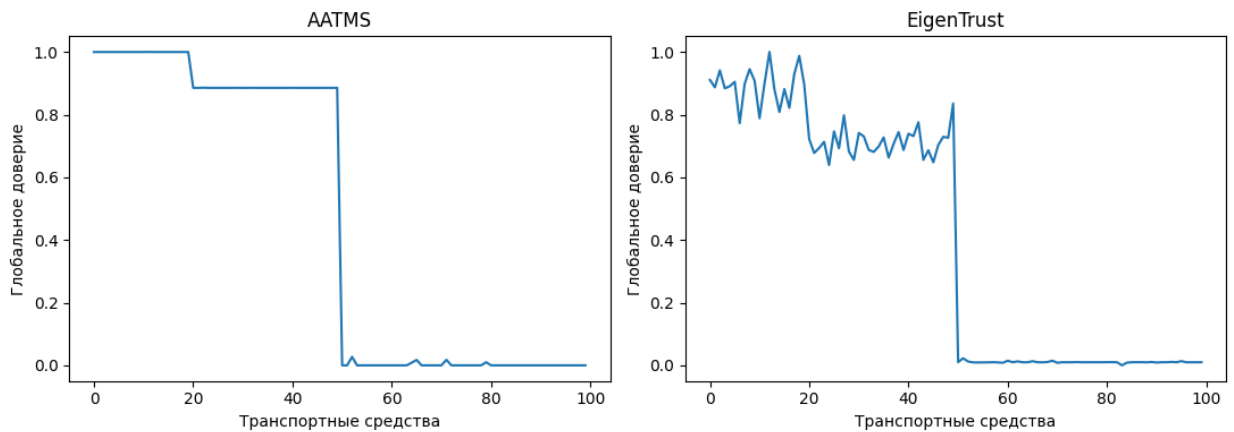


Рисунок 102 – Сценарий 2. Значения глобального доверия на 50 итерации работы алгоритмов

В сценарии 2 на 10 итерации оба алгоритма успешно выделили недобросовестные транспортные средства. Алгоритм AATMS к 10 итерации работы явно обнаружил и авторитетные транспортные средства. Алгоритм на основе EigenTrust с этой задачей справился частично лишь к 30 итерации.

Сценарий 3: сценарий взаимодействия транспортных средств при преобладающем количестве нарушителей над добросовестными транспортными средствами при отсутствии авторитетных транспортных средств. Параметры сценария: общее количество транспортных средств – 100; количество транспортных средств нарушителей – 60; количество доверенных транспортных средств – 0.

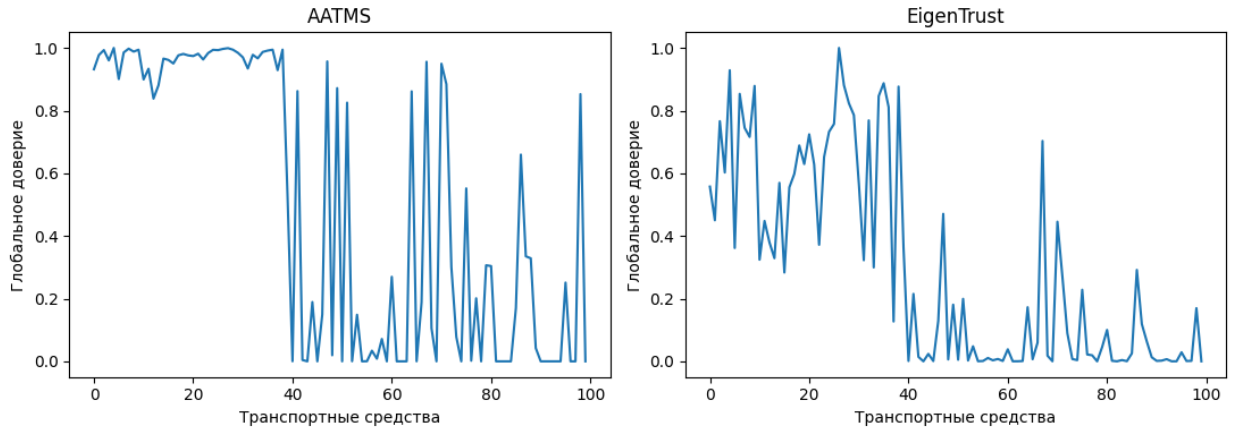


Рисунок 103 – Сценарий 3. Значения глобального доверия на 1 итерации работы алгоритмов

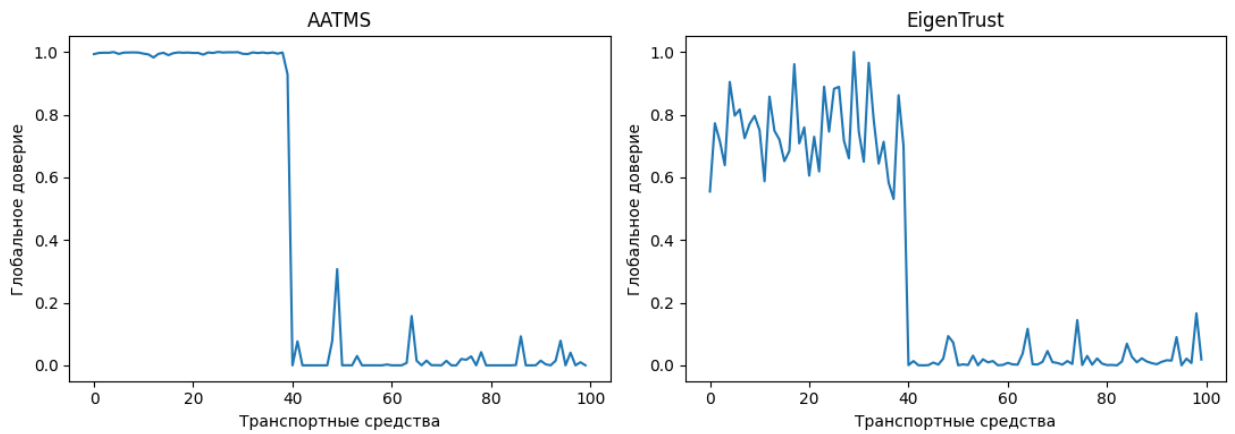


Рисунок 104 – Сценарий 3. Значения глобального доверия на 10 итерации работы алгоритмов

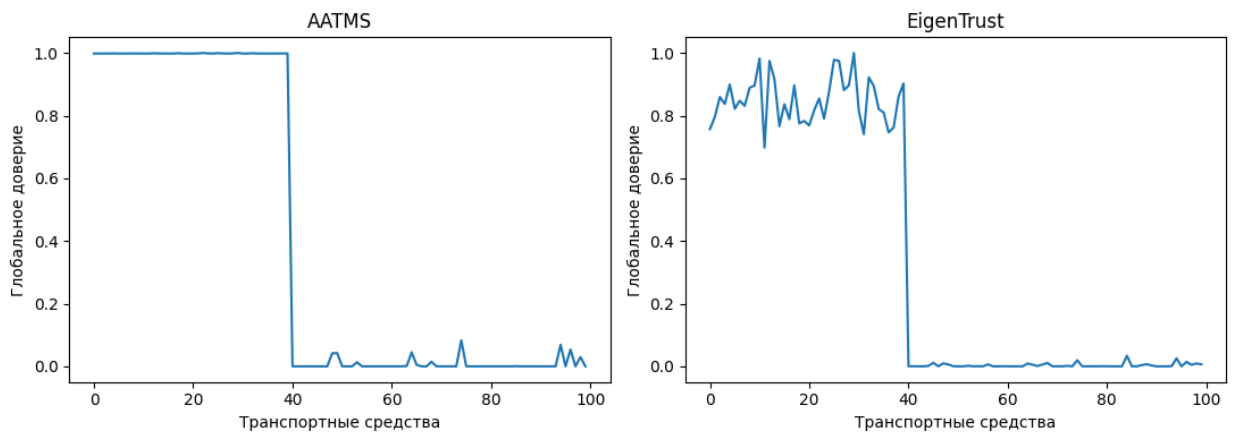


Рисунок 105 – Сценарий 3. Значения глобального доверия на 30 итерации работы алгоритмов

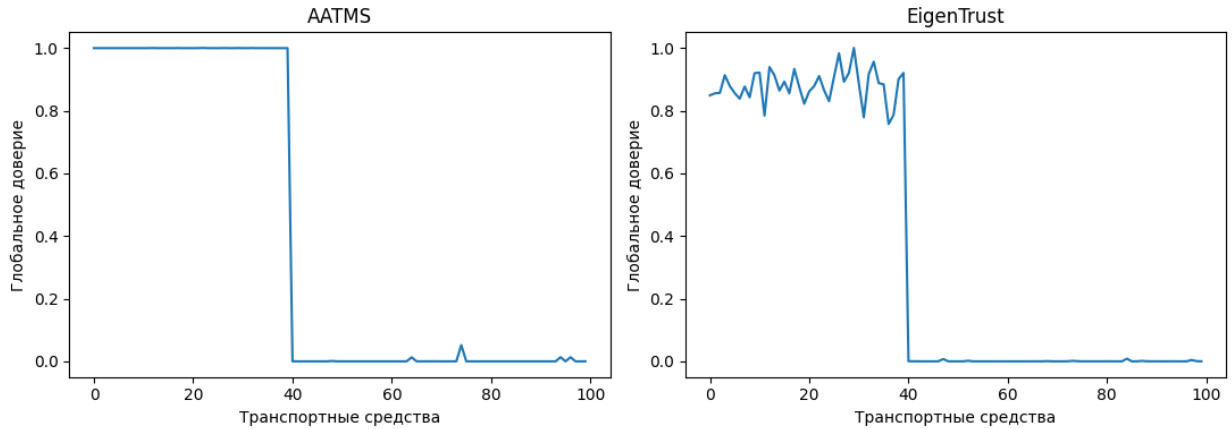


Рисунок 106 – Сценарий 3. Значения глобального доверия на 50 итерации работы алгоритмов

Третий сценарий взаимодействия показал, что оба алгоритма успешно справляются с обнаружением недобросовестных транспортных средств к 10 итерации. Преобладающее количество транспортных средств нарушителей и отсутствие авторитетных транспортных средств не оказывает значительного влияния на результат обнаружения аномального поведения.

Экспериментальные исследования в сценариях без атак отражают значения вектора глобального доверия транспортных средств, которые позволяют сделать выводы о качественном выявлении нарушителей.

Во всех рассмотренных сценариях без атак удалось выявить нарушителей вне зависимости от их количества и количества авторитетных транспортных средств.

Движение в сценарии с атаками

Атака on-off. Атака типа on-off предполагает, что злоумышленники в течении некоторого промежутка времени ведут себя как добросовестные транспортные средства. После заданной итерации недобросовестные транспортные средства начинают превышать скоростной режим. Экспериментальные исследования сценариев с данным видом атак направлены на оценку влияния исторических негативных взаимодействий при вычислении глобального доверия в сети.

Сценарий 4: сценарий взаимодействия транспортных средств при проведении атаки типа on-off. Параметры сценария: общее количество транспортных средств – 100; количество транспортных средств нарушителей – 30; количество доверенных транспортных средств – 10; атака on-off – включена; итерация начала проведения атаки on-off – 10.

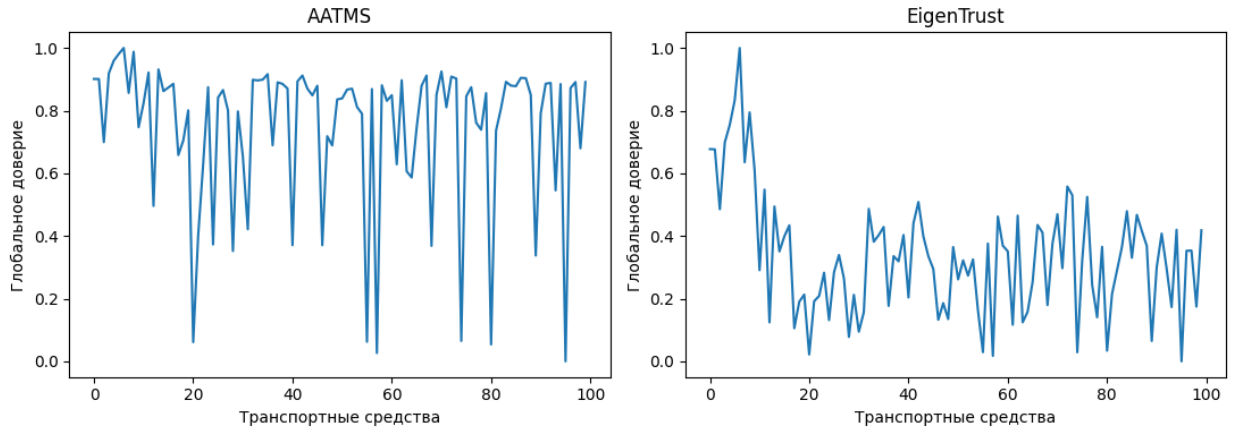


Рисунок 107 – Сценарий 4. Значения глобального доверия на 1 итерации работы алгоритмов

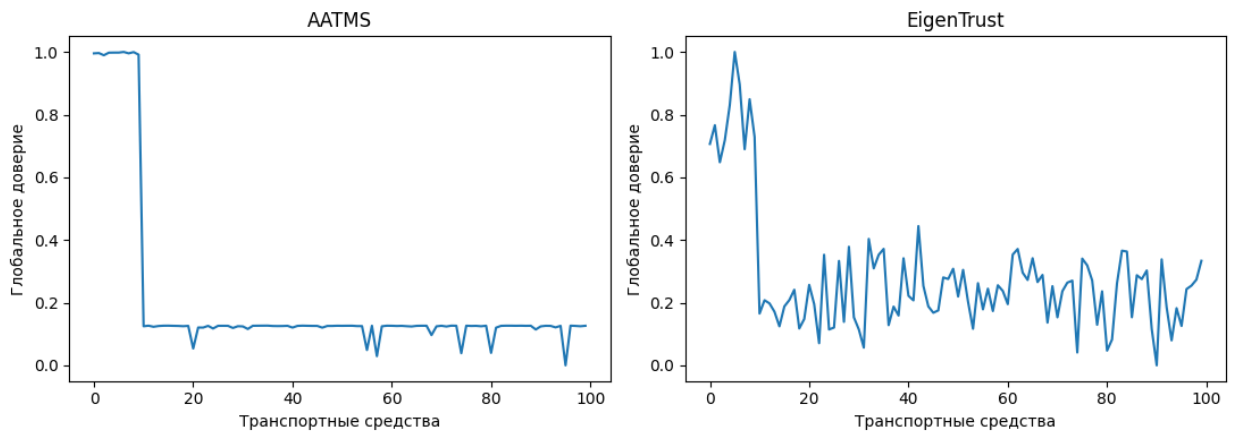


Рисунок 108 – Сценарий 4. Значения глобального доверия на 10 итерации работы алгоритмов

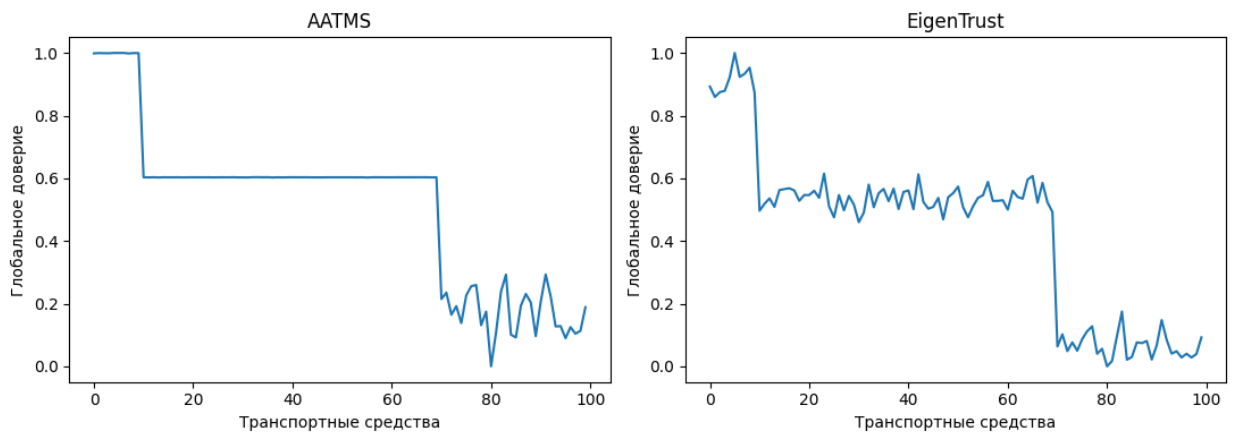


Рисунок 109 – Сценарий 4. Значения глобального доверия на 30 итерации работы алгоритмов

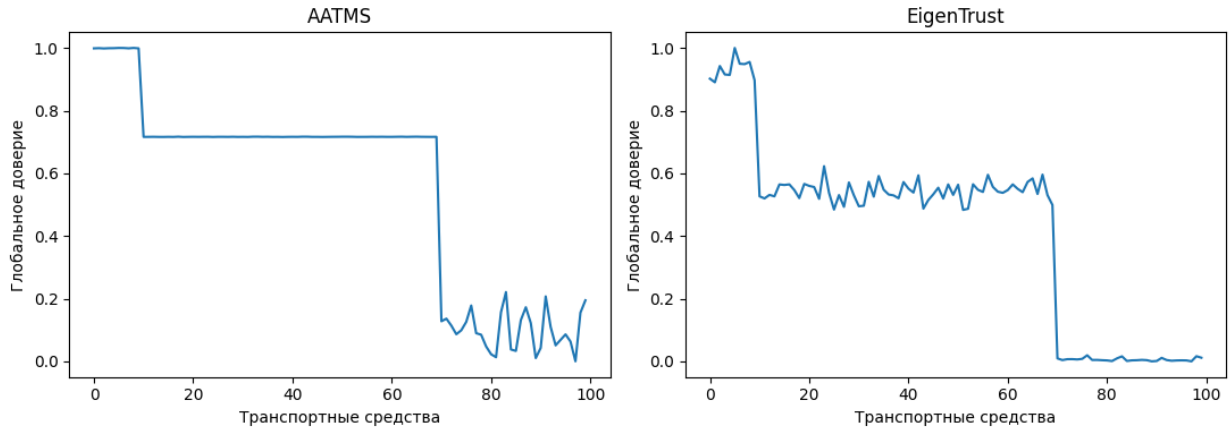


Рисунок 110 – Сценарий 4. Значения глобального доверия на 50 итерации работы алгоритмов

В 4 сценарии после 10 итерации работы алгоритма, происходит переключение логики работы нарушителей, они начинают превышать скоростной режим. Уже к 30 итерации, как показано на рисунке 109, можно сделать вывод, что злоумышленники были успешно определены.

Сценарий 5: сценарий взаимодействия транспортных средств при проведении атаки типа on-off с большим количеством злоумышленников и увеличенной итерацией переключения логики работы нарушителей. Параметры сценария: общее количество транспортных средств – 100; количество транспортных средств нарушителей – 50; количество доверенных транспортных средств – 0; атака on-off – включена; итерация начала проведения атаки on-off – 25.

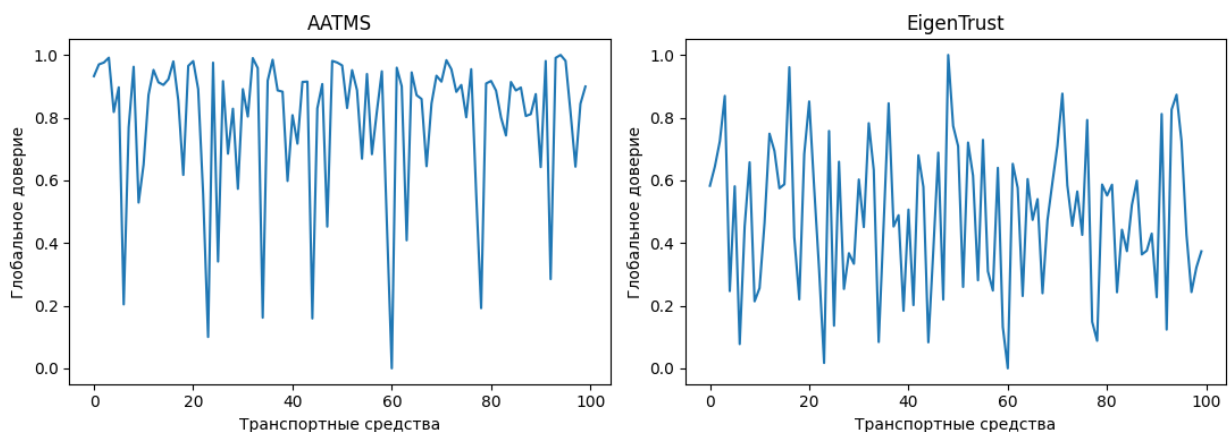


Рисунок 111 – Сценарий 5. Значения глобального доверия на 1 итерации работы алгоритмов

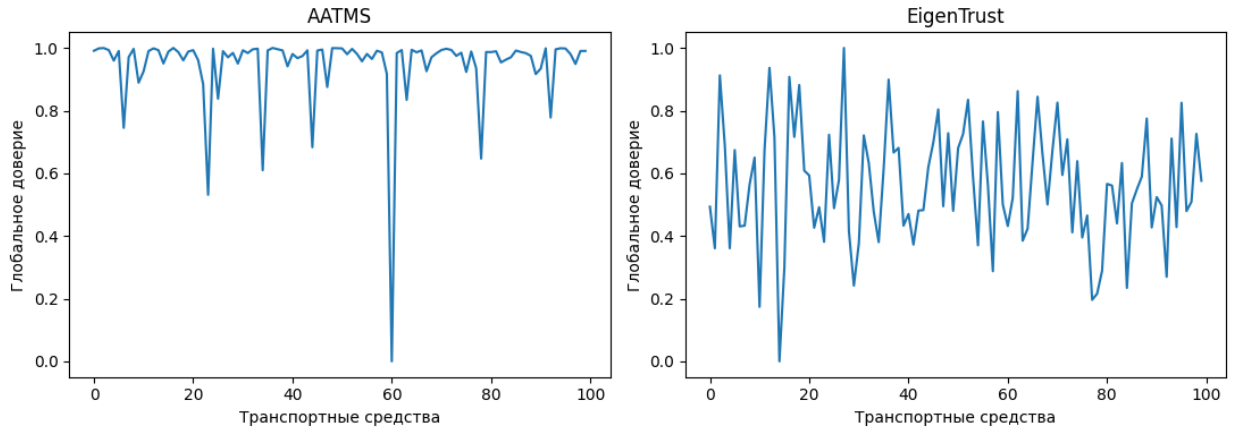


Рисунок 112 – Сценарий 5. Значения глобального доверия на 10 итерации работы алгоритмов

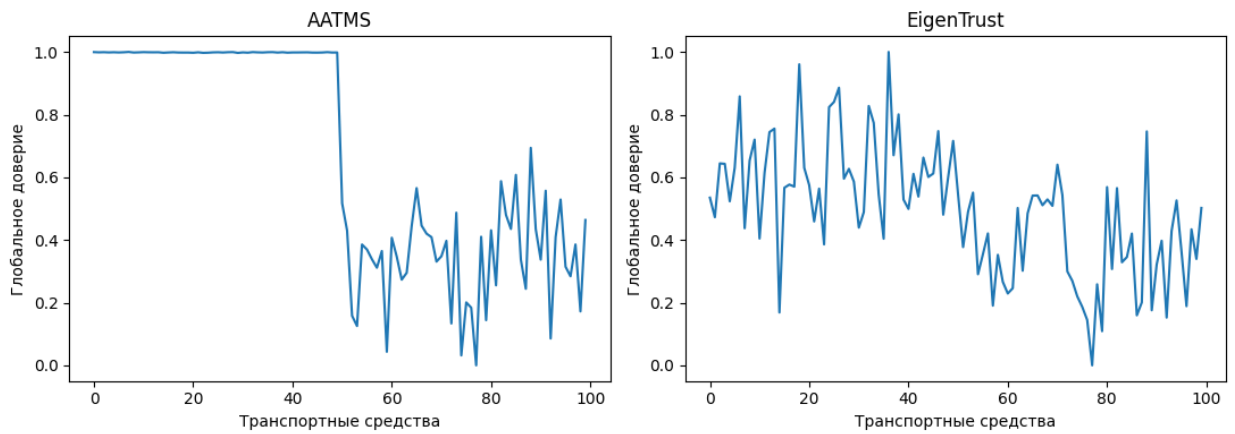


Рисунок 113 – Сценарий 5. Значения глобального доверия на 30 итерации работы алгоритмов

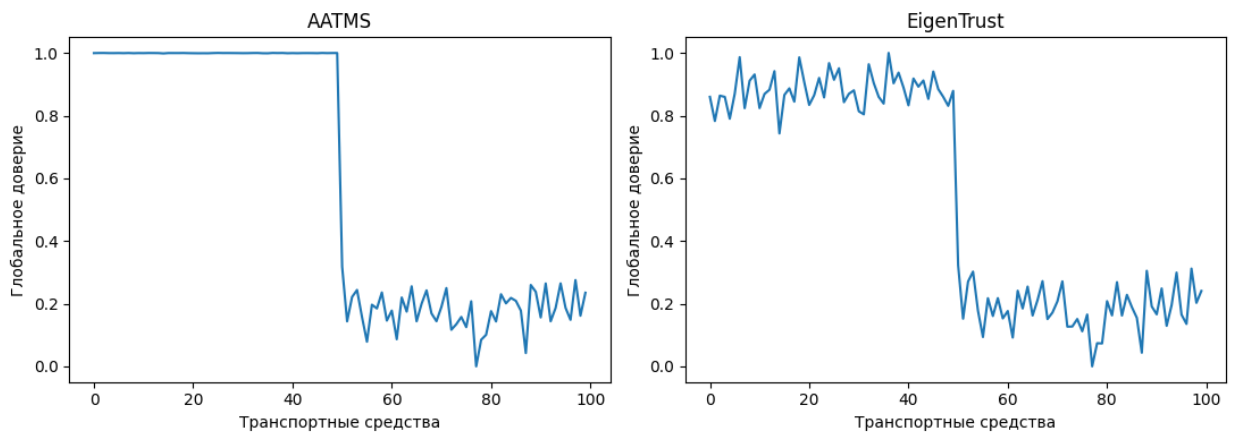


Рисунок 114 – Сценарий 5. Значения глобального доверия на 50 итерации работы алгоритмов

В 5 сценарии изменение логики нарушителей происходит на 25 итерации, как показано на рисунке 113. На графике видно, что за 5 последующих итерации алгоритму на основе AATMS удалось определить нарушителей. А алгоритм на основе EigenTrust к 30 итерации не справился с этой задачей даже частично. К заключительной 50 итерации оба алгоритма успешно справились с задачей.

Во всех рассмотренных сценариях с атакой типа on-off удалось выявить нарушителей вне зависимости от их количества, количества авторитетных транспортных средств и итерации переключения логики работы нарушителей.

Атака в сговоре. Атака в сговоре предполагает, что транспортные средства нарушители образуют союз и начинают сообщать ложную информацию другим транспортным средствам. То есть придерживаются логики: о своих всегда говорят хорошо, о чужих всегда плохо, вне зависимости от фактического скоростного режима другого транспортного средства.

Сценарий б: сценарий взаимодействия транспортных средств при проведении атаки в сговоре. Параметры сценария: общее количество транспортных средств – 100; количество транспортных средств нарушителей – 20; количество доверенных транспортных средств – 10; атака в сговоре – включена; атака on-off – отключена.

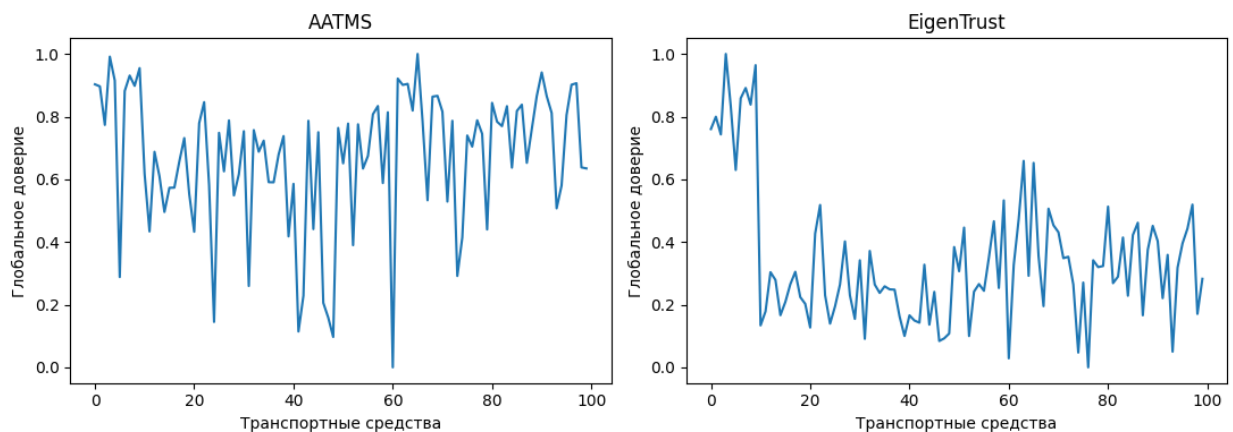


Рисунок 115 – Сценарий б. Значения глобального доверия на 1 итерации работы алгоритмов

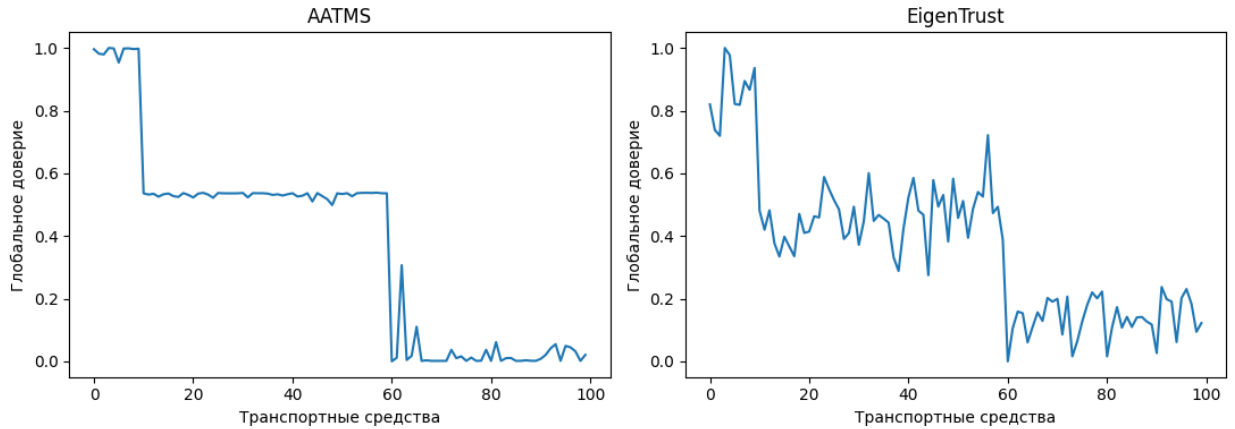


Рисунок 116 – Сценарий 6. Значения глобального доверия на 10 итерации работы алгоритмов

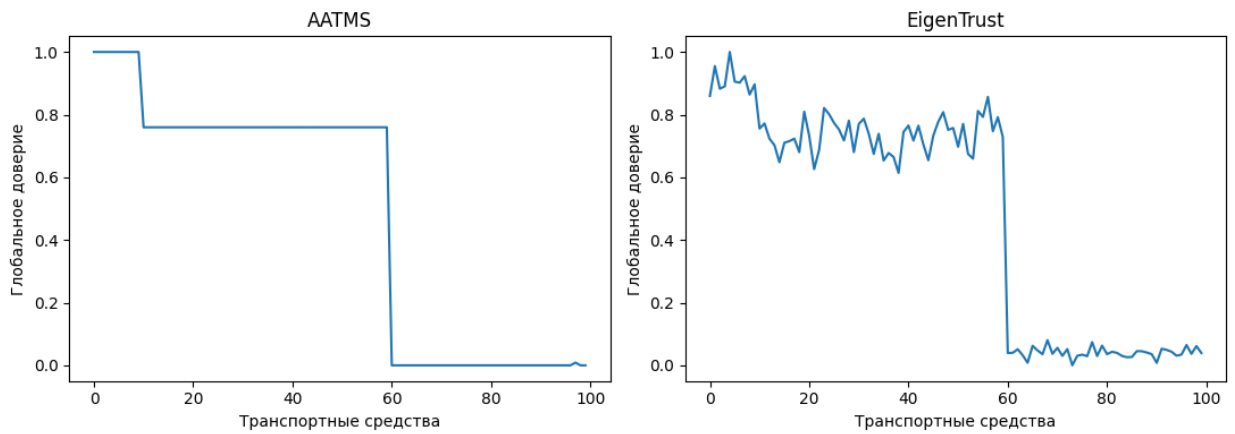


Рисунок 117 – Сценарий 6. Значения глобального доверия на 30 итерации работы алгоритмов

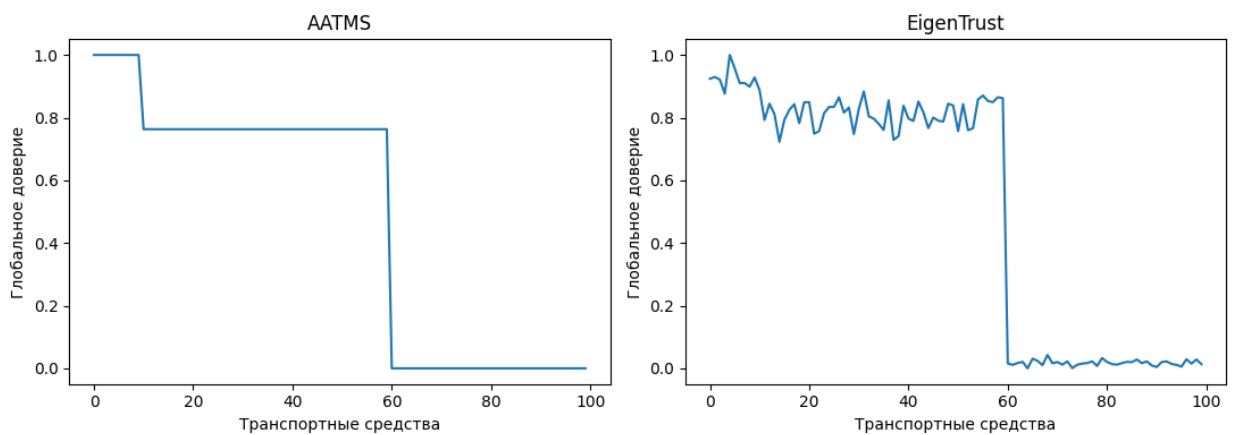


Рисунок 118 – Сценарий 6. Значения глобального доверия на 50 итерации работы алгоритмов

В 6 сценарии оба алгоритма к 10 итерации успешно обнаружили транспортные средства нарушители.

Сценарий 7: сценарий взаимодействия транспортных средств при проведении атаки в сговоре с преобладающим количеством транспортных средств нарушителей. Параметры сценария: общее количество транспортных средств – 100; количество транспортных средств нарушителей – 60; количество доверенных транспортных средств – 10; атака в сговоре – включена; атака on-off – отключена.

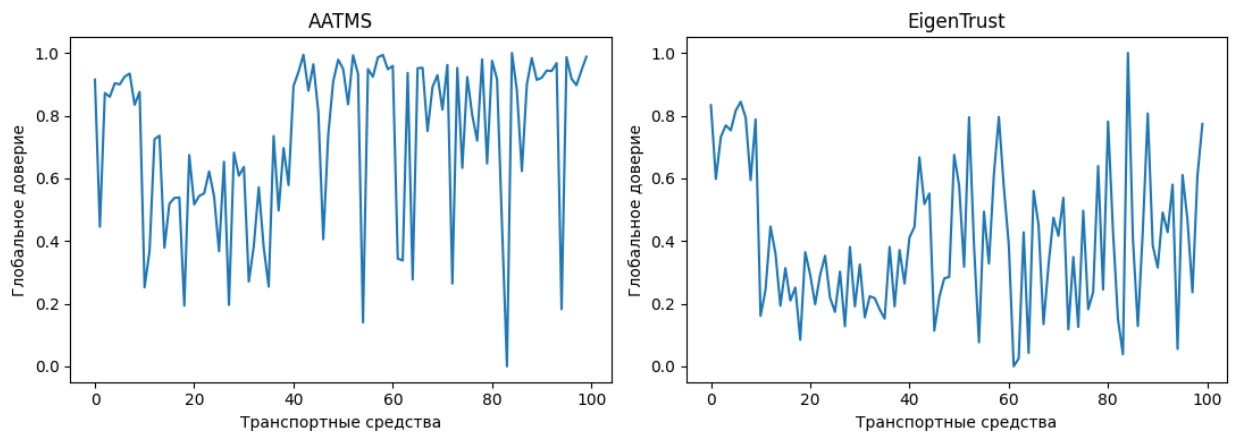


Рисунок 119 – Сценарий 7. Значения глобального доверия на 1 итерации работы алгоритмов

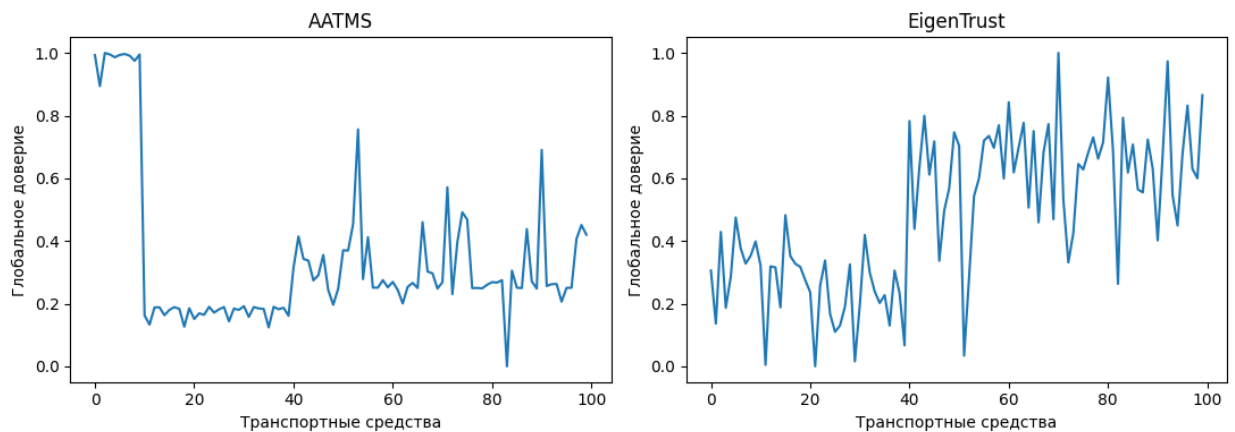


Рисунок 120 – Сценарий 7. Значения глобального доверия на 10 итерации работы алгоритмов

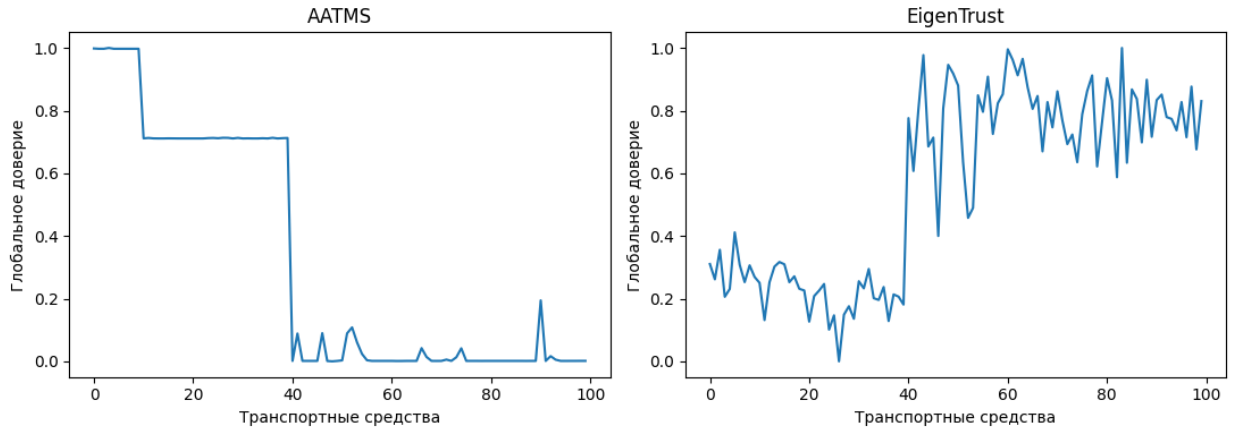


Рисунок 121 – Сценарий 7. Значения глобального доверия на 30 итерации работы алгоритмов

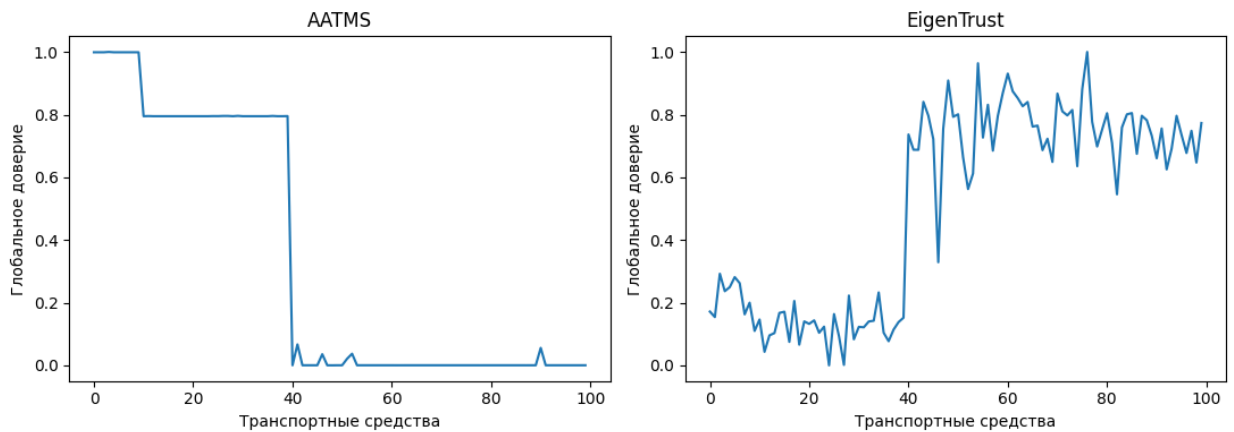


Рисунок 122 – Сценарий 7. Значения глобального доверия на 50 итерации работы алгоритмов

В 7 сценарии при значительном количестве нарушителей алгоритм на основе EigenTrust с задачей обнаружения не справился. Причиной тому послужил децентрализованный тип устройства алгоритма – гарантия безопасности сетей децентрализованного типа достигается большинством добросовестных участников.

Сценарий 8: сценарий взаимодействия транспортных средств с атакой в сговоре при отсутствии авторитетных транспортных средств. Параметры сценария: общее количество транспортных средств – 100; количество транспортных средств нарушителей – 30; количество доверенных транспортных средств – 0; атака в сговоре – включена; атака on-off – отключена.

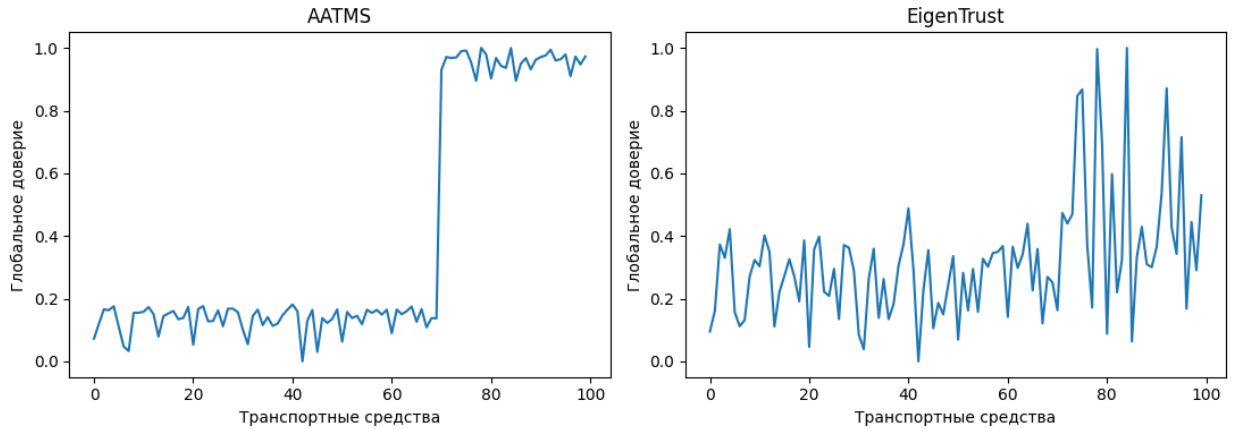


Рисунок 123 – Сценарий 8. Значения глобального доверия на 1 итерации работы алгоритмов

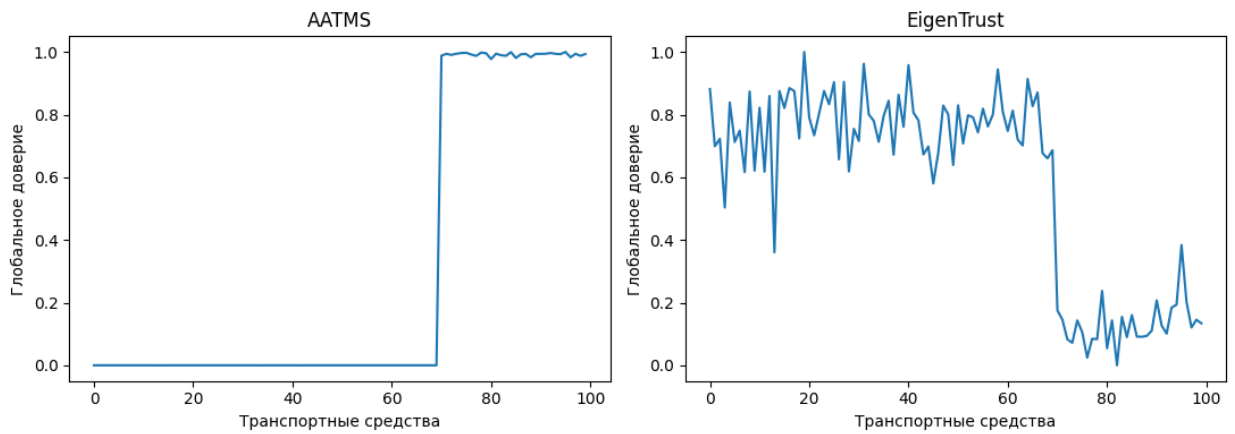


Рисунок 124 – Сценарий 8. Значения глобального доверия на 10 итерации работы алгоритмов

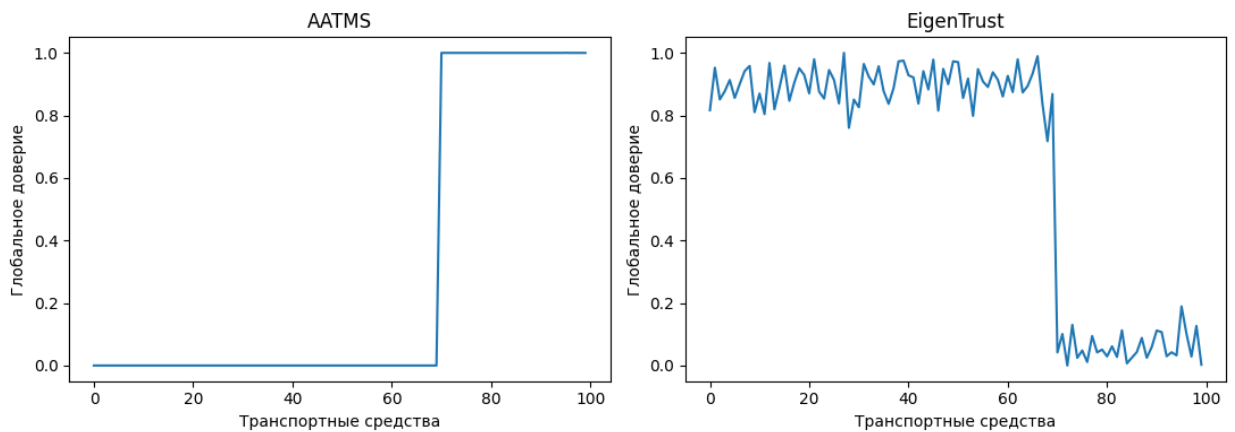


Рисунок 125 – Сценарий 8. Значения глобального доверия на 30 итерации работы алгоритмов

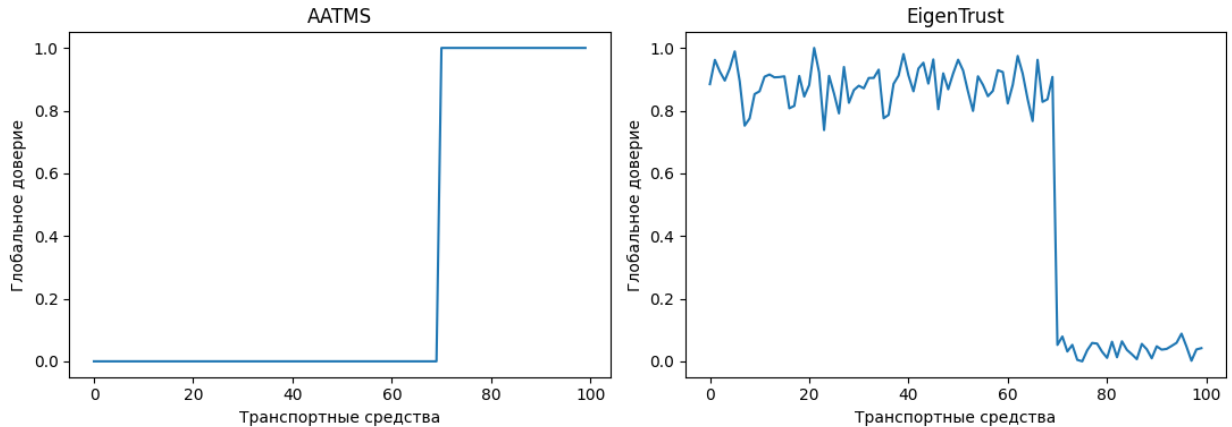


Рисунок 126 – Сценарий 8. Значения глобального доверия на 50 итерации работы алгоритмов

Сценарий 8 отражает ситуацию, когда в сети отсутствуют авторитетные транспортные средства. Алгоритм AATMS не предполагает защиту от данного типа атаки в случае отсутствия заведомо доверенных транспортных средств. Алгоритм на основе EigenTrust с задачей в заданных условиях справился уже к 10 итерации.

Экспериментальные исследования в рассмотренных сценариях (сценарии без атак, с атакой on-off и атакой в сговоре) показали, что оба алгоритма способны качественно выделять недобросовестные транспортные средства в сценариях без атак и атакой on-off. Сценарии с атакой в сговоре показали, что существуют ситуации, когда один из алгоритмов не способен либо частично способен справиться с задачей обнаружения аномального поведения транспортных средств:

- алгоритм на основе EigenTrust имеет децентрализованный тип устройства, следовательно, способен качественно выявлять нарушителей только в случае преобладающего большинства добросовестных транспортных средств при проведении атаки в сговоре;

- алгоритм на основе AATMS был спроектирован для противодействия всем рассмотренным видам атак только при наличии авторитетных транспортных средств. В сценариях без атак и атакой on-off данный алгоритм успешно обнаруживает нарушителей и без участия авторитетных транспортных средств в сценарии. В сценарии с атакой в сговоре данная особенность делает обнаружение недобросовестных транспортных средств невозможным.

Однако следует отметить, что сценарий, при котором преобладающие большинство образуют ТС-злоумышленники, является маловероятным в реальных условиях.

4.4.3 Обнаружение аномального поведения транспортных средств в колонне

В данном разделе рассматривается частная задача обеспечения информационной безопасности при движении транспортных средств в колонне. Для поддержания оптимальной дистанции в колонне необходим обмен информацией между транспортными средствами. В данном разделе рассматривается консенсусный подход как стратегия управления колонной транспортных средств (раздел 4.4.3.1). Этот подход учитывает расстояние, скорость и ускорение для определения стратегии управления. В разделе рассматривается совместная стратегия, основанная на алгоритме голосования для обнаружения вредоносных транспортных средств. Предлагается алгоритм обнаружения аномалий, позволяющий обнаружить вредоносное транспортное средство и повысить безопасность движения транспортной колонны (раздел 4.4.3.2). Исследования предложенного алгоритма в сценариях с различными типами вредоносных атак (спуфинг, фальсификация сообщений) представлены в разделе 4.4.3.3.

4.4.3.1 Постановка задачи

Рассмотрим транспортную колонну как группу из N транспортных средств, движущихся по одной полосе дороги. Сеть из N транспортных средств может быть представлена в виде ориентированного графа $G = (V, E, A)$ с множеством узлов $V = \{0, \dots, N - 1\}$, множеством ребер E , и матрицей смежности $A = \alpha_{N \times N}$, описывающей структуру связи. Предполагается, что $\alpha_{ij} = 1$, если существует связь между транспортными средствами i и j , $\alpha_{ij} = 0$ в противном случае. Определим матрицу степеней $D = \text{diag}\{\Delta_0 \Delta_1 \dots \Delta_{N-1}\}$, где $\Delta_i = \sum_{j \in V} \alpha_{ij}$. Набор вредоносных транспортных средств обозначим как $M = \{m_1, m_2 \dots m_p\}$, $p < N$.

Существуют различные топологии информационных потоков при движении транспортных средств в колонне: каждый-с-каждым, лидер-предшественник-последователь, предшественник-последователь, двунаправленная топология. В данном разделе рассматривается топология лидер-предшественник-последователь, показанная на рисунке 127.

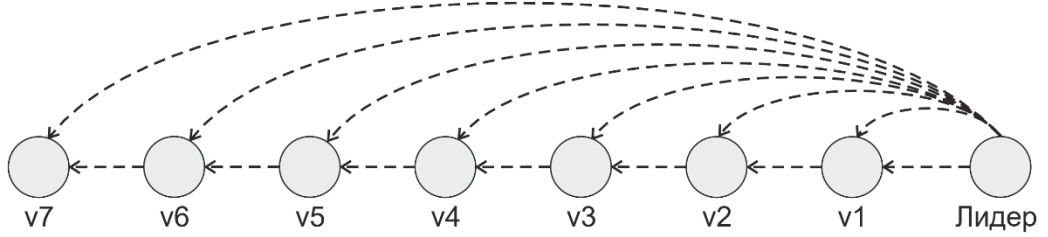


Рисунок 127 – Топология лидер-предшественник-последователь

Стратегия управления движением состоит в том, чтобы поддерживать скорость и требуемое расстояние между транспортными средствами. Динамику i -го транспортного средства можно описать следующим образом [286]:

$$\begin{aligned} \dot{x}_i(t) &= v_i(t), \\ \dot{v}_i(t) &= a_i(t), \\ \dot{a}_i(t) &= \frac{1}{T_i} [u_i(t)] \end{aligned} \quad (93)$$

где $x_i(t)$ – позиция;

$v_i(t)$ – скорость;

$a_i(t)$ – ускорение i -го транспортного средства в момент времени t ;

u_i – управляющее воздействие, выбранное для достижения желаемого ускорения;

T_i – константа трансмиссии.

Проблема консенсуса может быть описана следующим образом:

$$\begin{aligned} x_i(t) &\rightarrow x_0(t) - i \cdot s, \\ v_i(t) &\rightarrow v_0(t), \\ a_i(t) &\rightarrow a_0(t) \end{aligned} \quad (94)$$

где s – желаемое расстояние между двумя соседними транспортными средствами;

$i \cdot s$ – желаемое расстояние между лидером и i -м транспортным средством.

Для решения задачи управления используется модифицированный консенсусный алгоритм третьего порядка в следующем виде:

$$\begin{aligned} u_i(t) &= \frac{1}{\Delta_i} \sum_{j=0, j \notin M}^{N-1} \beta_1 \alpha_{ij} [x_i(t - \tau_j) - x_i(t) - (i - j)s + v_0(t - \tau_0)\tau_j] \\ &\quad + \frac{1}{\Delta_i} \sum_{j=0, j \notin M}^{N-1} \beta_2 \alpha_{ij} [v_j(t - \tau_j) - v_i(t)] + \frac{1}{\Delta_i} \sum_{j=0, j \notin M}^{N-1} \beta_3 \alpha_{ij} a_j(t - \tau_j), \end{aligned} \quad (95)$$

где $\beta_1, \beta_2, \beta_3$ – параметры контроллера.

В частности, вход управления для каждого транспортного средства вычисляется на основе ошибки положения относительно желаемого расстояния между транспортными средствами, ошибки скорости и желаемых значений ускорения. Кроме того, алгоритм консенсуса учитывает временные задержки связи τ_j .

В случае нарушения связи между транспортными средствами полученная информация может быть фальсифицирована. В следующем разделе описывается совместный подход, основанный на алгоритме голосования для обнаружения аномалий в полученных данных. Такой подход позволяет находить вредоносные транспортные средства и обновлять множество M .

4.4.3.2 Алгоритм обнаружения аномалий

Для обнаружения аномалий на каждом временном шаге каждое транспортное средство проверяет полученные данные и сравнивает фактические значения со средними. Алгоритм обнаружения аномалий может быть определен следующим образом (Алгоритм 15).

Алгоритм 15: Алгоритм обнаружения аномалий

Входные данные: $x_j(t), v_j(t), a_j(t), \tau_{ij}(t) \forall j = \overline{0, N-1}$

Выходные данные: Вход управления $u_i(t)$

// Объявление переменных

1: $d_{ij}(t) = [x_i(t - \tau_j) - x_i(t) - (i - j)s + v_0(t - \tau_0)\tau_j]$;

2: $v_{ij}(t) = [v(t - \tau_j) - v_i(t)]$;

3: $a_{ij}(t) = [a_i(t - \tau_j) - a_i(t)]$;

// Расчет средних ошибок

4: $\bar{d}_i = \frac{1}{\Delta_i} \sum_{j=0}^{N-1} \alpha_{ij} d_{ij}(t)$;

5: $\bar{v}_i = \frac{1}{\Delta_i} \sum_{j=0}^{N-1} \alpha_{ij} v_{ij}(t)$;

6: $\bar{a}_i = \frac{1}{\Delta_i} \sum_{j=0}^{N-1} \alpha_{ij} a_{ij}(t)$;

// Проверка данных

7: **for** $j = 0, 1, \dots, N - 1$ **do**

8: $\epsilon = \|\bar{d}_i(t) - d_{ij}(t)\| + \|\bar{v}_i(t) - v_{ij}(t)\| + \|\bar{a}_i(t) - a_{ij}(t)\|$;

9: **if** $\epsilon > \sigma$ **then**

10: $M = M \cup \{i\}$;

11: **end if**

12: **end for**

// Расчет управляющего входа

13: $u_i(t) = \dots$ // согласно (95)

Описанный алгоритм позволяет обнаружить вредоносные транспортные средства и вычислить управляющий вход, используя только доверенные данные.

В следующем разделе проводится экспериментальный анализ алгоритма обнаружения аномалий и оценивается эффективность предложенного подхода против типовых атак.

4.4.3.3 Экспериментальные исследования

Для оценки эффективности описанной стратегии управления рассматриваются сценарии движения транспортных средств в открытой среде моделирования PLEXE [372], расширяющая системы SUMO и Veins и позволяющая реалистично моделировать системы движения транспортных средств в колонне.

В сценарии рассматривается колонна из восьми транспортных средств (лидер и семь ТС-последователей), движущихся по одной полосе. В рассматриваемом сценарии движения лидер, движущийся с постоянной скоростью в 25 м/с, замедляется до 10 м/с в момент времени $t = 60$ с, сохраняет скорость в 10 м/с и разгоняется с 5 м/с^2 в момент времени $t = 120$ с. Моделируются типичные атаки прикладного уровня, описанные в [290]: спуфинг и фальсификация сообщений.

Состояние колонны по умолчанию

На первом этапе экспериментального анализа оцениваются ошибки скорости движения в номинальных условиях движения в колонне без атак (рисунок 128). Ошибки скорости уменьшаются, в то время как транспортные средства сходятся к желаемому межтранспортному расстоянию примерно через 55 секунд. После того как лидер начинает разгоняться/замедляться, погрешности скорости немного увеличиваются и достигают 0,1 м/с.

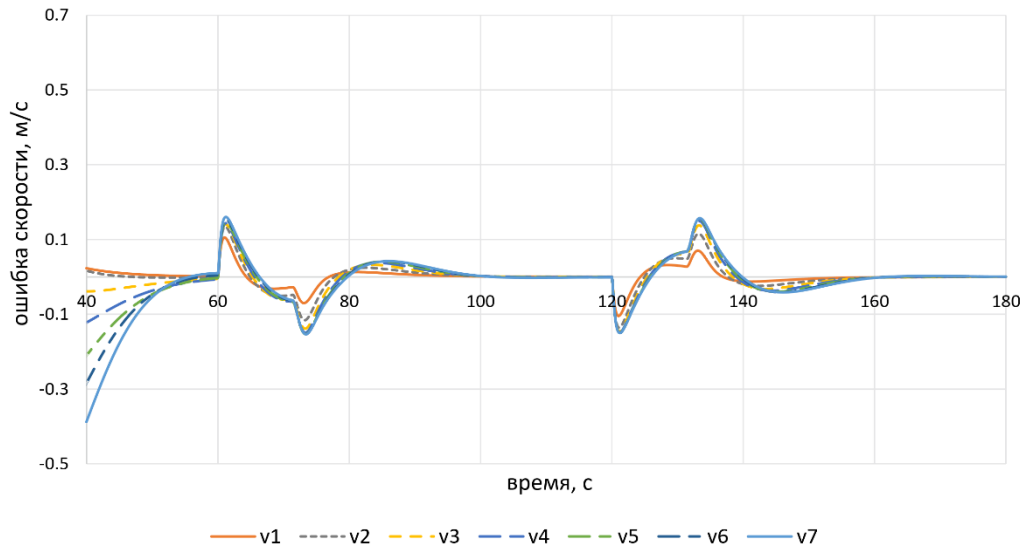


Рисунок 128 – Ошибки скорости при номинальных условиях без атак

Спуфинг

В данной атаке предполагается, что внутренний противник берет на себя управление транспортным средством. Эта атака может нарушить строй или даже вызвать столкновения. Чтобы смоделировать атаку спуфинга, в сценарии предполагается, что третий автомобиль начинает неправильно разгоняться с момента времени $t = 55$ секунд. Ошибки скорости при атаке спуфинга показаны на рисунке 129 в случае, когда алгоритм обнаружения аномалий (Алгоритм 15) неактивен. Результаты с включенной процедурой обнаружения аномалий показаны на рисунке 130.

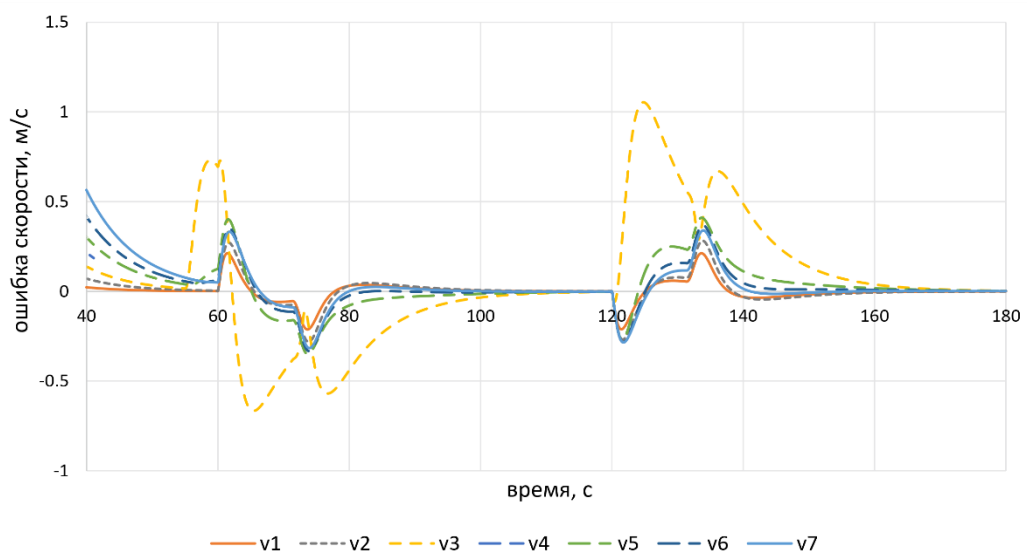


Рисунок 129 – Спуфинг атака без обнаружения аномалий

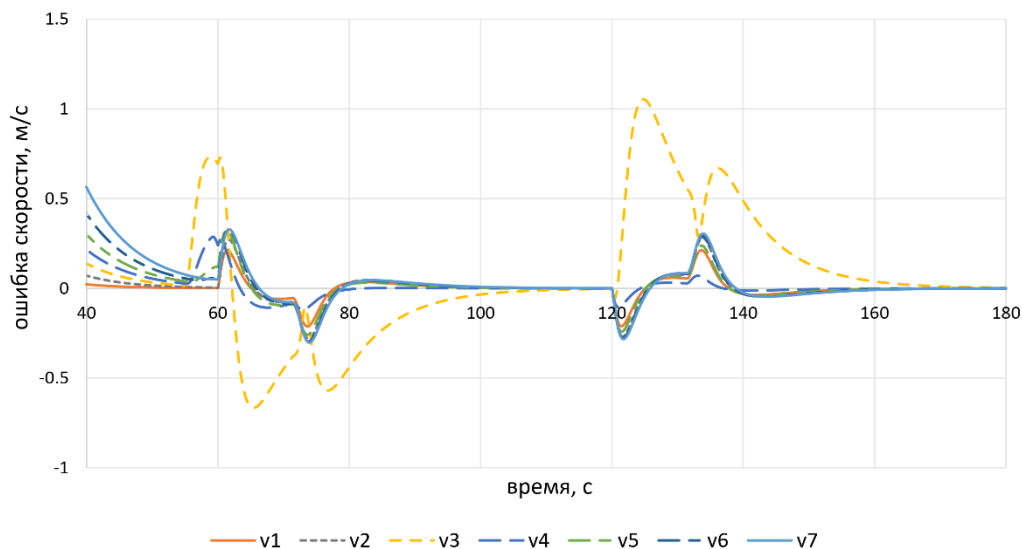


Рисунок 130 – Спуфинг атака с обнаружением аномалий

Можно заметить, что с работающим алгоритмом обнаружения аномалий вредоносное транспортное средство обнаруживается и эффект неправильного ускорения смягчается.

Фальсификация сообщений

При атаке фальсификации сообщения предполагается, что злоумышленник фальсифицирует содержание сообщения, но само транспортное средство ведет себя правильно. Чтобы смоделировать атаку фальсификации сообщения, в сценарии фальсифицируется положение четвертого транспортного средства (добавляется постоянное смещение, равное 5 м), начиная с момента времени $t = 55$ секунд. Ошибки положения без активного алгоритма обнаружения и с включенной процедурой обнаружения аномалий показаны на рисунке 131 и рисунке 132 соответственно.

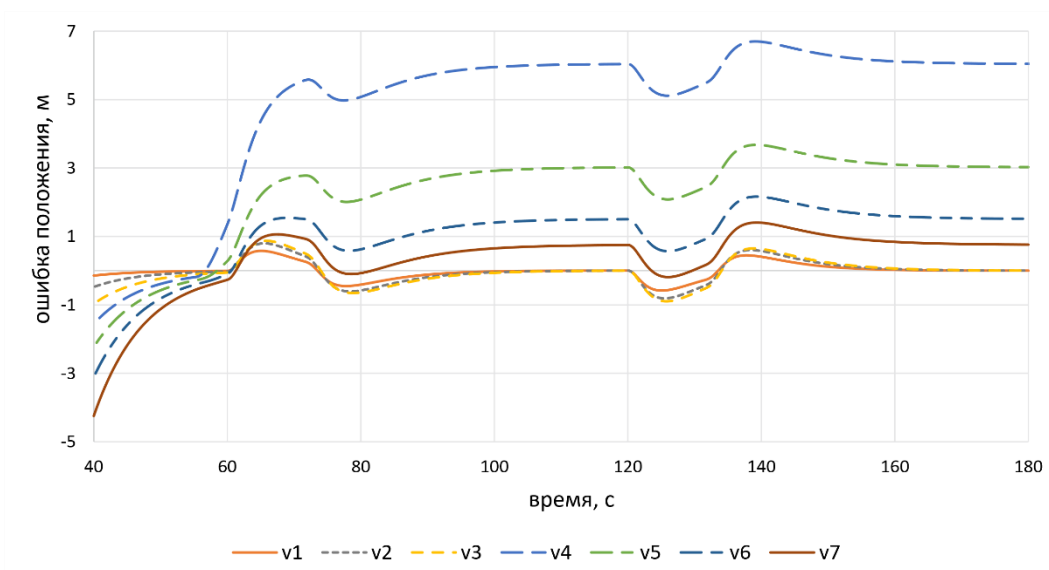


Рисунок 131 – Атака фальсификации сообщений без обнаружения аномалий

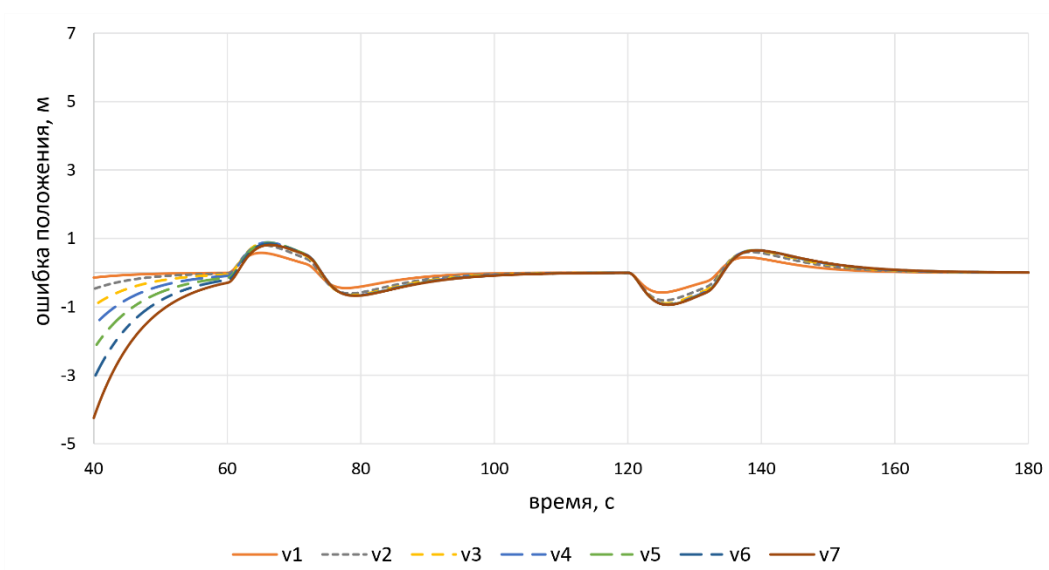


Рисунок 132 – Атака фальсификации сообщений с обнаружением аномалий

Предлагаемый подход позволяет обнаруживать вредоносные транспортные средства и исключать фальсифицированные сообщения при определении управляющего воздействия.

4.5 Выводы и результаты четвертого раздела

В рамках решения задач косвенного управления транспортным потоком и информирования в интеллектуальной транспортной системе получены следующие результаты:

В рамках решения задачи косвенного управления транспортным потоком в интеллектуальной транспортной системе получены следующие результаты:

1) предложен комплекс алгоритмических средств решения задач косвенного управления и информирования в интеллектуальных транспортных системах:

а) алгоритмы краткосрочного прогнозирования времени движения отдельных транспортных средств, учитывающие гетерогенную информацию о транспортной ситуации, прямо или косвенно влияющую на прогнозируемое время движения (пп. 4.1.1, 4.1.2);

б) алгоритм нахождения надежного маршрута движения, максимизирующего вероятность прибытия в пункт назначения в течение заранее определенного интервала времени с использованием актуальной и прогнозной информации о распределении транспортного потока в сети (п. 4.2.2), а также его ускоренная модификация с использованием устойчивых распределений Леви для описания времени прохождения сегментов дорожной сети (п. 4.2.3), позволяющая повысить скорость вычислений в среднем в 40 раз при ухудшении качественных показателей в среднем на 9%;

в) алгоритм маршрутизации транспортных средств в зависящей от времени транспортной сети на основе численного метода резервирования маршрутов (пп. 4.3.1, 4.3.2), а также его модификация для применения в гетерогенном транспортном потоке с учетом адаптивного светофорного регулирования (п. 4.3.3);

2) проведено экспериментальное исследование разработанных алгоритмов в системе микроскопического моделирования движения транспортных средств в различных синтетических и реальных сценариях моделирования;

3) результаты экспериментальных исследований подтвердили высокую эффективность разработанных алгоритмов и их преимущество по сравнению с современными подходами к решению задач косвенного управления и информирования.

По материалам данного раздела опубликованы работы [100, 208, 233, 242, 373-410].

5 Архитектура и реализация программного комплекса кооперативной ИТС

Разработанные математические методы и алгоритмы были реализованы в составе исследовательского программного комплекса кооперативной интеллектуальной транспортной системы, решающей задачи анализа, прогнозирования и управления транспортным потоком с гетерогенным составом транспортных средств с использованием подходов к обработке больших данных.

5.1 Требования к программному комплексу

В качестве основных требований, предъявляемых к разрабатываемому программному комплексу, были выделены следующие:

1) программное обеспечение должно быть построено на основе сервис-ориентированной архитектуры, допускающей адаптивную комбинацию различных методов прогнозирования и использование разнородных данных на основе методов и подходов к обработке больших данных;

2) архитектура и реализация программного обеспечения должна использовать децентрализованный подход к созданию системы и быть оптимизирована для работы на высокопроизводительном вычислительном кластере;

3) результаты работы аналитических сервисов, входящих в состав разрабатываемого программного комплекса, должны быть доступны пользователям посредством разработанного программного интерфейса.

Программный комплекс должен обеспечивать выполнение следующих функций:

- анализ и прогнозирование скорости движения на сегментах дорожной сети с использованием статических и актуальных данных о состоянии транспортных потоков, в т.ч. данных от подключенных транспортных средств;

- адаптивное управление транспортным потоком путем светофорного регулирования с учетом данных от подключенных и автономных подключенных транспортных средств, в т.ч. координированное управление сигналами светофорных объектов и траекториями движения подключенных и/или автономных подключенных транспортных средств;

- расчет времени прибытия общественных транспортных средств на остановочные пункты;

- расчет оптимального пути на индивидуальном транспорте в стохастической сети с учетом информации о прогнозном распределении транспортных потоков.

Для обеспечения указанных требований была разработана и реализована модульная архитектура исследовательского программного комплекса с использованием подходов к обработке больших данных в составе высокопроизводительного горизонтально масштабируемого вычислительного кластера.

5.2 Архитектура программного комплекса с использованием принципов обработки больших данных

5.2.1 Понятие больших данных

Обычно под понятием больших данных (Big Data) обозначают структурированные и неструктурированные данные огромных объёмов и значительного многообразия, эффективно обрабатываемые горизонтально масштабируемыми программными инструментами [411, 412]. Т.е. под термином «большие данные» понимают наборы данных, достаточно большие и сложные для того, чтобы их можно было обработать традиционными средствами работы с данными, например, реляционными СУБД.

Для обработки таких данных была необходима разработка отдельных инструментов, базовым принципом которых является горизонтальная масштабируемость, обеспечивающую обработку данных, распределённых на сотни и тысячи вычислительных узлов, без деградации производительности.

В 2001 году было предложено оценивать большие данные с использованием трех характеристик (правило 3V) [413]:

- 1) volume – физический объем данных;
- 2) velocity – скорость прироста данных и скорость быстрой обработки данных с целью получения результатов;
- 3) variety – вариативность, предполагает возможность одновременной обработки различных типов данных; Данные могут быть структурированными, неструктурированными или структурированными частично.

В дальнейшем появились интерпретации с «четырьмя V» (добавлялась veracity – достоверность как самого набора данных, так и результатов его анализа). Далее набор V еще расширился, добавлялись такие характеристики, как variability – изменчивость форматов, структуры или источников больших данных; value – ценность информации, которую можно получить путем обработки и анализа больших наборов данных, и т.д.

Во всех случаях в этих признаках подчёркивается, что определяющей характеристикой для больших данных является не только их физический объём, но другие

категории, существенные для представления о сложности задачи обработки и анализа данных.

Таким образом, цель использования технологий обработки больших данных заключается в следующем:

- 1) обеспечение хранения и управления большими объемами данных (в настоящее время существует необходимости эффективной обработки петабайт данных);
- 2) организация неструктурированных и слабоструктурированных данных, таких как тексты, изображения, видео и т.д.;
- 3) получение аналитических выводов, которые будут полезны для решения практических задач.

Можно выделить следующие базовые принципы работы с большими данными:

- 1) горизонтальная масштабируемость, обеспечивающая обработку данных таким образом, что их распределение по узлам вычислительного кластера не приводит к снижению производительности системы;
- 2) отказоустойчивость, которая минимизирует влияние возможных сбоев оборудования на работу с данными;
- 3) локальность данных, которая предполагает, что данные будут обработаны на том же вычислительном узле, на котором они хранятся, чтобы исключить сократить временные затраты на передачу данных от места хранения к месту обработки.

5.2.2 Инструментарий для обработки больших данных

В настоящее время существует большое количество разных инструментариев для обработки больших данных. На рисунке 133 представлена классификация инструментов по типу решаемых ими задач [411].

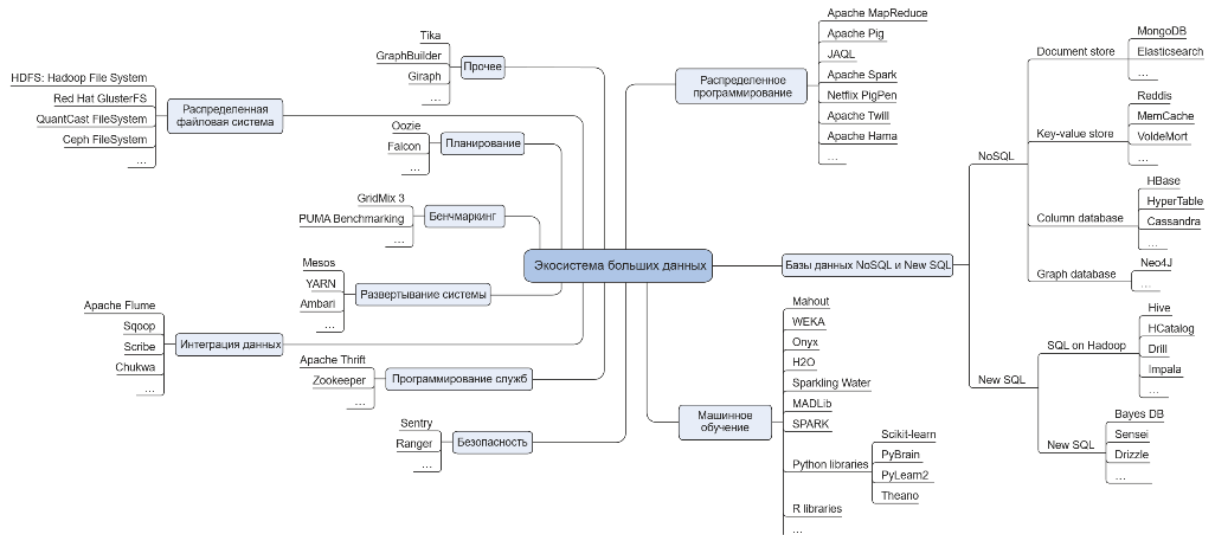


Рисунок 133 – Программное обеспечение для работы с большими данными

Распределенная файловая система предполагает хранение данных на нескольких узлах кластера. Распределенные файловые системы обладают рядом преимуществ:

- 1) способны хранить файлы, размер которых превышает размер диска отдельного узла системы, что позволяет разделить большие файлы на части и распределить по разным узлам кластера;
- 2) файлы в распределенной файловой системе автоматически реплицируются на нескольких серверах, что обеспечивает избыточность данных и возможность выполнения параллельных операций, а также доступность данных при выходе из строя узла кластера;
- 3) система легко масштабируется путем добавления новых узлов.

В настоящее время одной из наиболее популярных распределенных файловых систем является HDFS – это Hadoop File System. Но существуют и различные альтернативы.

Для использования данных, сохраненных в распределенной файловой системе, используются инструменты для распределенного программирования, которые, условно, перемещают код программы к данным, чтобы обеспечить принцип локальной обработки данных. Для обработки данных широко используются инструменты MapReduce и Apache Spark.

Для добавления и перемещения данных из одних источников в другой используются инструменты интеграции данных, такие как Apache Flume.

Для анализа данных с использованием методов машинного обучения используются как библиотеки языков Python или R, так и программное обеспечение типа Mahout или Apache Spark.

Для хранения огромных объемов данных предпочтительными являются NoSQL и NewSQL решения. Типы NoSQL баз данных – это документоориентированные базы данных типа MongoDB, системы хранения типа ключ-значение, такие как Redis, столбцовые базы данных, например, HBase, и графовые базы данных, например, Neo4j.

Кроме того, реализованы инструменты планирования для автоматизации повторяющихся операций и запуска заданий по событиям, инструменты развертывания системы, программирования служб и обеспечения безопасности данных.

5.2.3 Архитектура систем обработки больших данных

5.2.3.1 Компоненты архитектуры

На рисунке 134 показаны основные компоненты, входящие в архитектуру системы, осуществляющей обработку больших данных. Отдельные решения могут не содержать все компоненты в этой схеме.

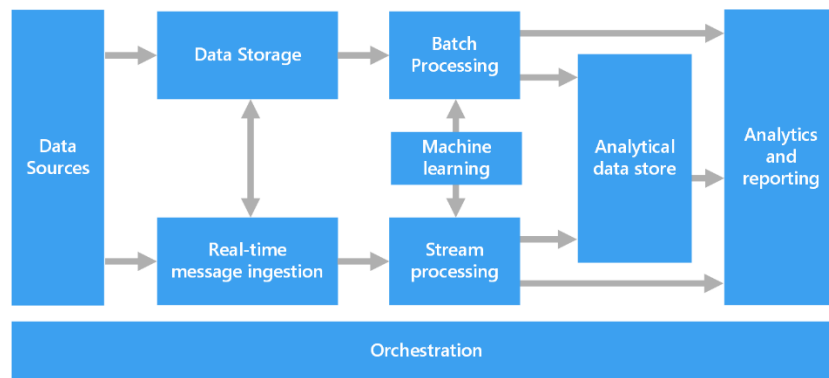


Рисунок 134 – Компоненты архитектуры больших данных

Архитектура содержит следующие компоненты:

1) источники данных (data sources). Все решения для обработки больших данных начинаются с одного или нескольких источников данных;

2) хранилище данных (data storage) – это распределенное файловое хранилище, предназначенное для хранения данных больших объемов в различных форматах для пакетной обработки. Такое хранилище часто называется озером данных;

3) пакетная обработка (batch processing) – процесс обработка данных, требующий больших временных ресурсов (фильтрация, статистическая обработка и другие задачи обработки). Для пакетной обработки часто применяется подход MapReduce;

4) прием сообщений в реальном времени (real-time message ingestion) – процесс сбора и сохранения поступающих в реальном времени данных для потоковой обработки (в режиме реального времени);

5) потоковая обработка (stream processing) – процесс обработки данных в режиме реального времени, включающий фильтрацию, статистическую обработку и другие задачи анализа поступающих сообщений;

6) хранилище аналитических данных (analytical data store) – используется для хранения обработанных структурированных данных в соответствии с форматом запросов для средств аналитики;

7) анализ и создание отчетов (analytics and reporting) – процесс предоставления обработанной информации в виде аналитических отчетов;

8) оркестрация (orchestration) – процесс, отвечающий за координацию и управление выполнением различных операций в системе.

5.2.3.2 Пакетная обработка

Обработка больших данных может выполняться либо в пакетном, либо в потоковом режиме. Пакетная обработка – это типичный сценарий работы с большими данными. Процесс пакетной обработки обычно включает несколько шагов. Сначала данные из источников загружаются в хранилище данных. Затем данные подвергаются обработке, которая может включать фильтрацию, трансформацию, агрегацию и другие операции. Обработка данных выполняется параллельно на нескольких вычислительных узлах для эффективной обработки данных. Важно отметить, что пакетная обработка может включать несколько итераций до того, как преобразованные результаты будут загружены в хранилище аналитических данных.

Пакетная обработка может выполняться для очень больших наборов данных и сложных сценариев обработки, которые требуют длительного времени выполнения. Пакетная обработка может выполняться достаточно эффективно если говорить о количестве обрабатываемых записей в единицу времени благодаря современным технологиям формату хранения, способу обработки и т.д. Но результаты обработки доставляются с задержкой.

На рисунке 135 представлена схематичная схема обработки: данные объединяются в блоки или пакеты и затем происходит обработка всего пакета данных.

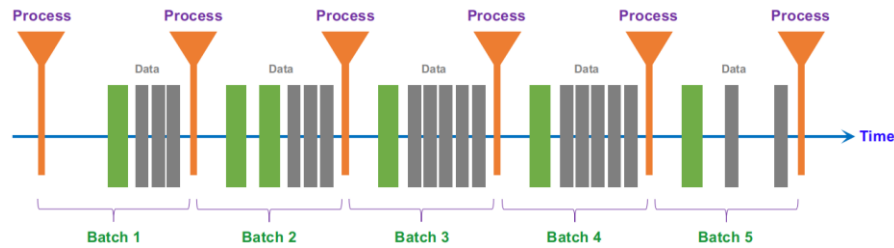


Рисунок 135 – Пакетная обработка данных

Архитектура системы пакетной обработки имеет следующие логические компоненты [414], показанные на рисунке 136.

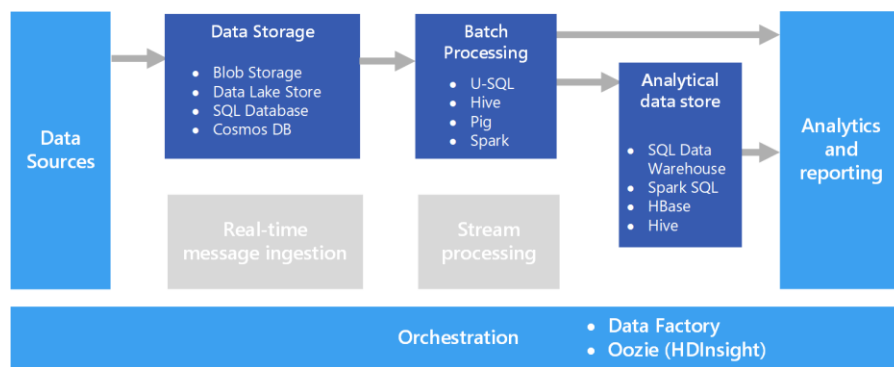


Рисунок 136 – Архитектура системы пакетной обработки данных

В качестве хранилища данных (data storage) обычно выступает распределенное хранилище файлов, которое может служить репозиторием для значительных объемов больших файлов в различных форматах. Также можно использовать различные базы данных. Для пакетной обработки данных (batch processing) может использоваться платформа Spark, которая поддерживает программы пакетной обработки, написанные на разных языках, включая Java, Scala и Python. Spark использует распределенную архитектуру для параллельной обработки данных в нескольких рабочих узлах. Также может использоваться SQL-подобный язык Hive, декларативный язык Pig. В качестве хранилища аналитических данных (analytics data store) могут использоваться NoSQL хранилище типа «семейство столбцов» HBase, система управления базами данных Apache Hive на основе платформы Hadoop и другие.

5.2.3.3 Поточковая обработка

Поточковая обработка выполняется для потоков данных, получаемых в реальном времени и обрабатываемых с минимальной задержкой для создания отчетов или автоматизированного реагирования в режиме реального времени (или приближенном к

реальному времени). Т.е. каждая поступившая порция данных обрабатывается сразу, как показано на рисунке 137.

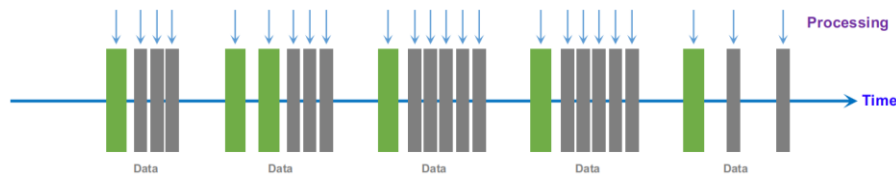


Рисунок 137 – Поточковая обработка данных

Поточковая обработка позволяет обеспечить непрерывную обработку входящего потока данных по мере поступления событий и выдавать результат обработки с минимальным временем задержки в несколько миллисекунд. Но такая обработка не сможет использоваться в сложных сценариях обработки данных или использовать данные из хранилища, т.е. результаты обработки могут быть менее точны или полны, чем при пакетной обработке.

Архитектура обработки в режиме реального времени состоит из следующих логических компонентов: средства сбора и сохранения сообщений в режиме реального времени, средства потоковой обработки, хранилище аналитических данных и средства создания и отчетов (рисунок 138) [415].

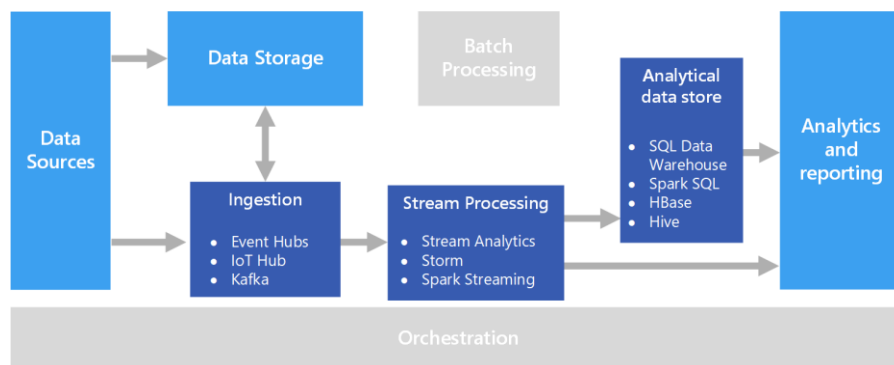


Рисунок 138 – Архитектура системы потоковой обработки данных

Для приема сообщений в реальном времени используются брокеры сообщений, такие, как Apache Kafka или RabbitMQ, которые поддерживают очереди сообщений, потоковую обработку с возможностью масштабирования до нескольких миллионов сообщений в секунду от множества отправителей сообщений, и перенаправление их множеству объектов-получателей. Для потоковой обработки данных могут использоваться системы Apache Storm или Spark Streaming.

5.2.3.4 Лямбда-архитектура

Часто необходимо совместить оба подхода к обработке данных, т.е. объединить потоковую обработку данных в режиме реального времени с результатами пакетной аналитики. Для этого была предложена лямбда-архитектура, которая является одним из наиболее популярных архитектурных решений для обработки больших данных.

Лямбда-архитектура [416] состоит из трёх взаимодополняющих уровней: уровень пакетной обработки, выполняющий обработку пакетов данных по расписанию, уровень скоростной обработки, выполняющий потоковую обработку данных в реальном времени, и уровень обслуживания, которые объединяет результаты пакетного и потокового представления для предоставления агрегированных данных пользователю по запросу (рисунок 139).

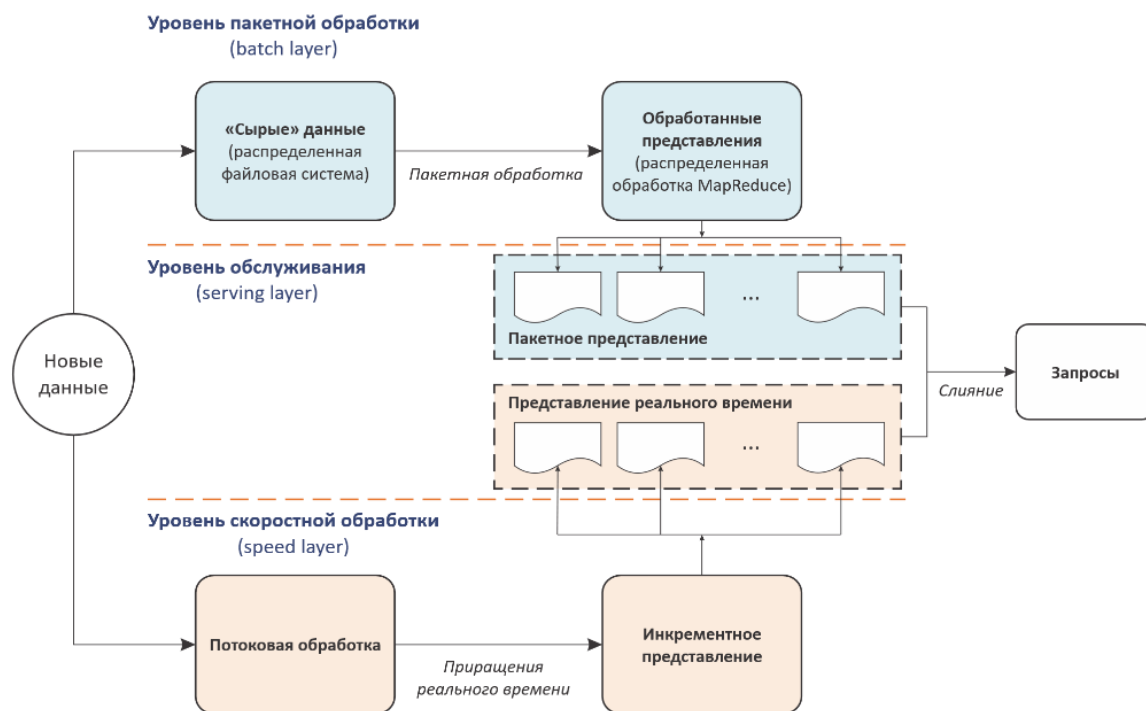


Рисунок 139 – Лямбда-архитектура

Уровень пакетной обработки представляет собой хранилище «сырых» данных. На этом уровне выполняется обработка по расписанию – через заранее заданные интервалы времени отправляются запросы к новым данным. Полученные данные добавляются к накопленному ранее архиву, не изменяя его предыдущие копии. Обычно уровень пакетной обработки реализуется на базе Apache Hadoop.

Уровень потоковой обработки осуществляет обработку поступающего в реальном времени потока данных. Работа в реальном времени допускает упрощенную обработку

данных и использование простых сценариев работы для быстрого анализа данных. В этом случае, можно получить, например, недостаточно точные или неполные данные. Однако эти погрешности с небольшим запозданием компенсируются уровнем пакетной обработки. Уровень скоростной обработки может быть выполнен с использованием инструментов: Apache Spark, Apache Storm и других.

Данные, полученные от уровней пакетной и потоковой обработки, сохраняются на уровне обслуживания. На этом слое выполняется обработка запросов от операторов и возвращение им заранее подготовленных или подготовленных «на лету» представлений. На стыке сервисного уровня и уровня скоростной обработки может работать как уже рассмотренная ранее документоориентированная СУБД MongoDB, так и ряд других систем, таких как Apache Cassandra или Apache HBase.

5.2.4 Программное обеспечение для работы с большими данными

5.2.4.1 Apache Hadoop

Для обеспечения возможности хранения больших данных в распределенных системах и эффективного управления всем кластером данных требовалось разработать соответствующее программное обеспечение. В 2005 году в Apache Software Foundation был разработан проект Apache Hadoop, который изначально был задуман как система хранения данных, способная выполнять задачи MapReduce. С тех пор Hadoop превратился в полноценный стек компьютерных технологий, способных обрабатывать данные на уровне пакетной обработки в лямбда-архитектуре.

Основными преимуществами стека технологий Hadoop являются следующие:

- 1) надежность, которая достигается путем поддержания избыточности данных и повторного применения задач обработки данных в случае сбоя;
- 2) отказоустойчивость, обеспечиваемая за счет обнаружения сбоев и автоматическим восстановлением;
- 3) горизонтальная масштабируемость, позволяющая распределять данные и проводить их обработку на отдельных узлах вычислительного кластера;
- 4) портируемость, т.е. возможность установки на различных устройствах и операционных системах.

Базовая инфраструктура Hadoop состоит из распределенной файловой системы (Hadoop Distributed File System, HDFS), менеджера ресурсов и системы выполнения распределенных программ. На практике такой подход позволяет работать с распределенной

файловой системой, используя те же подходы, что и при работе с локальной файловой системой, не заботясь о реальное расположение данных на разных серверах.

В Apache Hadoop центральное место занимают три компонента:

- 1) распределенная файловая система (Hadoop Distributed File System, HDFS);
- 2) метод распределенных вычислений MapReduce;
- 3) система управления ресурсами кластера YARN, которая осуществляет управление ресурсами и задачами кластера. Для этого создаются контейнеры для приложений, выполняется оценка их потребностей в ресурсах и выделение дополнительных ресурсов при необходимости;

На этой базе сформировалась экосистема приложений, показанная на рисунке 140. В экосистему входят, например, базы данных Hive и HBase или инфраструктура машинного обучения Mahout. Конечно, в рамках этой инфраструктуры при необходимости могут использоваться и другие компоненты, упомянутые при изучении архитектуры систем обработки больших данных, в т.ч. СУБД MongoDB или Cassandra.

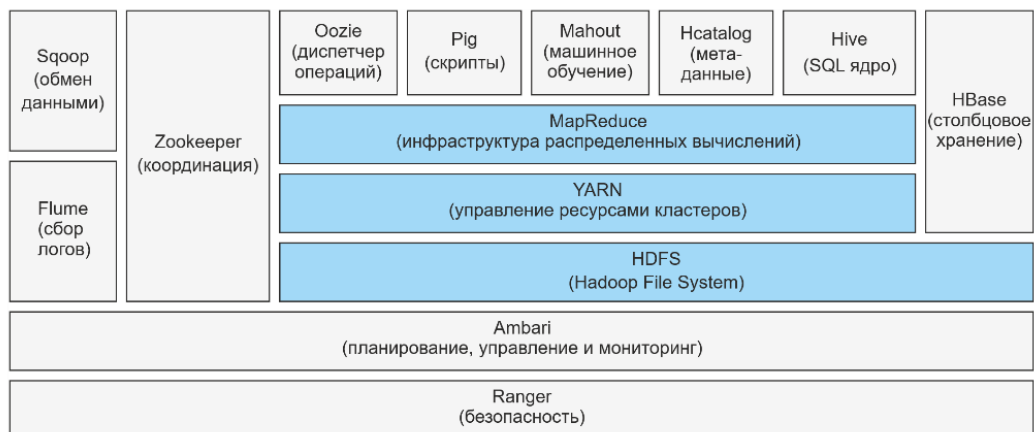


Рисунок 140 – Основные компоненты Apache Hadoop

Для реализации параллелизма Hadoop использует технологию MapReduce. Суть парадигмы MapReduce заключается в том, что большая задача разделяется на ряд небольших заданий, каждое из которых может быть выполнено на любом из узлов кластера.

Однако алгоритм MapReduce плохо подходит для интерактивного анализа, потому что данные записываются на диск между этапа вычислений. При работе с большими наборами данных запись обходится достаточно дорого.

5.2.4.2 Apache Spark

Чтобы повысить производительность обработки, был разработан фреймворк Apache Spark [417].

Apache Spark – фреймворк с открытым исходным кодом, предназначенный для выполнения распределённой обработки неструктурированных и слабоструктурированных данных. Apache Spark является частью экосистемы проектов Hadoop.

В отличие от классического обработчика из ядра Hadoop, реализующего двухуровневую концепцию MapReduce с хранением промежуточных данных на базе дискового хранилища, Spark обрабатывает данные в оперативной памяти, благодаря чему позволяет получать значительный выигрыш в скорости работы для некоторых классов задач, в частности, возможность многократного доступа к загруженным в память пользовательским данным делает библиотеку привлекательной для алгоритмов машинного обучения.

Apache Spark предоставляет программные интерфейсы для языков Java, Scala, Python и R. Следует отметить, что Apache Spark не решает задачи хранения файлов в распределённой файловой системе или задач управлением ресурсами. Для этих целей используются системы управления ресурсами, такие как YARN или Mesos. Apache Spark поддерживает несколько распределённых систем хранения данных, включая HDFS, OpenStack Swift, NoSQL-СУБД Cassandra и Amazon S3. Таким образом, Hadoop и Spark являются взаимодополняющими системами, где Hadoop отвечает за хранение данных и управление ресурсами, а Spark предоставляет возможности для эффективной обработки данных.

Apache Spark состоит из ядра и нескольких расширений. Spark SQL позволяет выполнять SQL-запросы над данными, Spark Streaming предоставляет возможности для обработки потоковых данных, Spark MLlib представляет набор библиотек машинного обучения, а GraphX предназначен для распределённой обработки графов. Эти расширения дополняют основные возможности Spark и расширяют его функциональность для различных сценариев обработки данных.

Стоит отметить, что Spark Streaming, в отличие от, например, Apache Storm, Flink или Samza, не обрабатывает потоки данных целиком. Вместо этого реализуется микропакетный подход (micro-batch), когда поток данных разбивается на небольшие пакеты временных интервалов.

Инфраструктура Spark, используемая в сочетании с инфраструктурой Hadoop, показана на рисунке 141.

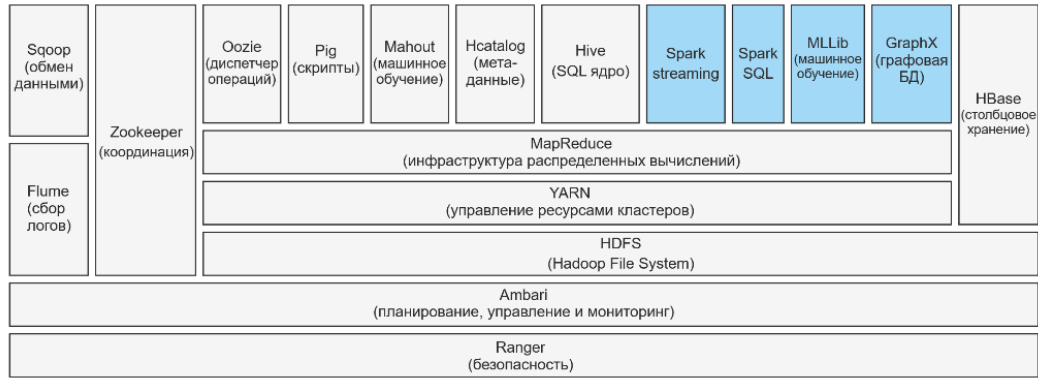


Рисунок 141 – Основные компоненты Apache Spark

5.2.4.3 Apache Kafka

Apache Kafka – это распределенная система обмена сообщениями с высокой пропускной способностью между компонентами программной системы, работающая по принципу «публикация–подписка» [418].

Можно выделить следующие особенности системы:

- 1) является распределенной, горизонтально-масштабируемой системой, обеспечивающей наращивание пропускной способности как при росте числа и нагрузки со стороны источников, так и количества систем-подписчиков;
- 2) обеспечивается публикация и подписка на потоки записей;
- 3) поддерживается отказоустойчивый способ хранения потоков записей за счет применения техники, сходной с журналами транзакций;
- 4) поток записей обрабатывается по мере появления записей с возможностью временного хранения данных для последующей пакетной обработки.

Использование брокера сообщений упрощает работу, когда в системе несколько источников данных или систем-подписчиков. В этом случае источник данных отправляет сообщений в брокер, которые передаются всем подписчикам, иначе бы источнику пришлось хранить информацию о всех подписчиках и отправлять им сообщения напрямую, решая все те задачи, которые решает брокер, например, надежность доставки сообщений.

Использование инструментов для обработки больших данных позволяет построить горизонтально-масштабируемую систему, способную обрабатывать большие объемы данных как в пакетном, так и в потоковом режиме.

5.2.5 Архитектура и реализация программного комплекса кооперативной интеллектуальной транспортной системы

Для реализации разработанного программного комплекса была выбрана лямбда-архитектура. В качестве источника данных используется система моделирования движения транспортных средств SUMO. Данные о параметрах движения поступают в брокер сообщений Apache Kafka. Далее, на уровне скоростной обработки используется инструмент Spark Streaming для чтения сообщений из Kafka, сохранения в распределенную систему HDFS, являющуюся частью экосистемы Hadoop, и в БД Cassandra.

Hadoop используется для распределенного хранения файлов и их пакетной обработки. БД Cassandra – это распределенная NoSQL база данных типа «семейство столбцов». В составе программного комплекса будет использоваться для хранения большого массива разнородных данных.

Для пакетной обработки используется инструмент Apache Spark, который будет выполнять задачи по усреднению данных, обучению моделей прогнозирования параметров транспортного потока и скорости движения отдельных (общественных и подключенных) транспортных средств. Apache Spark использует файловое хранилище HDFS для чтения исходных данных и сохраняет результат аналитической обработки в БД PostgreSQL. БД PostgreSQL была выбран для упрощенной обработки геопространственных данных, кроме того, эта СУБД позволяет отдавать данные в MVT формате, что позволяет легко отобразить их на карте.

На уровне обслуживания было разработано веб-приложение на языке Java с использованием библиотеки Spring Boot. Приложение реализует программно-прикладной интерфейс (API), принимает запросы пользователей и возвращает результат аналитической обработки.

Используемые технологии построения программного комплекса представлены на рисунке 142.

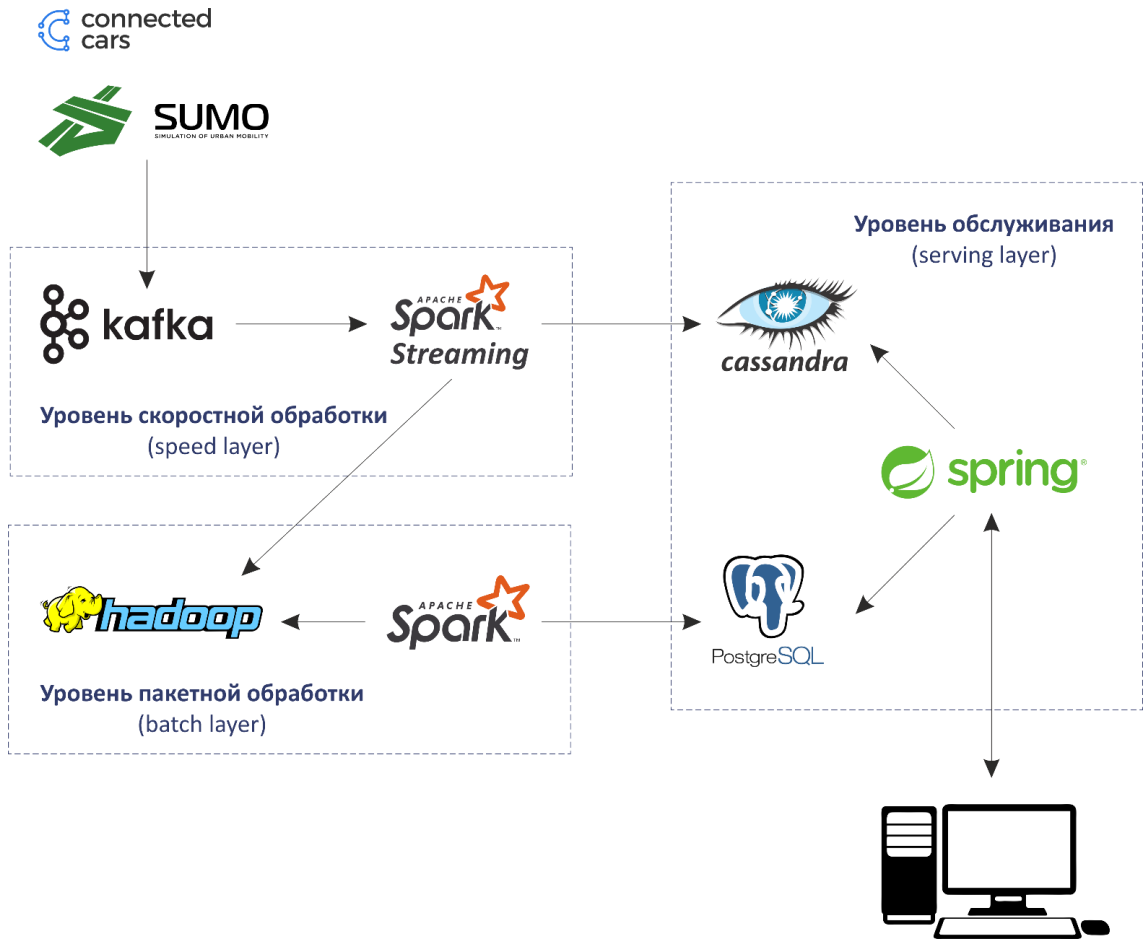


Рисунок 142 – Технологии программного комплекса

После развертывания компонентов системы можно подключиться к Spark, Hadoop или БД. При подключении к мастер-ноде Spark можно увидеть список рабочих процессов (рисунок 143). При подключении к главной ноде Hadoop можно увидеть список хранилищ данных (рисунок 144).

Spark Master at spark://spark-master:7077

URL: spark://spark-master:7077
 Alive Workers: 2
 Cores in use: 24 Total, 0 Used
 Memory in use: 48.0 GiB Total, 0.0 B Used
 Resources in use:
 Applications: 0 Running, 0 Completed
 Drivers: 0 Running, 0 Completed
 Status: ALIVE

Workers (2)

Worker Id	Address	State	Cores	Memory	Resources
worker-20220821064539-172.23.0.10-41071	172.23.0.10:41071	ALIVE	12 (0 Used)	24.0 GiB (0.0 B Used)	
worker-20220821064539-172.23.0.11-41013	172.23.0.11:41013	ALIVE	12 (0 Used)	24.0 GiB (0.0 B Used)	

Running Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
----------------	------	-------	---------------------	------------------------	----------------	------	-------	----------

Completed Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
----------------	------	-------	---------------------	------------------------	----------------	------	-------	----------

Рисунок 143 – Мастер-нода Spark

Hadoop Overview Datanodes Datanode Volume Failures Snapshot Startup Progress Utilities

Datanode Information

✔ In service ❌ Down 🔄 Decommissioning 🔴 Decommissioned 🔴 Decommissioned & dead
🔧 Entering Maintenance 🔴 In Maintenance 🔴 In Maintenance & dead

Datanode usage histogram

In operation

Show 25 entries Search:

Node	Http Address	Last contact	Last Block Report	Capacity	Blocks	Block pool used	Version
✔ datanode-1.9866 (172.23.0.8.9866)	http://datanode-1.9864	2s	0m	2.73 TB	0	0 B (0%)	3.1.3
✔ datanode-2.9866 (172.23.0.9.9866)	http://datanode-2.9864	1s	0m	2.73 TB	0	0 B (0%)	3.1.3

Рисунок 144 – Нода Hadoop

Источником данных является система моделирования SUMO. На рисунке 145 показан скриншот системы, желтым показаны транспортные средства.

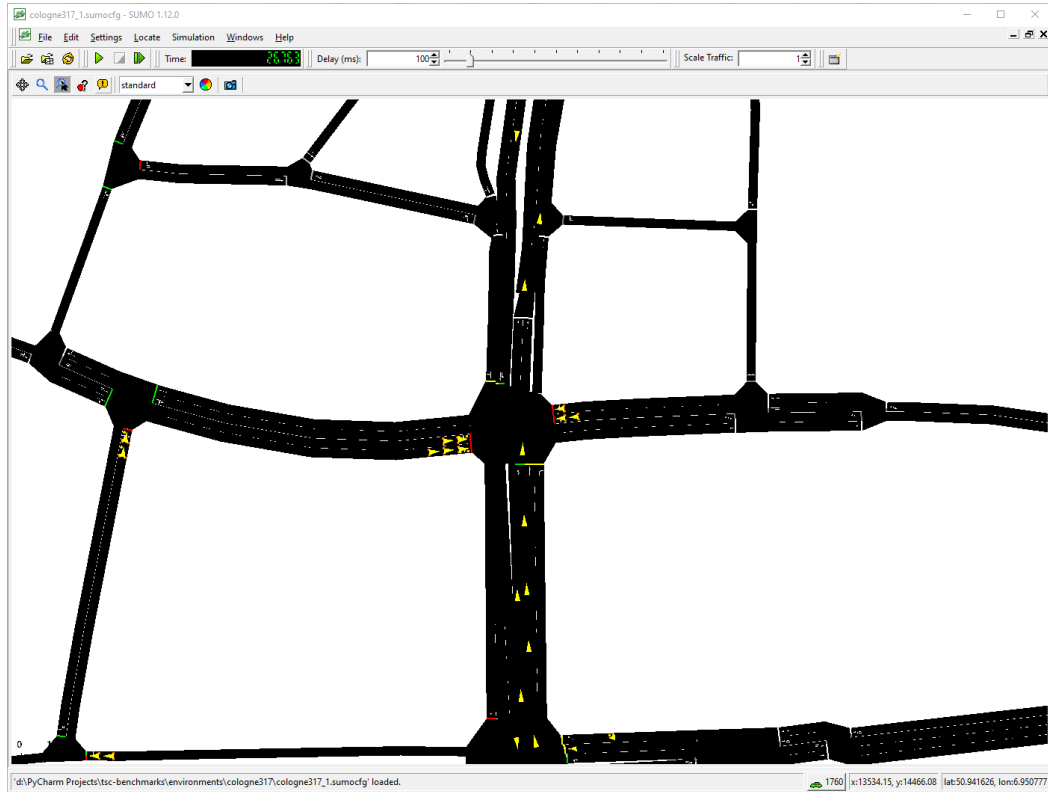


Рисунок 145 – Система моделирования SUMO

После развертывания системы была запущена передача сообщений из SUMO в Kafka. Далее были созданы несколько задач для потоковой обработки данных, пакетного усреднения скорости, пакетного прогнозирования скорости и обучения модели.

После отправки задач Apache Spark, можно увидеть статус их выполнения в панели управления Spark. В данном случае, задача обучения модели прогнозирования параметров транспортных потоков (ml-training) завершилась, задачи потоковой и пакетной обработки данных продолжают свое выполнение (рисунок 146).

Spark Master at spark://spark-master:7077

URL: spark://spark-master:7077
 Alive Workers: 2
 Cores in use: 24 Total, 18 Used
 Memory in use: 48.0 GiB Total, 6.0 GiB Used
 Resources in use:
 Applications: 3 Running, 1 Completed
 Drivers: 0 Running, 0 Completed
 Status: ALIVE

Workers (2)

Worker Id	Address	State	Cores	Memory	Resources
worker-20220821113711-172.23.0.10-33007	172.23.0.10:33007	ALIVE	12 (9 Used)	24.0 GiB (3.0 GiB Used)	
worker-20220821113712-172.23.0.11-36159	172.23.0.11:36159	ALIVE	12 (9 Used)	24.0 GiB (3.0 GiB Used)	

Running Applications (3)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
app-20220821113859-0003	(kill) speed-prediction	6	1024.0 MiB		2022/08/21 11:38:59	root	RUNNING	13 s
app-20220821113851-0002	(kill) batch-processor	6	1024.0 MiB		2022/08/21 11:38:51	root	RUNNING	21 s
app-20220821113732-0000	(kill) streaming-processor	6	1024.0 MiB		2022/08/21 11:37:32	root	RUNNING	1.7 min

Completed Applications (1)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
app-20220821113819-0001	ml-training	6	1024.0 MiB		2022/08/21 11:38:19	root	FINISHED	19 s

Рисунок 146 – Мастер-нода Spark после отправки задач

Для отображения результатов обработки было создано веб-приложение, которое отображает на карте исходные положения транспортных средств как точечные объекты (рисунок 147), либо результат анализа как векторные слои в формате MVT, каждый сегмент сети раскрашен в зависимости от скорости движения по сегменту (рисунок 148).

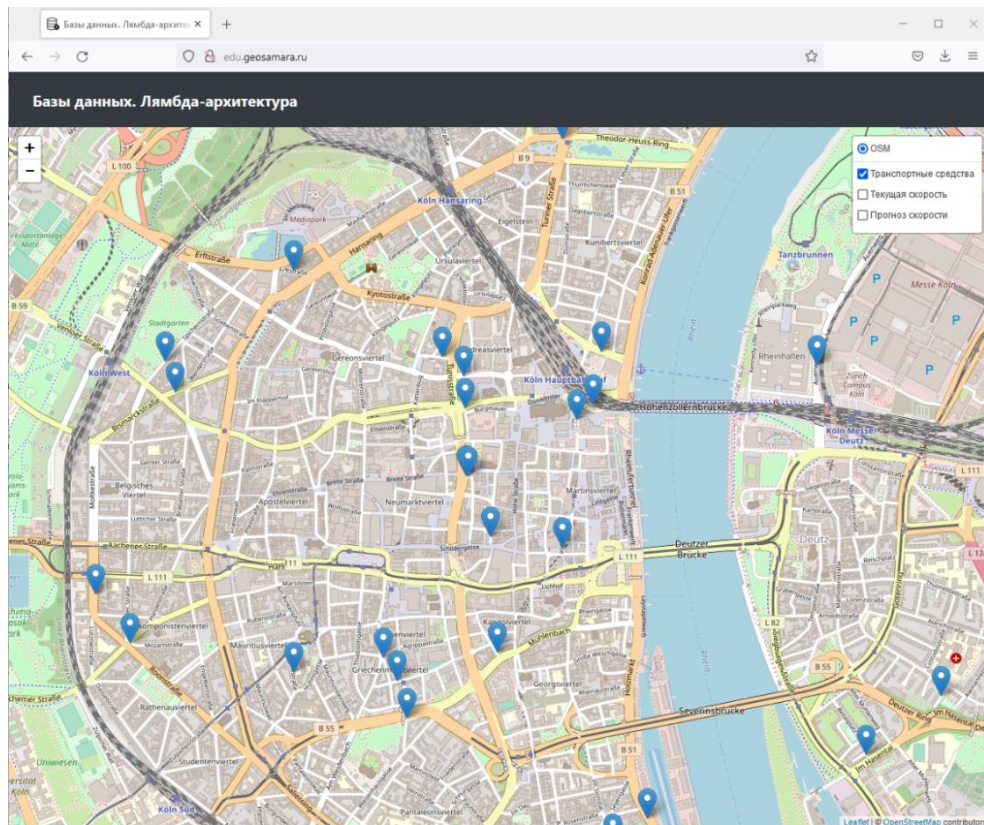


Рисунок 147 – Карта ТС в виде точечных объектов

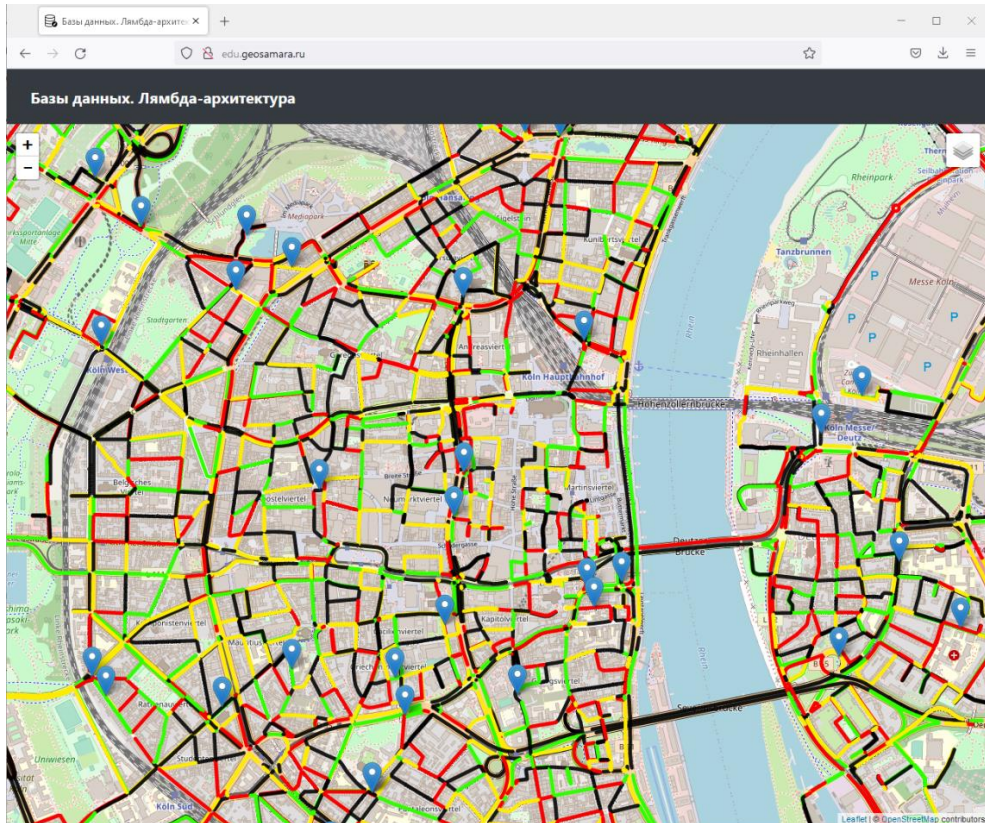


Рисунок 148 – Карта скоростей движения

Все компоненты предлагаемой системы поддерживают горизонтальное масштабирование, т.е. допускается добавление в кластер дополнительных вычислительных узлов в случае необходимости обработки сценариев с большим трафиком. При этом изменения в коде приложения не потребуются.

В следующих разделах подробнее рассмотрены программные модули, разработанные в рамках программного комплекса и решающие следующие задачи:

- анализ и прогнозирование скорости движения на сегментах дорожной сети с использованием статических и актуальных данных о состоянии транспортных потоков, в т.ч. данных от подключенных транспортных средств (раздел 5.3);

- адаптивное управление транспортным потоком путем светофорного регулирования с учетом данных от подключенных и автономных подключенных транспортных средств, в т.ч. координированное управление сигналами светофорных объектов и траекториями движения подключенных и/или автономных подключенных транспортных средств (раздел 5.4);

- расчет времени прибытия общественных транспортных средств на остановочные пункты и расчет оптимального маршрута движения на общественном транспорте с использованием информации о времени прибытия транспортных средств (раздел 5.5);

- расчет оптимального пути на индивидуальном транспорте в стохастической сети с учетом информации о прогнозном распределении транспортных потоков (раздел 5.6).

5.3 Программный модуль краткосрочного прогнозирования параметров транспортного потока

5.3.1 Назначение программного модуля

Программный модуль, реализующий математические методы и алгоритмы краткосрочного прогнозирования параметров транспортных потоков на основе технологии «больших данных» с использованием данных реального времени и архивных данных о состоянии транспортных потоков и методов машинного обучения (далее – ПМ КППП) предназначен для реализации методов краткосрочного прогнозирования параметров транспортных потоков и предоставления участнику дорожного движения результатов прогнозирования посредством Web-сервиса.

ПМ КППП предоставляет следующие основные функции и средства:

- анализ и краткосрочное прогнозирования параметров транспортных потоков на основе технологии «больших данных», методов машинного обучения с использованием данных реального времени и архивных данных о состоянии транспортных потоков;
- предоставления результатов прогнозирования участникам дорожного движения посредством Web-сервиса по протоколу WMS для выдачи географически привязанных изображений через Интернет и протоколу MVT для выдачи векторных объектов;
- возможность обновления статистики по архивным данным о состоянии транспортных потоков;
- возможность пересчета параметров используемой модели краткосрочного прогнозирования параметров транспортных потоков на основе технологии «больших данных» с использованием данных реального времени и архивных данных о состоянии транспортных потоков.

5.3.2 Описание логической структуры

Все модули, входящие в состав ПМ КППП, являются независимыми в том плане, что не требуется их выполнения в рамках определенной заранее последовательности, соответствующей некоторой логической структуре программы. Каждый программный модуль выполняет отдельную самостоятельную задачу.

Модуль обработки информации о транспортных потоках является основным модулем в составе ПМ КППП. Модуль реализует следующие функции:

- обновление информации о дорожной сети из файла с графом, формируемым в геоинформационной системе;

- привязка информации о параметрах транспортных потоков к сегментам транспортной сети, архивация данных;

- вызов модуля краткосрочного прогнозирования параметров транспортных потоков на основе технологии «больших данных» с использованием данных реального времени и архивных данных о состоянии транспортных потоков и сохранение прогнозной информации;

- предоставление результатов прогнозирования участникам дорожного движения посредством Web-сервиса по протоколу WMS, предназначенного для выдачи географически привязанных изображений через Интернет, и протоколу MVT для выдачи векторных объектов.

Модуль прогнозирования предназначен для краткосрочного прогнозирования параметров транспортных потоков на основе технологии «больших данных» с использованием данных реального времени и архивных данных о состоянии транспортных потоков и сохранение прогнозной информации. Реализация методов краткосрочного прогнозирования вынесена в отдельный модуль для уменьшения связности между модулями и возможности замены модели прогнозирования без необходимости изменения реализации других модулей. В рамках ПМ КППП используется алгоритм краткосрочного прогнозирования параметров транспортных потоков на основе метода непараметрической регрессии k ближайших соседей с применением модели распределенных вычислений MapReduce на фреймворке Apache Spark для обработки «больших данных» о состоянии транспортных потоков, а также его модификация на основе сверточной графовой нейронной сети.

Веб-модуль предназначен для объединения требуемых модулей ПМ КППП, для их развертывания в контейнере сервлетов Apache Tomcat и использования в качестве Web-сервиса в части предоставления участнику движения результатов работы программного модуля краткосрочного прогнозирования параметров транспортных потоков.

5.3.3 Схема работы программного модуля

Процесс работы модуля краткосрочного прогнозирования параметров транспортных потоков на основе технологии «больших данных» с использованием метода k ближайших соседей схематично показан на рисунке 149.

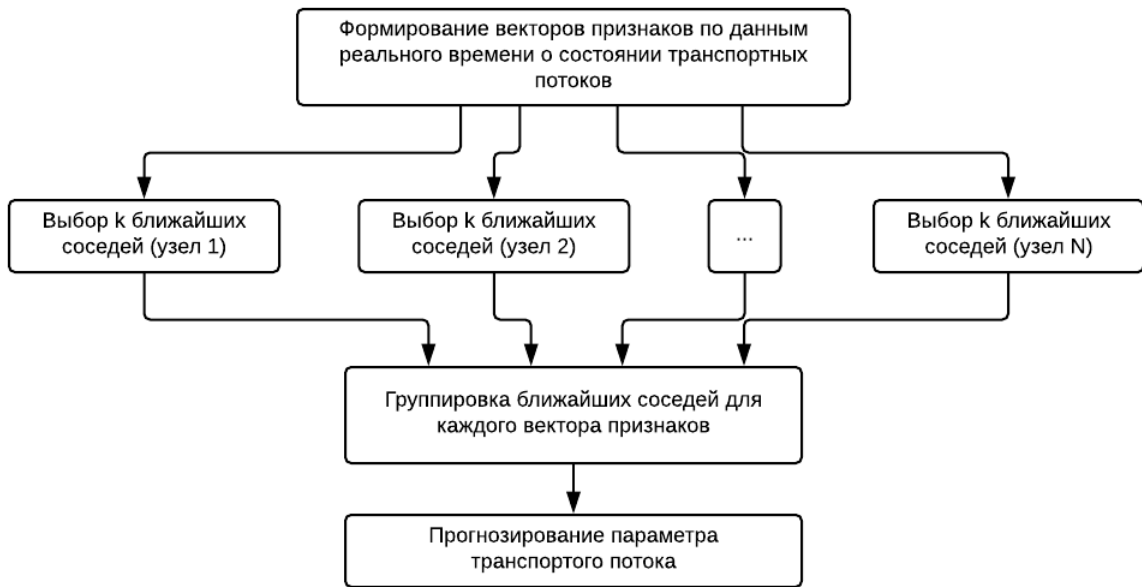


Рисунок 149 – Диаграмма этапов прогнозирования

Процесс состоит из следующих этапов:

1) по данным реального времени о состоянии транспортных потоков для каждого сегмента дорожной сети формируется вектор признаков с учетом информации о состоянии потоков для подграфа графа, которому принадлежит сегмент;

2) полученные вектора признаков распределяются между вычислительными узлами;

3) на каждом вычислительном узле для каждого вектора признаков формируется список локальных k ближайших соседей на основе архивных данных о состоянии транспортных потоков («больших данных»);

4) для каждого элемента тестовых данных множество списков локальных k ближайших соседей преобразуется в результирующий (глобальный) список k ближайших соседей;

5) используя полученные списки, вычисляется прогнозное значение параметра транспортного потока для каждого сегмента дорожной сети;

При использовании графовой нейронной сети прогнозы строятся распределенно для каждого подграфа сети.

5.3.4 Логическая модель программного модуля

UML-диаграмма основных классов, реализующих алгоритм краткосрочного прогнозирования параметров транспортных потоков на основе технологии «больших

данных» с использованием данных реального времени и архивных данных о состоянии транспортных потоков, показана на рисунке 150.

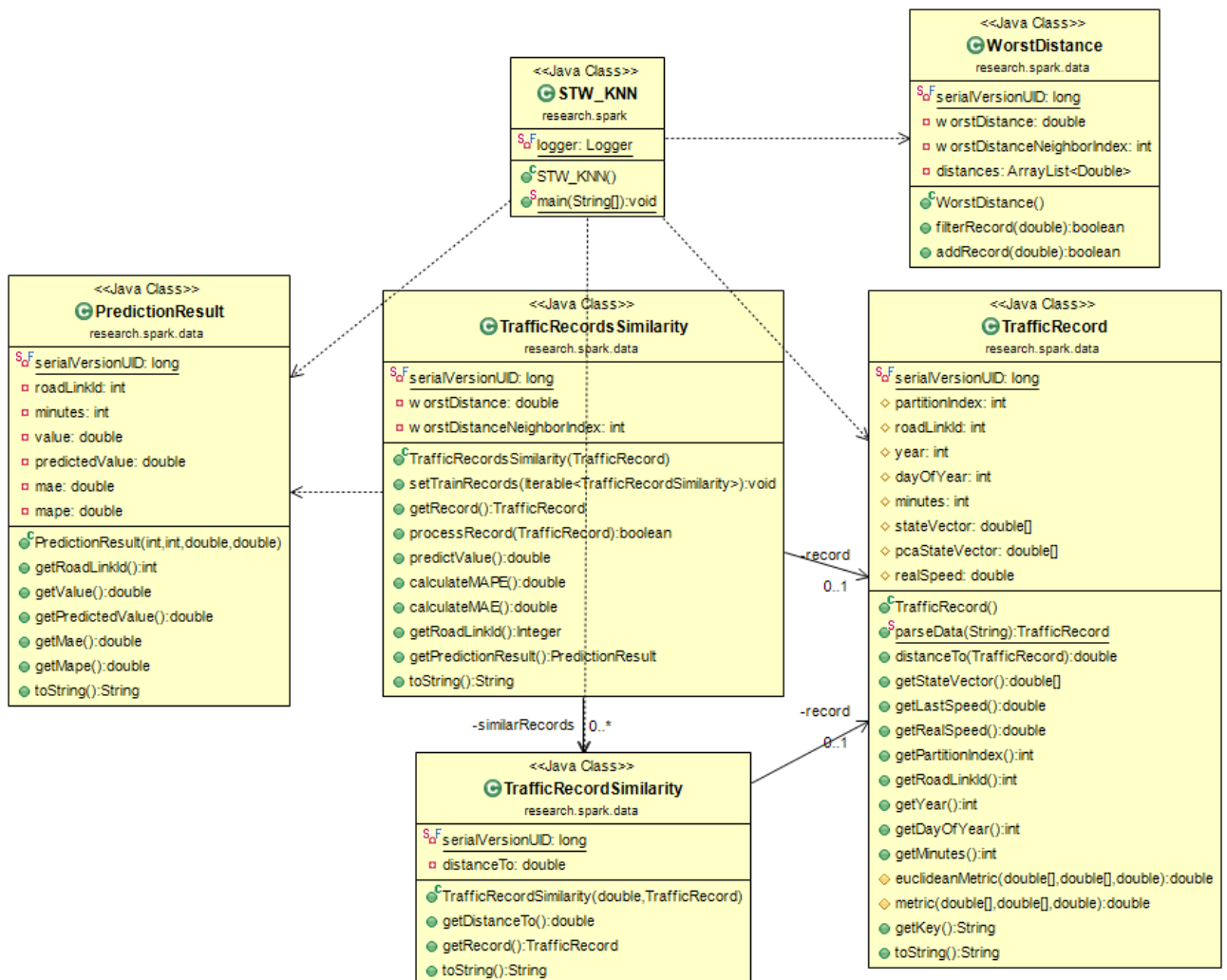


Рисунок 150 – Диаграмма классов для прогнозирования параметров транспортных потоков

Точкой входа в программу является класс `STW_KNN`, осуществляющий формирование векторов признаков, запуск процедуры нахождения локальных k ближайших соседей с помощью фреймворка Apache Spark и формирование прогнозных результатов.

Назначение основных классов программного модуля:

- класс `TrafficRecord` содержит информацию о векторе признаков по актуальным / архивным данным о состоянии транспортных потоков, помимо непосредственно части вектора признака на выбранном сегменте / в соответствующем сегменте подграфе графа, в классе хранится информация о моменте времени, для которого был сформирован вектор признаков;

- класс `TrafficRecordSimilarity` используется для хранения меры близости (расстояния между векторами признаков) между текущим и архивным векторами признаков для нахождения к ближайших соседей;

- класс `TrafficRecordsSimilarity` используется для хранения к ближайших соседей (архивных векторов признаков) для текущего вектора признаков;

- класс `WorstDistance` используется для фильтрации к ближайших соседей с худшей мерой близости;

- класс `PredictionResult` содержит результат прогнозирования параметра транспортного потока, а также наблюдаемое значение для расчета средней абсолютной и средней относительной ошибок прогноза.

5.3.5 Физическая модель данных программного модуля

Данные, необходимые для работы программного модуля, хранятся в SQL базе данных. В качестве СУБД была выбрана PostgreSQL.

Диаграмма таблиц, хранящих информацию, необходимую для составления прогнозов в модели краткосрочного прогнозирования параметров транспортных потоков, приведена на рисунке 151.

Используются следующие таблицы:

- `TRAFFIC_models` – сериализованные параметры модели прогнозирования динамики транспортных потоков;

- `CORE_dayTypes` – тип дня (будний день / выходной);

- `ROAD_links` – список дорожных сегментов с указанием начальных / конечных координат, длины и геометрии в системе координат Web Mercator;

- `TRAFFIC_records` – данные о значении скорости транспортного потока, усредненные за 10-минутный интервал времени. Данные хранятся для каждого сегмента улично-дорожной сети.

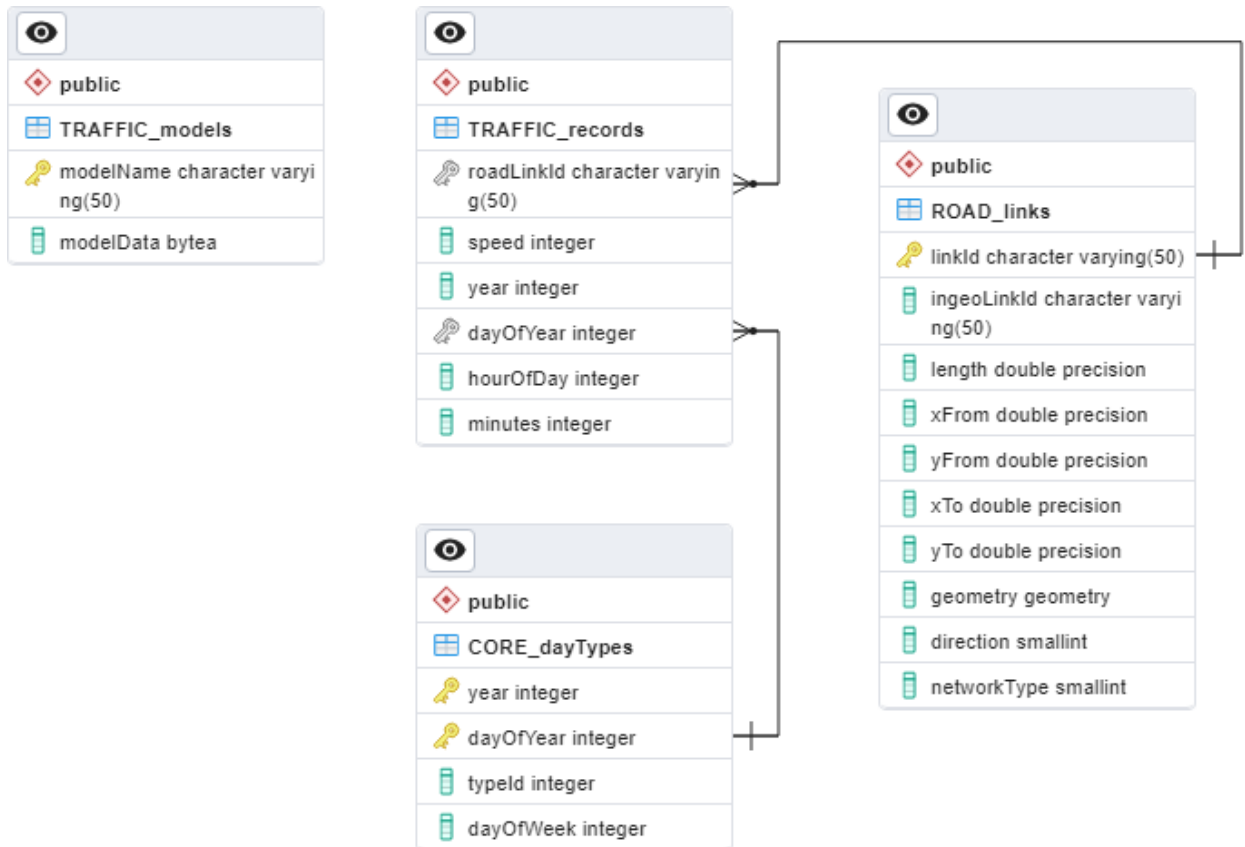


Рисунок 151 – Диаграмма таблиц для прогнозирования параметров транспортных потоков

5.3.6 Программный интерфейс модуля

Web-сервис предназначен для предоставления результатов прогнозирования участникам дорожного движения посредством Web-сервиса по протоколу WMS для выдачи географически привязанных изображений через Интернет и протоколу MVT для выдачи векторных объектов.

Web-сервис доступен по следующему относительному http-адресу: `/wms`.

Для получения информации о доступных слоях, поддерживаемых форматах и типах запросах необходимо выполнить запрос:

`/wms?SERVICE=WMS&VERSION=1.3.0&REQUEST=GetCapabilities`.

Результатом выполнения запроса является XML-документ стандартизированной структуры по стандарту WMS.

Для получения информации о значениях параметров транспортных потоков в виде геопривязанного изображения, необходимо выполнить запрос GetMap следующего вида:

```
/wms?SERVICE=WMS&VERSION=1.3.0&REQUEST=GetMap&LAYERS=0&STYLES=&WIDTH=256&HEIGHT=256&FORMAT=image/png&CRS=urn:ogc:def:crs:EPSG::3857&TRANSPARENT=TRUE&BBOX=5576779,7017154,5579402,7019777,
```

где REQUEST – тип запроса (GetMap);

LAYERS – идентификатор запрашиваемого слоя;

WIDTH – ширина возвращаемого изображения;

HEIGHT – высота возвращаемого изображения;

FORMAT – формат возвращаемого изображения;

CRS – EPSG-код запрашиваемой системы координат;

TRANSPARENT – прозрачность возвращаемого изображения;

BBOX – запрашиваемые границы области карты.

Пример возвращаемой информации о параметрах транспортных потоков в наглядном для пользователя – участника движения виде показан на рисунке 152.



Рисунок 152 – Ответ на запрос GetMap

Запрос векторных данных в формате MVT имеет следующий вид:

```
/mvt/LAYER/Z/X/Y.pbf
```

где LAYER – идентификатор запрашиваемого слоя, X, Y, Z – индексы запрашиваемой области карты.

5.4 Программный модуль адаптивного управления транспортным потоком путем светофорного регулирования

5.4.1 Назначение программного модуля

Программный модуль, реализующий математические методы адаптивного управления транспортным потоком путем светофорного регулирования с использованием как наблюдаемых, так и прогнозных параметров, описывающих состояние транспортного потока, а также данных от подключенных и автономных подключенных транспортных средств (далее – ПМ АУТП) предназначен для реализации методов адаптивного управления сигналами/фазами светофорных объектов и предоставления результатов посредством Web-сервиса.

ПМ АУТП предоставляет следующие основные функции и средства:

- получение информации о параметрах движения подключенных и автономных транспортных средств (положение, скорость, ускорение и маршрут);

- прогнозирование времени движения, необходимого транспортному средству для достижения перекрестка, с использованием выбранной модели прогнозирования (раздел 4.1.2);

- расчет фазы светофорного объекта с использованием выбранного алгоритма адаптивного управления: на основе обучения с подкреплением (раздел 3.2) или с использованием метода, максимизирующего прогнозное количество транспортных средств, которые проедут перекресток за выбранный интервал действия фазы (раздел 3.3);

- расчет траектории движения транспортных средств, согласованных с фазами светофорных объектов, позволяющих достичь координированного управления сигналами светофорных объектов и движением подключенных и/или автономных подключенных транспортных средств;

- предоставление результатов анализа посредством Web-сервиса с использованием разработанного программного интерфейса (API).

Все модули, входящие в состав ПМ АУТП, являются независимыми в том плане, что не требуется их выполнения в рамках определенной заранее последовательности, соответствующей некоторой логической структуре программы. Каждый программный модуль выполняет отдельную самостоятельную задачу. В то же время, сохраняется зависимость между модулями, обусловленная необходимостью использования общих данных и результатов их обработки разными модулями ПМ АУТП.

ПМ АУТП включает в себя следующие модули: модуль сбора информации о параметрах движения подключенных и автономных транспортных средств, модуль адаптивного выбора фазы светофорного цикла, модуль расчета траектории движения транспортных средств.

5.4.2 Логическая модель программного модуля

UML-диаграмма основных классов, реализующих математические методы и алгоритмы адаптивного управления сигналами светофоров, показана на рисунке 153.

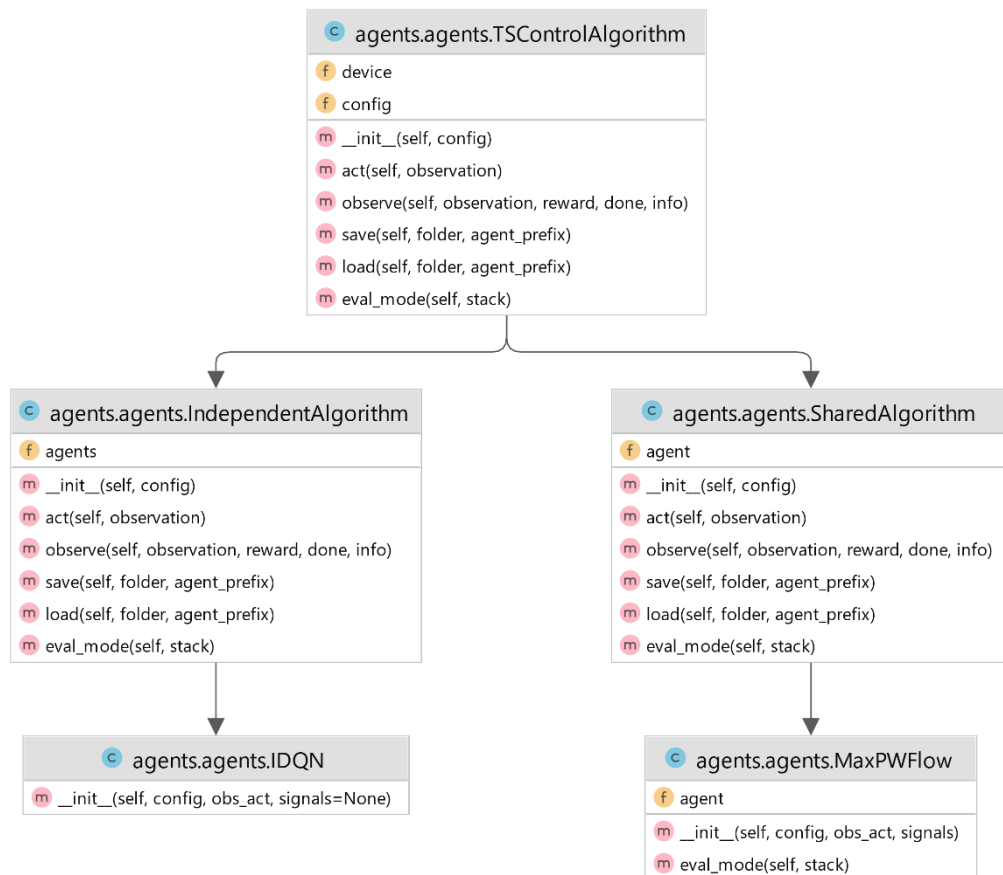


Рисунок 153 – Диаграмма классов для адаптивного управления сигналами светофоров

Назначение основных классов программного модуля:

- класс `TSControlAlgorithm` является базовым абстрактным классом, содержащим методы для получения информации о состоянии окружающей среды и выбора следующего действия агента, а также для сохранения и загрузки параметров агента;
- класс `IndependentAgent` является базовым классом для определения независимых агентов, управляющих отдельными светофорами;

- класс `SharedAgent` является базовым классом для определения агента, управляющих всеми светофорами;

- класс `IDQN` реализует алгоритм адаптивного управления: на основе обучения с подкреплением (раздел 3.2) с использованием подхода двойного Q-обучения;

- класс `MaxPWF1ow` реализует метод адаптивного управления, максимизирующий прогнозируемое количество транспортных средств, которые проедут перекресток за выбранный интервал действия фазы (раздел 3.3).

UML-диаграмма основных классов, реализующих математические методы и алгоритмы построения траектории движения, координированной с прогнозными сигналами светофоров, показана на рисунке 154.



Рисунок 154 – Диаграмма классов для построения траектории движения

Назначение основных классов программного модуля:

- класс `Trajectory` определяет траекторию движения транспортного средства;

- класс `Segment` определяет отдельный сегмент траектории транспортного средства;
- класс `Phases` определяет список фаз светофорного объекта;
- класс `Phase` определяет отдельную фазу светофорного объекта;
- класс `FeasibleTrajectory` определяет реализуемую траекторию транспортного средства.

5.4.3 Физическая модель данных программного модуля

Данные, необходимые для работы программного модуля, хранятся в NoSQL базе данных. В качестве СУБД была выбрана Apache Cassandra.

Диаграмма таблиц, хранящих информацию, необходимую для управления сигналами светофорных объектов, представлена на рисунке 163. Используются следующие таблицы:

- `traffic.vehicles` – данные о положении транспортных средств с моментом времени передачи координат;
- `traffic.actual_phases` – текущие фазы светофоров;
- `traffic.proposed_phases` – прогнозные фазы светофоров.

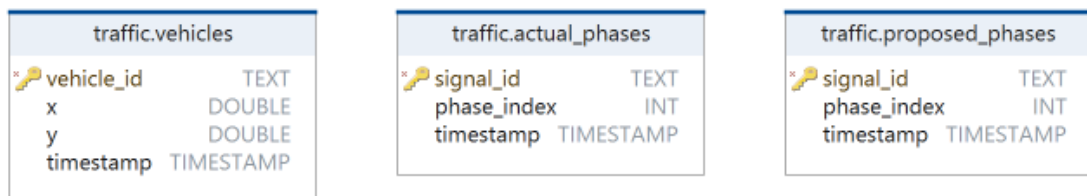


Рисунок 155 – Диаграмма таблиц для управления сигналами светофоров

5.4.4 Процедура обработки данных

Программный модуль реализован на основе лямбда-архитектуры, предназначенной для объединения потоковой обработки данных в режиме реального времени с результатами пакетной аналитики. Предлагаемая процедура обработки данных в составе программного модуля состоит из следующих шагов:

- 1) каждое транспортное средство отправляет информацию о своем движении (положение, скорость, ускорение и маршрут). Эти данные передаются в Apache Kafka, распределенному программному брокеру потоковых сообщений с открытым исходным кодом;

2) на уровне потоковой обработки (каждую секунду) данные о транспортных средствах обрабатываются и сохраняются в оперативной памяти;

3) на уровне пакетной обработки выполняется оценка времени движения для каждого транспортного средства и выбирается следующая фаза для каждого светофора, которая максимизирует прогнозируемые транспортные потоки (каждые две секунды). Результаты сохраняются в базе данных;

4) следующие фазы светофора запрашиваются и возвращаются из базы данных на обслуживающем уровне.

Исходя из требований к системе, можно распределить решаемые системой задачи по уровням обработки лямбда-архитектуры следующим образом:

1) на уровне скоростной обработки (speed layer) выполняется получение новых сообщений о параметрах движения транспортных средств, сохранение данных на диск для последующей пакетной обработки и в БД для выдачи пользователям. Эта обработка происходит в реальном времени, без задержек;

2) на уровне пакетной обработки (batch layer) осуществляется чтение информации с диска и

а) усреднение мгновенной скорости за 5-минутный интервал;

б) обучение модели прогнозирования скорости движения на основе накопленных данных;

в) прогнозирование скорости движения с использованием обученной модели прогнозирования и архивных/текущих данных о скорости движения;

г) сохранение результатов анализа в базу данных;

3) на уровне обслуживания (service layer) выполняется обработка http-запросов пользователей и выдача исходных и обработанных данных, уже сохраненных в базу данных.

5.5 Программный модуль расчета оптимального пути на общественном транспорте

5.5.1 Назначение программного модуля

Программный модуль, реализующий алгоритм расчета оптимального пути на общественном транспорте в стохастической сети с учетом прогнозной информации о времени прибытия транспортных средств (далее – ПМ РОПОТ), предназначен для реализации алгоритма прогнозирования времени прибытия общественных транспортных

средств на остановочные пункты и алгоритма расчета оптимального пути на общественном транспорте, а также предоставления результатов участнику движения посредством Web-сервиса.

ПМ РОПОТ предоставляет следующие основные функции и средства:

- анализ и краткосрочное прогнозирование времени прибытия общественного транспорта на остановочные пункты с использованием данных реального времени и архивных данных о движении транспортных средств;

- предоставление результатов прогнозирования участникам дорожного движения посредством Web-сервиса с использованием разработанного программного интерфейса (API);

- расчет оптимального пути на общественном транспорте в стохастической сети с учетом прогнозной информации о времени прибытия транспортных средств и с использованием индивидуальных предпочтений участника движения;

- предоставление данных расчета оптимального пути на общественном транспорте участникам дорожного движения посредством Web-сервиса с использованием разработанного программного интерфейса (API);

- возможность настройки и обучения модели прогнозирования времени прибытия общественного транспорта.

Прикладная задача прогнозирования времени прибытия общественных транспортных средств на остановочные пункты была реализована в составе программного модуля, разработанного по заказу МБУ «Центр информатизации городского округа Самара». К программному модулю предъявлялись следующие основные требования:

- 1) должны использоваться данные о положении транспортных средств из автоматизированной радионавигационной системы диспетчерского управления пассажирским транспортом «АСУ-Навигация», хранящей данные ГЛОНАСС-мониторинга общественного транспорта по территории г.о. Самара;

- 2) информация о времени прибытия транспортных средств на остановочные пункты должна быть доступна клиентам посредством сайта, мобильных приложений, электронных информационных табло на остановках с возможностью расширения типов клиентских приложений;

- 3) время обработки по всем транспортным средствам и остановкам должно быть гарантированно меньше периода обновления данных.

Аналитическая и прогнозируемая информация формируется с помощью разработанного программного модуля, который должен с высокой скоростью обрабатывать

запросы на получение прогнозов и других продуктов аналитики, поступающие от сайтов, мобильных приложений, информационного табло и других потребителей.

5.5.2 Исходные данные

Программный модуль получает информацию о движении ОТС из системы мониторинга подвижного состава, частью которых являются подсистемы получения информации с датчиков GPS/ГЛОНАСС, установленных на общественный транспорт. Исходными данными из системы мониторинга для решения задачи прогнозирования времени прибытия общественного транспорта на остановки являются:

- 1) перечень транспортных средств, оснащенных системой мониторинга движения;
- 2) запланированные диспетчерской службой маршруты движения транспортных средств;
- 3) координаты положений транспортных средств (в системе координат WGS-84) и моменты времени получения координат (x, y, t) ;
- 4) плановое расписание прохождения транспортом контрольных точек маршрутов.

В исходных данных существует ряд ограничений и недостатков:

- 1) точность получаемых координат от датчика спутникового мониторинга составляет в среднем 6-8 метров, однако в некоторых случаях может достигать 70 метров;
- 2) периодичность поступления информации о положении каждого транспортного средства в среднем составляет 30 секунд, строгой периодичности нет;
- 3) возможны временные отключения датчиков и потери части информации о положении транспортных средств;
- 4) назначенный маршрут движения не содержит информацию о направлении движения и коротких рейсах;
- 5) информация о геометрии маршрута движения отсутствует;
- 6) возможен уход транспортного средства со своего планового маршрута;
- 7) расписания прохождения контрольных точек составлены без учёта зависимости от сезона, времени дня, погодных условий;
- 8) количество контрольных точек на маршруте мало;
- 9) не существует выполняемого норматива, какое время транспорт должен проводить на остановке.

Количественные характеристики обрабатываемых данных следующие:

- 1) количество маршрутов, для которых ведется мониторинг – 100-150 штук;

- 2) количество остановок, для которых рассчитываются прогнозы прибытия – 1000-1500 штук;
- 3) длина маршрута – 10-50 остановок, 3-40 километров;
- 4) количество транспортных средств, оснащенных системой мониторинга движения – 1000-1500 штук;
- 5) усредненная периодичность обновления данных – 30 секунд;
- 6) количество запросов пользователей на получение аналитической и прогнозируемой информации – 5000 в минуту.

Взаимодействие с системой мониторинга подвижного состава «АСУ-Навигация» осуществляется напрямую с базой данных (БД). Диаграмма таблиц БД «АСУ-Навигация», используемых для получения необходимых данных, изображена на рисунке 156:

- `TransportTypes` – типы транспортных средств в системе. В системе выделяют следующие транспортные средства: автобус, трамвай, троллейбус, маршрутное такси;
- `tbMarhes` – список маршрутов;
- `CntrlPoints` – список контрольных точек, для которых составляется расписание их прохождения транспортными средствами;
- `tbNSITransport` – список транспортных средств в системе;
- `tbCardNariad` – список назначений транспортных средств на маршруты;
- `Nav_cur` – последние полученные координаты транспортного средства в системе координат WGS84, момент времени их получения.

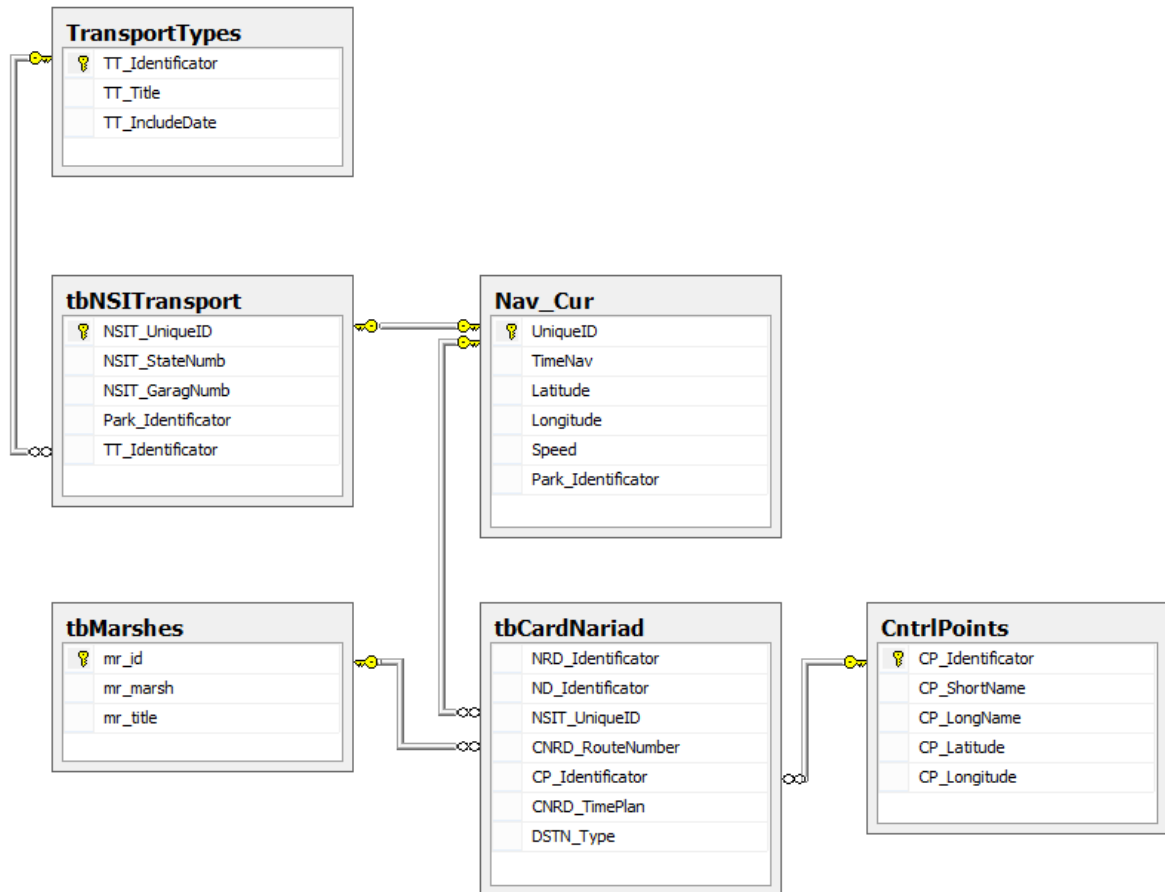


Рисунок 156 – Диаграмма таблиц БД «АСУ-Навигация»

5.5.3 Формирование модели транспортной системы города

Важным этапом в предварительной подготовке исходных данных является построение модели транспортной системы города. Модель представляет собой ориентированный граф, по структуре и геометрии повторяющий правила прохождения маршрутов транспортом.

В силу пространственного характера модели для подготовки исходных данных логично использовать геоинформационную систему. В качестве такой системы была выбрана ГИС «Ингео» со специальным модулем, реализующим автоматические процедуры по подготовке транспортных данных. В качестве базовой карты использовалась Единая цифровая картографическая основа Самары в составе слоев адресного плана, автомобильных дорог, трамвайных путей и путей метрополитена.

В нескольких слоях транспортной инфраструктуры была организована структура слоев, стилей (классов и подклассов пространственных объектов) и семантических данных для описания остановок и маршрутов, а затем внесены данные по следующим правилам:

1) остановочные пункты создавались как точечные объекты, с высокой точностью расположенные на месте остановочных павильонов;

2) маршруты автобусов, трамваев, троллейбусов и метро создавались в отдельных слоях по видам транспорта в виде объектов, содержащих в семантике упорядоченный список остановок в порядке их прохождения; два направления каждого маршрута создавались как два отдельных объекта;

3) геометрическая форма маршрутов рассчитывалась автоматической процедурой по слоям осевых линий дорог и трамвайных путей, объединяя кратчайшими путями интервалы между последовательными остановками; результаты проверялись вручную, при необходимости корректировались промежуточными обязательными точками.

Объекты остановок и маршрутов так же имели семантическое описание (название, направление) и классификационные номера, позволяющие однозначно устанавливать взаимосвязи между объектами.

Из подготовленных слоев транспортной инфраструктуры автоматической процедурой строился граф следующей структуры: остановки транспорта являлись вершинами графа, перегоны между остановками – ребрам графа. Такая структура графа позволяет:

- уточнять маршрут транспортного средства;
- уточнять координаты положения транспортного средства на маршруте для исправления погрешности GPS/ГЛОНАСС;
- рассчитывать время прибытия на остановки с учетом данных о расписании движения.

Кроме того, каждое ребро маршрутного графа дополнительно содержит атрибутивную информацию о сегментах дорожной сети, из которых оно состоит, в порядке следования сегментов, а также их длины. Наличие такой информации позволяет использовать в прогнозах информацию о прохождении одного дорожного сегмента транспортными средствами разных маршрутов.

5.5.4 Описание логической структуры

Все модули, входящие в состав ПМ РОПОТ, являются независимыми в том плане, что не требуется их выполнения в рамках определенной заранее последовательности, соответствующей некоторой логической структуре программы. Каждый программный модуль выполняет отдельную самостоятельную задачу. В то же время, сохраняется

зависимость между модулями, обусловленная необходимостью использования общих данных и результатов их обработки разными модулями ПМ РОПОТ.

ПМ РОПОТ включает в себя следующие модули: модуль обработки информации о движении общественного транспорта, модуль прогнозирования времени прибытия общественного транспорта, модуль расчета оптимального пути на общественном транспорте.

Модуль обработки информации о движении общественного транспорта реализует следующие функции:

- обновление информации об остановках и маршрутах общественного транспорта из файлов-классификаторов, формируемых в геоинформационной системе;
- установка связей между маршрутами общественного транспорта в ПМ РОПОТ и соответствующими маршрутами в системе мониторинга движения общественного транспорта;
- вызов модуля краткосрочного прогнозирования времени прибытия общественного транспорта на остановочные пункты с использованием данных реального времени и архивных данных о движении транспортных средств и сохранение прогнозной информации;
- предоставление результатов прогнозирования участникам дорожного движения посредством Web-сервиса с использованием разработанного программного интерфейса.

Модуль прогнозирования предназначен для краткосрочного прогнозирования времени прибытия общественного транспорта на остановочные пункты с использованием данных реального времени и архивных данных о движении транспортных средств. Реализация алгоритма краткосрочного прогнозирования вынесена в отдельный модуль для уменьшения связности между модулями и возможности замены модели прогнозирования без необходимости изменения реализации других модулей. В рамках ПМ РОПОТ используется разработанный алгоритм краткосрочного прогнозирования движения транспортных средств на основе рекуррентной нейронной сети с долгой краткосрочной памятью (сеть LSTM), учитывающий разнородную информацию, описывающую транспортную ситуацию, включая данные реального времени и статистические данные о движении транспортных средств.

Модуль расчета оптимального пути на общественном транспорте предназначен для нахождения оптимального пути на общественном транспорте в стохастической транспортной сети с учетом прогнозной информации о времени прибытия транспортных средств. Прогнозная информация о времени прибытия общественного получается из

модуля обработки информации о движении общественного транспорта. Критерий оптимальности определяется исходя из индивидуальных предпочтений участников движения.

Разработанные модули объединены в составе веб-модуля, предназначенного для развёртывания ПМ РОПОТ в контейнере сервлетов Apache Tomcat и использования в качестве Web-сервиса в части предоставления участнику движения результатов работы программного модуля (прогнозов прибытия общественного транспорта и оптимальных путей движения на общественном транспорте).

Блок схема взаимодействия основных компонентов ПМ РОПОТ представлена на рисунке 157. Процесс взаимодействия программных компонентов состоит из следующих этапов:

- модуль обработки информации о движении транспортных средств (ТС) непрерывно в режиме реального времени обрабатывает координаты транспортных средств (уточняет с использованием информации о дорожной сети, определяет маршрут движения ТС), обменивается обработанной информацией с модулем прогнозирования времени прибытия и возвращает прогнозную информацию о времени прибытия ТС на остановки при получении запроса посредством Web-сервиса информирования;

- модуль прогнозирования использует данные реального времени и статистические данные о движении транспортных средств для прогнозирования времени прибытия ТС на остановки с использованием разработанной LSTM-модели;

- модуль расчета оптимального пути реализует алгоритм поиска оптимального пути на общественном транспорте в стохастической сети с использованием прогнозной информации о движении транспортных средств. Расчет происходит при получении запроса от участника дорожного движения через Web-сервис информирования. Модуль расчета оптимального пути взаимодействует с модулем обработки информации о движении ТС для получения прогнозной информации о прибытии ТС на остановки.

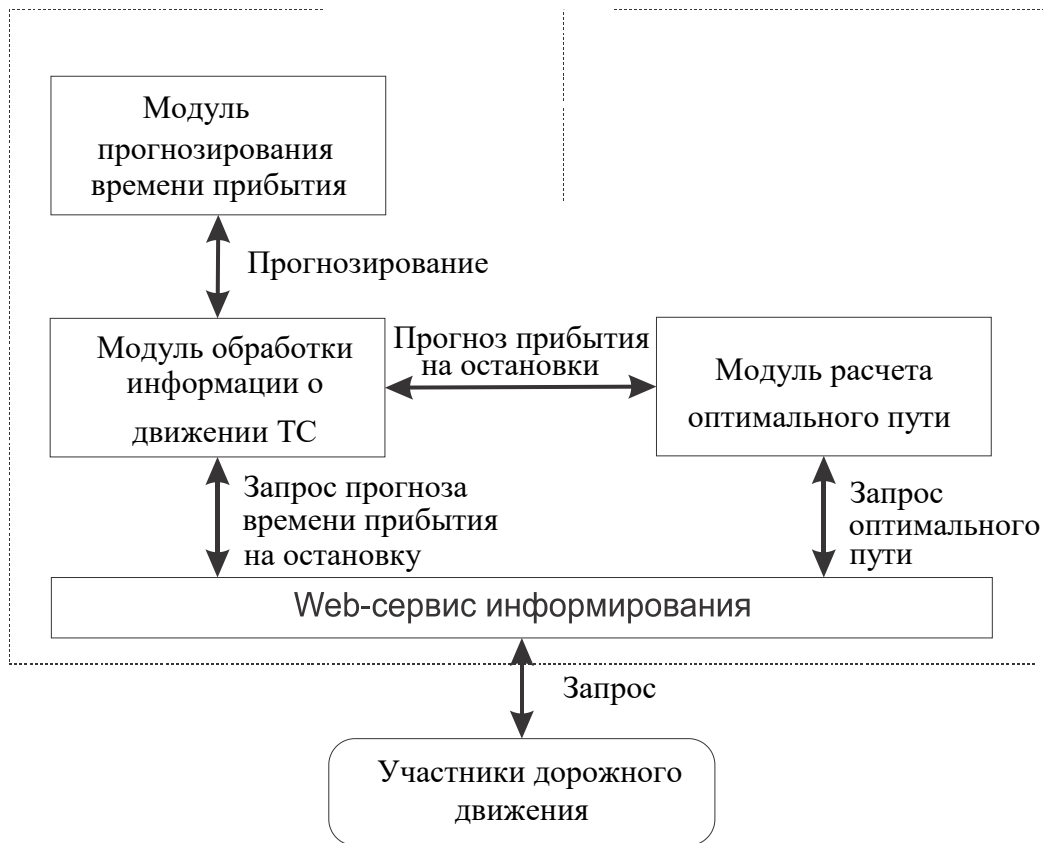


Рисунок 157 – Диаграмма взаимодействия

5.5.5 Описание работы программного модуля

Данные спутникового мониторинга поступают в программный комплекс в режиме реального времени, и для поддержания актуальности результатов прогноза требуется постоянный пересчёт. Чтобы снизить зависимость вычислительной нагрузки программного модуля от количества поступающих запросов, используется программная архитектура из двух циклов обработки и хранилища промежуточных результатов расчёта в оперативной памяти. Первый цикл периодически запрашивает новые данные спутникового мониторинга, проводит расчёт прогнозов, и сохраняет промежуточные результаты. Второй цикл обрабатывает поступающие запросы от потребителей, и отвечает на них по имеющимся промежуточным результатам. В таком варианте архитектуры нагрузка на программный комплекс может быть очень высока, а актуальность предоставляемых прогнозов никогда не бывает хуже периода обновления данных.

Алгоритм циклического пересчёта состоит из следующих последовательных шагов:

1) определяется полный набор транспортных средств, для которых ведется спутниковый мониторинг, у базы данных мониторинга запрашиваются последние положения каждого из них;

2) из набора выделяется подмножество транспортных средств, изменивших свое положение с предыдущего цикла пересчёта, для каждого них выполняются последующие шаги;

3) по модели транспортной системы города делается уточнение положения транспорта, в какой координате и на каком участке маршрута находится транспорт;

4) по текущему и предыдущим положениям транспорта определяется его состояние. Транспортное средство, чье состояние определено как «не на маршруте», не должно учитываться в прогнозах и при накоплении статистики;

5) по положению транспорта рассчитывается прогнозное время прибытия на все остановки по ходу следования с помощью модели адаптивной комбинации элементарных прогнозов;

б) в хранилище промежуточных результатов заносится новая порция прогнозов от рассматриваемого транспорта.

Важным свойством алгоритма является то, что расчёт требует значений времени прохождения только тех немногих участков маршрута, которые соответствуют текущему положению транспортного средства, появляется возможность рассчитывать эти характеристики моментально, не теряя в вычислительной эффективности всей системы.

Рассмотрим подробнее шаги алгоритма циклического пересчета результатов прогноза.

5.5.5.1 Уточнение положения транспортных средств.

Определение точных координат положения транспортного средства на маршрутной сети и выбор маршрута производится в два этапа. Первый, подготовительный этап состоит в получении списка возможных координат на дорожной сети для каждого из возможных маршрутов транспортного средства, а также пройденного транспортным средством расстояния от начала маршрута. На втором этапе маршрут и, соответственно, координаты положения транспортного средства уточняются по новой порции данных, полученных из передатчика навигационной системы.

На рисунке 158 показан результат определения положения транспортного средства на трамвайной маршрутной сети города. Синие точки показывают координаты, полученные от навигационных систем GPS/ГЛОНАСС. Видно, что координаты имеют погрешность и не точно определяют маршрут транспорта. Черные точки показывают проекции положений транспорта на возможные маршруты. После получения второй порции данных маршрут определен точно, направление движения показано стрелкой.

Далее, при получении новых координат транспортного средства, происходит их привязка к выбранному маршруту.

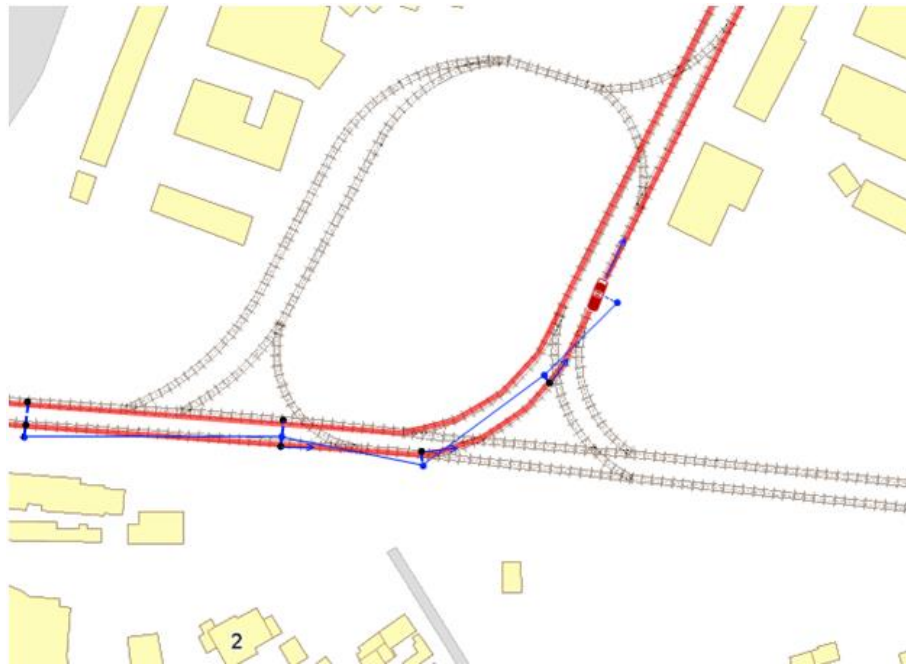


Рисунок 158 – Определение точных координат транспортного средства

5.5.5.2 Определение состояния транспортного средства

Будем считать, что транспорт может находиться в одном из следующих состояний:

1) неопределенное состояние. Транспорт находится в этом состоянии, пока его маршрут и положение на маршруте не уточнено по нескольким последовательным положениям;

2) движение. Транспорт находится в состоянии движения, если его положение на маршруте изменяется;

3) ожидание. Транспорт переходит в состояние ожидания, если его координаты не изменились за период обновления данных. Ситуация возможна, если транспорт находится на остановке, около светофора, либо движется в условиях дорожного затора;

4) непредвиденная ситуация. Возникает, если транспортное средство не изменяет свои координаты в течение нескольких циклов обновления данных (больше 10 минут). При переходе в это состояние транспорт необходимо удалить из прогнозов.

Диаграмма состояний транспортных средств с условиями перехода между состояниями изображена на рисунке 159.

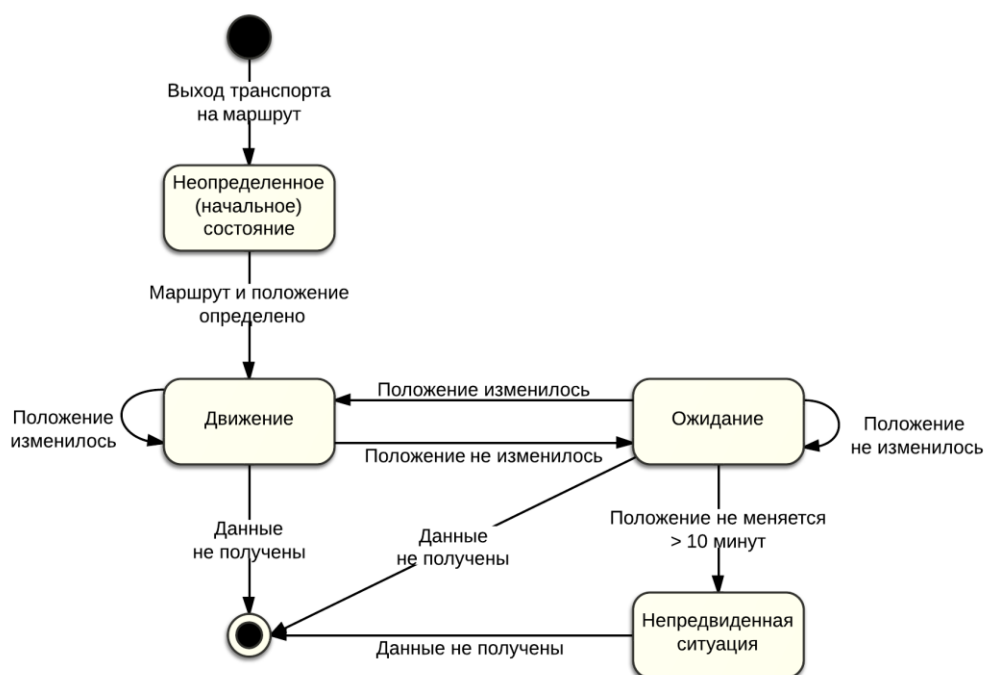


Рисунок 159 – Диаграмма состояний транспортного средства

UML-диаграмма классов, реализующих диаграмму состояний ТС, показана на рисунке 160. Классы конкретных состояний содержат информацию о том, при каких условиях и в какие состояния осуществляется переход объекта из текущего состояния, т.е. реализуют связь между состояниями, представленную на диаграмме (рисунок 160).

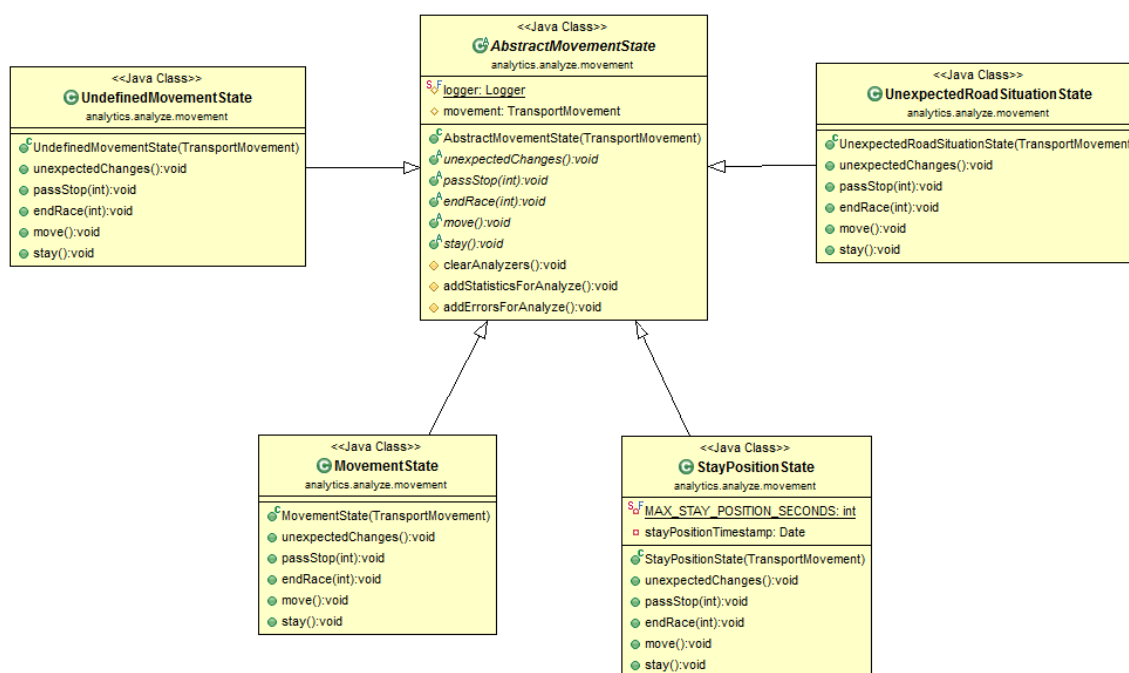


Рисунок 160 – UML-диаграмма классов-состояний

5.5.5.3 Прогноз времени прибытия

После определения маршрута и точных координат транспортного средства, необходимо рассчитать прогнозируемое время прибытия на остановочные пункты. Прогноз времени прибытия выполняется с использованием алгоритма прогнозирования, описанного в разделе 4.1.1.

UML-диаграмма основных классов, реализующих алгоритм краткосрочного прогнозирования времени прибытия общественного транспорта с использованием рекуррентной нейронной сети с архитектурой LSTM показана на рисунке 161.

Прогнозирование времени прибытия транспорта осуществляется в классе LSTMArrivalTimePredictor, который обрабатывает события, возникающие при движении транспортных средств для формирования вектора признаков LSTM-модели, а также реализует функцию predict для прогнозирования времени прибытия.

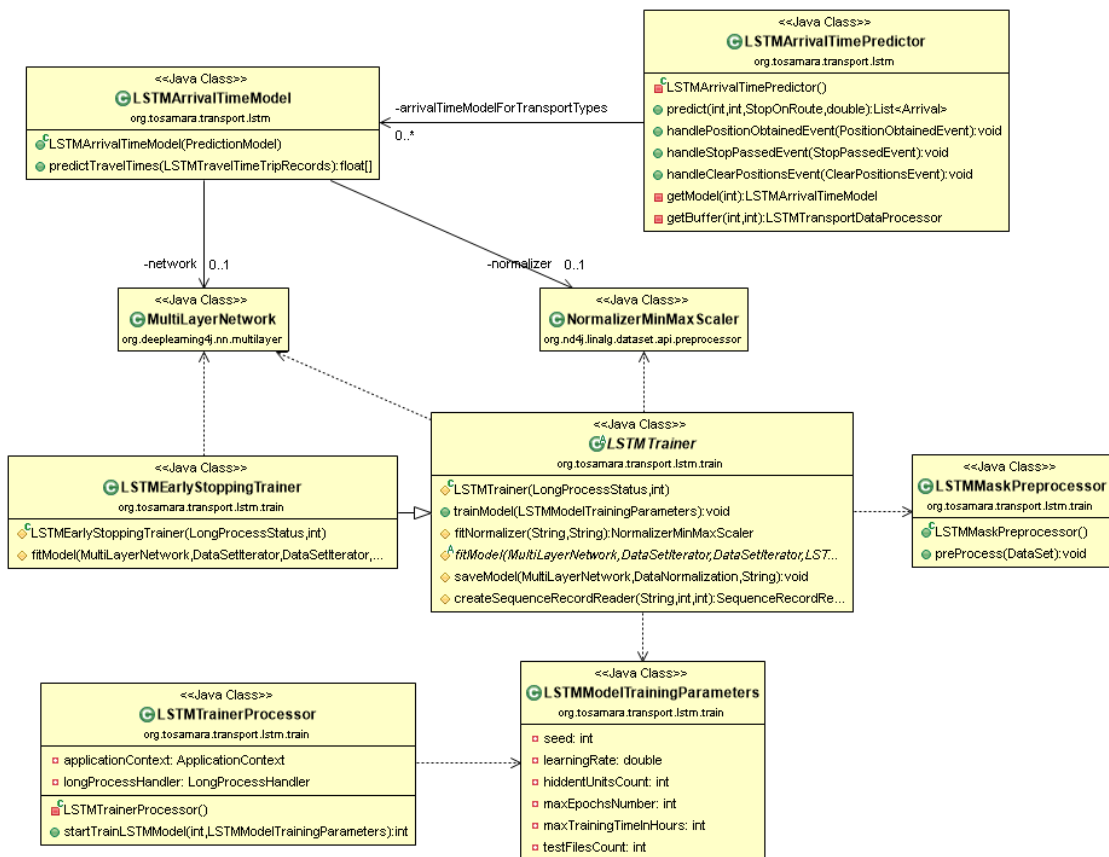


Рисунок 161 – UML-диаграмма для прогнозирования времени прибытия транспортных средств

Назначение основных классов программного модуля:

- класс MultiLayerNetwork хранит информацию о параметрах рекуррентной нейронной сети (слоях, весах нейронов и т.д.), используемой для прогнозирования;

- класс `NormalizerMinMaxScaler` используется для нормализации вектора признаков, используемого для прогнозирования;
- класс `LSTMMaskPreprocessor` используется для предобработки данных и маскирования векторов признаков различной длины ;
- класс `LSTMArrivalTimeModel` хранит обученную модель прогнозирования для отдельного типа транспорта;
- класс `LSTMModelTrainingParameters` хранит значения параметров, задаваемых оператором, для обучения моделей;
- класс `LSTMTrainer` является базовым классом для обучения моделей прогнозирования времени прибытия;
- класс `LSTMEarlyStoppingTrainer` позволяет осуществлять обучение модели прогнозирования с одновременной валидацией модели;
- класс `LSTMTrainerProcessor` используется для запуска процесса обучения модели по запросу оператора системы.

UML-диаграмма основных классов, реализующих алгоритм расчета оптимального пути на общественном транспорте, показана на рисунке 162. Точкой входа в программу является класс `PublicTransportGraphDecorator`, который хранит граф транспортной сети города и вызывает алгоритм нахождения оптимального пути на общественном транспорте, реализованный в классе `DijkstraShortestPathWeightFunction`.

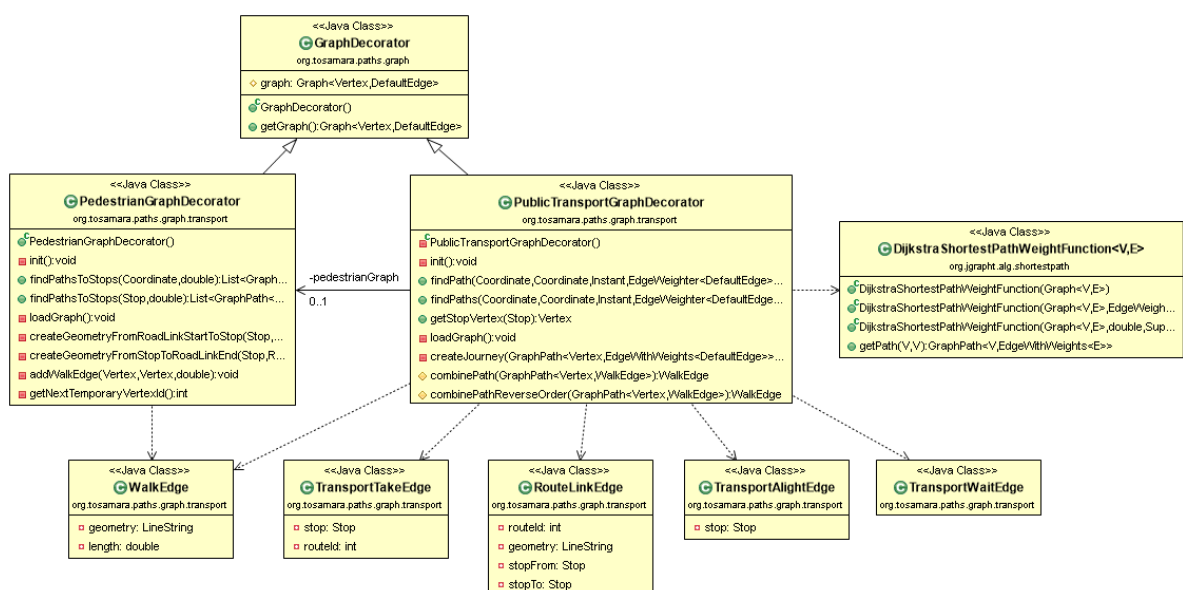


Рисунок 162 – UML-диаграмма классов для расчета оптимального пути на общественном транспорте

Назначение основных классов программного модуля:

- класс `GraphDecorator` – базовый класс для структуры типа граф;
- класс `PedestrianGraphDecorator` – пешеходный граф;
- класс `PublicTransportGraphDecorator` – транспортный граф;
- класс `WalkEdge` – ребро графа, описывающее переход пешком;
- класс `TransportTakeEdge` – ребро графа, описывающее посадку на транспорт;
- класс `RouteLinkEdge` – ребро графа, описывающее сегмент маршрутной сети между парой остановок;
- класс `TransportAlightEdge` – ребро графа, описывающее высадку из транспортного средства на остановке;
- класс `TransportWaitEdge` – ребро графа, описывающее ожидание транспортного средства на остановке с учетом прогнозной информации о времени прибытия;
- класс `DijkstraShortestPathWeightFunction` – класс, реализующий алгоритм поиска оптимального пути.

5.5.6 Физическая модель данных программного модуля

Данные, необходимые для работы программного модуля, хранятся в SQL базе данных. В качестве СУБД была выбрана PostgreSQL.

Диаграмма таблиц, хранящих информацию, необходимую для анализа движения общественного транспорта, представлена на рисунке 163. Используются следующие таблицы:

- `TRANSPORT_statistics` – статистические данные о скорости прохождения сегментов маршрутной сети между парой остановок;
- `TRANSPORT_stops` – классификатор остановок, используемый в ПМ РОПОТ;
- `TRANSPORT_stopsLinks` – связь классификатора остановок с контрольными точками в системе мониторинга подвижного состава;
- `TRANSPORT_routes` – классификатор маршрутов в ПМ РОПОТ;
- `TRANSPORT_routesLinks` – связь классификатора маршрутов в ПМ РОПОТ с маршрутами в системе мониторинга подвижного состава;
- `TRANSPORT_types` – типы транспортных средств;
- `TRAFFIC_schedule` – информация о прохождении контрольных точек на маршруте по расписанию движения.

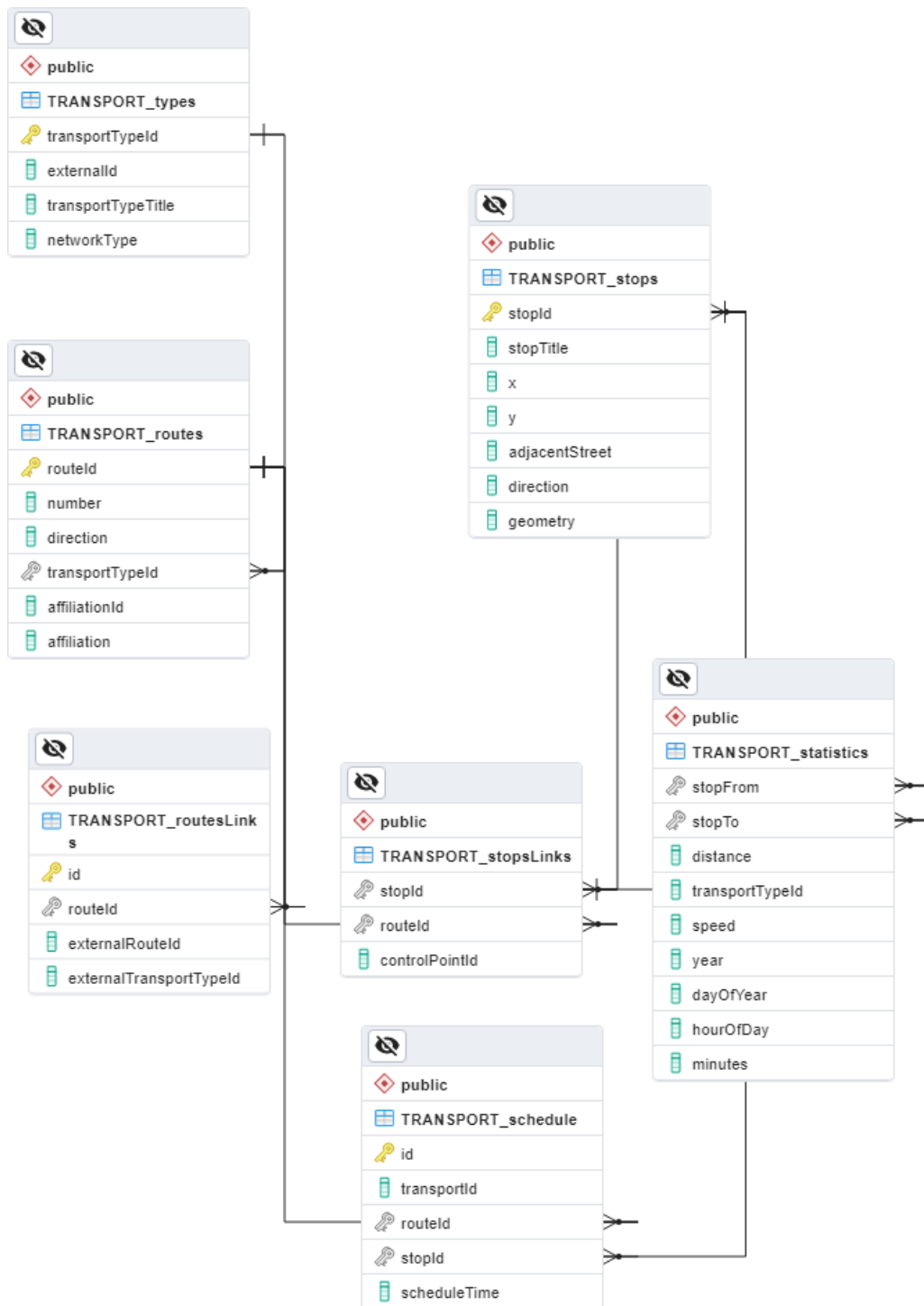


Рисунок 163 – Диаграмма таблиц для анализа движения общественного транспорта

Диаграмма таблиц, хранящих информацию, необходимую для прогнозирования времени прибытия общественного транспорта на остановки, представлена на рисунке 164. Используются следующие таблицы:

- CORE_models – хранение моделей прогнозирования времени прибытия в сериализованном виде;
- LSTM_arrivalModels – хранение связи модели прибытия из таблицы CORE_models и указанного типа транспорта.

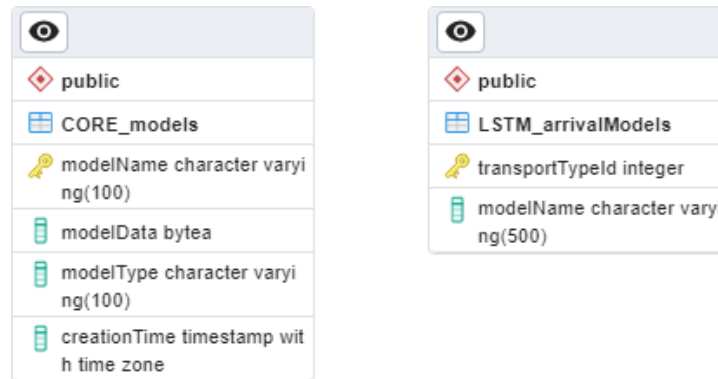


Рисунок 164 – Диаграмма таблиц для прогнозирования времени прибытия

Для расчета оптимального пути на общественном транспорте помимо описанных выше таблиц используются данные о дорожной сети, хранящиеся в таблице ROAD_links, показанной на рисунке 151.

5.5.7 Программный интерфейс модуля

Расчет оптимального пути на общественном транспорте в стохастической транспортной сети с учетом прогнозной информации о времени прибытия транспортных средств и с использованием индивидуальных предпочтений участников движения доступен посредством Web-сервиса с использованием разработанного программного интерфейса (API).

Web-сервис доступен по следующему относительному http-адресу: /api/v1/json.

Результатом выполнения запросов к API является json-документ разработанной структуры.

Формат запроса получения прогнозов времени прибытия общественного транспорта на остановку имеет следующий вид:

```
{ "method": "arrivalForecast",
  "deviceId": "j2d489h24f89hf9384f",
  "stopId": 9,
  "limit": 100 }
```

Формат запроса получения оптимального пути на общественном транспорте имеет следующий вид:

```
{
  "method": "startPublicTransportJourneySupport",
  "deviceid": "j2d489h24f89hf9384f",
  "startLocation": "POINT(50.337865 53.245659)",
  "endLocation": "POINT(50.358865 53.284659)",
  "endLocationSavedPlaceId": "54321",
  "tunedPreferences": {
    "availableTransport": [
      "bus",
      "tram",
      "taxi"
    ],
    "firstTransport": "tram",
    "secondTransport": "taxi",
    "groupTrips": "alone",
    "criterion": "probability"
  }
}
```

Пример запроса оптимального пути на общественном транспорте и ответа сервера показан на рисунках 13 и 14 соответственно.

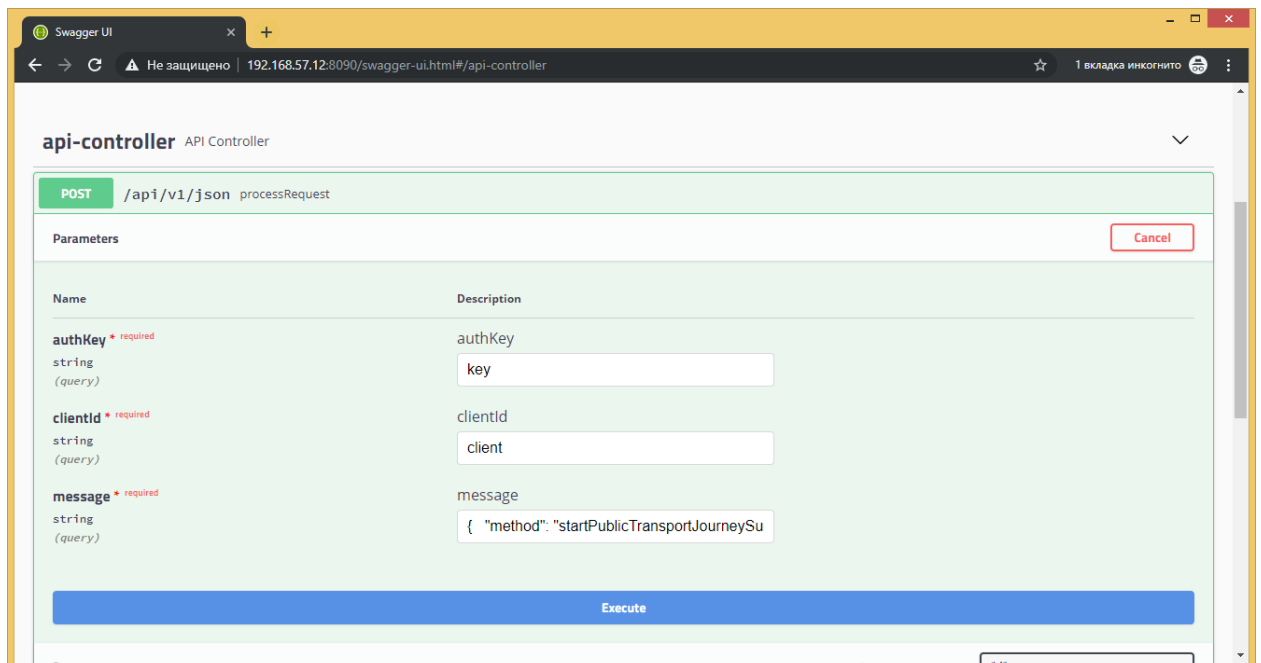


Рисунок 165 – Запрос оптимального пути на общественном транспорте

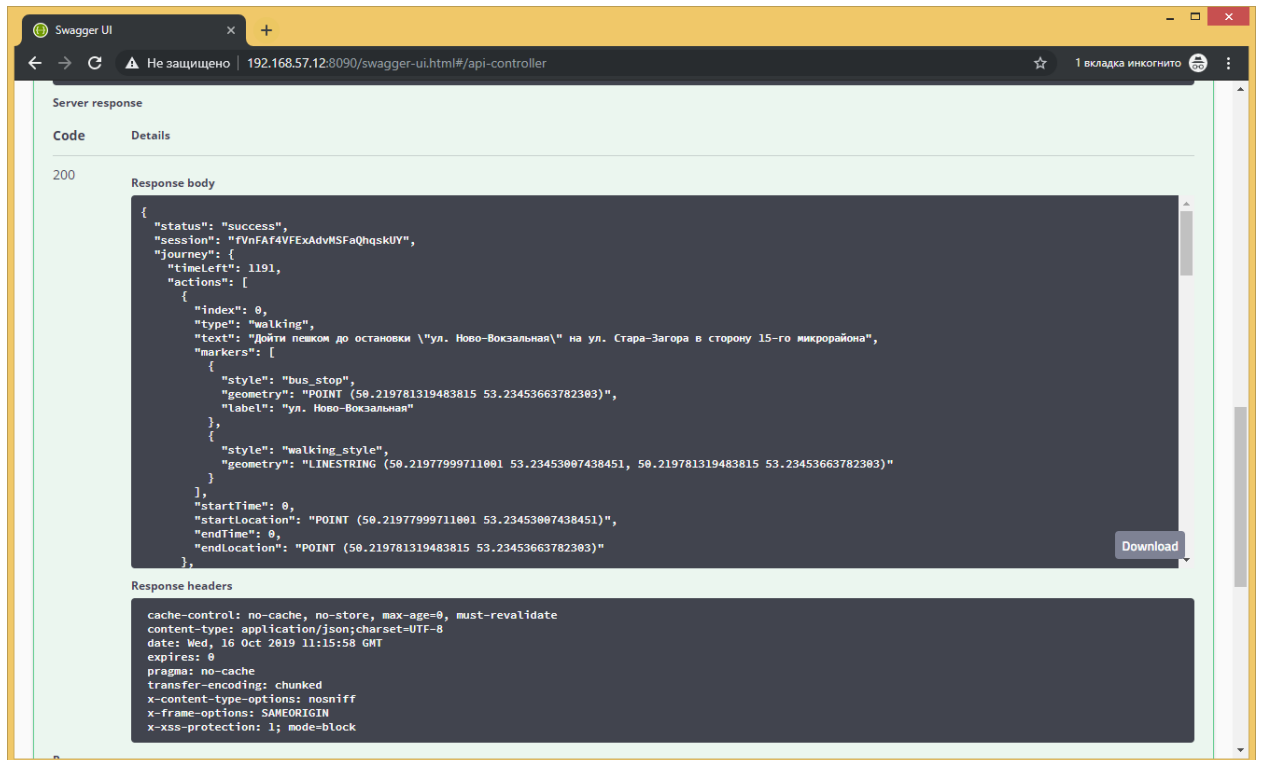


Рисунок 166 – Ответ сервера

В процессе работы ПМ РОПОТ формируется архив данных о движении транспортных средств для последующей настройки и обучения моделей прогнозирования времени прибытия транспортных средств.

В процессе работы web-сервиса предоставления участнику движения результатов работы программного модуля формируются результаты прогнозов прибытия общественного транспорта и оптимальных путей движения на общественном транспорте в формате JSON.

5.5.8 Взаимодействие с клиентскими приложениями

Прогнозы прибытия и другая аналитическая информация о движении ОТС должны быть доступны сторонним сервисам для информирования пассажиров, поэтому для взаимодействия сервисов с программным комплексом был разработан программный интерфейс. Обращение к программному интерфейсу происходит с помощью HTTP-протокола. Существование единого программного интерфейса позволяет разрабатывать новые типы клиентских приложений без внесения изменений в работу программного комплекса. В программном комплексе реализован интерфейс со следующими методами:

1) получение последнего известного положения подвижного состава общественного транспорта на маршруте;

2) получение информации о прибытии ближайших транспортных средств доступных маршрутов на выбранную остановку;

3) получение информации о прибытии транспортных средств выбранного маршрута на выбранную остановку;

4) получение расписания движения.

Получение прогнозов для заданной остановки выполняется в отдельном потоке и не затрагивает основной цикл пересчета положений ТС и формирования прогнозов. Состоит из двух шагов:

1) из хранилища делается выборка прогнозов всех маршрутов, проходящих через остановку;

2) набор прогнозов сортируется по времени, снабжается дополнительной информацией о прибывающих транспортных средствах (бортовой номер, местонахождение), и в определенном формате записи выдается в ответ на запрос.

В настоящее время программный комплекс используется для отображения информации о движении транспорта на сайте транспортного оператора Самары (рисунок 167), с помощью мобильных приложений (рисунок 168), на карте города (рисунок 169), с помощью электронных табло на остановочных пунктах.

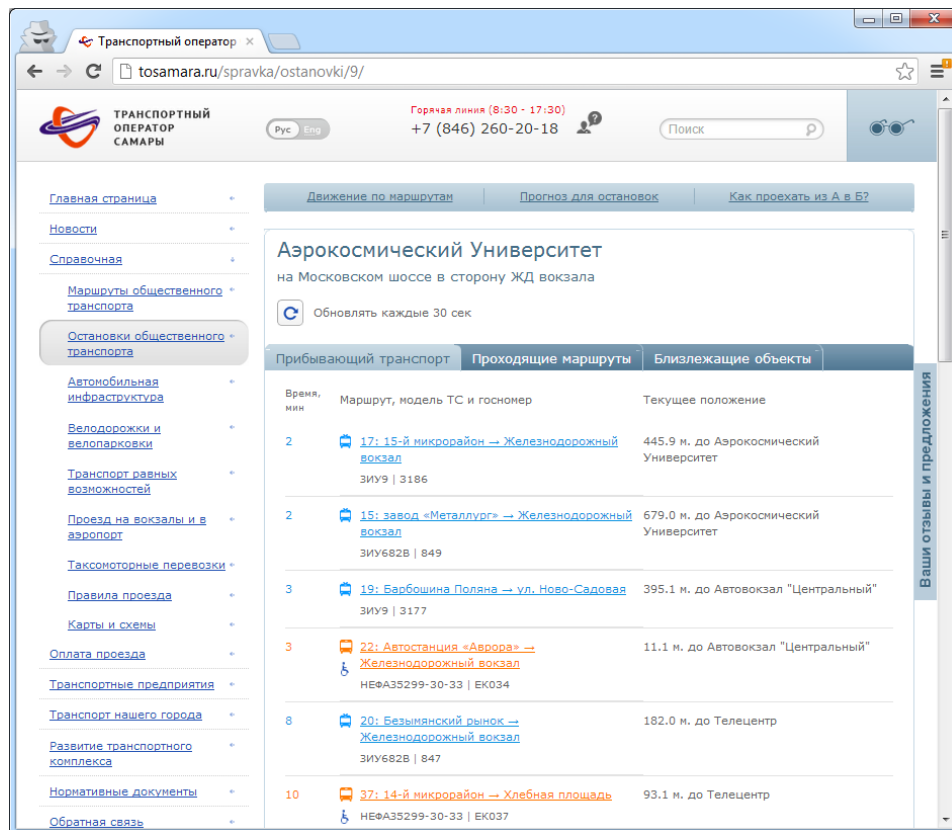


Рисунок 167 – Сайт транспортного оператора Самары

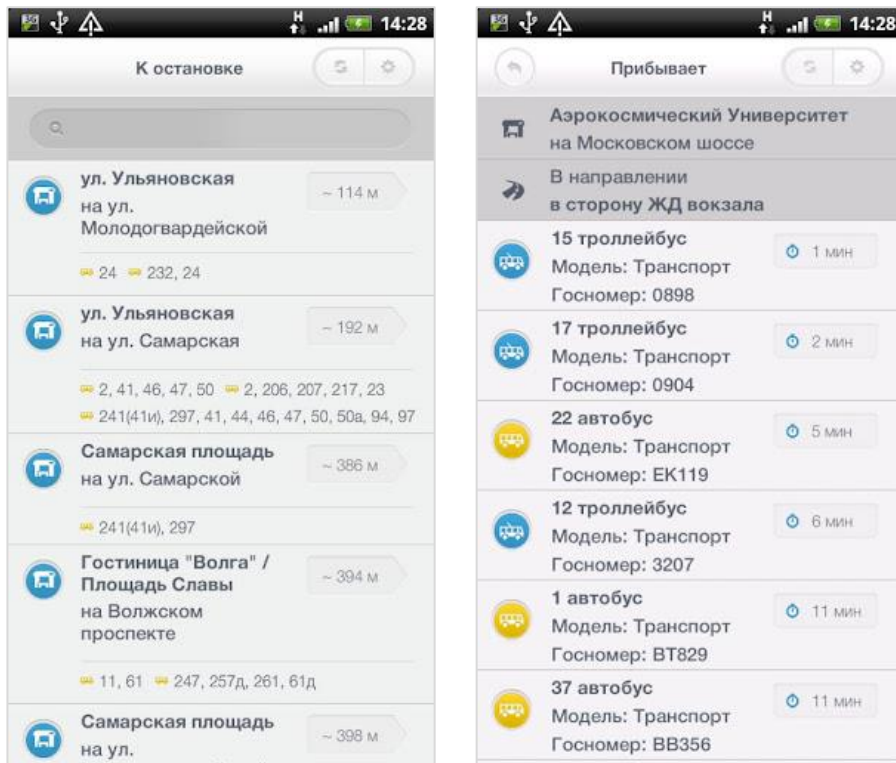


Рисунок 168 – Мобильное приложение

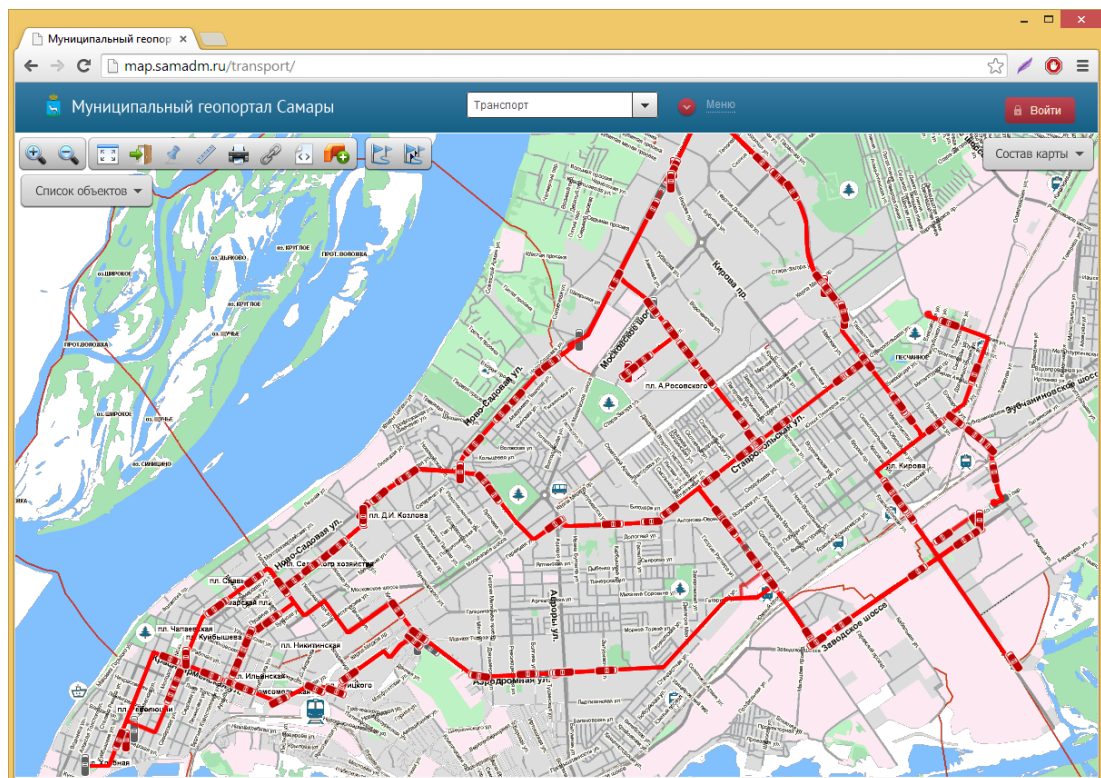


Рисунок 169 – Карта движения транспорта

5.6 Программный модуль расчета оптимального (в том числе – надежного) пути на индивидуальном транспорте в стохастической сети

5.6.1 Назначение программного модуля

Программный модуль, реализующий математические алгоритмы расчета оптимального (в том числе – надежного) пути на индивидуальном транспорте в стохастической сети с учетом информации о прогнозном распределении транспортных потоков (далее – ПМ РПИТ), предназначен для реализации алгоритма маршрутизации индивидуальных транспортных средств в стохастической транспортной сети, а также предоставления результатов участнику движения посредством Web-сервиса.

ПМ РПИТ предоставляет следующие основные функции и средства:

- 1) расчет оптимального (кратчайшего) пути, минимизирующего общее время поездки, и альтернативных путей на индивидуальном транспорте в стохастической сети с учетом информации о прогнозном распределении транспортных потоков;
- 2) расчет надежного пути на индивидуальном транспорте в стохастической сети, максимизирующего вероятность достижения пункта назначения в течение определенного промежутка времени, с учетом информации о прогнозном распределении транспортных потоков;
- 3) предоставление данных расчета оптимального пути на индивидуальном транспорте участникам дорожного движения посредством Web-сервиса с использованием разработанного программного интерфейса (API);
- 4) возможность настройки параметров работы модуля расчета оптимального пути на индивидуальном транспорте.

Все модули, входящие в состав ПМ РПИТ, являются независимыми в том плане, что не требуется их выполнения в рамках определенной заранее последовательности, соответствующей некоторой логической структуре программы. Каждый программный модуль выполняет отдельную самостоятельную задачу. В то же время, сохраняется зависимость между модулями, обусловленная необходимостью использования общих данных и результатов их обработки разными модулями ПМ РПИТ.

ПМ РПИТ включает в себя следующие модули: модуль расчета оптимального (кратчайшего) пути на индивидуальном транспорте, модуль расчета надежного пути на индивидуальном транспорте.

Кроме того, ПМ РПИТ имеет зависимости с программным модулем, реализующим алгоритм краткосрочного прогнозирования параметров транспортных потоков на основе

технологии «больших данных» с использованием данных реального времени и архивных данных о состоянии транспортных потоков (ПМ КППП). ПМ КППП используется для учета информации о прогнозном распределении транспортных потоков при решении задачи нахождения оптимального пути на индивидуальном транспорте.

Модуль расчета оптимального пути на индивидуальном транспорте предназначен для нахождения одного или нескольких маршрутов движения на индивидуальном транспорте между указанными точками отправления-назначения с учетом информации о прогнозном распределении транспортных потоков. Критерием оптимальности кратчайшего пути является минимизация времени движения. Критерием оптимальности надежного пути является максимизация вероятности прибытия в пункт назначения в течение определенного интервала времени (бюджета поездки).

Разработанные модули объединены в составе веб-модуля, предназначенного для развёртывания ПМ РПИТ в контейнере сервлетов Apache Tomcat и использования в качестве Web-сервиса в части предоставления участнику движения результатов работы программного модуля (оптимальных путей движения на индивидуальном транспорте).

Блок схема взаимодействия ПМ РПИТ с другими программными модулями представлена на рисунке 170.

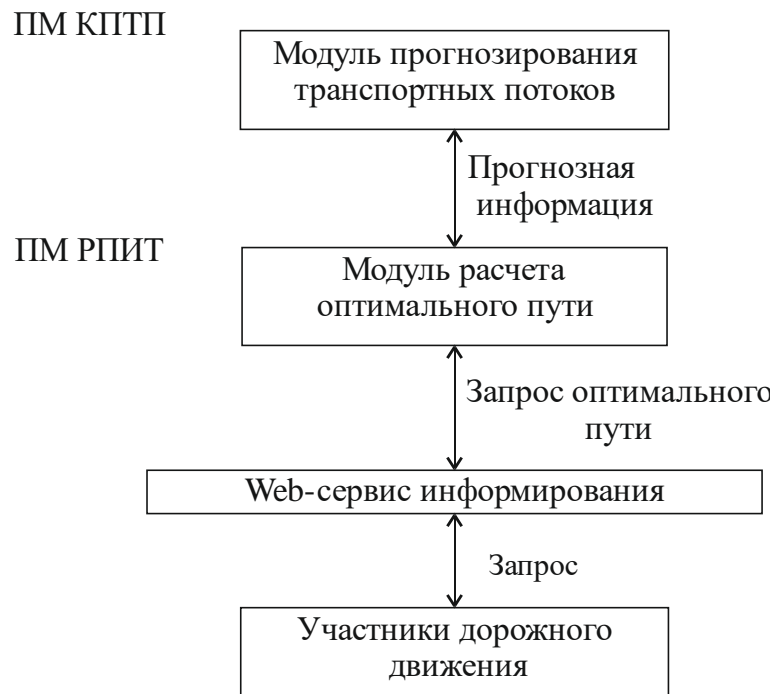


Рисунок 170 – Диаграмма взаимодействия

Процесс взаимодействия программных компонентов состоит из следующих этапов:

- модуль прогнозирования транспортных потоков в отдельном процессе осуществляет краткосрочное прогнозирование параметров потоков на основе технологии «больших данных» и методов машинного обучения с использованием данных реального времени и архивных данных о состоянии транспортных потоков (часть ПМ КППП);

- модуль расчета оптимального пути реализует алгоритм расчета оптимального (в том числе - надежного) пути на индивидуальном транспорте в стохастической сети с учетом информации о прогнозном распределении транспортных потоков. Расчет происходит при получении запроса от участника дорожного движения через Web-сервис информирования. Модуль расчета оптимального пути взаимодействует с модулем прогнозирования транспортных потоков для получения прогнозной информации о времени прохождения дорожных сегментов;

- результат обработки запроса возвращается участнику дорожного движения посредством Web-сервиса информирования с помощью разработанного программного интерфейса (API).

5.6.2 Логическая модель программного модуля

UML-диаграмма основных классов, реализующих выбор вариантов поиска оптимального (кратчайшего или надежного) пути на индивидуальном транспорте, показана на рисунке 171.

Запрос нахождения оптимального пути от участника движения через Web-сервис информирования передается для обработки в объект класса `StartPersonalTransportJourneyHandler`. Далее, в зависимости от типа запрашиваемого пути, выполняется нахождение кратчайшего пути с помощью класса `GraphHopperProcessor` или надежного пути с использованием класса `ReliablePathProcessor`.

Вычисление кратчайшего (по расстоянию или по времени) пути выполняется в классе `GraphHopperTOS`, который осуществляет обновление весов графа (времени прохождения сегментов дорожной сети), используя прогнозную информацию о распределении транспортных потоков, и нахождение одного или нескольких путей.

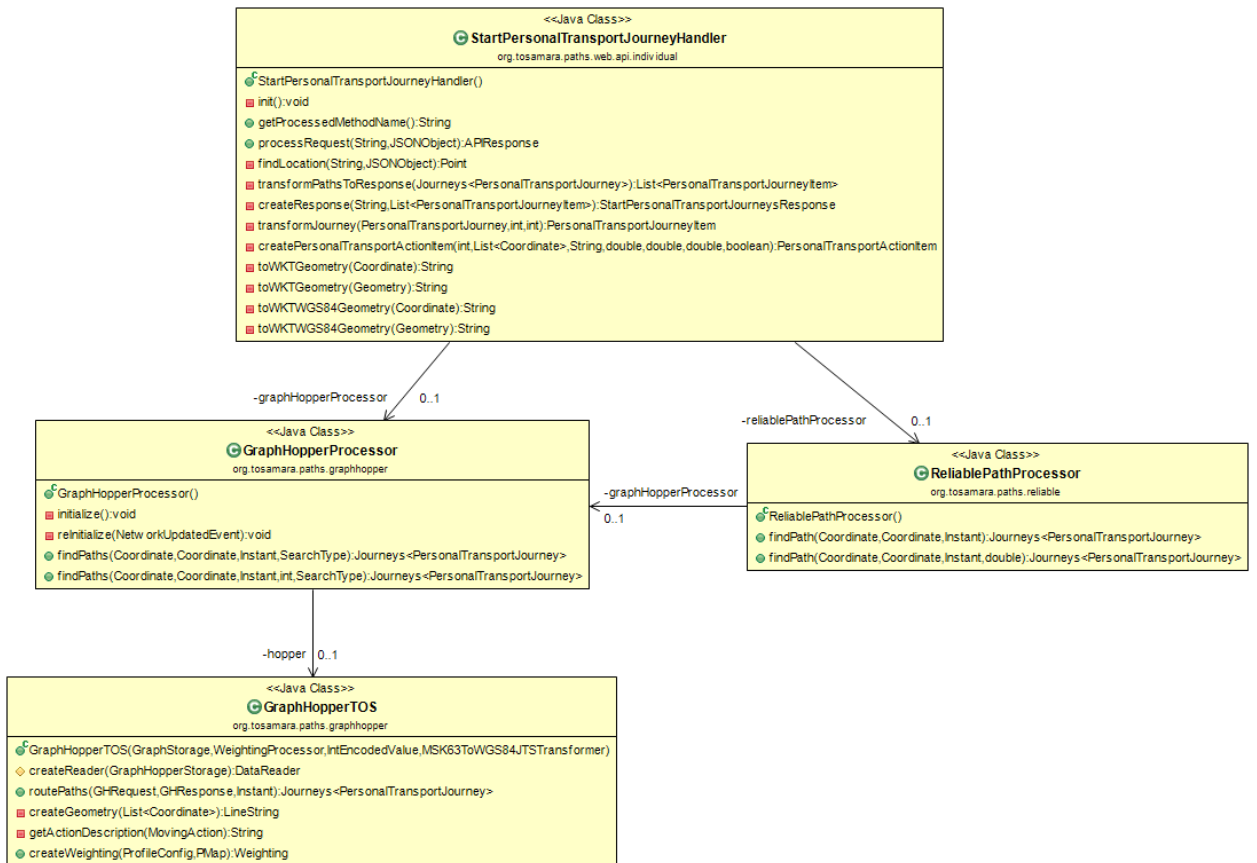


Рисунок 171 – UML-диаграмма выбора типа оптимального пути

Вычисление оптимального надежного пути выполняется в классе `ReliablePathProcessor` и связанных с ним классах. UML-диаграмма основных классов, реализующих поиск оптимального надежного пути на индивидуальном транспорте, показана на рисунке 172.

Назначение основных классов программного модуля:

- класс `ReliablePathProcessor` является точкой входа для обработки запроса нахождения оптимального надежного пути;
- класс `SOTARouteFinderLevy` реализует алгоритм нахождения надежного кратчайшего пути с использованием устойчивых распределений Леви;
- класс `ReachingNodeProbabilityLevy` хранит вероятности достижения конечной вершины (пункта назначения) из текущей вершины за оставшийся бюджет поездки;
- класс `LevyProbabilityCalculator` рассчитывает вероятность достижения конечной вершины из текущей за оставшийся бюджет поездки, используя устойчивые распределения Леви;

- класс `LevyApproximator` оценивает параметры распределения Леви для прохождения оставшегося пути.

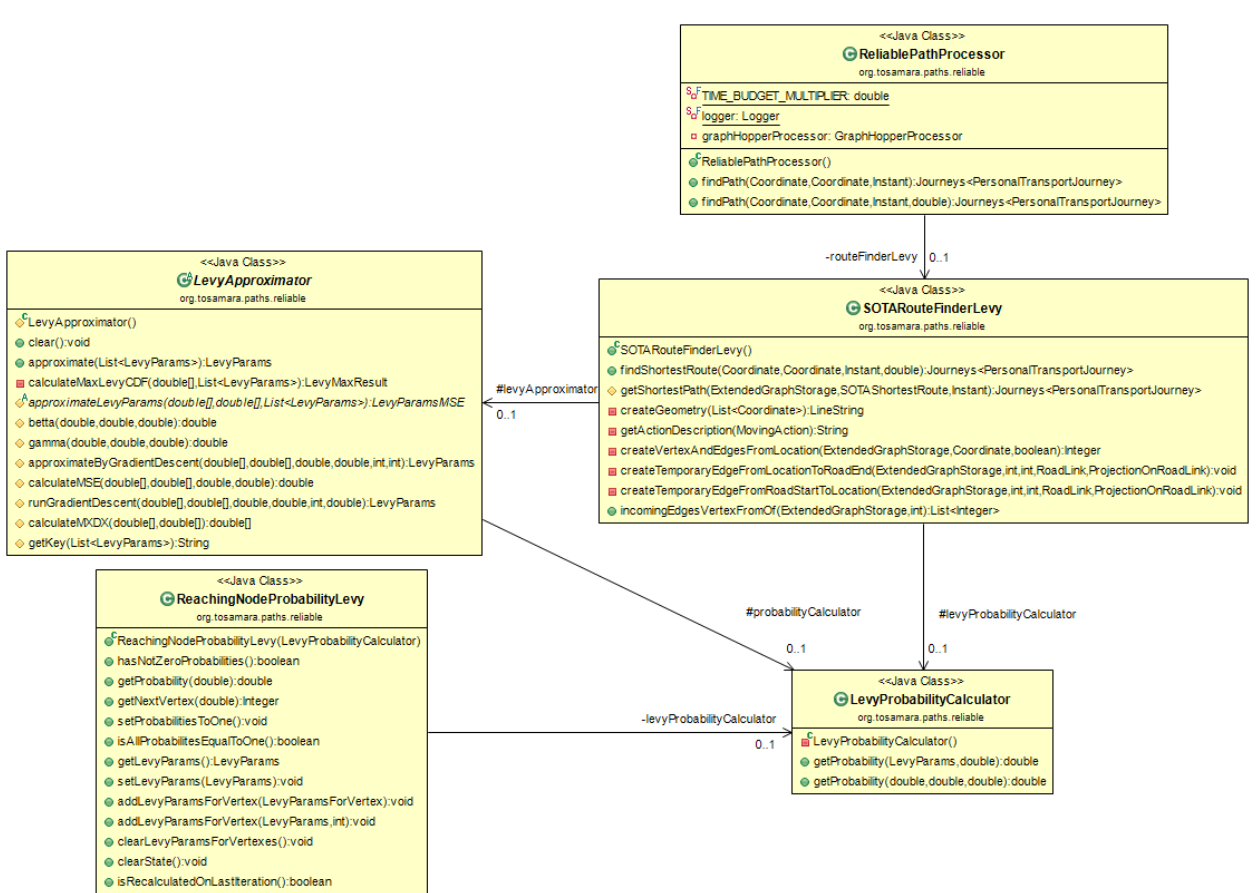


Рисунок 172 – UML-диаграмма классов для поиска надежного пути на индивидуальном транспорте

Согласно разработанному алгоритму (раздел 4.2.3), расчет параметров функции распределения Леви является наиболее вычислительно сложным этапом работы алгоритма и выполняется с использованием методов аппроксимации, либо путем предварительного вычисления аппроксимаций. UML-диаграмма классов, вычисляющих значения функции распределения Леви, представлена на рисунке 173.

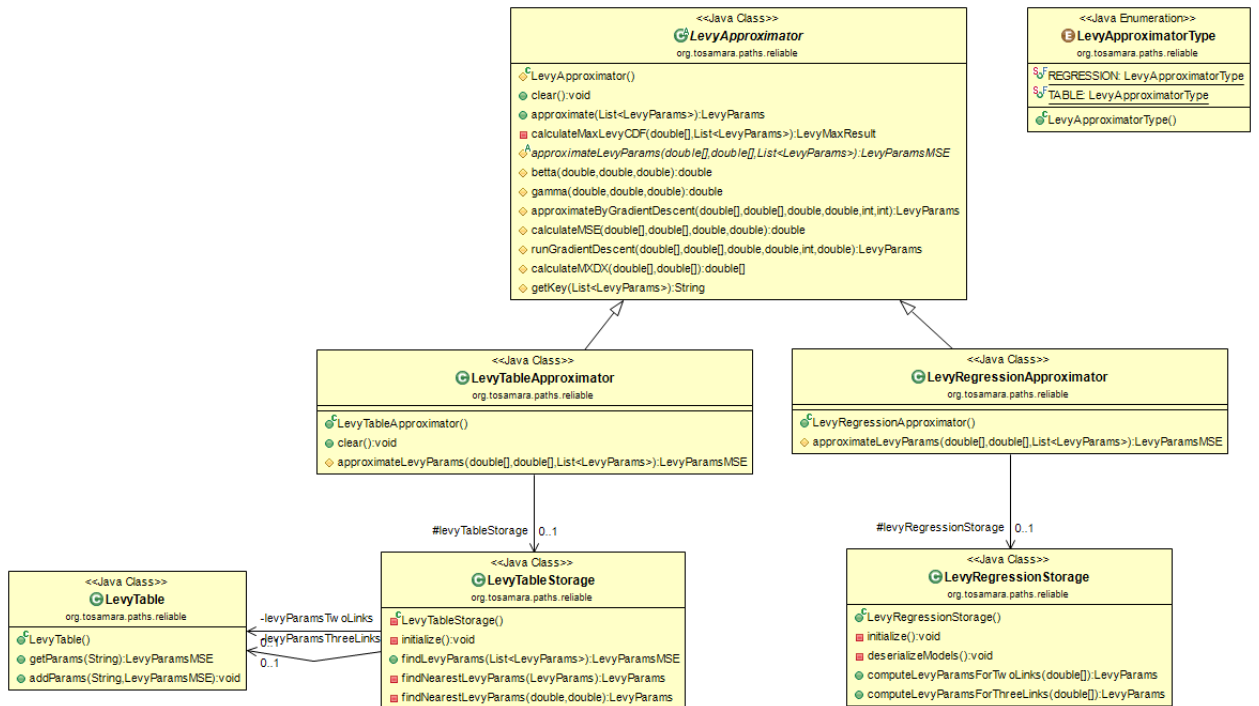


Рисунок 173 – UML-диаграмма классов, вычисляющих параметры функции распределения Леви

Назначение основных классов программного модуля:

- класс `LevyTableApproximator` рассчитывает и хранит в оперативной памяти предварительно рассчитанные параметры функции распределения Леви в виде таблицы, используя классы `LevyTableStorage` и `LevyTable`;
- класс `LevyRegressionApproximator` хранит регрессионные коэффициенты для вычисления итоговых параметров функции распределения Леви, используя класс `LevyRegressionStorage`, и рассчитывает параметры функции распределения.

5.6.3 Физическая модель данных программного модуля

Данные, необходимые для работы программного модуля, хранятся в SQL базе данных. В качестве СУБД была выбрана PostgreSQL.

Для формирования графа дорожно-транспортной сети и расчета оптимального пути на индивидуальном транспорте используются данные о дорожной сети, хранящиеся в таблице `ROAD_links`, показанной на рисунке 151.

5.6.4 Программный интерфейс модуля

Расчет оптимального пути на индивидуальном транспорте в стохастической сети с учетом информации о прогнозном распределении транспортных потоков доступен

посредством Web-сервиса с использованием разработанного программного интерфейса (API).

Web-сервис доступен по следующему относительному http-адресу: /api/v1/json.

Результатом выполнения запросов к API является json-документ разработанной структуры.

5.6.4.1 Кратчайший путь

Запрос кратчайшего пути на индивидуальном транспорте, минимизирующего общее время движения, а также альтернативных путей движения, имеет следующий вид:

```
{
  "method": "startPersonalTransportJourneySupport",
  "deviceid": "eqifo7414w",
  "startLocation": "POINT(50.104664862155914 53.19256170356393)",
  "endLocation": "POINT(50.18998056650162 53.20027453918263)",
  "tunedPreferences": {
    "usePIP": "true",
    "variantsCountLimit": 4,
    "showMultipleDirections": "true",
    "useTrafficJamsForecast": true,
    "criterion": "TIME"
  }
}
```

В запросе указываются точки отправления и прибытия, а также критерий оптимальности маршрута (в данном случае, "criterion"="TIME").

Пример запроса оптимального пути на индивидуальном транспорте и ответа сервера показан на рисунках 174 и 175 соответственно.

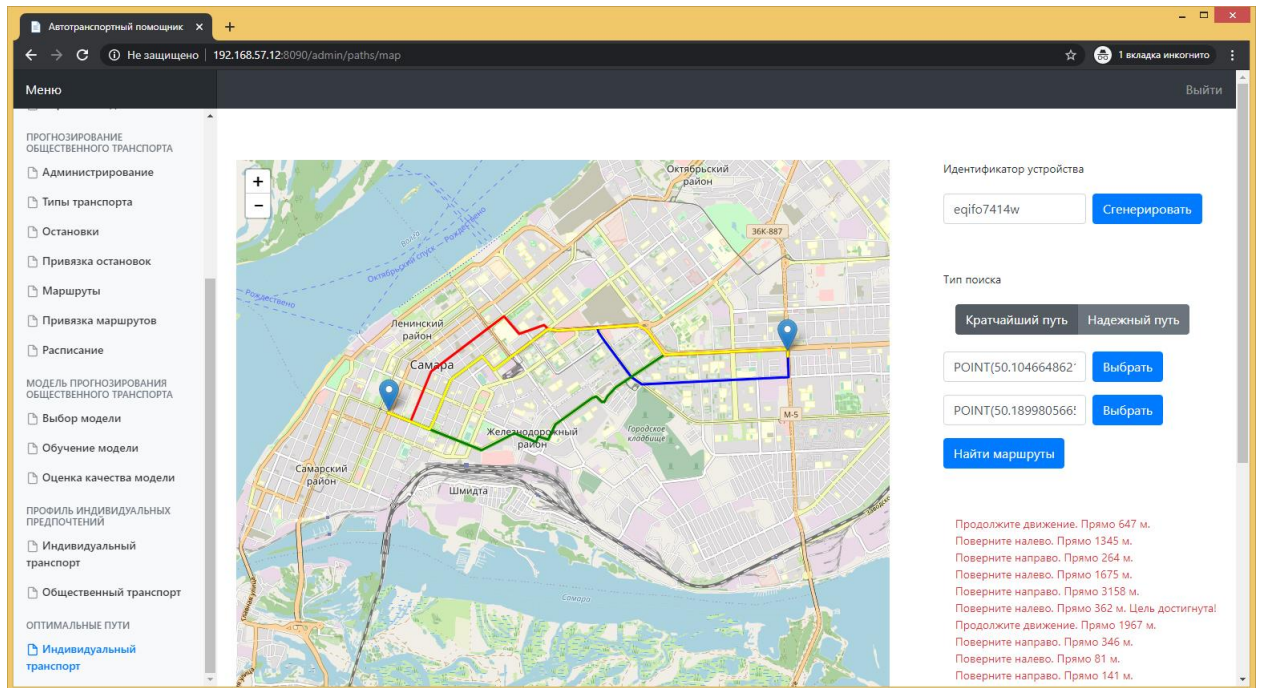


Рисунок 176 – Результат нахождения кратчайшего пути на индивидуальном транспорте

5.6.4.2 Надежный путь

Запрос надежного пути на индивидуальном транспорте, минимизирующего вероятность прибытия в пункт назначения в течение определенного промежутка времени (бюджета поездки), имеет следующий вид:

```
{
  "method": "startPersonalTransportJourneySupport",
  "deviceid": "eqifo7414w",
  "startLocation": "POINT(50.104664862155914 53.19256170356393)",
  "endLocation": "POINT(50.18998056650162 53.20027453918263)",
  "tunedPreferences": {
    "usePIP": "true",
    "variantsCountLimit": 4,
    "showMultipleDirections": "true",
    "useTrafficJamsForecast": true,
    "timeBudget": 1800,
    "criterion": "PROBABILITY "
  }
}
```

В запросе указываются точки отправления и прибытия, бюджет поездки, а также критерий оптимальности маршрута (в данном случае, "criterion"="PROBABILITY").

Пример запроса надежного пути на индивидуальном транспорте и ответа сервера показан на рисунках 177 и 178 соответственно.

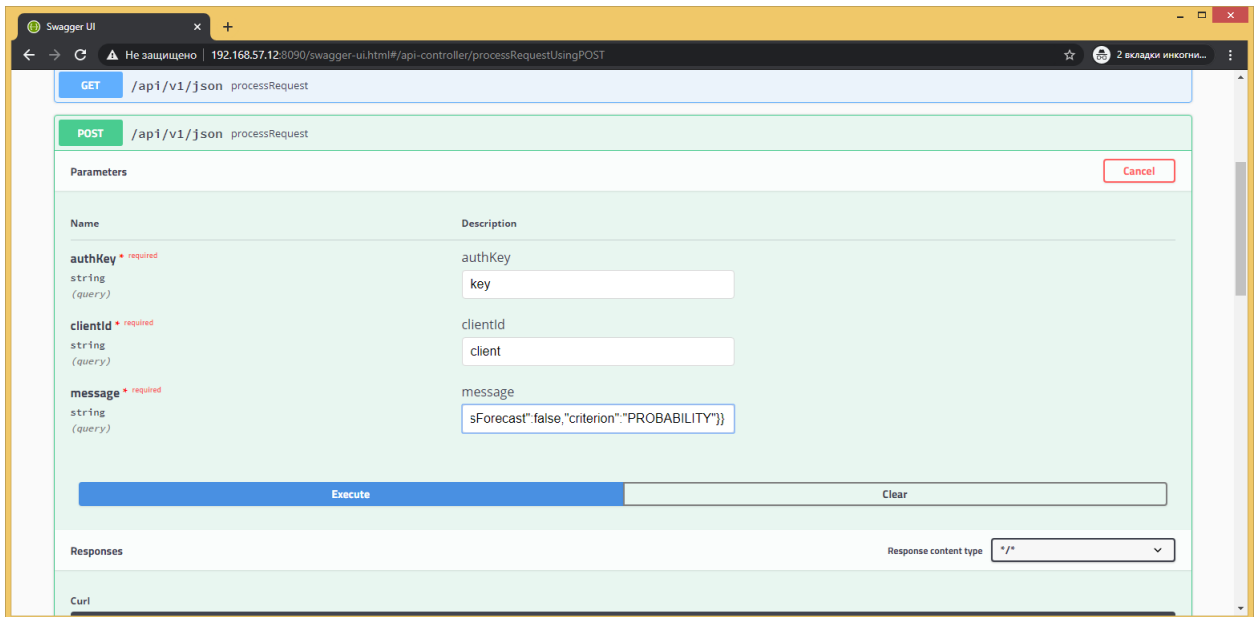


Рисунок 177 – Запрос надежного пути на индивидуальном транспорте

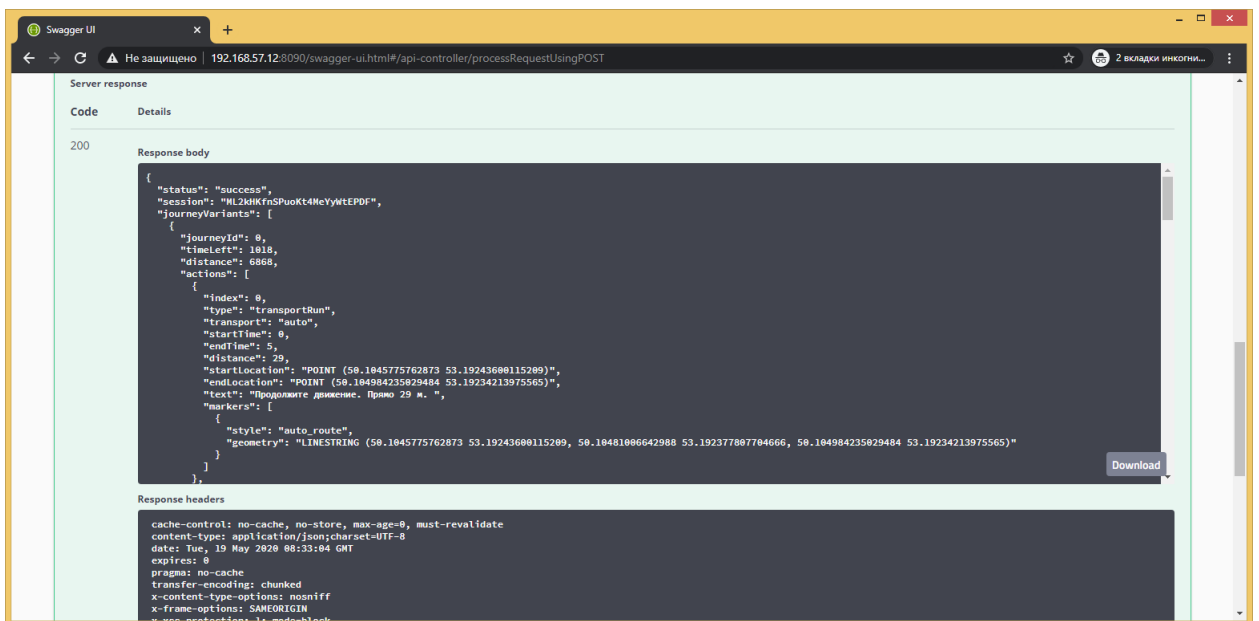


Рисунок 178 – Ответ сервера

Результат работы модуля нахождения надежного пути на индивидуальном транспорте можно отобразить на карте (рисунок 179).

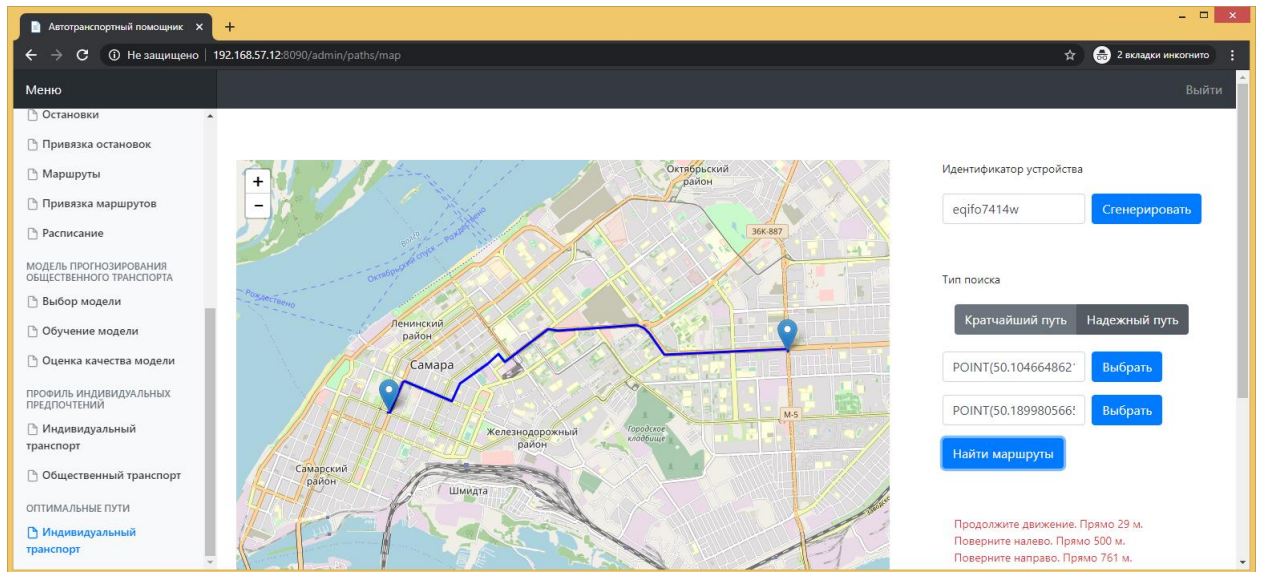


Рисунок 179 – Результат нахождения надежного пути на индивидуальном транспорте

В процессе работы web-сервиса предоставления участнику движения результатов работы программного модуля расчета оптимального (в том числе – надежного) пути на индивидуальном транспорте формируются результаты, описывающие оптимальные пути движения в формате JSON.

5.7 Выводы и результаты пятого раздела

В рамках данного раздела получены следующие результаты:

- 1) разработана архитектура и реализован программный комплекс кооперативной интеллектуальной транспортной системы с использованием подходов к обработке больших данных и методов машинного обучения, позволяющий объединить потоковую обработку данных в режиме реального времени с результатами пакетной аналитики (раздел 5.2);
- 2) в составе программного комплекса разработан и реализован программный модуль краткосрочного прогнозирования параметров транспортных потоков на основе технологии «больших данных» с использованием данных реального времени и архивных данных о состоянии транспортных потоков и методов машинного обучения (раздел 5.3);
- 3) в составе программного комплекса разработан и реализован программный модуль адаптивного управления транспортным потоком путем светофорного регулирования с использованием как наблюдаемых, так и прогнозных параметров, описывающих состояние транспортного потока, а также данных от подключенных и автономных подключенных транспортных средств (раздел 5.4);

4) в составе программного комплекса разработан и реализован программный модуль расчета оптимального пути на общественном транспорте в стохастической сети с учетом прогнозной информации о времени прибытия транспортных средств (раздел 5.5);

5) в составе программного комплекса разработан и реализован программный модуль расчета оптимального (в том числе – надежного) пути на индивидуальном транспорте в стохастической сети с учетом информации о прогнозном распределении транспортных потоков (раздел 5.6);

6) разработанный программный комплекс позволил получить результаты исследований математических методов и алгоритмов решения задач анализа, прогнозирования и управления транспортным потоком с гетерогенным составом транспортных средств, а также движения отдельных транспортных средств.

По материалам данного раздела опубликованы работы [193, 419-427].

ЗАКЛЮЧЕНИЕ

В диссертационной работе разработаны методы и алгоритмы обработки гетерогенной информации и адаптивного управления в интеллектуальной транспортной системе, позволяющие повысить эффективности использования транспортной инфраструктуры путем директивного и косвенного управления транспортным потоком с гетерогенным составом транспортных средств, а также движением отдельных транспортных средств. В работе были получены следующие основные результаты:

1) проведена формализация задачи директивного управления транспортным потоком с гетерогенным составом транспортных средств, определены информационные потоки и состав гетерогенной информации, используемой для решения задачи управления;

2) разработан комплекс алгоритмических средств (математический метод и алгоритмы) решения задачи краткосрочного прогнозирования параметров транспортного потока в транспортной сети с использованием графовых сверточных нейронных сетей и подходов к обработке больших данных на основе гетерогенной информации о состоянии транспортного потока и движении отдельных транспортных средств, позволяющий снизить ошибку прогнозирования транспортного потока до 6% по сравнению с современными (state-of-the-art) решениями задачи прогнозирования на открытых наборах данных;

3) разработан комплекс алгоритмических средств решения задачи директивного управления транспортным потоком, включая методы независимого и/или координированного адаптивного управления сигналами светофоров и движением подключенных автономных транспортных средств в транспортной сети. Экспериментальные исследования в среде моделирования показывают, что предложенные методы позволяют снизить средний расход топлива (до 4,2%), среднее время движения (до 5.3%) и среднее время ожидания (до 27%) по сравнению с современными методами решения задачи адаптивного управления светофорными объектами;

4) разработан комплекс алгоритмических средств решения задачи косвенного управления транспортным потоком и информирования в интеллектуальной транспортной системе:

- алгоритм определения надёжного пути в зависящей от времени стохастической (транспортной) сети, учитывающий информацию о пространственной и временной корреляции сегментов дорожной сети, текущую и прогнозную информацию о состоянии транспортного потока; ускоренная модификация алгоритма с использованием

распределения Леви, позволяющая повысить скорость вычислений в среднем в 40 раз при ухудшении качественных показателей в среднем на 9%;

- алгоритмы краткосрочного прогнозирования времени движения отдельных транспортных средств, учитывающие разнородную информацию о транспортной ситуации, прямо или косвенно влияющую на прогнозируемое время движения;

- алгоритм маршрутизации подключенных транспортных средств в транспортной сети на основе численного метода резервирования маршрутов, учитывающий стохастические свойства транспортной сети; модификация алгоритма для применения в гетерогенном транспортном потоке;

5) разработан и реализован программный комплекс кооперативной интеллектуальной транспортной системы, решающей задачи анализа, прогнозирования и управления транспортным потоком с гетерогенным составом транспортных средств с использованием подходов к обработке больших данных;

б) проведены экспериментальные исследования разработанных алгоритмических средств на натуральных и модельных данных, подтвердившие эффективность разработанных методов, алгоритмов и реализованного программного комплекса.

СПИСОК СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

Основные сокращения

CAV	– Автономное транспортное средство (англ.: Connected Autonomous Vehicle)
CV	– Подключенное транспортное средство (англ.: Connected Vehicle)
RL	– Обучение с подкреплением (англ.: Reinforcement Learning)
ГНС	– Глубокая нейронная сеть
ИНС	– Искусственная нейронная сеть
ИТС	– Интеллектуальная транспортная система
ОТС	– Общественное транспортное средство
ТС	– Транспортное средство
УДС	– Улично-дорожная сеть

Основные обозначения

\emptyset	– Пустое множество или отсутствие значений
N	– Множество натуральных чисел

СПИСОК ЛИТЕРАТУРЫ

1. Transport sector economic analysis // An official website of the European Union: [сайт]. – URL: https://joint-research-centre.ec.europa.eu/scientific-activities-z/transport-sector-economic-analysis_en (дата обращения 30.05.2022).
2. О производстве и использовании валового внутреннего продукта (ВВП) в 2021 году // Официальный сайт федеральной службы государственной статистики : [сайт]. – 2021. – URL: https://rosstat.gov.ru/storage/mediabank/59_08-04-2022.htm (дата обращения 30.05.2022).
3. Traffic congestion to cost the UK economy more than £300 billion over the next 16 years // Inrix: [сайт]. – 2014. – URL: <https://inrix.com/press-releases/traffic-congestion-to-cost-the-uk-economy-more-than-300-billion-over-the-next-16-years/> (дата обращения 12.04.2023).
4. The health impacts of weekday traffic: A health risk assessment of PM2.5 emissions during congested periods / W.J. Requia, C.D. Higgins, M.D. Adams [et al.] // Environment International. – February, 2018. – Vol. 111. – P. 164–176.
5. Zhang, K. Air pollution and health risks due to vehicle traffic / K. Zhang, S. Batterman // Science of The Total Environment. – April, 2013. – Vol. 450-451. – P. 307–316.
6. Stop-and-go traffic analysis: Theoretical properties, environmental impacts and oscillation mitigation / X. Li, J. Cui, S. An, M. Parsafard // Transportation Research Part B: Methodological. – December, 2014. – Vol. 70. – P. 319–339.
7. Наличие автомобильного транспорта // Единая межведомственная информационно-статистическая система (ЕМИСС): [сайт]. – 2022. – URL: <https://fedstat.ru/indicator/36228> (дата обращения 30.05.2022).
8. ГОСТ Р 56829-2015. Интеллектуальные транспортные системы. Термины и определения = Intelligent transport systems. Terms and definitions.: национальный стандарт Российской Федерации : издание официальное : утвержден и введен в действие Приказом Федерального агентства по техническому регулированию и метрологии : дата введения. 2016-06-01. – Москва: Стандартинформ, 2018. – 14 с.
9. Якимов, М.Р. Транспортное планирование: создание транспортных моделей городов / М.Р. Якимов. – Москва: Логос, 2013. – 188 с.
10. Российская Федерация. Правительство. О Транспортной стратегии Российской Федерации: Распоряжение Правительства РФ от 22.11.2008 № 1734-р / Российская Федерация. Правительство.
11. Российская Федерация. Правительство. Об утверждении Концепции обеспечения безопасности дорожного движения с участием беспилотных транспортных средств на автомобильных дорогах общего пользования: Распоряжение Правительства РФ от 25.03.2020 N 724-р / Российская Федерация. Правительство.

12. On-Road Automated Driving (ORAD) committee. Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles //SAE International: [сайт]. – 2021. – URL: https://www.sae.org/content/j3016_202104 (дата обращения 14.05.2023).
13. Automotive Autonomous Vehicles Market and Trend Analysis by Technology, Key Companies and Forecast, 2021-2036 // GlobalData : [сайт]. – 2023. – URL: <https://www.globaldata.com/store/report/automotive-autonomous-vehicles-market-analysis/> (дата обращения 14.05.2023).
14. Российская Федерация. Правительство. Об утверждении Стратегии развития автомобильной промышленности Российской Федерации на период до 2025 года: Распоряжение Правительства РФ от 28.04.2018 № 831-р / Российская Федерация. Правительство.– 2018.
15. A European strategy on Cooperative Intelligent Transport Systems, a milestone towards cooperative, connected and automated mobility // An official website of the European Union: [сайт]. – 2023. – URL: <https://eur-lex.europa.eu/homepage.html> (дата обращения 02.05.2023).
16. 2020 Autonomous Vehicle Technology Report// Wevolver: [сайт]. – 2023. – URL: <https://www.wevolver.com/article/2020.autonomous.vehicle.technology.report> (дата обращения 2.05.2023).
17. Autonomous Drive // Volvo: [сайт]. – 2023. – URL: <https://group.volvocars.com/company/innovation/autonomous-drive> (дата обращения 02.05.2023).
18. Autopilot // Tesla: [сайт] . – 2021. – URL: <https://www.tesla.com/autopilot> (дата обращения 1.04.2021).
19. Yandex Self-Driving Cars // Yandex: [сайт]. – 2021. – URL: <https://sdc.yandex.com> (дата обращения 01.04.2021).
20. Bundin, M. Legal framework for self-driving cars: the case of Russia / М. Bundin, А. Martynov, F. Romyantsev // Proceedings of the 13th International Conference on Theory and Practice of Electronic Governance. Athens Greece. – September, 2020. – P. 206–213.
21. НТИ “Автонет” провел успешные испытания V2X // НТИ Автонет: [сайт] . – 2021. – URL: <https://autonet-nti.ru/media/news/nti-avtonet-provel-uspeshnye-ispytaniya-v2x/> (дата обращения 02.04.2021).
22. V2I based cooperation between traffic signal and approaching automated vehicles/ B. Xu, X.J. Van, Y. Bian [et al.] // 2017 IEEE Intelligent Vehicles Symposium (IV) . – Los Angeles: CA, USA, June, 2017. – P. 1658–1664.
23. Integrated optimization of traffic signals and vehicle trajectories at isolated urban intersections/ C. Yu, Y. Feng, H.X. Liu [et al.] // Transportation Research Part B: Methodological. – June , 2018. – Vol. 112. – P. 89–112.

24. Applications of Artificial Intelligence in Transport: An Overview/ R. Abduljabbar, H. Dia, S. Liyanage, S.A. Bagloee // Sustainability. – January, 2019. – Vol. 11. – P. 189.
25. Chajka-Cadin, L. Intelligent Transportation Systems Deployment: Findings from the 2019 Connected Vehicle and Automated Vehicle Survey/ L. Chajka-Cadin, M. Petrella, S. Plotnick // United States. Department of Transportation. Intelligent Transportation Systems Joint Program Office (ITS JPO) . – 2020.
26. Applications of Artificial Intelligence and Machine learning in smart cities / Z. Ullah, F. Al-Turjman, L. Mostarda, R. Gagliardi // Computer Communications. – March, 2020. – Vol. 154. – P. 313–323.
27. Traffic Flow Prediction With Big Data: A Deep Learning Approach / Y. Lv, Y. Duan, W. Kang [et al.] // IEEE Transactions on Intelligent Transportation Systems. – 2014 . – P. 1–9.
28. Network-wide traffic signal control based on the discovery of critical nodes and deep reinforcement learning/ M. Xu, J. Wu, L. Huang [et al.] // Journal of Intelligent Transportation Systems. – January, 2020. – Vol. 24. – P. 1–10.
29. Multi-output bus travel time prediction with convolutional LSTM neural network/ N.C. Petersen, F. Rodrigues, F.C. Pereira [et al.] // Expert Systems with Applications. – April, 2019. – Vol. 120. – P. 426–435.
30. The AI Index 2021 Annual Report/ Zhang D., Mishra S., Brynjolfsson E. [et al.]. – arXiv, 2021. – arXiv:2103.06312 [cs].
31. Dimensions: [сайт]. – 2023. – URL: <https://www.dimensions.ai/> (дата обращения: 2.05.2023).
32. Введение в математическое моделирование транспортных потоков/ А.В. Гасников, С.Л. Кленов, Е.А. Нурминский [и др.]. – Москва: МФТИ, 2010. – 362 с.
33. Гасников, А.В. Модели равновесного распределения транспортных потоков в больших сетях/ А.В. Гасников, Е.В. Гасникова. – М.: URSS, 2023. – 240 с.
34. Finding equilibrium in two-stage traffic assignment model/ E.V. Kotliarova, A.V. Gasnikov, E.V. Gasnikova, D.V. Yarmoshik // Computer Research and Modeling. – April 2021. – Vol. 13. – P. 365–379.
35. Shvetsov, V.I. Algorithms for distributing traffic flows // Automation and Remote Control. – October, 2009. – Vol. 70. – P. 1728–1736.
36. Старожилец, В.М. Об одном подходе к статистическому моделированию транспортных потоков / В.М. Старожилец, Ю.В. Чехович // Журнал Вычислительной Математики И Математической Физики. – 2021. – Т. 61. – № 7. – С. 1220-1232.
37. Швецов, В.И. Математическое моделирование транспортных потоков // Автоматика и телемеханика. – 2003. – Т. 11. – С. 3-46.

38. Старожилец, В.М. Об одном подходе к статистическому моделированию транспортных потоков на МКАД и управлению въездами / В.М. Старожилец, Ю.В. Чехович // Автоматика и телемеханика. – Ноябрь, 2021. – С. 114–134.
39. Generalization Second Order Macroscopic Traffic Models via Relative Velocity of the Congestion Propagation / Y. Kholodov, A. Alekseenko, V. Kazorin, A. Kurzhanskiy // Mathematics. – August, 2021. – Vol. 9. – P. 2001.
40. Vlahogianni, E.I. Short-term traffic forecasting: Overview of objectives and methods / E.I. Vlahogianni, J.C. Golias, M.G. Karlaftis // Transport Reviews. – September, 2004. – Vol. 24. – P. 533–557.
41. Road Traffic Forecasting: Recent Advances and New Challenges / I. Lana, J. Del Ser, M. Velez, E.I. Vlahogianni // IEEE Intelligent Transportation Systems Magazine. – 2018. – Vol. 10. – P. 93–109.
42. Vlahogianni, E.I. Short-term traffic forecasting: Where we are and where we're going / E.I. Vlahogianni, M.G. Karlaftis, J.C. Golias // Transportation Research Part C: Emerging Technologies. – June, 2014. – Vol. 43. – P. 3–19.
43. Miglani, A. Deep learning models for traffic flow prediction in autonomous vehicles: A review, solutions, and challenges / A. Miglani, N. Kumar // Vehicular Communications. – December, 2019. – Vol. 20. – P. 100184.
44. Deep Learning on Traffic Prediction: Methods, Analysis, and Future Directions / X. Yin, G. Wu, J. Wei [et al.] // IEEE Transactions on Intelligent Transportation Systems. – June, 2022. – Vol. 23. – P. 4927–4943.
45. Karlaftis, M.G. Statistical methods versus neural networks in transportation research: Differences, similarities and some insights / M.G. Karlaftis, E.I. Vlahogianni // Transportation Research Part C: Emerging Technologies. – June, 2011. – Vol. 19. – P. 387–399.
46. Ahmed, M.S. Analysis of Freeway Traffic Time-Series Data by Using Box-Jenkins Techniques/M.S. Ahmed, A.R. Cook // Transportation Research Record. – 1979. – P. 1–9.
47. Shekhar, S. Adaptive Seasonal Time Series Models for Forecasting Short-Term Traffic Flow / S. Shekhar, B.M. Williams // Transportation Research Record: Journal of the Transportation Research Board. – January, 2007. – Vol. 2024. – P. 116–125.
48. Williams, B.M. Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results / B.M. Williams, L.A. Hoel // Journal of Transportation Engineering. – January, 2007. – Vol. 129. – P. 664–672.
49. Kumar, S.V. Short-term traffic flow prediction using seasonal ARIMA model with limited input data / S.V. Kumar, L. Vanajakshi // European Transport Research Review. – 2015. – Vol. 7.

50. Chandra, S.R. Predictions of Freeway Traffic Speeds and Volumes Using Vector Autoregressive Models / S.R. Chandra, H. Al-Deek // *Journal of Intelligent Transportation Systems*. – May, 2009. – Vol. 13. – P. 53–72.
51. Guo, J. Real-Time Short-Term Traffic Speed Level Forecasting and Uncertainty Quantification Using Layered Kalman Filters / J. Guo, B.M. Williams // *Transportation Research Record: Journal of the Transportation Research Board*. – January, 2010. – Vol. 2175. – P. 28–37.
52. Wang, Y. Real-time freeway traffic state estimation based on extended Kalman filter: a general approach / Y. Wang, M. Papageorgiou // *Transportation Research Part B: Methodological*. – February, 2005. – Vol. 39. – P. 141–167.
53. Kumar S.V. Traffic Flow Prediction using Kalman Filtering Technique 2017. – Vol. 187. – P. 582–587.
54. Wang, Y. Real-time freeway traffic state estimation based on extended Kalman filter: Adaptive capabilities and real data testing / Y. Wang, M. Papageorgiou, A. Messmer // *Transportation Research Part A: Policy and Practice*. – 2008. – Vol. 42. – P. 1340–1358.
55. Short-term traffic predictions on large urban traffic networks: Applications of network-based machine learning models and dynamic traffic assignment models / G. Fusco, C. Colombaroni, L. Comelli, N. Isaenko // *2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*. – Budapest: Hungary, June, 2015. – P. 93–101.
56. Smith, B.L. Comparison of parametric and nonparametric models for traffic flow forecasting / B.L. Smith, B.M. Williams, R.K. Oswald // *Transportation Research Part C: Emerging Technologies*. – 2002. – Vol. 10. – P. 303–321.
57. Urban traffic flow prediction using a fuzzy-neural approach / H. Yin, S.C. Wong, J. Xu, C.K. Wong // *Transportation Research Part C: Emerging Technologies*. – April, 2002. – Vol. 10. – P. 85–98.
58. Joint Feature Selection and Parameter Tuning for Short-Term Traffic Flow Forecasting Based on Heuristically Optimized Multi-layer Neural Networks / I. Laña, J. Del Ser, M. Vélez, I. Oregi // *In: Harmony Search Algorithm*. Singapore: Springer Singapore. – 2017. – Vol. 514. – P. 91–100.
59. Neural-network-based models for short-term traffic flow forecasting using a hybrid exponential smoothing and levenberg-marquardt algorithm / K.Y. Chan, T.S. Dillon, J. Singh, E. Chang // *IEEE Transactions on Intelligent Transportation Systems*. – 2012. – Vol. 13. – P. 644–654.
60. Kumar, K. Short term traffic flow prediction in heterogeneous condition using artificial neural network/ K. Kumar, M. Parida, V.K. Katiyar // *Transport*. – 2015. – Vol. 30. – P. 397–405.

61. Zheng, Z. Short-term traffic volume forecasting: A k-nearest neighbor approach enhanced by constrained linearly sewing principle component algorithm / Z. Zheng, D. Su // *Transportation Research Part C: Emerging Technologies*. – June, 2014. – Vol. 43. – P. 143–157.
62. A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting / P. Cai, Y. Wang, G. Lu [et al.] // *Transportation Research Part C: Emerging Technologies*. – January, 2016. – Vol. 62. – P. 21–34.
63. Agafonov, A. Short-Term Traffic Flow Forecasting Using a Distributed Spatial-Temporal k Nearest Neighbors Model / A. Agafonov, A. Yumaganov // *2018 IEEE International Conference on Computational Science and Engineering (CSE)* . – Bucharest: Romania, 2018. – P. 91–98.
64. Агафонов, А.А. Анализ больших данных в геоинформационной задаче краткосрочного прогнозирования параметров транспортного потока на базе метода к ближайших соседей / А.А. Агафонов, А.С. Юмаганов, В.В. Мясников // *Компьютерная оптика*. – 2018. – Т. 42. – С. 1101–1111.
65. Wu, C.H. Travel-time prediction with support vector regression / C.H. Wu, J.M. Ho, D.T. Lee // *IEEE Transactions on Intelligent Transportation Systems*. –2004. – Vol. 5. – P. 276–281.
66. Su, H. Short-term Traffic Flow Prediction Based on Incremental Support Vector Regression / H. Su, L. Zhang, S. Yu // *Third International Conference on Natural Computation (ICNC 2007)* . – Haikou: China, 2007. – P. 640–645.
67. Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions / M. Castro-Neto, Y.S. Jeong, M.K. Jeong, L.D. Han // *Expert Systems with Applications*. – 2009. – Vol. 36. – P. 6164–6173.
68. Fei, X. A bayesian dynamic linear model approach for real-time short-term freeway travel time prediction / X. Fei, C.C. Lu, K. Liu // *Transportation Research Part C: Emerging Technologies*. – December, 2011. – Vol. 19. – P. 1306–1318.
69. Short-term traffic flow prediction with linear conditional Gaussian Bayesian network: Traffic Flow Prediction, Bayesian Network, Linear Conditional Gaussian / Z. Zhu, B. Peng, C. Xiong, L. Zhang // *Journal of Advanced Transportation*. – October, 2016. – Vol. 50. – P. 1111–1123.
70. Sun, S. The Selective Random Subspace Predictor for Traffic Flow Forecasting / S. Sun, C. Zhang // *IEEE Transactions on Intelligent Transportation Systems*. – June, 2007. – Vol. 8. – P. 367–373.
71. 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. – Т. 38. – № 3. – С. 356-369.
72. Zhang, N. Seasonal Autoregressive Integrated Moving Average and Support Vector Machine Models: Prediction of Short-Term Traffic Flow on Freeways / N. Zhang, Y. Zhang, H. Lu // *Transportation Research Record: Journal of the Transportation Research Board.* – January, 2011. – Vol. 2215. – P. 85–92.
 73. Urban traffic flow forecasting through statistical and neural network bagging ensemble hybrid modeling / F. Moretti, S. Pizzuti, S. Panzieri, M. Annunziato // *Neurocomputing.* – November, 2015. – Vol. 167. – P. 3–7.
 74. Different methods of traffic forecast based on real data / R. Chrobok, O. Kaumann, J. Wahle, M. Schreckenber // *European Journal of Operational Research.* – June, 2004. – Vol. 155. – P. 558–568.
 75. Lana, I. Understanding daily mobility patterns in urban road networks using traffic flow analytics / I. Lana, J. Del Ser, I.I. Olabarrieta // *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium.* – Istanbul: Turkey, April, 2016. – P. 1157–1162.
 76. Improved deep hybrid networks for urban traffic flow prediction using trajectory data / Z. Duan, Y. Yang, K. Zhang [et al.] // *IEEE Access.* – 2018. – Vol. 6. – P. 31820–31827.
 77. Fu, R. Using LSTM and GRU neural network methods for traffic flow prediction / R. Fu, Z. Zhang, L. Li // *Proceedings - 2016 31st Youth Academic Annual Conference of Chinese Association of Automation.* – YAC 2016. – 2017. – P. 324–328.
 78. LSTM-based traffic flow prediction with missing data / Y. Tian, K. Zhang, J. Li [et al.] // *Neurocomputing.* – 2018. – Vol. 318. – P. 297–305.
 79. A Hybrid Deep Learning Model With Attention-Based Conv-LSTM Networks for Short-Term Traffic Flow Prediction / H. Zheng, F. Lin, X. Feng, Y. Chen // *IEEE Transactions on Intelligent Transportation Systems.* – 2020. – P. 1–11.
 80. Attention meets long short-term memory: A deep learning network for traffic flow forecasting / W. Fang, W. Zhuo, J. Yan [et al.] // *Physica A: Statistical Mechanics and its Applications.* – 2022. – Vol. 587.
 81. Zhang, Z. A deep network with analogous self-attention for short-term traffic flow prediction / Z. Zhang, X. Jiao // *IET Intelligent Transport Systems.* – July 2021. – Vol. 15. – P. 902–915.
 82. An attention-based deep learning model for traffic flow prediction using spatiotemporal features towards sustainable smart city / B. Vijayalakshmi, K. Ramar, N.Z. Jhanjhi [et al.] // *International Journal of Communication Systems.* – 2021. – Vol. 34.
 83. A joint temporal-spatial ensemble model for short-term traffic prediction / G. Zheng, W.K. Chai, V. Katos, M. Walton // *Neurocomputing.* – October, 2021. – Vol. 457. – P. 26–39.

84. TrafficGAN: Network-Scale Deep Traffic Prediction with Generative Adversarial Nets / Y. Zhang, S. Wang, B. Chen [et al.] // *IEEE Transactions on Intelligent Transportation Systems*. – 2021. – Vol. 22. – P. 219–230.
85. Graph convolutional networks: a comprehensive review / S. Zhang, H. Tong, J. Xu, R. Maciejewski // *Computational Social Networks*. – 2019. – Vol. 6.
86. Spectral networks and deep locally connected networks on graphs / J. Bruna, W. Zaremba, A. Szlam, Y. LeCun. – 2014.
87. Prokoptsev, N.G. Traffic flow speed prediction on transportation graph with convolutional neural networks / N.G. Prokoptsev, A.E. Alekseenko, Y.A. Kholodov // *Computer Research and Modeling*. – June, 2018. – Vol. 10. – P. 359–367.
88. Gao, H. Large-scale learnable graph convolutional networks / H. Gao, Z. Wang, S. Ji. – 2018. – P. 1416–1424.
89. Kipf, T.N. Semi-supervised classification with graph convolutional networks / T.N. Kipf, M. Welling – 2019.
90. Defferrard, M. Convolutional neural networks on graphs with fast localized spectral filtering / M. Defferrard, X. Bresson, P. Vandergheynst . – 2016. – P. 3844–3852.
91. Yu, B. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting / B. Yu, H. Yin, Z. Zhu. – 2018. – Vol. 2018. – P. 3634–3640.
92. Traffic Graph Convolutional Recurrent Neural Network: A Deep Learning Framework for Network-Scale Traffic Learning and Forecasting / Z. Cui, K. Henrickson, R. Ke, Y. Wang // *IEEE Transactions on Intelligent Transportation Systems*. – 2019. – P. 1–12.
93. Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting / S. Guo, Y. Lin, N. Feng [et al.] // *Proceedings of the AAAI Conference on Artificial Intelligence*. – July, 2019. – Vol. 33. – P. 922–929.
94. Gu, Y. STAGCN: Spatial–Temporal Attention Graph Convolution Network for Traffic Forecasting / Y. Gu, L. Deng // *Mathematics*. – January 2022. – Vol. 10. – P. 1599.
95. Guo, G. Short-term traffic speed forecasting based on graph attention temporal convolutional networks / G. Guo, W. Yuan // *Neurocomputing*. – 2020. – Vol. 410. – P. 387–393.
96. Agafonov A.A. Short-Term Traffic Data Forecasting: A Deep Learning Approach // *Optical Memory and Neural Networks*. – January, 2021. – Vol. 30. – P. 1–10.
97. LSTM variants meet graph neural networks for road speed prediction / Z. Lu, W. Lv, Y. Cao [et al.] // *Neurocomputing*. – 2020. – Vol. 400. – P. 34–45.
98. A distributed spatial–temporal weighted model on MapReduce for short-term traffic flow forecasting / D. Xia, B. Wang, H. Li [et al.] // *Neurocomputing*. – February, 2016. – Vol. 179. – P. 246–263.

99. Claes, R. A Decentralized Approach for Anticipatory Vehicle Routing Using Delegate Multiagent Systems / R. Claes, T. Holvoet, D. Weyns // *IEEE Transactions on Intelligent Transportation Systems*. – June, 2011. – Vol. 12. – P. 364–373.
100. Agafonov, A. Anticipatory Vehicle Routing in Stochastic Networks Using Multi-Agent System / A. Agafonov, V. Myasnikov, A. Borodinov // *2019 XXI International Conference Complex Systems: Control and Modeling Problems (CSCMP)*. – September, 2019. – P. 91–95.
101. Generalized velocity–density model based on microscopic traffic simulation / O. Derbel, T. Péter, B. Mourllion, M. Basset // *Transport*. – April, 2017. – Vol. 33. – P. 489–501.
102. Logistic modeling of the equilibrium speed–density relationship / H. Wang, J. Li, Q.Y. Chen, D. Ni // *Transportation Research Part A: Policy and Practice*. – July, 2011. – Vol. 45. P. 554–566.
103. Qu, X. On the stochastic fundamental diagram for freeway traffic: Model development, analytical properties, validation, and extensive applications / X. Qu, J. Zhang, S. Wang // *Transportation Research Part B: Methodological*. – October, 2017. – Vol. 104. – P. 256–271.
104. Bayesian Localization for Autonomous Vehicle using Sensor Fusion and Traffic Signs / S. Verentsov, E. Magerramov, V. Vinogradov [et al.] // *Proceedings of the 2017 International Conference on Robotics and Artificial Intelligence*. – Shanghai China, December, 2017. – P. 71–74.
105. Suh, B. Vehicle Speed Prediction for Connected and Autonomous Vehicles Using Communication and Perception / B. Suh, Y. Shao, Z. Sun // *2020 American Control Conference (ACC)*. – Denver: CO, USA, July, 2020. – P. 448–453.
106. Shao, Y. Eco-Approach With Traffic Prediction and Experimental Validation for Connected and Autonomous Vehicles / Y. Shao, Z. Sun // *IEEE Transactions on Intelligent Transportation Systems*. – March, 2021. – Vol. 22. – P. 1562–1572.
107. Predictive Eco-Driving Application Considering Real-World Traffic Flow / X. Zhang, T. Zhang, Y. Zou, G. Du, N. Guo // *IEEE Access*. – 2020. – Vol. 8. – P. 82187–82200.
108. Ostojic, M. Comprehensive Framework for Quantitative Performance Assessment of Signal Control Effectiveness using Vehicle Trajectories / M. Ostojic, A. Mittal, H.S. Mahmassani // *Transportation Research Record: Journal of the Transportation Research Board*. – December, 2020. – Vol. 2674. – P. 118–129.
109. Al-Mallah, R. Prediction of Traffic Flow via Connected Vehicles / R. Al-Mallah, A. Quintero, B. Farooq // *IEEE Transactions on Mobile Computing*. – 2020. – P. 1–10.
110. Rahman, M. Real time Traffic Flow Parameters Prediction with Basic Safety Messages at Low Penetration of Connected Vehicles / M. Rahman, M. Chowdhury, J. McClendon // *arXiv:1811.03562 [cs, stat]*. – May, 2019.

111. Kamal, M.A.S. Road-Speed Profile for Enhanced Perception of Traffic Conditions in a Partially Connected Vehicle Environment / M.A.S. Kamal, T. Hayakawa, J.I. Imura // *IEEE Transactions on Vehicular Technology*. – August, 2018. – Vol. 67. – P. 6824–6837.
112. The SCOOT on-line traffic signal optimization technique / P.B. Hunt, D.I. Robertson, R.D. Bretherton, M.C. Royle. – April, 1982. – Vol. 23. – P. 190–192.
113. Transport, Laboratory R.R. SCOOT: A Traffic Responsive Method of Coordinating Signals. TRRL Urban Networks Division / P.B. Hunt, D.I. Robertson, R.D. Bretherton, R.I. Winton. – 1981.
114. Lowrie P.R. SCATS: Sydney Co-Ordinated Adaptive Traffic System: a traffic responsive method of controlling urban traffic // *Roads and Traffic Authority NSW*. – 1990. – P. 28.
115. Webster F.V. *Traffic Signal Settings*. H.M. Stationery Office. – 1958.
116. Little, J. MAXBAND: A Program for Setting Signals on Arteries and Triangular Networks / J. Little, M. Kelson, N. Gartner // *Transportation Research Record Journal of the Transportation Research Board*. – 1981. – Vol. 795. – P. 40–46.
117. Li, M.T. Signal timing optimization for oversaturated networks using TRANSYT-7F / M.T. Li, A.C. Gan // *Transportation Research Record*. – 1999. – P. 118–126.
118. Review of road traffic control strategies / M. Papageorgiou, C. Kiakaki, V. Dinopoulou [et al.] // *Proceedings of the IEEE*. – December, 2003. – Vol. 91. – P. 2043–2067.
119. El-Tantawy, S. An agent-based learning towards decentralized and coordinated traffic signal control 2010 / S. El-Tantawy, B. Abdulhai. – P. 665–670.
120. Varaiya P. The Max-Pressure Controller for Arbitrary Networks of Signalized Intersections. Vol 2. // In: *Advances in Dynamic Network Modeling in Complex Transportation Systems*. – New (York): Springer New York, 2013. – P. 27–66.
121. Genders, W. Evaluating reinforcement learning state representations for adaptive traffic signal control / W. Genders, S. Razavi // *Procedia Computer Science*. – January, 2018. – Vol. 130. – P. 26–33.
122. Мясников, В.В. Детерминированная прогнозная модель управления сигналами светофоров в интеллектуальных транспортных и геоинформационных системах / В.В. Мясников, А.А. Агафонов, А.С. Юмаганов // *Компьютерная оптика*. – 2021. – Т. 45. – С. 917–925.
123. Ribeiro, I.M. The fully actuated traffic control problem solved by global optimization and complementarity / I.M. Ribeiro, M.L. Simões // *Engineering Optimization*. – February, 2016. – Vol. 48. – P. 199–212.
124. Cools, S.B. Self-Organizing Traffic Lights: A Realistic Simulation / S.B. Cools, C. Gershenson, B. D’Hooghe // In: *Advances in Applied Self-Organizing Systems*. – London: Springer, 2013. – P. 45–55.

125. A Survey on Traffic Signal Control Methods / H. Wei, G. Zheng, V. Gayah, Z. Li // arXiv:1904.08117 [cs, stat]. – January, 2020.
126. Savithramma, R.M. Road Traffic Signal Control and Management System : A Survey / R.M. Savithramma, R. Sumathi // 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS) . – December, 2020. – P. 104–110.
127. Traffic Signal Optimization Based on Fuzzy Control and Differential Evolution Algorithm / H. Lin, Y. Han, W. Cai, B. Jin // IEEE Transactions on Intelligent Transportation Systems. – 2022. – P. 1–12.
128. Jafari, S. Improving the Road and Traffic Control Prediction Based on Fuzzy Logic Approach in Multiple Intersections / S. Jafari, Z. Shahbazi, Y.C. Byun // Mathematics. – January, 2022. – Vol. 10. – 2832 p.
129. Diveev, A.I. Synthesis of intelligent control of traffic flows in urban roads based on the logical network operator method / A.I. Diveev, E.A. Sofronova // 2013 European Control Conference (ECC) . – Zurich, July, 2013. – P. 3512–3517.
130. Adaptive traffic light control on highway entrances / A. Alekseenko, Y. Kholodov, A. Kholodov [et al.] // 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC) . – Yokohama, October, 2017. – P. 1–6.
131. Control of Traffic Signals in a Model Predictive Control Framework / M.A.S. Kamal, J. Imura, A. Ohata [et al.] // IFAC Proceedings Volumes. – September, 2012. – Vol. 45. – P. 221–226.
132. Yazici, A.. A model predictive control approach for decentralized traffic signal control / A. Yazici, G. Seo, U. Ozguner // IFAC Proceedings Volumes. – January, 2008. – Vol. 41. – P. 13058–13063.
133. Nakanishi, H. Optimal traffic signal control for alleviation of congestion based on traffic density prediction by model predictive control / H. Nakanishi, T. Namerikawa // 2016 55th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE) . – September, 2016. – P. 1273–1278.
134. Cooperative Comfortable-Driving at Signalized Intersections for Connected and Automated Vehicles / X. Shen, X. Zhang, T. Ouyang [et al.] // IEEE Robotics and Automation Letters. – October, 2020. – Vol. 5. – P. 6247–6254.
135. Agafonov, A.A. Adaptive Traffic Signal Control Based on Neural Network Prediction of Weighted Traffic Flow / A.A. Agafonov, A.S. Yumaganov, V.V. Myasnikov // Optoelectronics, Instrumentation and Data Processing. – October, 2022. – Vol. 58. – P. 503–513.
136. Li, L. A survey of traffic control with vehicular communications / L. Li, D. Wen, D. Yao // IEEE Transactions on Intelligent Transportation Systems. – 2014. – Vol. 15. – P. 425–432.

137. Roess, R. Traffic engineering / R. Roess, E. Prassas, W. McShane // 4th ed. NJ, USA: Pearson/Prentice Hall. – 2010.
138. Gokulan, B.P. Distributed Geometric Fuzzy Multiagent Urban Traffic Signal Control / B.P. Gokulan, D. Srinivasan // IEEE Transactions on Intelligent Transportation Systems. – September, 2010. – Vol. 11. – P. 714–727.
139. Teodorović D. Swarm intelligence systems for transportation engineering: Principles and applications // Transportation Research Part C: Emerging Technologies. – December, 2008. – Vol. 16. – P. 651–667.
140. Mikami, S. Genetic reinforcement learning for cooperative traffic signal control / S. Mikami, Y. Kakazu // Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence. – Orlando: FL, USA, 1994. – P. 223–228.
141. Real-Time Optimization for Adaptive Traffic Signal Control Using Genetic Algorithms / J. Lee, B. Abdulhai, A. Shalaby, E.H. Chung // Journal of Intelligent Transportation Systems. – July, 2005. – Vol. 9. – P. 111–122.
142. Ricalde, E. Evolving Adaptive Traffic Signal Controllers for a Real Scenario Using Genetic Programming with an Epigenetic Mechanism / E. Ricalde, W. Banzhaf // 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA) . – Cancun, December, 2017. – P. 897–902.
143. IntelliLight: A Reinforcement Learning Approach for Intelligent Traffic Light Control / H. Wei, G. Zheng, H. Yao, Z. Li // Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. – London: United Kingdom, July, 2018. – P. 2496–2505.
144. CoLight: Learning Network-level Cooperation for Traffic Signal Control / H. Wei, N. Xu, H. Zhang [et al.] // Proceedings of the 28th ACM International Conference on Information and Knowledge Management. – Beijing: China, November, 2019. – P. 1913–1922.
145. Toward A Thousand Lights: Decentralized Deep Reinforcement Learning for Large-Scale Traffic Signal Control / C. Chen, H. Wei, N. Xu [et al.] // Proceedings of the AAAI Conference on Artificial Intelligence. – April, 2020. – Vol. 34. – P. 3414–3421.
146. Liu, Y. Intelligent traffic light control using distributed multi-agent Q learning / Y. Liu, L. Liu, W.P. Chen // 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC) . – Yokohama, October, 2017. – P. 1–8.
147. Network-wide traffic signal control optimization using a multi-agent deep reinforcement learning / Z. Li, H. Yu, G. Zhang [et al.] // Transportation Research Part C: Emerging Technologies. – April 2021. – Vol. 125. – 103059 p.
148. A Survey on Reinforcement Learning Models and Algorithms for Traffic Signal Control / K.L.A. Yau, J. Qadir, H.L. Khoo [et al.] // ACM Computing Surveys. – May, 2018. – Vol. 50. – P. 1–38.

149. Mannion, P. An Experimental Review of Reinforcement Learning Algorithms for Adaptive Traffic Signal Control / P. Mannion, J. Duggan, E. Howley // In: *Autonomic Road Transport Support Systems*. Cham: Springer International Publishing. – 2016. – P. 47–66.
150. Multi-Agent Deep Reinforcement Learning for Large-Scale Traffic Signal Control / T. Chu, J. Wang, L. Codecà, Z. Li // *IEEE Transactions on Intelligent Transportation Systems*. – March, 2019. – Vol. PP.
151. Application of Deep Reinforcement Learning in Traffic Signal Control: An Overview and Impact of Open Traffic Data / M. Gregurić, M. Vujić, C. Alexopoulos, M. Miletić // *Applied Sciences*. – June, 2020. – Vol. 1. – 4011 p.
152. Qadri, S.S.S.M. State-of-art review of traffic signal control methods: challenges and opportunities / S.S.S.M. Qadri, M.A. Gökçe, E. Öner // *European Transport Research Review*. – December, 2020. – Vol. 12. – 55 p.
153. Watkins, C.J.C.H. Q-learning / C.J.C.H. Watkins, P. Dayan // *Machine Learning*. – May, 1992. – Vol. 8. – P. 279–292.
154. van der Pol, E. Coordinated Deep Reinforcement Learners for Traffic Light Control / E. van der Pol, F.A. Oliehoek // *Proceedings of Learning, Inference and Control of Multi-Agent Systems (at NIPS 2016)*. – 2016.
155. Cooperative Deep Reinforcement Learning for Large-Scale Traffic Grid Signal Control / T. Tan, F. Bao, Y. Deng [et al.] // *IEEE Transactions on Cybernetics*. – June, 2020. – Vol. 50. – P. 2687–2700.
156. Adaptive Traffic Signal Control: Deep Reinforcement Learning Algorithm with Experience Replay and Target Network / J. Gao, Y. Shen, J. Liu [et al.]. – 2017.
157. Ducrocq, R. Deep Reinforcement Q-Learning for Intelligent Traffic Signal Control with Partial Detection / R. Ducrocq, N. Farhi // *International Journal of Intelligent Transportation Systems Research*. – February, 2023.
158. Boukerche, A. A Novel Reinforcement Learning-Based Cooperative Traffic Signal System Through Max-Pressure Control / A. Boukerche, D. Zhong, P. Sun // *IEEE Transactions on Vehicular Technology*. – February, 2022. – Vol. 71. – P. 1187–1198.
159. Double Deep Q-Network with a Dual-Agent for Traffic Signal Control / J. Gu, Y. Fang, Z. Sheng, P. Wen // *Applied Sciences*. – February, 2020. – Vol. 10. – 1622 p.
160. Hasselt, H.V. Deep reinforcement learning with double Q-Learning / H.V. Hasselt, A. Guez, D. Silver // *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. – Phoenix: Arizona, 2016. – P. 2094–2100.
161. Agafonov, A.A. Hybrid Prediction-Based Approach for Traffic Signal Control Problem / A.A. Agafonov, V.V. Myasnikov // *Optical Memory and Neural Networks*. – 2022. – Vol. 31. – 11 p.
162. Casas N. Deep Deterministic Policy Gradient for Urban Traffic Light Control. – 2017.

163. Intelligent Traffic Light via Policy-based Deep Reinforcement Learning / Y. Zhu, M. Cai, C.W. Schwarz [et al.] // International Journal of Intelligent Transportation Systems Research. – December 2022. – Vol. 20. – P. 734–744.
164. An Y. Traffic Signal Control Method Based on Modified Proximal Policy Optimization / Y. An, J. Zhang // 2022 10th International Conference on Traffic and Logistic Engineering (ICTLE). – August 2022. – P. 83–88.
165. Aslani M. Developing adaptive traffic signal control by actor–critic and direct exploration methods / M. Aslani, M.S. Mesgari, S. Seipel, M. Wiering // Proceedings of the Institution of Civil Engineers. – Transport. – October 2019. – Vol. 172. – P. 289–298.
166. A Deep Reinforcement Learning Approach to Traffic Signal Control With Temporal Traffic Pattern Mining / D. Ma, B. Zhou, X. Song, H. Dai // IEEE Transactions on Intelligent Transportation Systems. – August 2022. – Vol. 23. – P. 11789–11800.
167. Aslani M. Adaptive traffic signal control with actor–critic methods in a real-world traffic network with different traffic disruption events / M. Aslani, M.S. Mesgari, M. Wiering // Transportation Research Part C: Emerging Technologies. – 2017. – Vol. 85. – P. 732–752.
168. A survey and comparative study of simulators for vehicular ad hoc networks (VANETs) / F.J. Martinez, C.K. Toh, J.C. Cano [et al.] // Wireless Communications and Mobile Computing. – July 2011. – Vol. 11. – P. 813–828.
169. Parsimonious shooting heuristic for trajectory design of connected automated traffic part II: Computational issues and optimization / J. Ma, X. Li, F. Zhou [et al.] // Transportation Research Part B: Methodological. – January 2017. – Vol. 95. – P. 421–441.
170. Integrated optimal eco-driving on rolling terrain for hybrid electric vehicle with vehicle-infrastructure communication / J. Hu, Y. Shao, Z. Sun [et al.] // Transportation Research Part C: Emerging Technologies. – July 2016. – Vol. 68. – P. 228–244.
171. Asadi B. Predictive Cruise Control: Utilizing Upcoming Traffic Signal Information for Improving Fuel Economy and Reducing Trip Time / B. Asadi, A. Vahidi // IEEE Transactions on Control Systems Technology. – May 2011. – Vol. 19. – P. 707–714.
172. A Priority-Based Autonomous Intersection Management (AIM) Scheme for Connected Automated Vehicles (CAVs) / H. Zhang, R. Zhang, C. Chen [et al.] // Vehicles. – August 2021. – Vol. 3. – P. 533–544.
173. Eco approaching at an isolated signalized intersection under partially connected and automated vehicles environment / H. Jiang, J. Hu, S. An [et al.] // Transportation Research Part C: Emerging Technologies. – June 2017. – Vol. 79. – P. 290–307.
174. Forecasting the Passage Time of the Queue of Highly Automated Vehicles Based on Neural Networks in the Services of Cooperative Intelligent Transport Systems / V. Shepelev, S. Zhankaziev, S. Aliukov [et al.] // Mathematics. – January 2022. – Vol. 10. – P. 282.

175. Lee J. Development and Evaluation of a Cooperative Vehicle Intersection Control Algorithm Under the Connected Vehicles Environment / J. Lee, B. Park // *IEEE Transactions on Intelligent Transportation Systems*. – March 2012. – Vol. 13. – P. 81–90.
176. Managing connected and automated vehicles at isolated intersections: From reservation to optimization-based methods / C. Yu, W. Sun, H.X. Liu, X. Yang // *Transportation Research Part B: Methodological*. – April 2019. – Vol. 122. – P. 416–435.
177. Zhou F. Parsimonious shooting heuristic for trajectory design of connected automated traffic part I: Theoretical analysis with generalized time geography / F. Zhou, X. Li, J. Ma // *Transportation Research Part B: Methodological*. – 2017. – Vol. 95. – P. 394–420.
178. Eco-Trajectory Planning with Consideration of Queue along Congested Corridor for Hybrid Electric Vehicles / Z. Yang, Y. Feng, X. Gong [et al.] // *Transportation Research Record: Journal of the Transportation Research Board*. – September 2019. – Vol. 2673. – P. 277–286.
179. Barth M. Dynamic ECO-driving for arterial corridors / M. Barth, S. Mandava, K. Boriboonsomsin, H.Xia // *2011 IEEE Forum on Integrated and Sustainable Transportation Systems*. – Vienna, Austria. – June 2011. – P. 182–188.
180. Wang Q. Connected automated vehicle trajectory optimization along signalized arterial: A decentralized approach under mixed traffic environment / Q. Wang, Y. Gong, X. Yang // *Transportation Research Part C: Emerging Technologies*. – 2022. – Vol. 145.
181. Zhang L. Theory and Experiment of Cooperative Control at Multi-Intersections in Intelligent Connected Vehicle Environment: Review and Perspectives / L. Zhang, Y. Wang, H. Zhu // *Sustainability (Switzerland)*. – 2022. – Vol. 14.
182. Guo Q. Urban traffic signal control with connected and automated vehicles: A survey / Q. Guo, L. Li, X. Ban // *Transportation Research Part C: Emerging Technologies*. – 2019. – Vol. 101. – P. 313–334.
183. Signalized Intersection Control in Mixed Autonomous and Regular Vehicles Traffic Environment - A Critical Review Focusing on Future Control / M. Al-Turki, N.T. Ratrout, S.M. Rahman, K.J. Assi // *IEEE Access*. – 2022. – Vol. 10. – P. 16942–16951.
184. Nguyen D.D. Smart City Total Transport-Managing System: (A Vision Including the Cooperating, Contract-Based and Priority Transport Management) / D.D. Nguyen, J. Rohacs // *In: Industrial Networks and Intelligent Systems*. Cham: Springer International Publishing. – 2019. – P. 74–85.
185. Intelligent Total Transportation Management System for Future Smart Cities / D.D. Nguyen, J. Rohács, D. Rohács, A. Boros // *Applied Sciences*. – December 2020. – Vol. 10. – P. 8933.
186. Trajectory Planning Method for Mixed Vehicles Considering Traffic Stability and Fuel Consumption at the Signalized Intersection / S. Fang, L. Yang, T. Wang, S. Jing // *Journal of Advanced Transportation*. – January 2020. – Vol. 2020. – P. 1–10.

187. Guo Y. DRL-TP3: A learning and control framework for signalized intersections with mixed connected automated traffic / Y. Guo, J. Ma // *Transportation Research Part C: Emerging Technologies*. – November 2021. – Vol. 132. – P. 103416.
188. Agafonov A. Adaptive Traffic Signal Control Based on Maximum Weighted Traffic Flow / A. Agafonov, A. Yumaganov, V. Myasnikov // *2022 VIII International Conference on Information Technology and Nanotechnology (ITNT)*. – May 2022. – P. 1–6.
189. A Coupled Vehicle-Signal Control Method at Signalized Intersections in Mixed Traffic Environment / Y. Du, W. Shangguan, L. Chai // *IEEE Transactions on Vehicular Technology*. – 2021. – Vol. 70. – P. 2089–2100.
190. Tajalli M. Traffic Signal Timing and Trajectory Optimization in a Mixed Autonomy Traffic Stream / M. Tajalli, A. Hajbabaie // *IEEE Transactions on Intelligent Transportation Systems*. – July 2022. – Vol. 23. – P. 6525–6538.
191. Predicting Bus Arrival Time on the Basis of Global Positioning System Data / D. Sun, H. Luo, L. Fu [et al.] // *Transportation Research Record: Journal of the Transportation Research Board*. – January 2007. – Vol. 2034. – P. 62–72.
192. Jeong R. Prediction Model of Bus Arrival Time for Real-Time Applications / R. Jeong, L.R. Rilett // *Transportation Research Record: Journal of the Transportation Research Board*. – January 2005. – Vol. 1927. – P. 195–204.
193. Агафонов, А.А. Прогнозирование параметров движения городского пассажирского транспорта по данным спутникового мониторинга / А.А. Агафонов, А.В. Сергеев, А.В. Чернов // *Компьютерная оптика*. – 2012. – Т. 36. – № 3. – С. 453-458.
194. Dynamic multi-interval bus travel time prediction using bus transit data / H. Chang, D. Park, S. Lee [et al.] // *Transportmetrica*. – January 2010. – Vol. 6. – P. 19–38.
195. Performance Comparison of Bus Travel Time Prediction Models across Indian Cities / R. Jairam, B.A. Kumar, S.S. Arkatkar, L. Vanajakshi // *Transportation Research Record: Journal of the Transportation Research Board*. – December 2018. – Vol. 2672. – P. 87–98.
196. Chien S.I.J. Dynamic Bus Arrival Time Prediction with Artificial Neural Networks / S.I.J. Chien, Y. Ding, C. Wei // *Journal of Transportation Engineering*. – September 2002. – Vol. 128. – P. 429–438.
197. Comparison of model based and machine learning approaches for bus arrival time prediction / V. Kumar, B.A. Kumar, L. Vanajakshi, S.C. Subramanian // *Transportation Research Board 93rd Annual Meeting*. – 2014. – P. 14.
198. Bin Y. Bus Arrival Time Prediction Using Support Vector Machines / Y. Bin, Y. Zhongzhen, Y. Baozhen // *Journal of Intelligent Transportation Systems*. – December 2006. – Vol. 10. – P. 151–158.
199. Hybrid model for prediction of bus arrival times at next station / B. Yu, Z.Z. Yang, K. Chen, B. Yu // *Journal of Advanced Transportation*. – July 2010. – Vol. 44. – P. 193–204.

200. Xu H. Bus arrival time prediction with real-time and historic data / H. Xu, J. Ying // *Cluster Computing*. – December 2017. – Vol. 20. – P. 3099–3106.
201. A Dynamic Bus-Arrival Time Prediction Model Based on APC Data / M. Chen, X. Liu, J. Xia, S.I. Chien // *Computer-Aided Civil and Infrastructure Engineering*. – September 2004. – Vol. 19. – P. 364–376.
202. Dynamic Bus Travel Time Prediction Models on Road with Multiple Bus Routes / C. Bai, Z.R. Peng, Q.C. Lu, J. Sun // *Computational Intelligence and Neuroscience*. – 2015. – Vol. 2015. – P. 1–9.
203. van Hinsbergen C.P.I. Bayesian committee of neural networks to predict travel times with confidence intervals / C.P.I. van Hinsbergen, J.W.C. van Lint, H.J. van Zuylen // *Transportation Research Part C: Emerging Technologies*. – October 2009. – Vol. 17. – P. 498–509.
204. Method of predicting bus arrival time based on MapReduce combining clustering with neural network / J. Zhang, J. Gu, L. Guan, S. Zhang // *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)*. – March 2017. – P. 296–302.
205. Yu B. Hybrid model for multi-stop arrival time prediction / B. Yu, Z.Z. Yang, B. Yu // *Neural Network World*. – 2009. – Vol. 19. – P. 321–332.
206. Bus Arrival Time Prediction Using Support Vector Machine With Genetic Algorithm / M. Yang, C. Chen, L. Wang [et al.] // *Neural Network World*. – 2016. – Vol. 26. – P. 205–217.
207. A prediction model of bus arrival time at stops with multi-routes / T. Yin, G. Zhong, J. Zhang [et al.] // *Transportation Research Procedia*. – 2017. – Vol. 25. – P. 4623–4636.
208. Agafonov A. An Adaptive Algorithm for Public Transport Arrival Time Prediction Based on Hierarchical Regression / A. Agafonov, V. Myasnikov // *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. – September 2015. – P. 2776–2781.
209. Prediction of Bus Travel Time Using Random Forests Based on Near Neighbors: Prediction of bus travel time / B. Yu, H. Wang, W. Shan, B. Yao // *Computer-Aided Civil and Infrastructure Engineering*. – April 2018. – Vol. 33. – P. 333–350.
210. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data / X. Ma, Z. Tao, Y. Wang [et al.] // *Transportation Research Part C: Emerging Technologies*. – May 2015. – Vol. 54. – P. 187–197.
211. Duan Y. Travel time prediction with LSTM neural network / Y. Duan, L.V. Yisheng, F.Y. Wang // *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. – November 2016. – P. 1053–1058.
212. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting / Y. Li, R. Yu, C. Shahabi, Y. Liu // *arXiv:1707.01926 [cs, stat]*. – July 2017.

213. Gao S. Optimal routing policy problems in stochastic time-dependent networks / S. Gao, I. Chabini // *Transportation Research Part B: Methodological*. – February 2006. – Vol. 40. – P. 93–122.
214. Hall R.W. The Fastest Path through a Network with Random Time-Dependent Travel Times / R.W. Hall // *Transportation Science*. – August 1986. – Vol. 20. – P. 182–188.
215. Gao S. Real-time traveler information for optimal adaptive routing in stochastic time-dependent networks / S. Gao, H. Huang // *Transportation Research Part C: Emerging Technologies*. – April 2012. – Vol. 21. – P. 196–213.
216. Nie Y. Shortest path problem considering on-time arrival probability / Y. Nie, X. Wu // *Transportation Research Part B: Methodological*. – July 2009. – Vol. 43. – P. 597–613.
217. Samaranayake S. A tractable class of algorithms for reliable routing in stochastic networks / S. Samaranayake, S. Blandin, A. Bayen // *Transportation Research Part C: Emerging Technologies*. – February 2012. – Vol. 20. – P. 199–217.
218. Fu L. Expected shortest paths in dynamic and stochastic traffic networks / L. Fu, L. Rilett // *Transportation Research Part B: Methodological*. – 1998. – Vol. 32(7). – P. 499–516.
219. Chen A. Path finding under uncertainty / A. Chen, Z. Ji // *Journal of Advanced Transportation*. – September 2005. – Vol. 39. – P. 19–37.
220. Finding Reliable Shortest Paths in Road Networks Under Uncertainty / B.Y. Chen, W.H.K. Lam, A. Sumalee [et al.] // *Networks and Spatial Economics*. – June 2013. – Vol. 13. – P. 123–148.
221. Fan Y. Optimal Routing for Maximizing the Travel Time Reliability / Y. Fan, Y. Nie // *Networks and Spatial Economics*. – September 2006. – Vol. 6. – P. 333–344.
222. Nie Y. Arriving-on-Time Problem: Discrete Algorithm That Ensures Convergence / Y. Nie, Y. Fan // *Transportation Research Record: Journal of the Transportation Research Board*. – January 2006. – Vol. 1964. – P. 193–200.
223. Wu X. Modeling Heterogeneous Risk-Taking Behavior in Route Choice: A Stochastic Dominance Approach / X. Wu, Y. Nie // *Procedia – Social and Behavioral Sciences*. – 2011. – Vol. 17. – P. 382–404.
224. Chen P. Application of Finite Mixture of Regression Model with Varying Mixing Probabilities to Estimation of Urban Arterial Travel Times / P. Chen, K. Yin, J. Sun // *Transportation Research Record: Journal of the Transportation Research Board*. – January 2014. – Vol. 2442. – P. 96–105.
225. The α -reliable path problem in stochastic road networks with link correlations: A moment-matching-based path finding algorithm / P. Chen, R. Tong, G. Lu, Y. Wang // *Expert Systems with Applications*. – November 2018. – Vol. 110. – P. 20–32.

226. Chen B.Y. Finding the k reliable shortest paths under travel time uncertainty / B.Y. Chen, Q. Li, W.H.K. Lam // *Transportation Research Part B: Methodological*. – December 2016. – Vol. 94. – P. 189–203.
227. Reliable route guidance: A case study from Chicago / Y. Nie, X. Wu, J.F. Dillenburg, P.C. Nelson // *Transportation Research Part A: Policy and Practice*. – February 2012. – Vol. 46. – P. 403–419.
228. Most reliable path-finding algorithm for maximizing on-time arrival probability / B.Y. Chen, C. Shi, J. Zhang [et al.] // *Transportmetrica B: Transport Dynamics*. – July 2017. – Vol. 5. – P. 248–264.
229. Maximizing the Probability of Arriving on Time: A Practical Q-Learning Method / Z. Cao, H. Guo, J. Zhang [et al.] // *Proceedings of the AAAI Conference on Artificial Intelligence*. – February 2017. – Vol. 31.
230. Flajolet A. Robust Adaptive Routing Under Uncertainty / A. Flajolet, S. Blandin, P. Jaillet // *Operations Research*. – February 2018. – Vol. 66. – P. 210–229.
231. Yang L. Optimizing on-time arrival probability and percentile travel time for elementary path finding in time-dependent transportation networks: Linear mixed integer programming reformulations / L. Yang, X. Zhou // *Transportation Research Part B: Methodological*. – February 2017. – Vol. 96. – P. 68–91.
232. Zhang Y. An algorithm for reliable shortest path problem with travel time correlations / Y. Zhang, A. Khani // *Transportation Research Part B: Methodological*. – March 2019. – Vol. 121. – P. 92–113.
233. Агафонов, А.А. Метод определения надёжного кратчайшего пути в зависящей от времени стохастической сети и его применение в геоинформационных задачах управления транспортом / А.А. Агафонов, В.В. Мясников // *Компьютерная Оптика*. – 2016. – Т. 40. – С. 275–283.
234. Samaranayake S. Speedup Techniques for the Stochastic on-time Arrival Problem / S. Samaranayake, S. Blandin, A. Bayen. – 2012. – P. 14.
235. Niknami M. Tractable Pathfinding for the Stochastic On-Time Arrival Problem / M. Niknami, S. Samaranayake // In: *Experimental Algorithms*. Cham: Springer International Publishing. – 2016. – Vol 9685. – P. 231–245.
236. Sabran G. Precomputation techniques for the stochastic on-time arrival problem / G. Sabran, S. Samaranayake, A. Bayen // In: *2014 Proceedings of the Sixteenth Workshop on Algorithm Engineering and Experiments (ALENEX)*. – 2014. – P. 138–146.
237. Abeydeera M. GPU parallelization of the stochastic on-time arrival problem / M. Abeydeera, S. Samaranayake // *2014 21st International Conference on High Performance Computing (HiPC)*. – December 2014. – P. 1–8.

238. Çolak S. Understanding congested travel in urban areas / S. Çolak, A. Lima, M.C. González // *Nature Communications*. – March 2016. – Vol. 7. – P. 10793.
239. Schmitt E. Vehicle Route Guidance Systems: Classification and Comparison / E. Schmitt, H. Jula // *2006 IEEE Intelligent Transportation Systems Conference*. – 2006. – P. 242–247.
240. Multi-agent based vehicular congestion management / P. Desai, S.W. Loke, A. Desai, J. Singh // *2011 IEEE Intelligent Vehicles Symposium (IV)*. – June 2011. – P. 1031–1036.
241. Controlling road congestion via a low-complexity route reservation approach / C. Menelaou, P. Kolios, S. Timotheou [et al.] // *Transportation Research Part C: Emerging Technologies*. – August 2017. – Vol. 81. – P. 118–136.
242. Агафонов, А.А. Исследование численного метода резервирования маршрутов в геоинформационной задаче маршрутизации автономных транспортных средств / А.А. Агафонов, В.В. Мясников // *Компьютерная оптика*. – 2018. – Т. 42. – С. 912–920.
243. Proactive Vehicular Traffic Rerouting for Lower Travel Time / J. Pan, I.S. Popa, K. Zeitouni, C. Borcea // *IEEE Transactions on Vehicular Technology*. – October 2013. – Vol. 62. – P. 3551–3568.
244. Gokula Krishnan V. Analyze traffic forecast for decentralized multi agent system using I-ACO routing algorithm / V. Gokula Krishnan, N. Sankar Ram // *Journal of Ambient Intelligence and Humanized Computing*. – August 2018.
245. Lamouik I. Smart multi-agent traffic coordinator for autonomous vehicles at intersections / I. Lamouik, A. Yahyaouy, M.A. Sabri // *2017 International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*. – May 2017. – P. 1–6.
246. Multiagent-based cooperative vehicle routing using node pressure and auctions / M. Seshadri, Z. Cao, H. Guo [et al.] // *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. – October 2017. – P. 1–7.
247. System-Optimal Routing of Traffic Flows with User Constraints in Networks with Congestion / O. Jahn, R.H. Möhring, A.S. Schulz, N.E. Stier-Moses // *Operations Research*. – August 2005. – Vol. 53. – P. 600–616.
248. Groot N. Toward System-Optimal Routing in Traffic Networks: A Reverse Stackelberg Game Approach / N. Groot, B. De Schutter, H. Hellendoorn // *IEEE Transactions on Intelligent Transportation Systems*. – February 2015. – Vol. 16. – P. 29–40.
249. Varga L.Z. Benefit of Online Real-Time Data in the Braess Paradox with Anticipatory Routing / L.Z. Varga // *2016 IEEE International Conference on Autonomic Computing (ICAC)*. – July 2016. – P. 245–250.
250. Parametric modeling of the heteroscedastic traffic speed variance from loop detector data: Heteroscedastic traffic speed variance / H. Wang, Z. Li, D. Hurwitz, J. Shi // *Journal of Advanced Transportation*. – March 2015. – Vol. 49. – P. 279–296.

251. Raya M. The security of vehicular ad hoc networks / M. Raya, J.P. Hubaux. – 2005. – Vol. 2005. – P. 11–21.
252. Mejri M.N. Survey on VANET security challenges and possible cryptographic solutions / M.N. Mejri, J. Ben-Othman, M. Hamdi // Vehicular Communications. – April 2014. – Vol. 1. – P. 53–66.
253. Manvi S.S. A survey on authentication schemes in VANETs for secured communication / S.S. Manvi, S. Tangade // Vehicular Communications. – July 2017. – Vol. 9. – P. 19–30.
254. ECPP: Efficient conditional privacy preservation protocol for secure vehicular communications / R. Lu, X. Lin, H. Zhu [et al.]. – 2008. – P. 1903–1911.
255. Enhanced MAC-based efficient message authentication scheme over VANET / T. Shin, J. Jung, H. Kim, S.W. Lee. – 2014. – P. 110–113.
256. Zhang C. An efficient message authentication scheme for vehicular communications / C. Zhang, X. Lin, R. Lu [et al.] // IEEE Transactions on Vehicular Technology. – 2008. – Vol. 57. – P. 3357–3368.
257. Ibrahim S. Comparison on VANET authentication schemes: Public Key vs. Symmetric Key / S. Ibrahim, M.A. Hamdy // 2015 Tenth International Conference on Computer Engineering & Systems (ICCES). – December 2015. – P. 341–345.
258. An efficient identity-based batch verification scheme for vehicular sensor networks / C. Zhang, R. Lu, X. Lin [et al.]. – 2008. – P. 816–824.
259. Distributed Aggregate Privacy-Preserving Authentication in VANETs / L. Zhang, Q. Wu, J. Domingo-Ferrer [et al.] // IEEE Transactions on Intelligent Transportation Systems. – 2017. – Vol. 18. – P. 516–526.
260. ECPB: Efficient conditional privacy-preserving authentication scheme supporting batch verification for VANETs / Y. Wang, H. Zhong, Y. Xu, J. Cui // International Journal of Network Security. – 2016. – Vol. 18. – P. 374–382.
261. Guo J. A group signature based secure and privacy-preserving vehicular communication framework / J. Guo, J.P. Baugh, S. Wang. – 2007. – P. 103–108.
262. Wu H.T. RSU-based message authentication for vehicular ad-hoc networks / H.T. Wu, W.S. Hsieh // Multimedia Tools and Applications. – September 2013. – Vol. 66. – P. 215–227.
263. NERA: A new and efficient RSU based authentication scheme for VANETs / M. Bayat, M. Pournaghi, M. Rahimi, M. Barmshoory // Wireless Networks. – July 2020. – Vol. 26. – P. 3083–3098.
264. NECPPA: A novel and efficient conditional privacy-preserving authentication scheme for VANET / S.M. Pournaghi, B. Zahednejad, M. Bayat, Y. Farjami // Computer Networks. – April 2018. – Vol. 134. – P. 78–92.

265. Rajput U. A Hybrid Approach for Efficient Privacy-Preserving Authentication in VANET / U. Rajput, H. Eun, H. Oh. – 2017. – Vol. 5. – P. 17.
266. Efficient Message Authentication Scheme with Conditional Privacy-Preserving and Signature Aggregation for Vehicular Cloud Network / Y. Xie, F. Xu, D. Li, Y. Nie // *Wireless Communications and Mobile Computing*. – September 2018. – Vol. 2018. – P. 1–12.
267. Tan S. A Trust Management System for Securing Data Plane of Ad-Hoc Networks / S. Tan, X. Li, Q. Dong // *IEEE Transactions on Vehicular Technology*. – September 2016. – Vol. 65. – P. 7579–7592.
268. A Secure Trust Model Based on Fuzzy Logic in Vehicular Ad Hoc Networks With Fog Computing / S.A. Soleymani, A.H. Abdullah, M. Zareei [et al.] // *IEEE Access*. – 2017. – Vol. 5. – P. 15619–15629.
269. Li W. ART: An Attack-Resistant Trust Management Scheme for Securing Vehicular Ad Hoc Networks / W. Li, H. Song // *IEEE Transactions on Intelligent Transportation Systems*. – April 2016. – Vol. 17. – P. 960–969.
270. An Attack-Resistant Trust Inference Model for Securing Routing in Vehicular Ad Hoc Networks / H. Xia, S.S. Zhang, Y. Li [et al.] // *IEEE Transactions on Vehicular Technology*. – July 2019. – Vol. 68. – P. 7108–7120.
271. REPLACE: A Reliable Trust-Based Platoon Service Recommendation Scheme in VANET / H. Hu, R. Lu, Z. Zhang, J. Shao // *IEEE Transactions on Vehicular Technology*. – February 2017. – Vol. 66. – P. 1786–1797.
272. Lin B. A Cloud-Based Trust Evaluation Scheme Using a Vehicular Social Network Environment / B. Lin, X. Chen, L. Wang // *2017 24th Asia-Pacific Software Engineering Conference (APSEC)*. – December 2017. – P. 120–129.
273. Xiao Y. BayesTrust and VehicleRank: Constructing an Implicit Web of Trust in VANET / Y. Xiao, Y. Liu // *IEEE Transactions on Vehicular Technology*. – March 2019. – Vol. 68. – P. 2850–2864.
274. TFDD: A trust-based framework for reliable data delivery and DoS defense in VANETs / C.A. Kerrache, N. Lagraa, C.T. Calafate, A. Lakas // *Vehicular Communications*. – July 2017. – Vol. 9. – P. 254–267.
275. On the Human Factor Consideration for VANETs Security Based on Social Networks / C.A. Kerrache, N. Lagraa, A. Benslimane [et al.] // *2018 IEEE International Conference on Communications (ICC)*. – May 2018. – P. 1–6.
276. AATMS: An Anti-Attack Trust Management Scheme in VANET / J. Zhang, K. Zheng, D. Zhang, B. Yan // *IEEE Access*. – 2020. – Vol. 8. – P. 21077–21090.
277. Rajamani R. An experimental comparative study of autonomous and co-operative vehicle-follower control systems / R. Rajamani, S.E. Shladover // *Transportation Research Part C: Emerging Technologies*. – February 2001. – Vol. 9. – P. 15–31.

278. Darbha S. Intelligent cruise control systems and traffic flow stability / S. Darbha, K.R. Rajagopal // *Transportation Research Part C: Emerging Technologies*. – December 1999. – Vol. 7. – P. 329–352.
279. Lu X.Y. ACC/CACC - Control design, stability and robust performance / X.Y. Lu, J.K. Hedrick, M. Drew. – 2002. – Vol. 6. – P. 4327–4332.
280. Cooperative adaptive cruise control in real traffic situations / V. Milanés, S.E. Shladover, J. Spring [et al.] // *IEEE Transactions on Intelligent Transportation Systems*. – 2014. – Vol. 15. – P. 296–305.
281. Rolling horizon stochastic optimal control strategy for ACC and CACC under uncertainty / Y. Zhou, S. Ahn, M. Chitturi, D.A. Noyce // *Transportation Research Part C: Emerging Technologies*. – October 2017. – Vol. 83. – P. 61–76.
282. di Bernardo M. Distributed Consensus Strategy for Platooning of Vehicles in the Presence of Time-Varying Heterogeneous Communication Delays / M. di Bernardo, A. Salvi, S. Santini // *IEEE Transactions on Intelligent Transportation Systems*. – February 2015. – Vol. 16. – P. 102–112.
283. Olfati-Saber R. Consensus problems in networks of agents with switching topology and time-delays / R. Olfati-Saber, R.M. Murray // *IEEE Transactions on Automatic Control*. – September 2004. – Vol. 49. – P. 1520–1533.
284. A Consensus-Based Approach for Platooning with Intervehicular Communications and Its Validation in Realistic Scenarios / S. Santini, A. Salvi, A.S. Valente [et al.] // *IEEE Transactions on Vehicular Technology*. – March 2017. – Vol. 66. – P. 1985–1999.
285. Adaptive multi-agents synchronization for collaborative driving of autonomous vehicles with multiple communication delays / A. Petrillo, A. Salvi, S. Santini, A.S. Valente // *Transportation Research Part C: Emerging Technologies*. – January 2018. – Vol. 86. – P. 372–392.
286. A Third-Order Consensus Approach for Vehicle Platoon with Intervehicle Communication / J. Chen, D. Bai, H. Liang, Y. Zhou // *Journal of Advanced Transportation*. – July 2018. – Vol. 2018. – P. 1–10.
287. Security challenges and implementation mechanism for vehicular Ad Hoc network / M.K. Nasir, D. Hossain, S. Hossain [et al.] // *International Journal of Scientific & Technology Research (IJSTR)*. – 2013. – Vol. 2. – P. 156–161.
288. Security challenges in vehicular cloud computing / G. Yan, D. Wen, S. Olariu, M.C. Weigle // *IEEE Transactions on Intelligent Transportation Systems*. – March 2013. – Vol. 14. – P. 284–294.
289. Isaac J.T. Security attacks and solutions for vehicular ad hoc networks / J.T. Isaac, S. Zeadally, J.S. Cámara // *IET Communications*. – April 2010. – Vol. 4. – P. 894–903.

290. Petrillo A. A collaborative approach for improving the security of vehicular scenarios: The case of platooning / A. Petrillo, A. Pescapé, S. Santini // *Computer Communications*. – June 2018. – Vol. 122. – P. 59–75.
291. Alotibi F. Anomaly Detection in Cooperative Adaptive Cruise Control Using Physics Laws and Data Fusion / F. Alotibi, M. Abdelhakim // *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*. – September 2019. – P. 1–7.
292. Threat Detection for Collaborative Adaptive Cruise Control in Connected Cars / M. Jagielski, N. Jones, C.W. Lin [et al.] // *Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. – June 2018. – P. 184–189.
293. Impact of Security Attacks on Cooperative Driving Use Case: CACC Platooning / P.K. Singh, G. Saikamal Tabjul, M. Imran [et al.] // *TENCON 2018 - 2018 IEEE Region 10 Conference*. Jeju, Korea (South). – October 2018. – P. 0138–0143.
294. Ashley A. History of Intelligent Transportation Systems / A. Ashley, F. Shelley, L. Stephen // *U.S. Department of transportation intelligent transportation systems joint program office*. – 2016.
295. Intelligent Transportation Systems (ITS) Joint Program Office: Strategic Plan 2020–2025 / S. Chan-Edmiston, S. Fischer, S. Sloan, M. Wong // *U.S. Department of Transportation's*.
296. Patel P. A Survey on Intelligent Transportation System Using Internet of Things / P. Patel, Z. Narmawala, A. Thakkar // *Advances in Intelligent Systems and Computing*. – 2019. – Vol. 882. – P. 231–240.
297. Kaffash S. Big data algorithms and applications in intelligent transportation system: A review and bibliometric analysis / S. Kaffash, A.T. Nguyen, J. Zhu // *International Journal of Production Economics*. – January 2021. – Vol. 231. – P. 107868.
298. Big Data Analytics in Intelligent Transportation Systems: A Survey / L. Zhu, F.R. Yu, Y. Wang [et al.] // *IEEE Transactions on Intelligent Transportation Systems*. – January 2019. – Vol. 20. – P. 383–398.
299. Data-driven intelligent transportation systems: A survey / J. Zhang, F.Y. Wang, K. Wang [et al.] // *IEEE Transactions on Intelligent Transportation Systems*. – 2011. – Vol. 12. – P. 1624–1639.
300. Pan, S.J. A Survey on Transfer Learning / S.J. Pan, Q. Yang // *IEEE Transactions on Knowledge and Data Engineering*. – October, 2010. – Vol. 22. – P. 1345–1359.
301. Veres, M. Deep Learning for Intelligent Transportation Systems: A Survey of Emerging Trends / M. Veres, M. Moussa // *IEEE Transactions on Intelligent Transportation Systems*. – August, 2020. – Vol. 21. – P. 3152–3168.
302. Enhancing transportation systems via deep learning: A survey / Y. Wang, D. Zhang, Y. Liu [et al.] // *Transportation Research Part C: Emerging Technologies*. – February, 2019. – Vol. 99. – P. 144–163.

303. Zhu, T. Intelligent Transport Systems in China: Past, Present and Future / T. Zhu, Z. Liu // 2015 Seventh International Conference on Measuring Technology and Mechatronics Automation. – Nanchang, China, June 2015. – P. 581–584.
304. Next-generation innovation and development of intelligent transportation system in China / W. Huang, Y. Wei, J. Guo, J. Cao // Science China Information Sciences. – November, 2017. – Vol. 60. – P. 110201.
305. Rugged communications for intelligent transportation systems // Siemens: [сайт]. – 2017. – URL: <https://assets.new.siemens.com/siemens/assets/api/uuid:1f855245-36ee-43c1-bbff-66fa8fb8dcb4/bs-robust-communications-for-its-en.pdf> (дата обращения 17.05.2023).
306. Интеллектуальные транспортные системы. Состояние рынка. Круглый стол. Обзор решений // Techportal.ru: [сайт]. – 2023. – URL: <http://www.techportal.ru/review/security-on-transport/its/> (дата обращения 18.05.2023).
307. Концепция внедрения интеллектуальных транспортных систем в городских агломерациях // Федеральное дорожное агентство Росавтодор: [сайт]. – URL: <https://rosavtodor.gov.ru/storage/app/uploads/public/5e0/1e9/708/5e01e9708cf8e517182017.pdf> (дата обращения 17.05.2023).
308. Жанказиев, С.В. Научные основы и методология формирования интеллектуальных транспортных систем в автомобильно-дорожных комплексах городов и регионов: дис. ... д-р. техн. наук. // Московский автомобильно-дорожный государственный технический университет (МАДИ). – Москва. – 451 с.
309. Mikheeva, T.I. Methodology for the Formation of the Infrastructure of Intelligent Management of Transport Processes / T.I. Mikheeva, S.V. Mikheev, A.I. Chugunov // Proceedings of the 8th Scientific Conference on Information Technologies for Intelligent Decision Making Support (ITIDS 2020) . – Ufa, Stavropol, Khanty-Mansiysk: Russian Federation. – 2020.
310. Golovnin, O.K. Attribute-driven network-centric urban transport process control system modeling / O.K. Golovnin, T.I. Mikheeva // Journal of Physics: Conference Series. – December, 2018. – Vol. 1096. – P. 012199.
311. Sumalee, A. Smarter and more connected: Future intelligent transportation system / A. Sumalee, H.W. Ho // IATSS Research. – July, 2018. – Vol. 42. – P. 67–71.
312. Structural analysis of network traffic flows / A. Lakhina, K. Papagiannaki, M. Crovella [et al.] // ACM SIGMETRICS Performance Evaluation Review. – 2004. – Vol. 32. – № 1. – P. 61-72.
313. Jolliffe, I.T. Principal Component Analysis. 2nd ed. – New York: Springer. – P. 487.
314. Mai, T. Short-term traffic flow forecasting using dynamic linear models / T. Mai, B. Ghosh, S. Wilson // In Irish Transport Research Network.

315. The application of space-time ARIMA model on traffic flow forecasting / S.H. Lin, H.Q. Huang, D.Q. Zhu, T.Z. Wang // Machine Learning and Cybernetics, 2009 International Conference on. – 2009. – Vol. 6. – P. 3408-3412.
316. Mai, T. Multivariate short term traffic flow forecasting using Bayesian vector autoregressive moving average model / T. Mai, B. Ghosh, S. Wilson // Transportation Research Board 91st Annual Meeting. – 2012. – Vol. 12. – P. 1-16.
317. Kamarianakis, I. Traffic Flow Modeling and Forecasting Through Vector Autoregressive and Dynamic Space Time Models / I. Kamarianakis, P. Prastacos // Proceeding of the 13th Mini-EURO Conference. – 2002. – P. 137-142.
318. Min, W. Real-time road traffic prediction with spatio-temporal correlations / W. Min, L. Wynter // Transportation Research Part C: Emerging Technologies. – 2011. – Vol. 19. – № 4. – P. 606-616.
319. Box, G.E. Time Series Analysis: Forecasting and Control. 4th ed. Wiley / G.E. Box, G.M. Jenkins, G.C. Reinsel. – 2008. – P. 784.
320. Chatfield, C. The Analysis of Time Series: An Introduction. London: Chapman and Hall. – 2003. – P. 352.
321. Traffic Flow Forecasting by Combination of SVM with PCA / S. Zhanquan, P. Jingshan, Z. Zhanjun [et al.] // J. Highway Transp. Res. Dev. . – Vol. 4. – № 2, 20120. – P. 103-107.
322. Apache Spark™ : [сайт]. – 2023. – URL: <https://spark.apache.org/> (дата обращения 29.03.2023).
323. Dean, J. MapReduce: simplified data processing on large clusters / J. Dean, S. Ghemawat // Communications of the ACM. – January, 2008. – Vol. 51. – P. 107–113.
324. Hammond, D.K. Wavelets on graphs via spectral graph theory / D.K. Hammond, P. Vandergheynst, R. Gribonval // Applied and Computational Harmonic Analysis. – 2011. – Vol. 30. – P. 129–150.
325. Hochreiter, S. Long Short-Term Memory / S. Hochreiter, J. Schmidhuber // Neural Comput. – November, 1997. – Vol. 9. – P. 1735–1780.
326. Freeway performance measurement system: Mining loop detector data / C. Chen, K. Petty, A. Skabardonis [et al.] // Transportation Research Record. – 2001. – P. 96–102.
327. Microscopic Traffic Simulation using SUMO / P.A. Lopez, E. Wiessner, M. Behrisch [et al.] // 2018 21st International Conference on Intelligent Transportation Systems (ITSC) . – Maui HI. – November, 2018. – P. 2575–2582.
328. Chakirov, A. Enriched Sioux Falls Scenario with Dynamic and Disaggregate Demand. Singapore ETH Centre (SEC): Future Cities Laboratory / A. Chakirov, P.J. Fourie . – 2014.

329. Agafonov, A. Traffic Flow Prediction Using Graph Convolution Neural Networks // 2020 10th International Conference on Information Science and Technology (ICIST). – September, 2020. – P. 91–95.
330. Agafonov, A. Spatio-Temporal Graph Convolutional Networks for Short-Term Traffic Forecasting / A. Agafonov, A. Yumaganov // 2020 International Conference on Information Technology and Nanotechnology (ITNT). – Samara, Russia, May, 2020. – P. 1–6.
331. Agafonov, A., Spatial-Temporal K Nearest Neighbors Model on MapReduce for Traffic Flow Prediction / A. Agafonov, A. Yumaganov // Intelligent Data Engineering and Automated Learning – IDEAL 2018. – Madrid: Spain, 2018. – Vol. 11314 LNCS. – P. 253–260.
332. Agafonov, A.A. Short-term traffic flow forecasting using a distributed spatial-temporal model / A.A. Agafonov, A.S. Yumaganov // CEUR Workshop Proceedings. – 2018. – Vol. 2210. – P. 402–409.
333. Агафонов, А.А. Прогнозирование транспортного потока с использованием графовых нейронных сетей / А.А. Агафонов, А.С. Юмаганов // Сборник трудов по материалам VI Международной конференции и молодежной школы в 4 т. – Самара, 2020. – Т. 4. – С. 143–152.
334. Agafonov, A.A. A public transport departure time prediction algorithm based on operation strategies and real-time monitoring data // CEUR Workshop Proceedings. – 2018. – Vol. 2210. – P. 75–81.
335. Agafonov, A. Traffic flow forecasting algorithm based on combination of adaptive elementary predictors / A. Agafonov, V. Myasnikov // Communications in Computer and Information Science. – 2015. – Vol. 542. – P. 163–174.
336. Agafonov, A. Structural optimization of the travel time prediction model based on hierarchical regression / A. Agafonov, V. Myasnikov // CEUR Workshop Proceedings. – 2016. – Vol. 1710. – P. 1–9.
337. Agafonov, A. Traffic Signal Control: a Double Q-learning Approach / A. Agafonov, V. Myasnikov // Proceedings of the 16th Conference on Computer Science and Intelligence Systems. – FedCSIS 2021. – September, 2021. – P. 365–369.
338. Hasselt, H.V. Double Q-learning // In: Advances in Neural Information Processing Systems 23. – 2010. – P. 2613–2621.
339. RESCO // Github: [сайт]. – 2021. – URL: <https://github.com/Pi-Star-Lab/RESCO> (дата обращения 10.04.2023).
340. TAPASCologne - SUMO Documentation // SUMO : [сайт]. – 2023. – URL: <https://sumo.dlr.de/docs/Data/Scenarios/TAPASCologne.html> (дата обращения 15.02.2023).

341. Ault, J. Learning an Interpretable Traffic Signal Control Policy / J. Ault, J.P. Hanna, G. Sharon. – arXiv. – 2020. – arXiv:1912.11023 [cs, stat].
342. Islam, S.M.A.B.A. A real-time network-level traffic signal control methodology with partial connected vehicle information / S.M.A.B.A. Islam, A. Hajbabaie, H.M.A. Aziz // Transportation Research Part C: Emerging Technologies. – December, 2020. – Vol. 121. – P. 102830.
343. Scalable trust-region method for deep reinforcement learning using Kronecker-factored approximation / Y. Wu, E. Mansimov, S. Liao [et al.] // arXiv:1708.05144 [cs] . – August, 2017.
344. PFRL // Github: [сайт]. – 2022. URL: <https://github.com/pfnet/pfml> (дата обращения 24.02.2022).
345. Krauß, S. Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics. – 1998.
346. HBEFA - Handbook Emission Factors for Road Transport : [сайт]. – 2023. – URL: <https://www.hbefa.net/e/index.html> (дата обращения 14.02.2023).
347. Ault, J. Reinforcement Learning Benchmarks for Traffic Signal Control / J. Ault, G. Sharon // Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks. – December, 2021. – Vol. 1.
348. Agafonov, A. Agent-Based Traffic Signal Control Using a Reinforcement Learning Approach / A. Agafonov, A. Yumaganov // 2021 International Conference on Information Technology and Nanotechnology (ITNT) . – September, 2021. – P. 1– 4.
349. Agafonov, A. Comparison of Traffic Signal Control Algorithms in a Large-Scale Traffic Simulation Environment / A. Agafonov, E. Efimenko // 2022 VIII International Conference on Information Technology and Nanotechnology (ITNT) . – May, 2022. – P. 1–5.
350. Yumaganov, A. Vehicle Trajectory Planning in the Problem of Traffic Flow Control at Signalized Intersections / A. Yumaganov, A. Agafonov // 2022 VIII International Conference on Information Technology and Nanotechnology (ITNT) . – May, 2022. – P. 1– 4.
351. Агафонов, А.А. Адаптивное управление дорожными сигналами на основе нейросетевого прогноза максимального взвешенного потока / А.А. Агафонов, А.С. Юмаганов, В.В. Мясников // Автометрия. – 2022. – Т. 58. – С. 85–97.
352. Agafonov, A. An Algorithm for Cooperative Control of Traffic Signals and Vehicle Trajectories / A. Agafonov, A. Yumaganov, V. Myasnikov // 2022 4th International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA). – November 2022. – P. 675–680.

353. Agafonov, A. Cooperative Control for Signalized Intersections in Intelligent Connected Vehicle Environments / A. Agafonov, A. Yumaganov, V. Myasnikov // *Mathematics*. – January, 2023. – Vol. 11. – P. 1540.
354. Агафонов, А.А. Совместное управление сигналами светофоров и траекториями движения транспортных средств / А.А. Агафонов, А.С. Юмаганов // *Информатика и автоматизация*. – Январь, 2023. – Т. 22. – С. 5–32.
355. Petrenkov, D. Anomaly Detection in Vehicle Platoon with Third-Order Consensus Control / D. Petrenkov, A. Agafonov // *2021 Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBREIT)*. – May, 2021. – P. 0463–0466.
356. Агафонов, А.А. Управление сигналами светофоров с использованием обучения с подкреплением / А.А. Агафонов, А.С. Юмаганов // *Сборник трудов по материалам VII Международной конференции и молодежной школы. Информационные технологии и нанотехнологии (ИТНТ-2021)*. – 2021. – Т. 2. – С. 22282.
357. Юмаганов, А.С. Сравнение подходов к созданию системы автономного управления транспортным средством / А.С. Юмаганов, А.А. Агафонов // *Сборник трудов по материалам VII Международной конференции и молодежной школы. Информационные технологии и нанотехнологии (ИТНТ-2021)*. – 2021. – Т. 2. – 22302 с.
358. Агафонов, А.А. Сравнение алгоритмов управления сигналами светофоров в крупномасштабном сценарии моделирования движения транспортных средств / А.А. Агафонов, Е.Ю. Ефименко // *Сборник трудов по материалам VIII Международной конференции и молодежной школы. Информационные технологии и нанотехнологии (ИТНТ-2022)*. – 2022. – Т. 3. – С. 31382.
359. Агафонов, А.А. Метод адаптивного управления дорожными сигналами на основе максимального взвешенного потока / А.А. Агафонов, А.С. Юмаганов, В.В. Мясников // *Сборник трудов по материалам VIII Международной конференции и молодежной школы. Информационные технологии и нанотехнологии (ИТНТ-2022)*. – 2022. – Т. 3. – С. 31202.
360. Agafonov, A. Efficiency of Adaptive Traffic Signal Control in a Partially Connected Vehicle Environment / A. Agafonov, A. Yumaganov, V. Myasnikov // *2023 IX International Conference on Information Technology and Nanotechnology (ITNT)*. – Samara: Russian Federation, April, 2023. –P. 1–4.
361. Yumaganov, A. Cooperative Application of Vehicular Traffic Rerouting Method and Adaptive Traffic Signal Control Method / A. Yumaganov, A. Agafonov, V. Myasnikov // *2023 IX International Conference on Information Technology and Nanotechnology (ITNT)*. – Samara: Russian Federation, April, 2023. – P. 1–5.

362. Chollet, F. Keras // Keras: [сайт]. – 2015. – URL: <https://keras.io> (дата обращения 23.05.2023).
363. Kingma, D.P. Adam: A Method for Stochastic Optimization / D.P. Kingma, J. Ba. – arXiv . – 2017. – arXiv:1412.6980 [cs].
364. Saw, K. Literature Review of Traffic Assignment: Static and Dynamic / K. Saw, B.K. Katti, G. Joshi // International Journal of Transportation Engineering. – April, 2015. – Vol. 2.
365. Wang, H. Speed-Density Relationship: from Deterministic to Stochastic / H. Wang, J. Li, Q.Y. Chen // Transportation Research Board 88th Annual Meeting. – 2009. – P. 1–20.
366. (U.S.) NRC, editor. Highway capacity manual. Washington, D.C: Transportation Research Board, National Research Council. – 2000.
367. Recent Development and Applications of SUMO - Simulation of Urban MObility / D. Krajzewicz, J. Erdmann, M. Behrisch, L. Bieker . – December, 2012. – Vol. 5. – P. 128–138.
368. Deep Learning System for Vehicular Re-Routing and Congestion Avoidance / P. Perez-Murueta, A. Gómez-Espinosa, C. Cardenas, M. Gonzalez-Mendoza // Applied Sciences. – July, 2019. – Vol. 9. – 2717 p.
369. Krauss, S. Metastable states in a microscopic model of traffic flow / S. Krauss, P. Wagner, C. Gawron // Physical Review E. – May, 1997. – Vol. 55. – P. 5597–5602.
370. Sommer, C. Bidirectionally coupled network and road simulation for improved IVC analysis / C. Sommer, R. German, F. Dressler. // IEEE Transactions on Mobile Computing. – 2011. – Vol. 10. – P. 3–15.
371. Kamvar S.D. The Eigentrust algorithm for reputation management in P2P networks / S.D. Kamvar, M.T. Schlosser, H. Garcia-Molina // Proceedings of the 12th international conference on World Wide Web . – New York, NY, USA. 2003. – P. 640–651.
372. Plexe: A platooning extension for Veins / M. Segata, S. Joerer, B. Bloessl [et al.] // 2014 IEEE Vehicular Networking Conference (VNC) . – December, 2014. – P. 53–60.
373. Agafonov, A. GPU implementation of the stochastic on-time arrival routing algorithm / A. Agafonov, A. Maksimov, A. Borodinov // CEUR Workshop Proceedings. – 2020. – Vol. 2665. – P. 164–167.
374. Agafonov, A.A. Bus Arrival Time Prediction Using Recurrent Neural Network with LSTM Architecture / A.A. Agafonov, A.S. Yumaganov // Optical Memory and Neural Networks. – July, 2019. – Vol. 28. – P. 222–230.
375. Agafonov, A.A. Approximation of the road segments travel time using Levy distributions in the reliable shortest path problem / A.A. Agafonov, V.V. Myasnikov, A.I. Maksimov // Journal of Physics: Conference Series. – November, 2019. – Vol. 1368. – P. 032008.

376. Agafonov, A. Bus Arrival Time Prediction with LSTM Neural Network / A. Agafonov, A. Yumaganov // *Advances in Neural Networks – ISNN 2019*. – Moscow: Russia. – 2019. – Vol. 11554. – P. 11–18.
377. Агафонов, А.А. Многоагентная система маршрутизации транспортных средств в стохастической транспортной сети / А.А. Агафонов, В.В. Мясников, А.А. Бородин // *Труды XXI Международной конференции "Проблемы управления и моделирования в сложных системах"*. – Самара, 2019. – Т. 2. – С. 200–205.
378. Agafonov, A.A. Performance comparison of machine learning methods in the bus arrival time prediction problem / A.A. Agafonov, A.S. Yumaganov // *CEUR Workshop Proceedings*. – Samara, 2019. – Vol. 2416. – P. 57–62.
379. Agafonov, A. Stochastic on-time arrival problem with levy stable distributions / A. Agafonov, V. Myasnikov // *2019 4th International Conference on Intelligent Transportation Engineering (ICITE)*. – 2019. – P. 227–231.
380. Agafonov, A.A. Vehicle routing algorithms based on a route reservation approach / A.A. Agafonov, V.V. Myasnikov // *Journal of Physics: Conference Series*. – 2018. – Vol. 1096. – 012029 p.
381. Агафонов, А.А. Метод определения надежного кратчайшего пути в стохастической сети с использованием параметрически заданных устойчивых распределений вероятностей / А.А. Агафонов, В.В. Мясников // *Труды СПИИРАН*. – Июнь, 2019. – Т. 18. – С. 558–582.
382. Agafonov, A. A Route Reservation Approach for an Autonomous Vehicles Routing Problem / A. Agafonov, A. Borodinov // *MATEC Web of Conferences*. – 2018. – Vol. 220. – 02004 p.
383. Agafonov, A. Efficiency comparison of the routing algorithms used in centralized traffic management systems / A. Agafonov, V. Myasnikov // *Procedia Engineering*. – 2017. – Vol. 201. – P. 265–270.
384. Abdulganiev, I. Acceleration of the reliable shortest path algorithm in a time-dependent stochastic transport network / I. Abdulganiev, A. Agafonov // *CEUR Workshop Proceedings*. – 2017. – Vol. 1901. – P. 1–5.
385. Agafonov, A. Reliable routing in stochastic time-dependent network with the use of actual and forecast information of the traffic flows / A. Agafonov, V. Myasnikov // *IEEE Intelligent Vehicles Symposium, Proceedings*. – 2016. – Vol. 2016. – P. 1168–1172.
386. Агафонов, А.А. Алгоритм оценки времени прибытия общественного транспорта с использованием адаптивной композиции элементарных прогнозов / А.А. Агафонов, В.В. Мясников // *Компьютерная оптика*. – 2014. – Т. 38. – № 2. – С. 356–369.
387. Агафонов, А.А. Использование параметрически заданных устойчивых распределений вероятностей в задаче нахождения надежного кратчайшего пути / А.А. Агафонов, В.В.

- Мясников, А.И. Максимов // Сборник трудов V международной конференции и молодежной школы «Информационные технологии и нанотехнологии» (ИТНТ-2019) . – Самара, 2019. – Т. 2. – С. 556–562.
388. Агафонов, А.А. Исследование эффективности вычисления надежного кратчайшего пути с использованием GPU / А.А. Агафонов, А.И. Максимов, А.А. Бородин // Сборник трудов по материалам VI Международной конференции и молодежной школы в 4 т. – Самара, 2020. – Т. 2. – С. 732–738.
389. Агафонов, А.А., Мясников В.В. Маршрутизация автономных транспортных средств в зависящих от времени транспортных сетях // Информационные технологии и нанотехнологии (ИТНТ-2018): сборник трудов IV международной конференции и молодежной школы. – Самара, 2018. – С. 1205–1213.
390. Агафонов, А.А. Сравнение эффективности алгоритмов маршрутизации транспортных средств, используемых в централизованных системах управления / А.А. Агафонов, В.В. Мясников // Информационные технологии и нанотехнологии (ИТНТ-2017): сборник трудов III международной конференции и молодежной школы. – Самара, 2017. – С. 552–556.
391. Агафонов, А.А. Сравнение методов машинного обучения в задаче прогнозирования движения общественного транспорта / А.А. Агафонов, А.С. Юмаганов // Сборник трудов V международной конференции и молодежной школы «Информационные технологии и нанотехнологии» (ИТНТ-2019) . – Самара, 2019. – Т. 4. – С. 761–768.
392. Agafonov, A.A. Using satellite monitoring and statistical data to predict arrival time of city public transport / A.A. Agafonov, A.V. Chernov, A.V. Sergeev // Pattern Recognition and Image Analysis. – 2015. – Vol. 25. – P. 385–388.
393. Agafonov, A. Short-term Traffic Flow Prediction in a Partially Connected Vehicle Environment / A. Agafonov, V. Myasnikov // 2021 3rd International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA) . – 2021. – P. 968-972.
394. Agafonov, A. 3D Objects Detection in an Autonomous Car Driving Problem / A. Agafonov, A. Yumaganov // 2020 International Conference on Information Technology and Nanotechnology (ITNT) . – Samara: Russia, May, 2020. – P. 1–5.
395. Borodin, A. A Method of Preference and Utility Elicitation By Pairwise Comparisons and its Application to Intelligent Transportation Recommendation Systems / A. Borodin, A. Agafonov, V. Myasnikov // 2020 10th International Conference on Information Science and Technology (ICIST) . – September, 2020. – P. 77–85.
396. Yumaganov, A. An Improved Map Matching Algorithm Based on Dynamic Programming Approach. / A. Yumaganov, A. Agafonov, V. Myasnikov // In: Information Technology for

- Management: Towards Business Excellence. Cham: Springer International Publishing. – 2021. – Vol 413. – P. 87–102.
397. Kovalev, K. Authentication Scheme in Vehicular Ad Hoc Networks Based on Road Side Unit Infrastructure / K. Kovalev, A. Agafonov // 2021 International Conference on Information Technology and Nanotechnology (ITNT) . – September, 2021. – P. 1–4.
398. Abdulganiev, I. Automatic checking of road network models / I. Abdulganiev, A. Agafonov // CEUR Workshop Proceedings. – 2016. – Vol. 1638. – P. 249–255.
399. Agafonov, A. Determining the Optimal Number of Features for User Preference Elicitation in Recommender Systems / A. Agafonov, A. Yumaganov, V. Myasnikov // 2021 International Conference on Information Technology and Nanotechnology (ITNT) . – September, 2021. – P. 1–5.
400. Yumaganov, A., Map Matching Algorithm Based on Dynamic Programming Approach / A. Yumaganov, A. Agafonov, V. Myasnikov // 2020 15th Conference on Computer Science and Information Systems (FedCSIS) . – September, 2020. – P. 563–566.
401. Borodinov, A.A. Public transport route planning in the stochastic network based on the user individual preferences / A.A. Borodinov, A.S. Yumaganov, A.A. Agafonov // CEUR Workshop Proceedings. – Samara, 2019. – Vol. 2391. – P. 66–71.
402. Агафонов, А.А. Алгоритм оценки времени отправления общественного транспорта с использованием стратегий управления движением // Информационные технологии и нанотехнологии (ИТНТ-2018): сборник трудов IV международной конференции и молодежной школы. – Самара, 2018. – С. 1214–1220.
403. Абдулганиев, И.И. Исследование ускорения метода определения надёжного кратчайшего пути в зависящей от времени стохастической транспортной сети / И.И. Абдулганиев, А.А. Агафонов // Информационные технологии и нанотехнологии (ИТНТ-2017): сборник трудов III международной конференции и молодежной школы. – Самара, 2017. – С. 490–495.
404. Бородинов, А.А. Построение маршрута движения на общественном транспорте с учетом индивидуальных предпочтений участников движения / А.А. Бородинов, А.С. Юмаганов, А.А. Агафонов // Сборник трудов V международной конференции и молодежной школы «Информационные технологии и нанотехнологии» (ИТНТ-2019). – Самара, 2019. – Т. 2. – С. 342–348.
405. Агафонов, А.А. Сравнение методов детектирования трехмерных объектов в задаче автономного вождения транспортных средств / А.А. Агафонов, А.С. Юмаганов // Сборник трудов по материалам VI Международной конференции и молодежной школы в 4 т. – Самара, 2020. – Т. 2. – С. 277–284.
406. Ковалев, К.А. Схема аутентификации в сетях Vanet с использованием инфраструктуры придорожных блоков / К.А.Ковалев, А.А. Агафонов // Сборник

- трудов по материалам VII Международной конференции и молодежной школы. Информационные технологии и нанотехнологии (ИТНТ-2021) . – 2021. – Т. 2. – С. 23852.
407. Юмаганов, А.С. Оптимизация траектории движения транспортных средств в задаче управления транспортными потоками на перекрестке / А.С. Юмаганов, А.А. Агафонов // Сборник трудов по материалам VIII Международной конференции и молодежной школы. Информационные технологии и нанотехнологии (ИТНТ-2022) . – 2022. – Т. 3. – С. 31922.
408. Абдулганиев, И.И. Автоматическое тестирование модели транспортной сети / И.И. Абдулганиев, А.А. Агафонов // Материалы Международной конференции и молодежной школы. Материалы Международной конференции и молодежной школы. Информационные технологии и нанотехнологии (ИТНТ-2016) . – 2016. – С. 390–397.
409. Agafonov, A. Reconstruction of User Trips on Public Transport Using Indirect Information / A. Agafonov, V. Myasnikov // Proceedings of ITNT 2021 - 7th IEEE International Conference on Information Technology and Nanotechnology. – 2021.
410. Agafonov, A. Connected Vehicles Travel Time Prediction in a Scenario with Adaptive Traffic Light Control / A. Agafonov, E. Efimenko // 2023 IX International Conference on Information Technology and Nanotechnology (ITNT) . – Russian Federation: Samara, April, 2023. – P. 1–4.
411. Силен, Д. Основы Data Science и Big Data. Python и наука о данных/ Д. Силен, М. Али, М. Арно. – СПб: Питер, 2017. – С. 336.
412. Buyya, R. Big data: principles and paradigms / R. Buyya, R.N. Calheiros, A.V. Dastjerdi // Cambridge (MA): Elsevier/Morgan Kaufmann. – 2016.
413. Laney, D. 3D Data Management: Controlling Data Volume, Velocity, and Variety. – February 2001. – Vol. 6.
414. Batch processing// Microsoft: [сайт]. – 2023. – URL: <https://learn.microsoft.com/en-us/azure/architecture/data-guide/big-data/batch-processing> (дата обращения 27.04.2023).
415. Real-time processing // Microsoft: [сайт]. – 2023. – URL: <https://learn.microsoft.com/en-us/azure/architecture/data-guide/big-data/real-time-processing> (дата обращения 23.05.2023).
416. Осипов, Д.Л. Технологии проектирования баз данных. – М.: ДМК Пресс, 2019.
417. Изучаем Spark: молниеносный анализ данных / Х. Карау, Э. Конвински, П. Венделл, М. Захария. – М.: ДМК Пресс, 2015.
418. Narkhede, N. Kafka: the definitive guide: real-time data and stream processing at scale. / N. Narkhede, G. Shapira, T. Palino // First edition ed. Sebastopol (CA): O'Reilly Media, 2017.

419. Agafonov, A. Adaptive Traffic Light Control Using a Distributed Processing Approach // 2022 IEEE International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON) . – November, 2022. – P. 186–189.
420. Свидетельство о государственной регистрации программы для ЭВМ № 2016616120 Российская Федерация. Прогнозирование параметров транспортных потоков: № 2016613312: заявлено 08.04.2016 : опубликовано 06.06.2016 / Мясников В.В., Агафонов А.А.; заявитель СГАУ. – 1 с.
421. Свидетельство о государственной регистрации программы для ЭВМ № 2019610695 Российская Федерация. Программный модуль краткосрочного прогнозирования параметров транспортных потоков на основе технологии «Больших данных» с использованием данных реального времени и архивных данных о состоянии транспортных потоков: № 2018665023: заявлено 24.12.2018 : опубликовано 15.01.2019 / Агафонов, А.А., Юмаганов А.С., Мясников В.В., Бородинов А.А.; заявитель Самарский университет. – 1 с.
422. Свидетельство о государственной регистрации программы для ЭВМ № 2021666358 Российская Федерация. Программный модуль краткосрочного прогнозирования параметров смешанных транспортных потоков: № 2021665497 : заявлено 04.10.2021 : опубликовано 13.10.2021 / Агафонов А.А., Мясников В.В., Юмаганов А.С., Юзькив Р.Р. ; заявитель Самарский университет. – 1 с.
423. Свидетельство о государственной регистрации программы для ЭВМ № 2021666359 Российская Федерация. Программный модуль управления сигналами светофора на основе детерминированной прогнозной модели : № 2021665500: заявлено 04.10.2021: опубликовано 13.10.2021 / Агафонов А.А., Мясников В.В., Юмаганов А.С., Юзькив Р.Р. ; заявитель Самарский университет. – 1 с.
424. Свидетельство о государственной регистрации программы для ЭВМ № 2022663558 Российская Федерация. Программный модуль управления сигналами светофора с использованием модели глубокой нейронной сети: № 2022662536: заявлено 29.06.2022 : опубликовано 15.07.2022 / Агафонов А.А., Мясников В.В., Юмаганов А.С., Ефименко Е.Ю., Ганеева Ю.Х. ; заявитель Самарский университет. – 1 с.
425. Свидетельство о государственной регистрации программы для ЭВМ № 2022663570 Российская Федерация. Программный модуль нахождения рекомендованной траектории движения автономных транспортных средств с учетом информации о светофорном цикле: № 2022662537: заявлено 29.06.2022: опубликовано 15.07.2022 / Агафонов А.А., Мясников В.В., Юмаганов А.С., Козлов Д.А., Ганеева Ю.Х. ; заявитель Самарский университет. – 1 с.
426. Свидетельство о государственной регистрации программы для ЭВМ № 2020612379 Российская Федерация. Программный модуль расчета оптимального надежного пути на индивидуальном транспорте в стохастической сети с использованием

параметрически заданных устойчивых распределений вероятностей: № 2020611546: заявлено 14.02.2020 : опубликовано 20.02.2020 / Агафонов А.А., Мясников В.В. ; заявитель Самарский университет. – 1 с.

427. Свидетельство о государственной регистрации программы для ЭВМ № 2019660218 Российская Федерация. Программный модуль прогнозирования времени прибытия общественного транспорта с использованием рекуррентной нейронной сети: № 2019618913: заявлено 19.07.2019: опубликовано 02.08.2019 / Агафонов А.А., Юмаганов А.С., Бородинов А.А. ; заявитель Самарский университет. – 1 с.

ПРИЛОЖЕНИЕ А
ИСПОЛЬЗОВАНИЕ РЕЗУЛЬТАТОВ ДИССЕРТАЦИИ

УТВЕРЖДАЮ

Заместитель генерального директора
АО «Самара-Информспутник»

Н.И. Глумов



2023 г.

АКТ

об использовании результатов диссертации А.А. Агафонова
«Методы и алгоритмы обработки гетерогенной информации и адаптивного
управления в интеллектуальной транспортной системе»
в акционерном обществе «Самара-Информспутник»

Комиссия в составе ведущего инженера, к.т.н. А.В. Чернова и ведущего инженера-математика А.В. Сергеева, рассмотрев диссертацию А.А. Агафонова «Методы и алгоритмы обработки гетерогенной информации и адаптивного управления в интеллектуальной транспортной системе», представленную на соискание ученой степени доктора технических наук по специальности «2.3.1 – Системный анализ, управление и обработка информации, статистика», подтверждает, что разработанные в диссертационной работе алгоритмы и программные средства были использованы в составе программного обеспечения, разрабатываемого в рамках НИОКР по теме «Разработка информационного и программного обеспечения сервиса «робот-картограф» работы с большими геоданными» по договору №213ГРЦТС10-D5/65159 от 24 декабря 2020 года между акционерным обществом «Самара-информспутник» и федеральным государственным бюджетным учреждением «Фонд содействия развитию малых форм предприятий в научно-технической сфере» (Фонд содействия инновациям).

Применение указанных алгоритмов и программных средств позволило повысить эффективность решения задачи интеллектуального совмещения гетерогенных данных из различных источников и их совместного анализа, а также оценки транспортной доступности на личном и общественном транспорте.

Ведущий инженер, к.т.н.

А.В. Чернов

Ведущий инженер-математик

А.В. Сергеев

УТВЕРЖДАЮ

Генеральный директор
МКУ «Центр обеспечения»

Н.И. Волков

" 04 " сентября 2023г.


АКТ

об использовании результатов диссертации А.А. Агафонова

"Методы и алгоритмы обработки гетерогенной информации и адаптивного
управления в интеллектуальной транспортной системе"в муниципальном казенном учреждении городского округа Самара «Центр
обеспечения»

Комиссия в составе руководителя Управления информационных ресурсов и технологий Аппарата Администрации городского округа Самара А.Ю. Николаева и заместителя генерального директора МКУ «Центр обеспечения» А.Н. Илясова, рассмотрев диссертацию А.А. Агафонова «Методы и алгоритмы обработки гетерогенной информации и адаптивного управления в интеллектуальной транспортной системе», представляемую на соискание ученой степени доктора технических наук по специальности 2.3.1 – Системный анализ, управление и обработка информации, статистика, подтверждает, что разработанный в диссертационной работе алгоритм и программный модуль краткосрочного прогнозирования движения общественного транспорта были использованы в составе программного обеспечения интернет-портала Транспортного оператора Самары (www.tosamara.ru).

Применение указанных программных средств позволило решить задачу прогнозирования времени прибытия общественных транспортных средств на остановочные пункты на территории г.о. Самара.

Руководитель Управления информационных
ресурсов и технологий Аппарата
Администрации городского округа Самара

А.Ю. Николаев

Заместитель генерального директора
МКУ «Центр обеспечения»

А.Н. Илясов

УТВЕРЖДАЮ:

Первый проректор - проректор по научно-исследовательской работе федерального государственного автономного образовательного учреждения высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева»



 А.Б. Прокофьев

8 августа 2023 г.

АКТ

об использовании результатов диссертации А.А. Агафонова
«Методы и алгоритмы обработки гетерогенной информации и адаптивного управления в интеллектуальной транспортной системе»
в федеральном государственном автономном образовательном учреждении высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева» (Самарский университет)

Комиссия в составе научного руководителя научно-исследовательской лаборатории геоинформатики и информационной безопасности (НИЛ-55), д.т.н. В.В. Сергеева и директора института информатики и кибернетики, д.т.н. А.В. Куприянова, рассмотрев диссертацию А.А. Агафонова «Методы и алгоритмы обработки гетерогенной информации и адаптивного управления в интеллектуальной транспортной системе», представленную на соискание ученой степени доктора технических наук по специальности «2.3.1 – Системный анализ, управление и обработка информации, статистика», подтверждает, что разработанные в диссертационной работе методы, алгоритмы и программные модули были использованы:

- при выполнении федеральной целевой программы «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2014-2020 годы» по теме «Персональный цифровой автотранспортный помощник» (уникальный идентификатор проекта RFMEFI57518X0177) соглашение о предоставлении субсидий от 26 ноября 2018 г. № 075-02-2018-205;
- в проекте «Методы и алгоритмы совместного и координированного управления сигналами светофоров и подключенными автономными транспортными средствами в транспортной сети», финансируемого Российским научным фондом в рамках соглашения № 21-11-00321 от 19.04.2021 г. между Самарским университетом и Российским научным фондом (шифр темы 17Г-P023-055);
- в курсе лекций по дисциплине «Базы данных» по специальности 10.05.03 Информационная безопасность автоматизированных систем.

Научный руководитель НИЛ-55, д.т.н.



В.В. Сергеев

Директор института информатики и кибернетики, д.т.н.



А.В. Куприянов



УТВЕРЖДАЮ:

Руководитель Института систем
обработки изображений Российской
академии наук – филиала федерального
государственного учреждения
«Федеральный научно-
исследовательский центр
«Кристаллография и фотоника»
Российской академии наук»

Н.М. Казанский

01 сентября 2023г.

АКТ

об использовании результатов диссертационной работы А.А. Агафонова
«Методы и алгоритмы обработки гетерогенной информации и адаптивного
управления в интеллектуальной транспортной системе»
в Институте систем обработки изображений Российской академии наук —
филиале федерального государственного учреждения «Федеральный научно-
исследовательский центр «Кристаллография и фотоника» Российской академии
наук»

Комиссия в составе заместителя руководителя по науке и инновациям ИСОИ РАН — филиала ФНИЦ «Кристаллография и фотоника» РАН, д.ф.-м.н. Р.В. Скиданова и ученого секретаря ИСОИ РАН — филиала ФНИЦ «Кристаллография и фотоника» РАН, д.ф.-м.н. В.В. Котляра, рассмотрев диссертацию старшего научного сотрудника А.А. Агафонова «Методы и алгоритмы обработки гетерогенной информации и адаптивного управления в интеллектуальной транспортной системе», представленную на соискание ученой степени доктора технических наук по специальности «2.3.1 – Системный анализ, управление и обработка информации, статистика», подтверждает, что результаты диссертации были использованы при выполнении госзадания по теме НИР 42.1. «Разработка методов интеллектуального анализа и криптозащиты изображений в задачах обработки данных дистанционного зондирования Земли».

Заместитель руководителя по науке
и инновациям ИСОИ РАН — филиала ФНИЦ
«Кристаллография и фотоника» РАН,
д.ф.-м.н.



Р.В. Скиданов

Ученый секретарь ИСОИ РАН — филиала ФНИЦ
«Кристаллография и фотоника» РАН,
д.ф.-м.н.



В.В. Котляр