

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Казанский национальный исследовательский
технический университет–КАИ»

На правах рукописи



Мингалиев Заид Зульфатович

**МЕТОДЫ И МОДЕЛИ АДАПТИВНОГО ПРОГНОЗИРОВАНИЯ
НЕОДНОРОДНЫХ ВРЕМЕННЫХ РЯДОВ В УСЛОВИЯХ
ОГРАНИЧЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ РЕСУРСОВ**

1.2.2 Математическое моделирование,
численные методы и комплексы программ

Диссертация
на соискание ученой степени
кандидата технических наук

Научный руководитель:
доктор технических наук, доцент
Новикова Светлана Владимировна

Казань 2026

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	5
1 ОБЗОР СУЩЕСТВУЮЩИХ МОДЕЛЕЙ И МЕТОДОВ ДЛЯ КЛАСТЕРИЗАЦИИ И ПРОГНОЗИРОВАНИЯ ВРЕМЕННЫХ РЯДОВ.....	18
1.1 Кластеризация временных рядов: теоретические основы и классификация методов.....	19
1.1.1 Методы на основе измерения расстояния.....	20
1.1.2 Методы на основе вероятностных подходов.....	23
1.1.3 Методы на основе глубокого обучения.....	29
1.2 Прогнозирование временных рядов: теоретические основы и классификация.....	32
1.2.1 Классические статистические методы.....	33
1.2.2 Методы машинного обучения.....	35
1.2.3 Глубокие нейросетевые модели.....	37
1.3. Постановка задачи исследования.....	40
1.4. Выводы.....	41
2 МЕТОДЫ И МОДЕЛЬ АДАПТИВНОГО ПРОГНОЗИРОВАНИЯ ВРЕМЕННЫХ РЯДОВ.....	42
2.1 Метод адаптивной кусочно-линейной аппроксимации временных рядов с динамическим выбором шага.....	43
2.1.1 Формальная постановка задачи.....	44
2.1.2 Описание алгоритма.....	45
2.1.3 Анализ вычислительной сложности.....	47
2.2 Метод кластеризации временных рядов на основе вычисления статических и динамических характеристик.....	47
2.2.1 Этап предобработки данных.....	48

2.2.2	Вычисление временных характеристик	48
2.2.3	Нормализация и кластеризация	52
2.3	Гибридная нейросетевая модель прогнозирования временных рядов ...	53
2.3.1	Предварительная обработка временных рядов	54
2.3.2	Кластеризация временных рядов.....	54
2.3.3	Обучение специализированных нейросетевых моделей.....	55
2.3.4	Прогнозирование новых временных рядов	55
2.4	Выводы.....	57
3	ПРОГРАММНЫЙ КОМПЛЕКС РЕАЛИЗАЦИИ РАЗРАБОТАННЫХ МЕТОДОВ И АЛГОРИТМОВ	59
3.1	Адаптивный модуль аппроксимации	60
3.2	Автоматический синтаксический анализатор данных биологического мониторинга.....	64
3.3	Программный комплекс модели для прогнозирования с использованием кластеризации временных рядов.....	67
3.4	Выводы.....	71
4	ВЫЧИСЛИТЕЛЬНЫЕ ЭКСПЕРИМЕНТЫ ПО ПРАКТИЧЕСКОМУ ИСПОЛЬЗОВАНИЮ РАЗРАБОТАННЫХ МЕТОДОВ, МОДЕЛИ И АЛГОРИТМОВ.....	73
4.1	Вычислительные эксперименты по применению метода адаптивной кусочно-линейной аппроксимации временных рядов.....	73
4.1.1	Кусочно-линейная аппроксимация сложных траекторий управления с переменным шагом на примере управления беспилотными летательными подводными аппаратами.....	75

4.1.2 Кусочно-линейная аппроксимация нелинейной функции, описывающей изменение концентрации экзогенного инсулина в плазме крови.....	89
4.1.3 Кусочно-линейная аппроксимация траектории движения автомобиля для цифрового имитационного моделирования транспортных процессов	95
4.2 Вычислительные эксперименты по применению метода адаптивной кластеризации и гибридной модели прогнозирования временных рядов ...	99
4.2.1 Применение для формирования нейроконтроллера, обеспечивающего устойчивость БПЛА в условиях атмосферной турбулентности.....	101
4.2.2 Применение для прогнозирования уровня гликемии пациентов с сахарным диабетом	122
4.3 Выводы.....	135
ЗАКЛЮЧЕНИЕ	137
СПИСОК ЛИТЕРАТУРЫ	140
ПРИЛОЖЕНИЯ.....	155

ВВЕДЕНИЕ

Актуальность темы. Прогнозирование временных рядов – одна из центральных задач математического моделирования: оно применяется всюду, где нужно анализировать изменение показателей во времени и предсказывать их дальнейшее поведение.

На практике данные редко обладают идеальными свойствами: временным рядам присущи изменчивость, шумы, пропуски и сложные нелинейные зависимости между наблюдениями, а параметры самого процесса нередко меняются со временем, из-за чего модель, хорошо работающая на одном участке данных, теряет точность на другом. Совокупность этих факторов существенно усложняет построение надежных моделей прогнозирования.

Помимо точности прогноза, от модели требуется способность адаптироваться к изменяющимся условиям, сохранять устойчивость при наличии шумов и укладываться в ограниченное время вычислений. Особенно это важно для систем поддержки принятия решений, работающих в режиме реального времени.

Среди современных инструментов прогнозирования хорошо зарекомендовали себя рекуррентные нейронные сети, в частности сети с долговременной краткосрочной памятью (LSTM-модели): благодаря способности учитывать долгосрочные временные зависимости они во многих задачах превосходят классические статистические методы. Однако единая нейросетевая модель не всегда обеспечивает требуемое качество прогнозирования – при работе с разнородными по структуре и характеру поведения временными рядами универсальные модели теряют точность, а их обучение и использование требует значительных вычислительных ресурсов.

Один из способов повысить качество прогнозирования в такой ситуации – предварительная структуризация временных рядов, в частности их кластеризация по динамическим и статистическим характеристикам, а также снижение размерности данных за счет адаптивной аппроксимации. Именно

сочетание кластеризации, предварительной обработки и специализированных прогнозных моделей позволяет учитывать неоднородность данных, повышать устойчивость прогнозов и снижать вычислительную сложность алгоритмов.

В диссертации разработанные методы апробированы на двух прикладных задачах – физиологических временных рядах, включая данные непрерывного мониторинга уровня глюкозы у пациентов с сахарным диабетом, и временных рядах оптимального управления беспилотными системами в условиях внешних помех. Эти области выбраны как показательные, но не единственно возможные: предложенные модели и алгоритмы ориентированы на широкий класс задач прогнозирования временных рядов и могут быть применены в различных областях науки и техники.

Отсюда вытекает научная задача диссертационного исследования – разработка методов и моделей адаптивного прогнозирования временных рядов, основанных на их предварительной кластеризации и использовании нейросетевых архитектур и обеспечивающих одновременно повышение точности прогнозов и снижение затрат на вычислительные операции.

Степень разработанности темы. Существующие методы прогнозирования временных рядов принято делить на три группы: классические статистические методы, алгоритмы машинного обучения и модели глубокого обучения.

Классические статистические методы – модели ARIMA, экспоненциальное сглаживание, фильтр Калмана – долгое время оставались основным инструментом анализа и прогнозирования временных рядов; основы построения моделей линейных временных зависимостей заложены еще в работах Бокса и Дженкинса [1] и Калмана [2]. При этом, как показывают Хайндман и Афанасопулос [3], при выраженной нелинейности, высокой изменчивости или смене характера процесса точность подобных моделей снижается.

Более высокую гибкость в задачах прогнозирования дают методы машинного обучения – градиентный бустинг, случайные леса, метод опорных

векторов. В работах Бреймана [4], Фридмана [5] и Кушнарера [6] показано, что подобные алгоритмы способны выявлять сложные закономерности, особенно при большом количестве признаков, но их эффективность во многом определяется качеством подготовки данных: для обучения, как правило, требуются значительные объемы данных и их предварительная обработка.

Высокую точность прогнозирования временных рядов на сегодняшний день показывают глубокие нейросетевые модели – LSTM (Хохрайтер и Шмидхубер [7]), сверточные нейросети (Лекун [8]) и скрытые марковские модели (Рабинер [9]). Так, Чжу и Перес-Гандия в своей работе [10] показали высокую эффективность LSTM-моделей при предсказании уровня глюкозы с учетом предыдущих измерений, данных о питании и физической активности. Ограничением для выбора глубоких нейросетевых моделей является высокая вычислительная сложность, что затрудняет применение таких моделей в условиях реального времени на маломощных устройствах.

Сохраняется и проблема адаптации моделей к индивидуальным особенностям данных: универсальные модели не всегда способны учитывать специфику конкретного процесса, из-за чего их эффективность может существенно различаться в зависимости от характера входных данных. Это в равной мере характерно как для прогнозирования физиологических показателей – в частности, уровня глюкозы у пациентов с сахарным диабетом, так и для задач оптимального управления динамическими объектами, включая беспилотные летательные аппараты, где динамика данных определяется режимами функционирования и внешними возмущениями. Для таких случаев разработка адаптивных методов прогнозирования, учитывающих характерные паттерны поведения конкретных временных рядов, представляется актуальной задачей.

Дополнительная сложность связана с самими данными, поступающими от внешних сенсоров: они могут содержать пропуски и артефакты, регистрироваться с разной частотой. Эти вопросы поднимаются, в частности, в исследованиях Бунеску и Дзеккин [11], где рассматриваются методы

предварительной обработки данных, полученных с внешних сенсорных устройств.

В настоящей работе предлагается комплексный подход к прогнозированию временных рядов, ориентированный на повышение точности и адаптивности предсказаний за счет предварительной кластеризации временных рядов: она позволяет учитывать структурные особенности данных, формировать более однородные группы наблюдений и снижать вычислительную сложность анализа благодаря обучению специализированных прогнозных моделей для каждого кластера.

В связи с этим **научной задачей**, решаемой в диссертации, является разработка методов и моделей адаптивного прогнозирования временных рядов на основе глубокого обучения, а также программного комплекса для их практической реализации.

Объект исследования: временные ряды, характеризующие динамику сложных процессов.

Предмет исследования: методы и модели адаптивного прогнозирования временных рядов на основе глубокого обучения.

Целью работы является повышение точности и адаптация прогнозов к индивидуальной динамике временных рядов.

Для достижения поставленной цели в работе решаются следующие **задачи**:

- 1) анализ и обзор современных методов прогнозирования временных рядов, включая статистические, машинно-обучающие и нейросетевые подходы;
- 2) разработка метода адаптивной кусочно-линейной аппроксимации временных рядов с динамическим выбором шага;
- 3) разработка метода кластеризации временных рядов на основе вычисления статических и динамических характеристик, отражающих их динамическую структуру;

4) разработка модели адаптивного прогнозирования временных рядов на основе LSTM в комбинации с методом кластеризации временных рядов и методов кусочно-линейной аппроксимации;

5) реализация программного комплекса, обеспечивающего полный цикл обработки временных рядов: предобработку, кластеризацию и прогнозирование;

б) проведение вычислительных экспериментов для оценки точности, устойчивости и вычислительной эффективности разработанных методов;

7) апробация предложенных решений на реальных временных рядах в различных предметных областях.

Соответствие диссертации паспорту научной специальности. В диссертации разработаны и реализованы в виде комплекса программ модели и методы предварительной обработки и прогнозирования временных рядов. Для оценки их адекватности и эффективности приведены результаты вычислительных экспериментов.

Следовательно, результаты диссертационного исследования соответствуют следующим пунктам паспорта научной специальности 1.2.2. Математическое моделирование, численные методы и комплексы программ:

п 2. Разработка, обоснование и тестирование эффективных вычислительных методов с применением современных компьютерных технологий: разработаны, обоснованы и протестированы вычислительные методы адаптивной кластеризации и кусочно-линейной аппроксимации временных рядов;

п 3. Реализация эффективных численных методов и алгоритмов в виде комплексов проблемно-ориентированных программ для проведения вычислительного эксперимента: реализация разработанных методов и алгоритмов в виде комплекса программ, на базе которого проведены вычислительные эксперименты по оценке эффективности разработанного алгоритмического обеспечения и возможности его практического использования для решения поставленных задач;

п 8. Комплексные исследования научных и технических проблем с применением современной технологии математического моделирования и вычислительного эксперимента: разработана гибридная нейросетевая модель прогнозирования временных рядов с применением адаптивных методов предварительной кластеризации и кусочно-линейной аппроксимации;

п 9. Постановка и проведение численных экспериментов, статистический анализ их результатов, в том числе с применением современных компьютерных технологий: проведены численные эксперименты с последующим статистическим анализом их результатов для оценки эффективности разработанных методов и алгоритмов.

Новизна полученных результатов заключается в следующем:

Научная новизна работы заключается в интеграции методов кластеризации временных рядов с глубокими нейросетевыми моделями, оптимизированными для работы в условиях ограниченных вычислительных ресурсов, а именно:

1. Разработан метод адаптивной кластеризации временных рядов, базирующийся на анализе их статистических и динамических характеристик, отличающийся возможностью выявления однородных групп временных рядов с учетом индивидуальных и контекстно-зависимых особенностей их динамики, что позволяет повысить точность последующего прогнозирования (п. 2 паспорта научной специальности 1.2.2).

2. Разработан метод адаптивной кусочно-линейной аппроксимации временных рядов, базирующийся на сегментации данных с переменным шагом по заданному критерию точности, отличающийся снижением размерности данных при сохранении их структурных свойств, что позволяет уменьшить вычислительные затраты на последующих этапах обработки и прогнозирования (п. 2 паспорта научной специальности 1.2.2);

3. Разработана гибридная нейросетевая модель прогнозирования временных рядов, базирующаяся на интеграции LSTM-архитектур, методов кластеризации и адаптивной аппроксимации, отличающаяся использованием

предварительной структуризации данных для формирования скрытых представлений, что позволяет повысить устойчивость прогнозов к шумам и пропускам, а также снизить вычислительную сложность модели (п. 8 паспорта научной специальности 1.2.2);

4. Разработан программный комплекс для прогнозирования временных рядов, базирующийся на интеграции методов кластеризации, адаптивной кусочно-линейной аппроксимации и гибридных нейросетевых моделей, отличающийся модульной архитектурой и возможностью настройки под различные классы временных данных, что позволяет реализовать адаптивный подход к прогнозированию в системах реального времени (п. 3 паспорта научной специальности 1.2.2);

5. Проведены численные эксперименты по решению задач прогнозирования неоднородных временных рядов, базирующиеся на использовании синтетических и реальных наборов данных с последующим статистическим анализом результатов, отличающиеся комплексной оценкой точности, устойчивости и вычислительной эффективности моделей, что позволяет обоснованно оценить применимость и эффективность разработанных методов и алгоритмов для решения поставленных задач (п. 9 паспорта научной специальности 1.2.2);

В отличие от существующих универсальных моделей прогнозирования, предложенный подход обеспечивает учет структурной неоднородности временных рядов, снижение вычислительных затрат и повышение устойчивости прогнозов в условиях шумов и пропусков данных.

Теоретическая значимость работы состоит в развитии методов анализа и прогнозирования временных рядов с учетом их структурных и вероятностных свойств, а также в обосновании эффективности предварительной кластеризации для повышения качества нейросетевого прогнозирования.

Практическая значимость заключается в возможности применения разработанных методов в интеллектуальных системах прогнозирования и

поддержки принятия решений в различных предметных областях. В качестве прикладного примера показана эффективность предложенного подхода при прогнозировании физиологических временных рядов, включая данные мониторинга уровня глюкозы, а также рядов оптимального управления беспилотными аппаратами.

Методы исследования. Для решения поставленных задач в диссертационной работе использовались предиктивные методы аппроксимации и регрессии, математической статистики, математического и нейросетевого моделирования, кластерного анализа, а также методы оптимизации и объектно-ориентированного программирования.

Положения, выносимые на защиту:

1. Метод адаптивной кластеризации временных рядов, позволяющий группировать временные ряды различной длины и структуры для повышения точности последующих прогнозов.

2. Метод адаптивной кусочно-линейной аппроксимации временных рядов, позволяющий уменьшить вычислительные затраты при прогнозировании.

3. Гибридная нейросетевая модель прогнозирования временных рядов, снижающая вычислительную сложность и повышающая устойчивость прогнозов к шумам и пропускам.

4. Архитектура оригинального комплекса программ, позволяющая реализовать разработанные модели, методы и алгоритмы.

Степень достоверности и апробация результатов. Достоверность научных результатов подтверждена вычислительными экспериментами и результатами практического использования. Диссертационная работа выполнена при поддержке Министерства науки и высшего образования в рамках исполнения обязательств по НИОКТР 126020516513–4, рег. номер FZSU-2026–0007.

Апробация работы. Основные положения и результаты диссертационной работы докладывались и обсуждались на следующих конференциях:

1) Современные цифровые технологии: проблемы, решения, перспективы: национальная (с международным участием) научно-практическая конференция, Казань, 19–20 мая 2022 года;

2) Цифровые системы и модели: теория и практика проектирования, разработки и использования: Международная научно-практическая конференция, Казань, 10–11 апреля 2025 года;

3) Профессиональные коммуникации в научной среде - фактор обеспечения качества исследований: XV Всероссийская научно-практическая конференция, Альметьевск, 16 апреля 2025 года.

4) XXVI Туполевские чтения (школа молодых ученых): Международная молодёжная научная конференция, Казань, 09–10 ноября 2023 года.

5) Приоритетные направления инновационной деятельности в промышленности: сборник научных статей по итогам девятой международной научной конференции, Казань, 29–30 сентября 2020 года.

6) Кулагинские чтения: техника и технологии производственных процессов: XX Международная научно-практическая конференция, Чита, 30 ноября – 04 декабря 2020 года.

Реализация результатов диссертации:

1) внедрены в ФГБОУ ВО «Казанский национальный исследовательский технический университет им. А.Н. Туполева-КАИ» и используются в учебном процессе при изучении дисциплин для подготовки магистров по направлению «Прикладная математика и информатика» при проведении лекционных занятий и лабораторных работ по дисциплинам «Компьютерное моделирование процессов и систем», «Теория принятия решений» и «Теория игр и исследование операций»;

2) использованы в работе Сектора безопасности дорожного движения и контрольно-надзорной деятельности Министерства Транспорта и дорожного

хозяйства Республики Татарстан в системе поддержки принятия решений для моделирования траекторий движения транспортных потоков на дорожных развязках сложных конфигураций на территории Республики Татарстан;

3) использованы при реализации научного направления деятельности Центра разработки и сопровождения информационных систем ГБУ «Безопасность дорожного движения» при разработке научных и практических основ моделирования транспортных потоков;

4) использованы в работе отделения эндокринологии в системе поддержки принятия решений для прогнозирования уровня гликемии у больных инсулинозависимым диабетом, получающих помповую инсулинотерапию;

5) использованы в работе Научно-производственного центра беспилотных авиационных систем (НПЦ БАС) Татарстана для имитации сложных траекторий движения и моделирования управления беспилотных летательных аппаратов.

Публикации. Материалы диссертации опубликованы в 10 научных работах общим объемом 4,38 п.л. (в т.ч. лично автором – 3,1 п.л.), из них 5 статей в журналах из перечня рецензируемых научных изданий, рекомендованных высшей аттестационной комиссией Министерства науки и высшего образования Российской Федерации – «Современные информационные технологии и ИТ-образование», «Вестник Тверского государственного университета», «Вестник Технологического университета». Зарегистрировано 4 свидетельства о государственной регистрации программ для ЭВМ (модули «Адаптивный модуль аппроксимации», «Программа анализа и прогнозирования смешанных временных рядов», «Программа кластеризации и адаптивного прогнозирования временных рядов на основе LSTM», «Программа синтаксического анализа больших данных биологического мониторинга»).

Структура и объем диссертации. Диссертация изложена на 163 страницах машинописного текста, содержит 38 рисунков, 33 таблицы, состоит

из введения, четырех глав, заключения, списка использованной литературы из 118 наименований и приложений.

Сведения о личном вкладе автора. Постановка научной задачи, содержание диссертации и все представленные в ней результаты получены лично автором. Подготовка к публикации некоторых результатов проводилась совместно с соавторами, причем вклад диссертанта определяющий.

Во введении обоснована актуальность темы исследования, сформулированы научная задача, цель и задачи работы, а также изложены основные положения, выносимые на защиту.

В первой главе проведен обзор существующих моделей и методов кластеризации и прогнозирования временных рядов: рассмотрены как классические, так и современные подходы с учетом специфики данных – их высокой изменчивости, пропущенных значений и малой информативности отдельных измерений. По результатам анализа преимуществ и ограничений различных методов выявлены пробелы в применении существующих подходов к прогнозированию неоднородных временных рядов, что позволило обосновать необходимость специализированных алгоритмов и программного обеспечения для их адаптивной кластеризации и прогнозирования с учетом индивидуальных и контекстно-зависимых особенностей данных; на этой основе сформулированы задачи исследования.

Во второй главе представлены разработанные методы кластеризации и прогнозирования временных рядов. Предложен метод адаптивной кусочно-линейной аппроксимации с переменным шагом, снижающий вычислительные затраты при сохранении заданной точности, а также метод кластеризации на основе статистических и динамических характеристик с использованием алгоритма DBSCAN, позволяющий формировать однородные группы данных. Для прогнозирования разработана адаптивная LSTM-модель, обучаемая отдельно для каждого кластера; эффективность такого подхода подтверждена сравнительным анализом, результаты которого приведены в главе 4.

В третьей главе представлены разработанные программные комплексы, реализующие методы и модели кластеризации и прогнозирования временных рядов. Для апробации метода расчета переменного шага создан адаптивный модуль аппроксимации, поддерживающий как жадный, так и адаптивный режимы разбиения функций на сегменты с сохранением заданной точности; реализован также программный комплекс для прогнозирования временных рядов с предварительной кластеризацией, обеспечивающий модульность, высокую производительность и возможность адаптации под различные задачи и типы временных данных.

В четвертой главе описаны вычислительные эксперименты, проведенные для проверки применимости и точности разработанных методов и алгоритмов в различных предметных областях.

В разделе 4.1 описываются вычислительные эксперименты по применению метода адаптивной кусочно-линейной аппроксимации временных рядов:

- для законов управления гибридным летательно-подводным аппаратом типа окто-квадрокоптер;
- нелинейной функции, описывающей динамику концентрации экзогенного инсулина в плазме крови пациента с сахарным диабетом 1 типа;
- траектории движения автомобиля для цифрового имитационного моделирования.

В разделе 4.2. описаны эксперименты по применению разработанного метода адаптивной кластеризации и гибридной модели прогнозирования временных рядов. Проведены вычислительные эксперименты по формированию многопоточного нейросетевого контроллера для расчёта управляющих параметров автопилота легкого самолета при развороте, а также по прогнозированию уровня глюкозы в крови пациентов, больных сахарным диабетом 1 типа, на основе данных инвазивных сенсоров инсулиновых помп для смешанного временного ряда, состоящего из значений уровня глюкозы крови, активного инсулина в крови и активных углеводов. Экспериментально

подтверждена вычислительная эффективность предложенного программного комплекса, что делает возможным его применение в системах реального времени и в условиях ограниченных вычислительных ресурсов.

В заключении сформулированы основные научные и практические результаты работы, намечены направления перспективных исследований.

В приложении 1 представлены акты о внедрении и использовании результатов диссертационного исследования.

В приложении 2 представлены свидетельства о регистрации программ для ЭВМ.

1 ОБЗОР СУЩЕСТВУЮЩИХ МОДЕЛЕЙ И МЕТОДОВ ДЛЯ КЛАСТЕРИЗАЦИИ И ПРОГНОЗИРОВАНИЯ ВРЕМЕННЫХ РЯДОВ

Временные ряды являются одной из наиболее распространенных форм представления данных в различных прикладных областях: от анализа финансовых показателей и мониторинга промышленного оборудования до диагностики заболеваний и управления движением беспилотных аппаратов.

Неоднородные временные ряды особенно сложны для анализа – в них динамика всё время меняется. Например, у беспилотника полёт может идти спокойно, а потом резко меняются условия – ветер усиливается или аппарат переходит на другой режим работы. У пациента с диабетом картина похожая: уровень глюкозы долго держится в норме, а потом вдруг начинает стремительно падать. Если пытаться описать такие разные участки одной и той же моделью, она будет ошибаться – ведь то, что верно для спокойного режима, не годится для резкого скачка. Поэтому тут не обойтись без адаптивных методов: они должны подстраиваться под локальные особенности и учитывать, что на каждом участке поведение данных своё.

В этой главе проводится обзор существующих методов кластеризации и прогнозирования временных рядов. Рассматриваются три основных класса подходов: методы на основе измерения расстояния (евклидово расстояние, динамическая трансформация временной шкалы, корреляционные меры), вероятностные модели (авторегрессионные процессы, скрытые марковские модели, байесовские подходы) и методы глубокого обучения (рекуррентные и сверточные нейронные сети, вариационные автокодировщики). Для каждого из направлений анализируются сильные и слабые стороны с точки зрения работы с неоднородными данными и вычислительной эффективности, что формирует теоретическую базу для разработки оригинальных методов, представленных в Главе 2.

1.1 Кластеризация временных рядов: теоретические основы и классификация методов

Кластеризация временных рядов представляет собой одну из ключевых задач анализа данных, заключающуюся в группировке временных последовательностей на основе их структурных и статистических характеристик [12]. Данный процесс позволяет выявлять сходные закономерности в данных, что упрощает последующий анализ, обработку и прогнозирование временных рядов [13]. Основная цель кластеризации – разделение совокупности временных рядов на группы, внутри которых объекты обладают схожими свойствами, а различия между группами являются значимыми [13].

Временные ряды, как объект кластерного анализа, характеризуются динамическими изменениями значений во времени, что накладывает дополнительные ограничения на применение традиционных методов кластеризации, разработанных для статических данных. В связи с этим в научной литературе предлагаются специализированные методы кластеризации временных рядов, учитывающие их временную структуру, корреляционные зависимости, наличие сезонности и стохастических компонент [13].

Современные подходы к кластеризации временных рядов можно классифицировать по нескольким признакам, включая тип используемой метрики расстояния, метод построения кластеров и способ представления временного ряда. В соответствии с этими характеристиками в литературе [31] выделяются следующие категории методов кластеризации:

- 1) методы на основе измерения расстояния;
- 2) модели на основе вероятностных подходов;
- 3) методы на основе машинного обучения.

Данная классификация позволяет систематизировать существующие подходы и определить область их применения в зависимости от структуры данных и целей анализа. В дальнейшем рассмотрении будет проведен

детальный обзор каждой из представленных групп методов с акцентом на их применимость к задачам прогнозирования временных рядов.

1.1.1 Методы на основе измерения расстояния

Методы кластеризации временных рядов, основанные на измерении расстояния, представляют собой один из наиболее распространенных классов подходов к группировке временных последовательностей [12]. Их основная идея заключается в вычислении меры схожести (или различия) между парами временных рядов и последующем объединении схожих рядов в группы [13]. Качество кластеризации в данных методах во многом определяется выбором метрики расстояния, которая должна адекватно отражать особенности временной динамики исследуемых процессов [31].

В практическом применении наибольшее распространение получили следующие меры расстояния:

- 1) евклидово расстояние;
- 2) расстояние с использованием динамического временного искажения;
- 3) корреляционные меры расстояния [13].

Выбор конкретной метрики определяется особенностями данных и целями исследования [31]; ниже эти метрики рассматриваются подробнее – с точки зрения точности, вычислительных затрат и применимости к неоднородным временным рядам.

Одной из самых простых и распространённых мер является евклидово расстояние – оно определяется как корень из суммы квадратов разностей между соответствующими точками двух рядов [17]:

$$d(X, Y) = \sqrt{\sum_{i=1}^N (x_i - y_i)^2} \quad (1)$$

где

$X = (x_1, x_2, \dots, x_N)$ и $Y = (y_1, y_2, \dots, y_N)$ – два временных ряда длины N ;

x_i и y_i – значения временных рядов в момент времени i .

Гораздо более гибкий инструмент сравнения – динамическая трансформация временной шкалы (DTW): в отличие от евклидова расстояния, она умеет адаптивно выравнивать ряды и точнее улавливать их структурное сходство [13, 18].

В основе DTW лежит динамическое программирование: ось времени одного ряда деформируется относительно другого так, чтобы минимизировать сумму различий между соответствующими точками [15], что особенно полезно при смещениях, растяжениях или сжатиях временной шкалы [19]. Для этого строится матрица стоимости выравнивания, где каждый элемент отражает разницу между значениями точек двух рядов [20]; оптимальное выравнивание находится по рекуррентному соотношению:

$$D(i, j) = d(x_i, y_i) + \min(D(i - 1, j), D(i, j + 1), D(i - 1, j - 1)) \quad (2)$$

где:

- $D(i, j)$ – минимальная стоимость пути до точки (i, j) ;
- $d(x_i, y_i)$ – расстояние между значениями точек x_i и y_i ;
- $\min(D(i - 1, j), D(i, j + 1), D(i - 1, j - 1))$ – выбор наилучшего пути

на основе предыдущих значений.

Когда матрица построена, остаётся найти путь минимальной стоимости. Он и задаёт итоговое соответствие между точками двух рядов.

Наконец, третья группа мер – корреляционные меры расстояния: в отличие от подходов, сравнивающих абсолютные значения, они оценивают, насколько согласованно меняются два ряда во времени [23, 24].

Одной из корреляционных мер является попарная корреляция Пирсона, которая вычисляется по следующей формуле:

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}} \quad (3)$$

где:

x_i и y_i – значения элементов временного ряда X и Y в момент времени i ;

\bar{x} и \bar{y} – средние значения рядов X и Y соответственно.

Значение коэффициента корреляции r находится в диапазоне от -1 до 1 , где $r = 1$ означает полную положительную линейную зависимость, $r = -1$ – полную отрицательную зависимость, а $r = 0$ означает отсутствие линейной связи между рядами [23].

Для учета эффекта запаздывания изменений во временных рядах может применяться кросс-корреляция [25], которая вычисляется по формуле:

$$r = \frac{\sum_{i=1}^{n-k} (x_i - \bar{x})(y_{i+k} - \bar{y})}{\sqrt{\sum_{i=1}^{n-k} (x_i - \bar{x})^2 \sum_{i=1}^{n-k} (y_{i+k} - \bar{y})^2}} \quad (4)$$

где r_k – это корреляция для сдвига k , а остальные параметры аналогичны корреляции Пирсона [25].

Они просты и вычислительно нетребовательны [26], поэтому хорошо подходят для предварительного анализа данных и обработки больших массивов информации [27]. Сравнительный анализ методов приведен в таблице 1.

Таблица 1 – Сравнение мер расстояния для кластеризации временных рядов

Мера	Преимущества	Недостатки
Евклидово расстояние	Простота вычислений и низкая сложность; понятный принцип расчёта; высокая эффективность для рядов одинаковой длины без сдвигов [18]	Высокая чувствительность к временным сдвигам [18]; требует приведения рядов к одинаковой длине; слабо учитывает форму ряда при деформациях [17]
DTW	Учитывает временные смещения и разную скорость протекания процессов [13]; устойчива к растяжениям, сжатиям и сдвигам; допускает сравнение рядов разной длины [21]	Высокая вычислительная сложность [22]; чувствительность к выбросам и шуму; зависимость от параметров ограничения пути; не всегда очевидная интерпретация [18]
Корреляционные меры	Простота и низкая вычислительная сложность [26]; удобны для предварительного анализа больших массивов данных [27]	Чувствительность к выбросам и шуму [23]; не отражают структуру ряда при нелинейной связи [24]

1.1.2 Методы на основе вероятностных подходов

Вероятностные методы применяются при кластеризации временных рядов, поскольку позволяют учитывать случайный характер данных, наличие шумов и изменчивость исследуемых процессов [28]. В отличие от методов, основанных на непосредственном сравнении рядов, они описывают механизм формирования данных и объединяют объекты в группы по схожести статистических характеристик [28] – это особенно ценно, когда ряды содержат пропуски, аномалии или смену тенденций, которые трудно уловить традиционными метриками расстояния [29].

Основная идея вероятностного подхода в том, что каждый временной ряд рассматривается как реализация некоторого случайного процесса [30]. Для его описания используются разные математические модели, выбор которых определяется особенностями исследуемых данных; наиболее распространены среди них авторегрессионные модели, скрытые марковские модели и байесовский подход [31].

У вероятностных методов есть очевидный плюс – они умеют учитывать неопределённость, случайные колебания и скрытые закономерности в данных, что делает анализ более гибким [28, 30]. Однако обучение таких моделей может быть ресурсоёмким, особенно при многомерных распределениях или большом числе параметров [31].

Автокорреляционные модели могут выявлять зависимости между значениями ряда и его предыдущими состояниями, что позволяет эффективно описывать временные зависимости, что важно для кластеризации, когда необходимо разделить ряды на группы, обладающие схожими динамическими характеристиками [32, 33]. Суть таких моделей в довольно простой идее: текущее значение ряда можно выразить через комбинацию предыдущих значений и случайную погрешность [34]. Это позволяет эффективно прогнозировать поведение ряда на основе его предыдущих состояний и, в

контексте кластеризации, использовать эти прогнозы для классификации и группировки временных рядов [35].

К основным типам автокорреляционных моделей относятся:

- 1) модель авторегрессии (AR);
- 2) модель скользящего среднего (MA);
- 3) модель авторегрессии и скользящего среднего (ARMA);
- 4) модель авторегрессии и интегрированного скользящего среднего (ARIMA).

Модель авторегрессии описывает текущее значение временного ряда как линейную комбинацию его предыдущих значений. Формула для модели авторегрессии порядка p имеет следующий вид:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} + \varepsilon_t \quad (5)$$

где:

- X_t – текущее значение временного ряда;
- $\phi_1, \phi_2, \dots, \phi_p$ – коэффициенты авторегрессии;
- ε_t – случайная ошибка (обычно принимается, что ошибка имеет нормальное распределение с нулевым математическим ожиданием и постоянной дисперсией).

Модель скользящего среднего представляет текущее значение временного ряда как линейную комбинацию случайных ошибок на предыдущих шагах. Формула для модели скользящего среднего порядка q выглядит следующим образом:

$$X_t = \mu + \varepsilon_t + \Theta_1 \varepsilon_{t-1} + \Theta_2 \varepsilon_{t-2} + \dots + \Theta_q \varepsilon_{t-q} \quad (6)$$

где:

- μ – среднее значение временного ряда;
- $\Theta_1, \Theta_2, \dots, \Theta_q$ – коэффициенты скользящего среднего;
- ε_t – случайные ошибки.

Модель ARMA объединяет элементы авторегрессии и скользящего среднего. Она описывает временной ряд как комбинацию как его предыдущих

значений, так и предыдущих случайных ошибок. Формула модели $ARMA(p, q)$ имеет вид:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} + \varepsilon_t + \Theta_1 \varepsilon_{t-1} + \Theta_2 \varepsilon_{t-2} + \dots + \Theta_q \varepsilon_{t-q} \quad (7)$$

где p и q – порядок авторегрессии и скользящего среднего соответственно.

Модель $ARIMA$ используется для анализа нестационарных временных рядов. Она включает в себя дифференцирование данных для их стационаризации. Модель $ARIMA(p, d, q)$ описывает временной ряд с помощью авторегрессии, дифференцирования и скользящего среднего, где d – порядок дифференцирования. Формула модели $ARIMA$:

$$(1 - \phi_1 B - \dots - \phi_p B^p)(1 - B)^d X_t = (1 + \Theta_1 B + \dots + \Theta_q B^q) \varepsilon_t \quad (8)$$

где:

- B – оператор сдвига;
- $B^k X_t = X_{t-k}$.

К преимуществам автокорреляционных моделей можно отнести математическую строгость и хорошую интерпретируемость, позволяющую проследить, каким образом формируются прогнозы [33, 34], невысокую вычислительную сложность, делающую модели пригодными для работы с большими объемами данных [35], а также достаточно точную работу со стационарными временными рядами [36].

Вместе с тем эти модели чувствительны к стационарности данных [37]: при наличии трендов или выраженной сезонности требуется предварительная обработка, иначе результаты могут быть искажены [33], а сложные нелинейные процессы такие модели описывают заметно хуже [34].

Модели скрытых марковских процессов используются для анализа и кластеризации временных рядов, когда наблюдаемые данные зависят от скрытых состояний системы [38]. В основе подхода лежит предположение о том, что система в каждый момент времени находится в одном из нескольких состояний, а переходы между ними описываются марковским процессом

первого порядка. Это означает, что вероятность перехода зависит только от текущего состояния и не учитывает всю предшествующую историю [39].

В контексте кластеризации марковские процессы позволяют рассматривать каждый временной ряд как последовательность состояний и распределений, характерных для конкретного типа поведения. При использовании данного подхода ряды объединяются в кластеры не по прямому сходству значений, а по близости их скрытых динамических характеристик [39].

Скрытая марковская модель описывается пятью основными компонентами [38]:

- множество скрытых состояний $S = \{s_1, s_2, \dots, s_N\}$, где N – количество возможных состояний, в которых может находиться система;
- множество наблюдаемых значений $O = \{o_1, o_2, \dots, o_M\}$, которые зависят от скрытых состояний;
- матрица переходов между состояниями $A = \{a_{ij}\}$, $a_{ij} = P(s_j | s_i)$ – вероятность перехода из состояния s_i в состояние s_j ;
- матрица вероятностей эмиссии $B = \{b_j(o_k)\}$, где $b_j(o_k) = P(o_k | s_j)$ – вероятность наблюдать значение o_k , если система находится в состоянии s_j ;
- начальное распределение состояний $\pi = \{\pi_i\}$, где $\pi_i = P(s_i)$ – вероятность того, что система начнет свою эволюцию в состоянии s_i .

Полная вероятность наблюдаемой последовательности $O = (o_1, o_2, \dots, o_T)$ вычисляется суммированием по всем возможным последовательностям скрытых состояний:

$$P(O|\lambda) = \sum_S P(O|S, \lambda) P(S|\lambda) \quad (9)$$

где $\lambda = (A, B, \pi)$ – параметры модели.

Для эффективного использования скрытых марковских процессов в анализе временных рядов применяются три основных алгоритма [29, 38-39]. Алгоритм Витерби применяется для поиска наиболее вероятной

последовательности скрытых состояний, соответствующей наблюдаемым данным, что позволяет интерпретировать структуру временного ряда. Алгоритм «вперед-назад» используется для вычисления вероятностей нахождения системы в каждом состоянии на каждом шаге времени и дает более детализированную картину динамики процесса. Алгоритм Баум–Уэлча предназначен для обучения модели и оценки ее параметров на основе данных, обеспечивая настройку матриц переходов и эмиссий.

Байесовские модели – это класс вероятностных методов, основанных на теореме Байеса, которые активно используются для анализа временных рядов, в том числе для кластеризации [40]. Их ключевая особенность заключается в учете неопределенности параметров и возможности их последовательного обновления по мере поступления новых данных [41]. Благодаря этому такие модели хорошо подходят для работы с шумными, неполными и нестабильными временными рядами, где структура данных может изменяться во времени.

Основу байесовского подхода составляет теорема Байеса:

$$P(\Theta|D) = \frac{P(D|\Theta)P(\Theta)}{P(D)} \quad (10)$$

где:

- $P(\Theta|D)$ – апостериорное распределение параметров модели Θ при условии наблюдаемых данных D ;
- $P(D|\Theta)$ – функция правдоподобия, вероятность получения данных D при фиксированном значении параметров Θ ;
- $P(\Theta)$ – априорное распределение параметров модели, отражающее наши предположения о параметрах до получения данных;
- $P(D)$ – нормирующий множитель, обеспечивающий корректность распределения данных.

В отличие от методов, где параметры фиксируются на этапе построения модели, байесовский подход позволяет учитывать их неопределенность и корректировать оценки по мере накопления информации [42].

В задачах кластеризации временных рядов байесовские методы реализуются в нескольких направлениях [41]. Одним из них является байесовская кластеризация на основе скрытых марковских моделей, где каждый кластер интерпретируется как скрытое состояние системы с собственными вероятностными характеристиками. Другой подход основан на использовании гауссовских смесей, в которых данные описываются набором нормальных распределений с оценкой параметров в вероятностной постановке. Также применяются методы Монте-Карло, позволяющие приближенно вычислять апостериорные распределения и оценивать вероятность принадлежности временных рядов к различным кластерам.

Сравнительный анализ методов приведен в таблице 2.

Таблица 2 – Сравнение вероятностных методов кластеризации временных рядов

Метод	Преимущества	Недостатки
Автокорреляционные модели (AR/MA/ARMA/ARIMA)	Математическая строгость и хорошая интерпретируемость [33, 34]; невысокая вычислительная сложность [35]; точная работа со стационарными рядами [36]	Чувствительность к стационарности данных [37]; требуют предобработки при трендах и сезонности [33]; хуже описывают нелинейные процессы [34]
Скрытые марковские модели	Учитывают скрытые состояния системы и вероятностные переходы между ними [38]; эффективны для процессов со скрытой структурой [39]	Значительные вычислительные затраты и тщательная настройка параметров [38]; чувствительность к выбору числа скрытых состояний [39]
Байесовские модели	Высокая гибкость: подходят как для простых, так и для сложных вероятностных структур [38, 43]	Высокая вычислительная сложность из-за итераций и сэмплирования [43]; зависимость результата от выбора априорных распределений [41]

1.1.3 Методы на основе глубокого обучения

Методы кластеризации временных рядов, основанные на глубоких нейронных сетях, представляют собой класс подходов, позволяющих учитывать сложные нелинейные зависимости в данных и извлекать представления, обладающие высокой дискриминативной способностью [44]. В отличие от традиционных методов, оперирующих непосредственно с исходными временными рядами или их признаковыми представлениями, глубокие модели способны автоматически выделять информативные признаки, что повышает качество кластеризации и снижает необходимость в предварительной обработке данных [45].

Основная идея данного подхода заключается в применении многослойных нейронных сетей для трансформации исходных временных рядов в компактное латентное пространство, где временные последовательности представляются в виде векторных представлений (эмбеддингов). В дальнейшем полученные эмбеддинги могут быть использованы для кластеризации с помощью стандартных алгоритмов, таких как K-средних (K-Means), DBSCAN, иерархическая кластеризация или более специализированных методов, учитывающих структуру временных данных [46].

В зависимости от используемой архитектуры нейронной сети методы глубинного обучения для кластеризации временных рядов можно разделить на несколько групп:

- 1) подходы с использованием рекуррентных нейронных сетей (RNN, LSTM, GRU);
- 2) сверточные нейросетевые модели (CNN);
- 3) гибридные методы и вариационные автокодировщики.

У методов глубокого обучения есть общая черта: они хорошо масштабируются на ряды высокой размерности и не требуют ручного конструирования признаков [46], но взамен требуют значительных

вычислительных ресурсов, чувствительны к выбору гиперпараметров и склонны к переобучению при нехватке данных. Как это соотношение плюсов и минусов выглядит для каждой из трёх архитектур – рекуррентных, сверточных сетей и вариационных автокодировщиков – показано в таблице 1.3.

Рекуррентные нейронные сети способны учитывать долговременные зависимости во временных рядах, что делает их особенно полезными при анализе последовательностей с выраженными нелинейными зависимостями [47].

В задаче кластеризации временных рядов часто используется подход, при котором рекуррентные нейронные сети применяются в качестве кодировщика, трансформирующего временной ряд в вектор фиксированной длины (эмбединг). Далее этот эмбединг передается в алгоритм кластеризации, такой как K-средних [48].

Одним из популярных подходов является Sequence-to-Sequence Autoencoder (Seq2Seq AE), в котором применяется LSTM-автокодировщик [47]. Кодировщик обрабатывает входной временной ряд и создает векторное представление, а затем оно используется для обучения кластеризующей модели. Такой подход позволяет эффективно группировать временные ряды с учетом их динамической структуры.

Сверточные нейронные сети традиционно применяются в области обработки изображений, однако их эффективность была продемонстрирована и в задачах анализа временных рядов. Основное преимущество сверточных нейронных сетей заключается в их способности выделять локальные паттерны во временных данных, что особенно полезно для задач кластеризации временных последовательностей с повторяющимися шаблонами [49].

Одним из эффективных методов является сверточная кластеризация временных рядов, в котором сверточные сети используются для извлечения ключевых признаков временных рядов, а затем полученные признаки передаются в алгоритм кластеризации. В этом подходе временные ряды

преобразуются в эмбединги, которые лучше подходят для дальнейшей группировки [50].

Дополнительно, сверточные нейронные сети можно комбинировать с рекуррентными сетями, получая гибридные архитектуры. Например, модель CNN-LSTM использует CNN для выделения локальных признаков, а затем LSTM для моделирования временной динамики, что повышает точность кластеризации.

Вариационные автокодировщики (VAE) являются расширением классических автокодировщиков, в которых латентное пространство задается не фиксированными векторами, а вероятностными распределениями [51]. Это позволяет учитывать стохастическую природу временных рядов и улучшать устойчивость модели.

В задачах кластеризации автокодировщики применяются в рамках глубинных методов разбиения данных, где важную роль играет форма латентного пространства [52].

Повышение качества кластеризации достигается также за счет гибридных архитектур, сочетающих разные типы нейронных сетей. [53]. Например, комбинация сверточных сетей в извлечении локальных признаков и вариационных автокодировщиков в представлении данных как вероятностного распределения делает кластеризацию более устойчивой к шумам.

Сравнительный анализ методов приведен в таблице 3.

Таблица 3 – Сравнение методов глубокого обучения для кластеризации временных рядов

Метод	Преимущества	Недостатки
Рекуррентные нейронные сети (RNN/LSTM/GRU)	Учитывают долговременные зависимости, эффективны для последовательностей со сложной нелинейной динамикой [47]	Последовательные вычисления не распараллеливаются, что увеличивает время обучения; нужны большие размеченные выборки; единая модель не учитывает структурную неоднородность рядов
Сверточные нейронные сети (CNN)	Хорошо выделяют локальные повторяющиеся паттерны во временных данных [49]	Ориентированы на фиксированное рецептивное поле, плохо подходят для рядов с заранее неизвестными границами сегментов
Вариационные автокодировщики и гибридные подходы	Учитывают стохастическую природу данных и повышают устойчивость модели [51]; гибридные архитектуры дополнительно снижают чувствительность к шумам [53]	Значительная вычислительная сложность из-за одновременного обучения нескольких компонентов и оптимизации сложной функции потерь

1.2 Прогнозирование временных рядов: теоретические основы и классификация

Прогнозирование временных рядов – это задача анализа данных, связанная с построением моделей для предсказания будущих значений на основе уже наблюдаемых исторических данных. Оно широко используется в сферах деятельности, где необходимо учитывать изменение процессов во времени и их динамику [54, 55].

Существующие методы прогнозирования временных рядов можно классифицировать по различным критериям, включая математический аппарат, используемый для моделирования динамики ряда, а также подход к обработке временной зависимости [31]. В рамках данной классификации выделяются следующие основные группы методов:

- 1) классические статистические методы;
- 2) методы машинного обучения;
- 3) глубокие нейросетевые модели [47].

Разнообразие существующих подходов позволяет адаптировать методы прогнозирования под конкретные особенности данных, обеспечивая высокую точность предсказаний.

1.2.1 Классические статистические методы

Классические статистические методы прогнозирования временных рядов представляют собой фундаментальный подход к анализу и предсказанию динамических процессов, основанный на математическом моделировании временных зависимостей. Данные методы применяются для прогнозирования значений временного ряда на основе его исторических данных, учитывая такие свойства, как тренд, сезонность и случайные флуктуации [2].

Основной принцип классических статистических методов заключается в формализации структуры временного ряда с использованием параметрических моделей, описывающих его динамику.

В данной категории можно выделить несколько ключевых классов моделей [1]:

- 1) авторегрессионные модели (AR);
- 2) модели скользящего среднего (MA);
- 3) модели ARIMA;
- 4) модели экспоненциального сглаживания (ETS).

Авторегрессионная (AR) модель описывает текущее значение временного ряда как линейную комбинацию его предыдущих значений. Предполагается, что наблюдения формируют стационарный процесс, а сами зависимости являются линейными [56].

Общая форма авторегрессионной модели задаётся в следующем виде:

$$X_t = c + \sum_{i=1}^p \varphi_i X_{t-i} + \varepsilon_t, \quad (11)$$

где X_t – значение временного ряда в момент t , c – константа, φ_i – параметры модели, а ε_t – белый шум.

Модели скользящего среднего описывают значение временного ряда через накопленные ошибки предыдущих прогнозов. Такой подход применяется для анализа процессов, в которых значительная роль принадлежит случайным возмущениям и шумовой составляющей [56].

Общая форма модели скользящего среднего задается в следующем виде:

$$X_t = \mu + \sum_{i=1}^q \theta_i \varepsilon_{t-i} + \varepsilon_t, \quad (12)$$

где μ – математическое ожидание процесса, θ_i – параметры модели, ε_t – белый шум.

Модели ARIMA (Autoregressive Integrated Moving Average) объединяет авторегрессионную модель и модель скользящего среднего, а также включает этап дифференцирования ряда для приведения его к стационарному виду [56].

Общая формула $ARIMA(p, d, q)$ выглядит следующим образом:

$$\phi(L)(1 - L)^d X_t = \theta(L)\varepsilon_t, \quad (13)$$

где L – оператор задержки, $\phi(L)$ – полином AR, $\theta(L)$ – полином MA.

Модели экспоненциального сглаживания позволяют моделировать уровни, тренды и сезонность временного ряда [56].

Простейшая форма экспоненциального сглаживания задается выражением:

$$\hat{X}_{t+1} = \alpha X_t + (1 - \alpha)\hat{X}_t, \quad (14)$$

где α – коэффициент сглаживания.

Сравнительный анализ методов приведен в таблице 4.

Таблица 4 – Сравнение классических статистических методов прогнозирования

Модель	Преимущества	Недостатки
AR	Простота и понятная интерпретация; хорошо выявляет автокорреляционные структуры и длительные зависимости в стационарных рядах	Требуется стационарности данных и линейности зависимостей; теряет точность при выраженных трендах или нелинейных закономерностях
MA	Хорошо описывает процессы с выраженной случайной (шумовой) составляющей	Менее наглядная интерпретация по сравнению с AR; требует анализа автокорреляционной функции для выбора порядка модели
ARIMA	Универсальность, применима к рядам с выраженным трендом; включает дифференцирование для приведения ряда к стационарному виду	Требуется тщательного подбора параметров p, d, q ; плохо справляется с выраженной сезонностью без дополнительных расширений
ETS (экспоненциальное сглаживание)	Хорошо адаптируется к изменениям данных, учитывает тренд и сезонность [58]	Теряет точность при долгосрочном прогнозировании; лучше работает на коротких интервалах

1.2.2 Методы машинного обучения

Методы машинного обучения позволяют автоматически выявлять зависимости в данных без необходимости заранее задавать конкретную математическую модель и применимы для анализа сложных нелинейных и многомерных связей, включая влияние внешних факторов на динамику временных рядов [58, 59].

В отличие от классических статистических подходов, такие методы не требуют строгого выполнения условий стационарности или линейности

зависимостей и позволяют работать с большими массивами данных [60, 61]. Это позволяет им адаптироваться к изменяющимся условиям и выявлять скрытые паттерны, которые сложно описать аналитически [58].

В прогнозировании временных рядов используются регрессионные модели с регуляризацией, деревья решений и ансамблевые методы. Они могут учитывать как значения самого ряда, так и дополнительные признаки, включая лаги, временные метки и внешние индикаторы [62, 63].

В то же время у этой группы методов есть и недостатки. Для стабильной работы требуется большое количество данных в обучающей выборке, иначе возрастает риск переобучения [58]. Также результаты моделей сложнее интерпретировать по сравнению с классическими статистическими подходами, что затрудняет объяснение полученных прогнозов [63]. Наконец, обучение некоторых моделей, особенно нейронных сетей, может требовать значительных вычислительных ресурсов.

Сравнительный анализ методов приведен в таблице 5.

Таблица 5 – Сравнение методов машинного обучения для прогнозирования временных рядов

Метод	Преимущества	Недостатки
Регрессия с регуляризацией	Простота и интерпретируемость коэффициентов [65]; регуляризация снижает переобучение и повышает устойчивость к нестационарным данным [66]	Ограниченная способность моделировать сложные нелинейные процессы [64]
Деревья решений и ансамблевые методы	Высокая точность, учитывают сложные нелинейные взаимодействия признаков [67-69]	Риск переобучения при малом объеме данных; сравнительно низкая интерпретируемость [70]
Метод опорных векторов	Описывает сложные нелинейные зависимости за счёт ядерных функций [71]	Чувствительность к выбору ядра и гиперпараметров [72]; проблемы вычислительной сложности на больших наборах данных

1.2.3 Глубокие нейросетевые модели

Глубокие нейросетевые модели – класс алгоритмов машинного обучения, способных выявлять сложные нелинейные зависимости в данных с учётом их временной структуры. Многослойная архитектура таких моделей позволяет учитывать как краткосрочные, так и долгосрочные зависимости ряда [73].

В отличие от классических статистических методов, которые требуют явного задания структуры модели (например, порядков авторегрессии и скользящего среднего), глубокие сети автоматически извлекают релевантные временные паттерны в процессе обучения [74]. Это позволяет им адаптироваться к сложным, нелинейным и многомерным данным [73].

Основной вычислительный элемент нейросетевой модели – искусственный нейрон, который формирует выход через взвешенную сумму входов и функцию активации:

$$y = \sigma \left(\sum_{i=1}^n w_i x_i + b \right), \quad (15)$$

где x_i – входные признаки, w_i – веса, b – смещение, σ – функция активации, а y – выход нейрона.

Рекуррентные нейронные сети предназначены для обработки последовательных данных и временных рядов. Их ключевая особенность заключается в наличии скрытого состояния h_t , которое кодирует информацию о предыдущих моментах времени и позволяет учитывать зависимость текущего значения от всей истории наблюдений.

Математически работа сети описывается следующим образом:

$$h_t = \sigma(W_h x_t + U_h h_{t-1} + b_h), \quad y_t = W_y h_t + b_y, \quad (16)$$

где x_t – вход в момент времени t , h_{t-1} – скрытое состояние предыдущего шага, W_h , U_h , W_y – матрицы весов, b_h , b_y – смещения, y_t – прогнозное значение.

Основное преимущество рекуррентных сетей заключается в их способности учитывать последовательный характер данных и выявлять зависимости различной длительности [75]. При этом стандартные рекуррентные сети имеют ограничение – проблему исчезающих и взрывающихся градиентов [76, 77]. При обучении на длинных последовательностях градиенты при обратном распространении ошибки могут либо быстро уменьшаться до нуля, либо, наоборот, резко возрасти, что делает обучение на длинных последовательностях затруднительным [78].

Сеть с долгосрочной краткосрочной памятью (LSTM) – специализированная архитектура рекуррентных нейронных сетей, способная эффективно моделировать долгосрочные зависимости за счёт введения механизмов шлюзов (gates): входного, выходного и забытого [7].

Ключевая инновация LSTM заключается в наличии состояния ячейки, которое проходит через всю цепочку временных шагов с минимальными линейными преобразованиями, что позволяет градиенту эффективно распространяться назад во времени без затухания. Шлюзы, управляемые сигмоидными функциями, решают, какая информация должна быть сохранена, забыта или передана на выход. Коэффициент забвения определяет, какую часть предыдущего состояния ячейки следует забыть; входной коэффициент регулирует, какая новая информация будет добавлена; выходной коэффициент определяет, какая информация из состояния ячейки будет передана в скрытое состояние.

Основные уравнения LSTM:

$$\begin{aligned}
 f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \\
 i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \\
 \tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C), \\
 C_t &= f_t * C_{t-1} + i_t * \tilde{C}_t, \\
 o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \\
 h_t &= o_t * \tanh(C_t),
 \end{aligned} \tag{17}$$

где f_t – коэффициент забвения, i_t – входной коэффициент, \tilde{C}_t – кандидат состояния, C_t – состояние ячейки, o_t – выходной коэффициент, h_t – скрытое состояние.

Обучение LSTM выполняется методом обратного распространения через время с использованием градиентного спуска [78].

Преимущества LSTM заключаются в способности моделировать как краткосрочные, так и долгосрочные зависимости, эффективно управляя потоком информации через шлюзы. Благодаря механизму постоянной ошибки (Constant Error Carousel) LSTM может удерживать информацию на протяжении более 1000 временных шагов [79].

Основные ограничения – высокая вычислительная сложность, необходимость большого объёма данных и сложность настройки гиперпараметров, таких как размер скрытого состояния, число слоев и параметры оптимизатора [80].

Многослойный персептрон (MLP) представляет собой полносвязную архитектуру нейронной сети, состоящую из входного слоя, нескольких скрытых слоев и выходного слоя [73]. Каждый нейрон скрытого слоя получает на вход выходы предыдущего слоя и применяет функцию активации:

$$h^{(l)} = \sigma(W^{(l)}h^{(l-1)} + b^{(l)}), \quad (18)$$

где $h^{(l-1)}$ – выход предыдущего слоя, $W^{(l)}$ и $b^{(l)}$ – веса и смещение слоя l , σ – функция активации. Выходной слой формирует прогнозное значение временного ряда.

Для временных рядов входными признаками обычно служат значения предыдущих наблюдений (лаговые признаки) и дополнительные внешние индикаторы.

Обучение модели осуществляется методом обратного распространения ошибки с использованием градиентного спуска. Алгоритм аналогичен обучению LSTM, но без учета временной зависимости скрытых состояний.

Сравнительный анализ методов приведен в таблице 6.

Таблица 6 – Сравнение глубоких нейросетевых моделей прогнозирования временных рядов

Модель	Преимущества	Недостатки
RNN	Учитывает последовательный характер данных, выявляет зависимости разной длительности [75]	Проблема исчезающих/взрывающихся градиентов затрудняет обучение на длинных последовательностях [76, 77, 78]
LSTM	Моделирует и краткосрочные, и долгосрочные зависимости благодаря механизму шлюзов; удерживает информацию свыше 1000 временных шагов [79]	Высокая вычислительная сложность, потребность в больших объёмах данных, сложная настройка гиперпараметров [80]
MLP	Гибкая архитектура, работает с многомерными признаками; при простой структуре иногда точнее и быстрее сложных моделей [81]; универсальный аппроксиматор [82]	Не учитывает долгосрочные зависимости естественным образом, чувствителен к масштабу данных, требует явного формирования лаговых признаков [83, 84]

1.3. Постановка задачи исследования

Анализ существующих моделей и методов кластеризации и прогнозирования временных рядов показал, что классические статистические модели обладают высокой интерпретируемостью и вычислительной эффективностью, однако ограничены в возможности описания сложных нелинейных зависимостей. Методы машинного обучения и глубокие нейросетевые модели обеспечивают более высокую точность прогнозирования, но требуют значительных вычислительных затрат и не всегда эффективно работают с неоднородными временными рядами, а значит не существует эффективных подходов, позволяющих осуществить точное прогнозирование для рядов с высокой изменчивостью с минимальным потреблением вычислительных ресурсов.

Следовательно, для повышения точности и адаптации прогнозов к индивидуальной динамике неоднородных временных рядов необходимо разработать специальные методы и модели адаптивного прогнозирования, а также программное обеспечение для их практической реализации в условиях ограниченных вычислительных ресурсов, на основе глубокого обучения.

1.4. Выводы

В первой главе диссертационной работы выполнен анализ существующих методов кластеризации и прогнозирования временных рядов, рассмотрены принципы работы с временными рядами и особенности их обработки в условиях неоднородности, высокой зашумленности и возможной смены режимов поведения.

Проведена классификация методов кластеризации: рассмотрены подходы на основе измерения расстояния, вероятностные модели и методы глубокого обучения. Методы на основе метрик расстояния отличаются простотой реализации и вычислительной эффективностью, но чувствительны к временным сдвигам и шумам; вероятностные модели лучше учитывают стохастическую природу процессов и позволяют выявлять скрытые закономерности, однако требуют более сложной настройки и значительных вычислительных затрат.

Методы машинного обучения и глубокого обучения демонстрируют высокую эффективность при работе со сложными нелинейными зависимостями, но их применение сопряжено с необходимостью больших объемов данных и высокой вычислительной сложностью.

В целом можно заключить, что существующие подходы не всегда обеспечивают одновременно высокую точность и приемлемую вычислительную эффективность при работе с неоднородными временными рядами, а значит требуются более адаптивные методы кластеризации и прогнозирования, способные учитывать структуру данных и работать в условиях ограниченных вычислительных ресурсов.

2 МЕТОДЫ И МОДЕЛЬ АДАПТИВНОГО ПРОГНОЗИРОВАНИЯ ВРЕМЕННЫХ РЯДОВ

Проведенный в первой главе анализ существующих подходов к кластеризации и прогнозированию временных рядов выявил ряд ограничений. Классические статистические модели предъявляют строгие требования к стационарности и линейности исследуемых процессов. Методы машинного обучения и глубокие нейросетевые архитектуры способны моделировать сложные нелинейные зависимости, однако их применение в условиях ограниченных вычислительных ресурсов затруднено.

В данной главе представлены оригинальные методы и модели, разработанные для решения задач анализа и прогнозирования неоднородных временных рядов в условиях ограниченных вычислительных ресурсов. Основная идея предлагаемых подходов заключается в переходе от анализа сырых временных последовательностей к работе с их компактными параметрическими представлениями, что позволяет существенно снизить размерность данных без потери ключевой информации.

В главе 2.1 представлен метод динамического определения сегментов переменной длины для кусочно-линейной аппроксимации с заданной точностью. В главе 2.2 детально рассматривается метод кластеризации временных рядов на основе вычисления временных характеристик, позволяющий перейти от высокоразмерных последовательностей к интерпретируемым векторам признаков. Наконец, в главе 2.3 представлена обобщенная модель прогнозирования, интегрирующая разработанные методы в единую двухэтапную стратегию: параметризация и кластеризация с последующим обучением специализированных LSTM-моделей для каждого выделенного кластера.

Разработанные методы ориентированы на решение прикладных задач, связанных с обработкой медицинских данных (прогнозирование гликемии) и управлением техническими системами (аппроксимация траекторий

беспилотных аппаратов), однако предложенные подходы обладают достаточной универсальностью для применения в других предметных областях.

2.1 Метод адаптивной кусочно-линейной аппроксимации временных рядов с динамическим выбором шага

Кусочно-линейная аппроксимация представляет собой метод, направленный на приближенное представление сложных функциональных зависимостей с помощью линейных сегментов. Эффективность данного подхода существенно зависит от выбора шага разбиения, который, в свою очередь, влияет на точность аппроксимации. Применение фиксированного шага может привести к снижению точности на участках с высокой кривизной, в то время как использование переменного шага позволяет более гибко адаптироваться к локальным особенностям данных.

В современных исследованиях, посвященных аппроксимации нелинейных функций с помощью кусочно-линейных моделей, доминирует подход с постоянным шагом разбиения. В рамках этого метода интервал разбивается на подинтервалы с фиксированным шагом h , который не превышает заданного порога допустимой погрешности $h \leq \varepsilon$. Данный подход предполагает выделение лишнего количества подинтервалов, особенно в случаях, когда функция имеет относительно прямолинейные участки. Это приводит к избыточному количеству операций для вычислений и снижению эффективности моделирования [85].

Методы с переменным шагом разбиения изучены сравнительно слабо. На практике такие подходы чаще всего строятся на заранее заданных предположениях о форме функции или особенностях моделируемого процесса. Из-за этого их применение ограничено в задачах, где поведение функции заранее неизвестно и требуется адаптация к локальным изменениям.

При анализе длительных временных рядов, в том числе физиологических данных, часто возникает необходимость точно описывать нелинейную динамику процессов. В этом случае классические методы кусочно-линейной аппроксимации с фиксированным шагом имеют ряд недостатков. Во-первых, на участках, где изменения происходят плавно, используется избыточное число сегментов, что приводит к лишним вычислениям. Во-вторых, на участках с резкими изменениями фиксированный шаг не позволяет добиться достаточной точности. В-третьих, при обработке протяженных и неоднородных рядов такие методы в целом оказываются недостаточно эффективными.

Использование переменного шага разбиения позволяет адаптировать размер сегментов к локальной сложности функции: более крупные интервалы применяются на гладких участках, а более мелкие – там, где наблюдаются быстрые изменения. Такой подход особенно полезен в задачах, связанных с персонализированным анализом физиологических данных, например при моделировании динамики гликемии, где требуется высокая точность описания индивидуальных особенностей пациента и их изменений во времени.

2.1.1 Формальная постановка задачи

Рассмотрим эмпирическую нелинейную одномерную функцию $y=f(x)$, заданную таблично на интервале $[a_1; a_n]$ в виде пары векторов-столбцов координат входных значений $X^n = (x_1, x_2 \dots x_n)^T$ и соответствующих координат выходных значений $Y^n = (y_1, y_2 \dots y_n)^T$. Здесь $x_1 = a_1$, $x_n = a_n$, n – количество опорных точек-узлов таблично заданной функции. Пары «вход-выход» образуют матрицу T размерности $(n \times 2)$ следующего вида:

$$T = \begin{pmatrix} x_1 & y_1 \\ \dots & \dots \\ x_n & y_n \end{pmatrix} = [X^n | Y^n] \quad (19)$$

Введем понятие сегмента.

Сегмент – это подматрица матрицы T , включающая в себя последовательные строки матрицы T с некоего номера u до номера v . Сегменты не пересекаются между собой. В совокупности сегменты образуют матрицу T .

Обозначим сегмент следующим образом:

$$S^m(u, v) = \begin{pmatrix} x_u & y_u \\ \dots & \dots \\ x_v & y_v \end{pmatrix} \quad (20)$$

$$S^m \cap S^l = \emptyset \quad (21)$$

$$\bigcup_{m=1}^Q S^m = T \quad (22)$$

Здесь: Q – количество сегментов, m – порядковый номер сегмента, u – индекс начального элемента сегмента, v – индекс конечного элемента сегмента.

Требуется декомпозировать исходную матрицу T на сегменты таким образом, чтобы:

1. Объединение всех сегментов образует исходную матрицу T .
2. Для каждого сегмента $S^m(u, v)$ линейная аппроксимация $\hat{y} = A + Bx$ обеспечивает коэффициент детерминации $R^2 \geq \varepsilon$, где ε – заданная точность.
3. Длина каждого сегмента $h_m = d[S^m(u, v)]$ максимальна при соблюдении условия точности.

2.1.2 Описание алгоритма

Для решения поставленной задачи был разработан численный метод определения на матрице T сегментов максимально возможной длины при условии, что внутри сегмента линейный многочлен аппроксимирует исходную функцию с заданной точностью ε . В качестве критерия точности используется коэффициент детерминации R^2 , вычисляемый для линейной регрессионной

функции внутри сегмента. В ходе исполнения алгоритма производится два цикла вычислений:

1. Внешний цикл отвечает за сокращение исходного интервала поиска $[a_1, a_n]$ по мере того, как будут выделяться последовательные сегменты аппроксимации.

2. Внутренний цикл отвечает за выделение границ очередного сегмента внутри интервала поиска.

Пошаговое описание алгоритма [86]:

Шаг 0. Установить начальные значения: номер определяемого сегмента $m = 1$, индекс первого элемента определяемого сегмента $u = 1$. Задать параметр точности ε .

Начало внешнего цикла (по сегментам):

Шаг 1. Установить $v = n$ (конец интервала поиска).

Начало внутреннего цикла (по длине сегмента):

Шаг 2. Для сегмента $S^m(u, v)$ построить линейную регрессионную функцию $\hat{y} = A + Bx$ с коэффициентами, определяемыми согласно методу наименьших квадратов:

$$A = \frac{\sum_{i=u}^v y_i \sum_{i=u}^v x_i^2 - \sum_{i=u}^v x_i \sum_{i=u}^v x_i y_i}{(v - u + 1) \sum_{i=u}^v x_i^2 - (\sum_{i=u}^v x_i)^2} \quad (23)$$

$$B = \frac{(v - u + 1) \sum_{i=u}^v x_i y_i - \sum_{i=u}^v x_i \sum_{i=u}^v y_i}{(v - u + 1) \sum_{i=u}^v x_i^2 - (\sum_{i=u}^v x_i)^2} \quad (24)$$

Шаг 3. Вычислить значения регрессионной функции в узлах сетки сегмента $S^m(u, v)$:

$$\begin{aligned} \hat{y}_i &= A + Bx_i \\ i &= \overline{u, v} \end{aligned} \quad (25)$$

Шаг 4. Вычисляется коэффициент детерминации для сегмента $S^m(u, v)$:

$$R^2 = 1 - \frac{\sum_{i=u}^v (y_i - \hat{y}_i)^2}{\sum_{i=u}^v (y_i - \bar{y})^2} \quad (26)$$

$$\bar{y} = \frac{\sum_{i=u}^v y_i}{(v - u + 1)} \quad (27)$$

Шаг 5. Если $R^2 < \varepsilon$ то $v = v - 1$. Возврат к шагу 2. Иначе переход к шагу 6.

Конец внутреннего цикла (условия завершения)

Шаг 6. Если $v < n$, то увеличиваем номер сегмента $m = m + 1$, устанавливаем начало следующего сегмента $u = v + 1$, возврат к Шагу 1. Иначе – $Q = m$, завершить формирование сегментов.

Конец внешнего цикла.

2.1.3 Анализ вычислительной сложности

В наихудшем случае количество итераций внутреннего цикла ограничено $O(n)$, где n – длина интервала. Это обусловлено последовательным уменьшением длины сегмента до достижения требуемой точности.

Максимальное количество сегментов Q ограничено $O(n)$, что обусловлено полным покрытием интервала. Это гарантирует, что алгоритм не будет выполняться дольше, чем необходимо.

Общая вычислительная сложность алгоритма составляет $O(n^2)$ в наихудшем случае. Однако на практике она значительно ниже за счёт эффективной адаптации длины сегментов к локальным особенностям функции, что позволяет существенно сократить количество итераций.

2.2 Метод кластеризации временных рядов на основе вычисления статических и динамических характеристик

В задачах анализа неоднородных данных, развернутых во времени, важна способность выделять группы схожих паттернов для последующей адаптации прогнозных моделей. В диссертационном исследовании предложен метод, преодолевающий ограничения, выделенные в главе 1, за счет:

- явного учета временной структуры рядов через вычисление вектора характеристик временного ряда, описывающих поведение и динамику его изменения;
- снижения размерности данных без потери значимых паттернов изменения динамики;
- вычислительной эффективности по сравнению с моделями глубокого обучения.

2.2.1 Этап предобработки данных

На первом этапе, прежде чем перейти к вычислению характеристик, временные ряды проходят последовательную предварительную обработку:

- 1) ряд делится на сегменты равной длины с количеством L точек в каждом;
- 2) заполнение пропущенных значений методом линейной или кубической интерполяции между соседними наблюдаемыми точками. Выбор метода будет зависеть от частоты пропусков и гладкости ряда;
- 3) применяется сглаживание шума методом скользящего среднего с окном размера w :

$$\hat{y}_t = \frac{1}{w} \sum_{i=t-w+1}^t y_i, \quad (36)$$

где \hat{y}_t – сглаженное значение в точке t . Это уменьшает влияние краткосрочных артефактов, не искажая долгосрочные тренды.

2.2.2 Вычисление временных характеристик

После этапа предварительной обработки каждый временной ряд представляется в виде совокупности сегментов фиксированной длины L . Для каждого сегмента вычисляется вектор временных характеристик, которые

описаны ниже. Они отражают его статистические, динамические и трендовые свойства. Использование свертки векторов временных характеристик позволяет перейти от анализа высокоразмерных временных последовательностей к анализу компактных и интерпретируемых признаков представлений, сохраняя при этом значимую информацию о поведении временного ряда.

Постановка задачи следующая. Пусть сегмент временного ряда задается последовательностью:

$$y = \{y_1, \dots, y_L\} \quad (28)$$

где y_t – значение анализируемого показателя в момент времени t .

Для каждого сформированного блока вычисляется набор статистических и временных характеристик: минимальное и максимальное значения, среднее и медиана, дисперсия, коэффициенты асимметрии и эксцесса, автокорреляция первого порядка, площадь под кривой и коэффициент наклона тренда. Совокупность этих признаков описывает особенности поведения временного ряда на рассматриваемом участке и используется при последующей кластеризации.

Минимальное и максимальное значения сегмента определяются следующим образом:

$$y_{min} = \min_{1 \leq t \leq L} y_t, y_{max} = \max_{1 \leq t \leq L} y_t \quad (29)$$

Эти показатели задают диапазон изменения значений внутри сегмента и позволяют определить его экстремальные значения.

Среднее значение сегмента рассчитывается по формуле:

$$\mu = \frac{1}{L} \sum_{t=1}^L y_t \quad (30)$$

Оно отражает общий уровень исследуемого показателя на данном интервале.

Медиана определяется как центральное значение упорядоченной последовательности:

$$\text{med}(y) = \begin{cases} y_{(k)}, L = 2k - 1, \\ \frac{y_{(k)} + y_{(k+1)}}{2}, L = 2k, \end{cases} \quad (31)$$

где $y_{(k)}$ – элементы, упорядоченные по возрастанию.

В отличие от среднего, медиана менее чувствительна к выбросам и поэтому лучше отражает центральную тенденцию при наличии резких скачков значений.

Чтобы понять, насколько сильно значения разбросаны вокруг среднего, считаем дисперсию:

$$\sigma^2 = \frac{1}{L} \sum_{t=1}^L (y_t - \mu)^2 \quad (32)$$

Это стандартная мера изменчивости: чем она больше, тем сильнее значения «расползаются» от среднего.

Дальше идут характеристики формы распределения. Коэффициент асимметрии (скошенности) рассчитывается по формуле:

$$\gamma_1 = \frac{1}{L} \sum_{t=1}^L \left(\frac{y_t - \mu}{\sigma} \right)^3 \quad (33)$$

Он показывает, перекошено ли распределение влево или вправо: например, если «хвост» тянется вправо, асимметрия будет положительной. Это помогает заметить несимметричность поведения ряда на сегменте.

Коэффициент эксцесса определяется следующим образом:

$$\gamma_2 = \frac{1}{L} \sum_{t=1}^L \left(\frac{y_t - \mu}{\sigma} \right)^4 - 3 \quad (34)$$

Его используют, чтобы оценить «остроту» пика и наличие тяжёлых хвостов. Повышенный эксцесс нередко сигнализирует о сегментах с резкими всплесками или аномальными значениями – то есть о том, что распределение отличается от нормального.

Для оценки зависимости между соседними наблюдениями используется автокорреляция первого порядка:

$$r_1 = \frac{\sum_{t=2}^L (y_t - \mu)(y_{t-1} - \mu)}{\sum_{t=1}^L (y_t - \mu)^2} \quad (35)$$

Данная метрика позволяет определить, насколько текущее значение связано с предыдущим. Если автокорреляция высокая, ряд меняется плавно; если низкая – в нём много резких колебаний и «рывков».

Суммарный вклад сегмента удобно оценивать через площадь под кривой. Её приближённо вычисляем как дискретную сумму:

$$S = \sum_{t=1}^L y_t \Delta t \quad (36)$$

Δt – шаг дискретизации, который для равномерных измерений принимается равным единице. Эта величина показывает общий «объём» показателя на всём отрезке.

Наконец, чтобы понять общую направленность сегмента – растёт он, падает или почти стоит на месте, – строим простую линейную регрессию:

$$y_t = at + b + \varepsilon_t, \quad (37)$$

где параметр a определяется методом наименьших квадратов:

$$a = \frac{\sum_{t=1}^L (t - \bar{t})(y - \mu)}{\sum_{t=1}^L (t - \bar{t})^2}, \quad \bar{t} = \frac{L + 1}{2} \quad (38)$$

Знак и величина a сразу дают представление о тренде: если положительный – сегмент в среднем растёт, если отрицательный – падает, близкий к нулю – почти не меняется. Это один из самых наглядных показателей для последующего прогнозирования: он помогает быстро отнести сегмент к одной из базовых динамик.

В результате использование данных характеристик позволяет эффективно группировать временные сегменты по структурному сходству [87].

2.2.3 Нормализация и кластеризация

После выделения параметров данные подвергаются кластеризации. Для этого используется метод DBSCAN, при этом данные предварительно масштабируются с помощью нормализации [88] для того, чтобы параметры, имеющие различные масштабы, не влияли на результаты кластеризации. Кластеризация позволяет группировать временные ряды с схожими характеристиками, что создает основу для создания отдельных прогнозных моделей для каждой группы.

Фрагмент данных распределения записей по кластерам представлен ниже.

Таблица 7 – Параметры временных рядов, распределенные по кластерам (фрагмент данных, полученный в вычислительных экспериментах)

Ряд	Min	Max	Среднее	Медиана	Дисперсия	...	Площадь	Тренд	Кластер
1	76	134,8	93,4971	87	268,598	...	93471,5	-0,04052	2
2	97	134	115,051	111,1	128,165	...	114904,5	-0,02207	2
3	84	135	110,856	113	312,149	...	110720,5	-0,05264	2
4	84	132	109,243	110	269,784	...	109129,5	-0,02338	2
5	55	160	112,751	119,5	1190,361	...	112664	-0,0854	0
6	85	163	135,128	142	616,3681	...	134903	-0,05	0
7	88	160	126,172	127	270,434	...	126040	0,05591	2
8	160	220	186,266	180,1	343,444	...	186047,5	0,05917	3
9	67	187	112,154	111	1063,401	...	112211	-0,01012	0
10	102	156	125,127	127,1	214,733	...	125057,5	-0,0197	2
11	137	176	158,835	157,4	116,403	...	158638	0,01230	1
12	145	175	159,736	162	75,564	...	159593,5	0,02604	1
13	135	176	146,844	142	121,029	...	146726,5	-0,02375	1
14	100	137	115,315	113	164,926	...	115262	0,00556	2
15	105	156	125,608	127	217,192	...	125542	0,02208	2

Этот метод уменьшает потери информации, характерные для векторных представлений, и снижает вычислительную сложность по сравнению с

глубокими нейросетевыми моделями. В результате улучшается качество кластеризации и создаются более точные прогнозные модели для каждого кластера [89].

2.3 Гибридная нейросетевая модель прогнозирования временных рядов

В предыдущем разделе был разработан метод кластеризации временных рядов на основе вычисления статистических и динамических характеристик, позволяющий выделять группы временных рядов со схожими структурными свойствами. Полученные результаты могут быть использованы для повышения эффективности методов прогнозирования за счёт учета неоднородности временных данных.

Целью данного раздела является разработка гибридной нейросетевой модели прогнозирования временных рядов, основанной на совместном использовании методов кластеризации и рекуррентных нейронных сетей. Основная идея предлагаемого подхода заключается в том, что временные ряды, обладающие схожими характеристиками, могут иметь близкие закономерности динамики. Это позволяет обучать специализированные прогнозные модели для отдельных групп временных рядов, что повышает точность прогнозирования по сравнению с использованием одной универсальной модели.

Предлагаемая гибридная модель состоит из нескольких взаимосвязанных этапов обработки данных:

- 1) предварительная обработка временных рядов;
- 2) формирование признакового представления временных рядов;
- 3) кластеризация временных рядов;
- 4) обучение специализированных нейросетевых моделей;
- 5) прогнозирование новых временных рядов.

2.3.1 Предварительная обработка временных рядов

Перед обучением нейронных сетей выполняется предварительная подготовка данных: пропущенные значения заполняются с помощью линейной интерполяции, а для уменьшения влияния случайных шумов временные ряды сглаживаются фильтром скользящего среднего. Затем каждый временной ряд делится на сегменты фиксированной длины. Это позволяет анализировать его поведение не только в целом, но и на отдельных локальных участках.

Для каждого сегмента временного ряда вычисляется набор статистических и динамических характеристик согласно методу, описанному в главе 2.2.2. Полученные характеристики формируют вектор признаков, который позволяет унифицировать данные для последующей кластеризации и снизить размерность задачи, что особенно важно при работе с большими объёмами данных.

2.3.2 Кластеризация временных рядов

На следующем этапе выполняется кластеризация полученных признаков векторов, используя алгоритм DBSCAN. Он позволяет выделять группы данных с близкими характеристиками без необходимости заранее задавать число кластеров.

Перед выполнением кластеризации выполняется нормализация признаков методом min–max масштабирования. Это необходимо, чтобы привести все значения к единому диапазону значений и исключить влияние различных масштабов на результаты кластеризации. В результате кластеризации формируются группы временных рядов, обладающих сходными статистическими и динамическими свойствами.

Полученный номер кластера используется как дополнительная структурная информация, отражающая особенности динамики соответствующих временных рядов.

2.3.3 Обучение специализированных нейросетевых моделей

Для каждого полученного кластера обучается отдельная модель прогнозирования на основе рекуррентной нейронной сети с долгой краткосрочной памятью (LSTM). Использование отдельных моделей для различных кластеров позволяет учитывать различия в характере динамики временных рядов и адаптировать прогнозные модели к конкретным типам поведения данных. Такой подход принципиально отличается от классической стратегии, где одна универсальная модель применяется ко всем данным.

Обучение нейросетевых моделей выполняется методом распространения ошибки во времени. В качестве функции потерь используется функция Хубера, обладающая повышенной устойчивостью к выбросам по сравнению с квадратичной функцией ошибки.

2.3.4 Прогнозирование новых временных рядов

При прогнозировании нового временного ряда выполняется последовательность действий:

- 1) вычисляются те же характеристики, что использовались на этапе формирования признаков;
- 2) определяется кластер, к которому относится данный временной ряд;
- 3) выбирается соответствующая обученная LSTM-модель;
- 4) выполняется прогноз будущих значений временного ряда.

В результате прогнозирование выполняется специализированной моделью, адаптированной к особенностям конкретного типа временных рядов.

Структура нейросетевых моделей, используемых для прогнозирования временных рядов, представлена на рисунке 1.

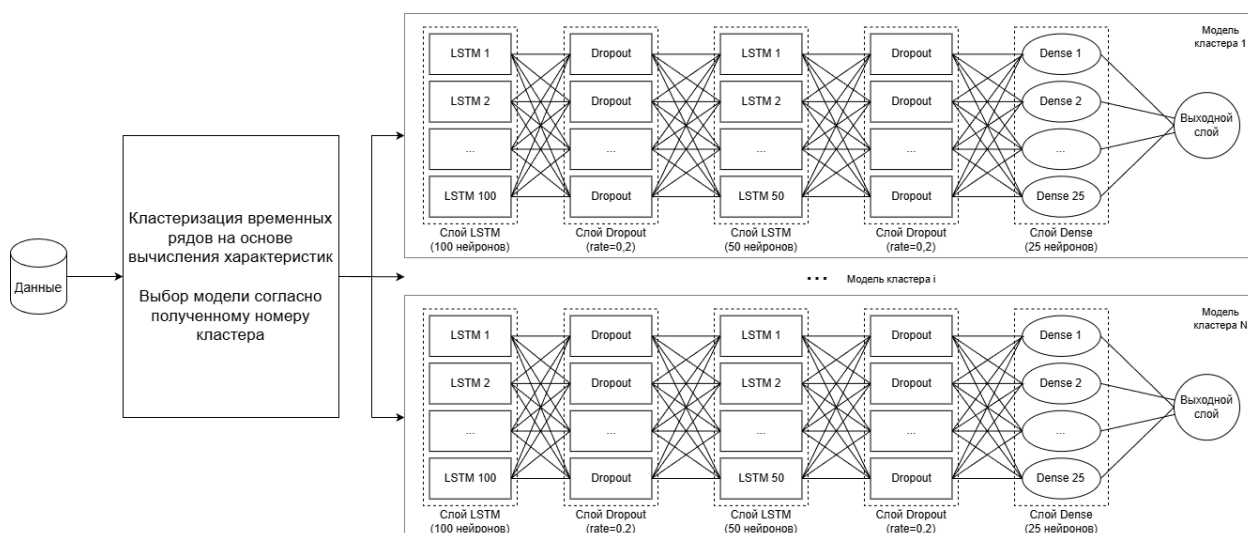


Рисунок 1 – Архитектура гибридной LSTM-модели прогнозирования временных рядов (разработанная в рамках диссертации и внедренная в программный комплекс)

Для каждого кластера мы обучаем свою модель на базе LSTM – она нужна, чтобы работать с последовательными данными и ловить зависимости в динамике показателей.

Начинаем с первого LSTM-слоя на 100 нейронах: тут модель вытаскивает базовые закономерности – вроде устойчивых циклов или типичных колебаний. Сразу после него ставим Dropout с коэффициентом 0,2: по сути, в процессе обучения он случайным образом «выключает» часть нейронов, чтобы модель не подстраивалась под шум и случайные совпадения.

Дальше идёт второй LSTM-слой, уже на 50 нейронах. Он берёт то, что выделил первый слой, и ищет более тонкие, неочевидные связи — то есть прорабатывает динамику детальнее. И тут тоже стоит Dropout 0,2 – чтобы дополнительно застраховаться от переобучения.

После рекуррентных слоёв подключаем полносвязный слой Dense из 25 нейронов: он агрегирует признаки и приводит их к виду, удобному для финального прогноза. На выходе модель даёт ожидаемое значение ряда на нужный горизонт.

Итого в архитектуре – два LSTM-слоя (100 и 50 нейронов), после каждого – Dropout 0,2, далее Dense-слой на 25 нейронов и выходной слой.

Принцип работы метода прогнозирования на основе гибридной LSTM-модели показан на рисунке 2.

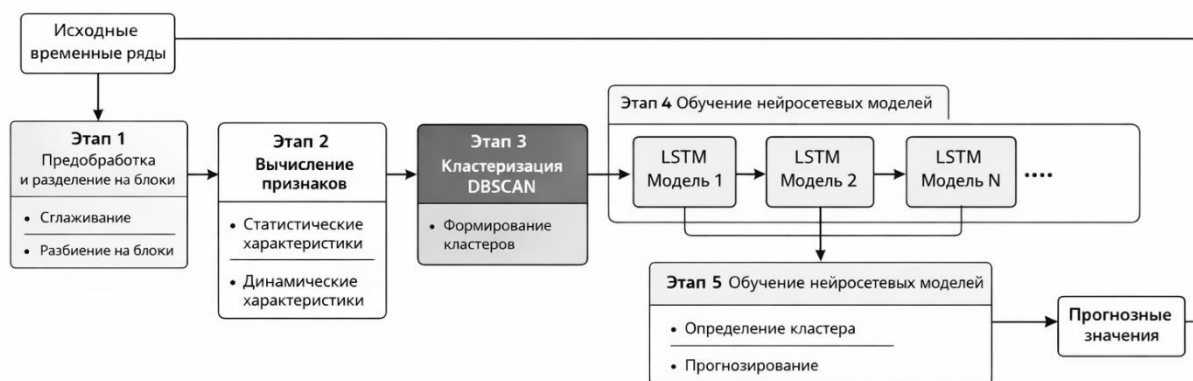


Рисунок 2 – Схема функционирования гибридной нейросетевой модели прогнозирования временных рядов (разработанная в рамках диссертации)

Предложенная гибридная модель позволяет учитывать особенности различных временных рядов при их кластеризации и последующем прогнозировании. Обучение отдельных моделей для каждого кластера дает возможность точнее описывать динамику данных и повышать качество прогнозов для временных рядов с различными характеристиками [86, 89, 90].

2.4 Выводы

Во второй главе диссертации изложены методы и модели, ориентированные на адаптивное прогнозирование неоднородных временных рядов. Ключевой акцент сделан на том, чтобы одновременно повысить точность прогнозов и уменьшить вычислительную нагрузку при работе с данными.

Предложен подход к динамическому выделению сегментов переменной длины для аппроксимации временных рядов линейными участками. Его суть в том, что границы сегментов не задаются жёстко, а определяются автоматически – алгоритм ориентируется на локальные особенности ряда:

резкие изменения, плато, изгибы. За счёт этого удаётся заметно сократить объём обрабатываемых данных, сохранив при этом ключевые черты динамики процесса.

Отдельно разработан метод кластеризации временных рядов, который опирается на анализ характеристик выделенных сегментов. На этапе предобработки для каждого ряда формируют вектор признаков – в него включают как статистические показатели, так и характеристики динамики. Затем к этим векторам применяют алгоритм DBSCAN: он группирует ряды со схожим поведением, позволяя явно учесть их неоднородность.

С учётом результатов кластеризации построена гибридная модель прогнозирования на базе рекуррентных сетей LSTM. Идея в том, чтобы не использовать одну универсальную модель, а для каждого кластера обучать свою – с учётом типичных для этой группы особенностей. Такой подход помогает точнее отражать специфику разных типов временных рядов и в итоге улучшать качество прогнозов.

В совокупности во второй главе сформирован набор согласованных методов: адаптивная аппроксимация данных с динамическим разбиением на сегменты, кластеризация рядов по их динамическим характеристикам и гибридная нейросетевая модель прогнозирования. Эти методы формируют теоретическую основу для построения программного комплекса, реализация которого рассматривается в следующей главе диссертационной работы.

3 ПРОГРАММНЫЙ КОМПЛЕКС РЕАЛИЗАЦИИ РАЗРАБОТАННЫХ МЕТОДОВ И АЛГОРИТМОВ

Теоретические методы и модели, представленные в главе 2, требуют практической реализации для проведения вычислительных экспериментов и последующего внедрения в прикладные системы. Эффективность разработанных алгоритмов может быть полностью раскрыта только при их корректной программной реализации, учитывающей особенности архитектуры целевых устройств и требования к вычислительной эффективности. В связи с этим разработка специализированного программного обеспечения, реализующего предложенные методы, является необходимым этапом диссертационного исследования.

В данной главе представлено описание программного комплекса, созданного для апробации и практического применения разработанных методов адаптивного прогнозирования неоднородных временных рядов. Архитектура комплекса отражает логическую структуру предложенных в главе 2 подходов и включает три основных компонента, каждый из которых реализует соответствующий метод или его часть.

Первый компонент – **адаптивный модуль аппроксимации** (Глава 3.1) – реализует метод динамического определения сегментов переменной длины для кусочно-линейной аппроксимации, описанный в главе 2.1.

Второй компонент – **автоматический синтаксический анализатор** (Глава 3.2) – предназначен для извлечения данных из неструктурированных лог-файлов и формирования обучающих выборок в удобном табличном представлении.

Третий, наиболее сложный компонент – **программный комплекс модели прогнозирования с использованием кластеризации временных рядов** (Глава 3.3) – реализует двухэтапную методологию, представленную в главах 2.2 и 2.3. Комплекс включает три функциональных блока: предобработки данных, обучения и прогнозирования, каждый из которых

реализован в виде специализированных Python-скриптов, интегрированных с ASP.NET Core API-контроллером.

Для каждого модуля приводятся сведения об используемых инструментальных средствах и библиотеках, архитектуре программной реализации, а также примеры работы и выходных данных. Ряд разработанных программных компонентов подтвержден свидетельствами о государственной регистрации программ для ЭВМ, что свидетельствует о практической значимости выполненной работы.

3.1 Адаптивный модуль аппроксимации

Для апробирования разработанного метода адаптивной кусочно-линейной аппроксимации временных рядов с динамическим выбором шага была разработана программа, предназначенная для расчета переменного шага для кусочно-линейной аппроксимации эмпирической нелинейной функции с заданной точностью. Точность определяется как значение коэффициента линейной модели на каждом выделенном в процессе работы программы сегменте. Программа предназначена для апробации метода, описанного в главе 2.1.

Программный модуль для адаптивной аппроксимации был разработан с использованием следующих инструментальных средств разработки: интегрированная среда разработки Visual Studio 2022 Community Edition, модульная платформа для разработки программного обеспечения .NET 7, высокоуровневый язык программирования C#. Для генерации электронных таблиц использовалась библиотека Microsoft.Office.Interop.Excel. Архитектура разработанного программного решения представлена ниже.

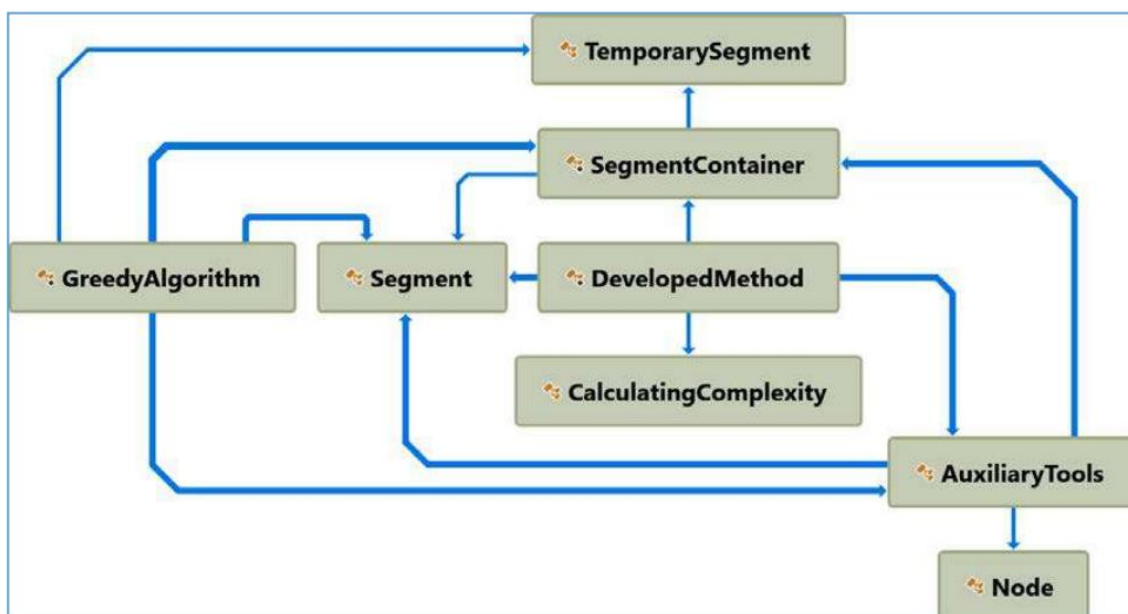


Рисунок 3 – Архитектура программного модуля адаптивной кусочно-линейной аппроксимации временных рядов

Адаптивный модуль аппроксимации состоит из следующих функциональных частей [91]:

1) Node – узел таблицы, хранящий в себе номер участка, коэффициенты линейной модели, номер начального узла, номер конечного узла;

2) AuxiliaryTools – класс, отвечающий за вычисление метрик, которые определяют адекватность построенной кусочно-линейной модели;

3) Segment – выделенный участок нелинейной функции, созданный в процессе динамического разбиения и хранящий таблицу координат каждой точки сегмента, номер участка, коэффициенты линейной модели, практические значения функции, вычисленный коэффициент детерминации, индекс текущей позиции, номер начального узла, номер конечного узла, длину сегмента;

4) SegmentContainer – коллекция полученных в процессе разбиения сегментов;

5) TemporarySegment – временная сущность, которая используется для хранения длины сегмента, наихудшего коэффициента детерминации и допустимого значения коэффициента детерминации;

6) CalculatingComplexity – класс, используемый для вычисления количества выполненных арифметических операций: сложений, вычитаний, умножений, делений;

7) DevelopedMethod – класс, инкапсулирующий логику разработанного метода аппроксимации (ядро модуля);

8) GreedyAlgorithm – класс, инкапсулирующий логику жадного алгоритма с выделением сегментов постоянной длины.

Результат работы приложения при выборе жадного алгоритма представлен ниже.

Number Segment	i	j	A	B	L	min R2	ε
0	0	53	-0.794577474269221	4.2087631	26	0.9596288035272	0.996
1	0	8	-0.357057295475281	4.0137696	17	0.95136743525044	0.996
2	8	16	-0.855573424001769	4.4791370	13	0.939052347905393	0.996
3	16	24	-1.10534892386494	4.9166168	10	0.92222615028022	0.996
4	24	32	-0.873959399588022	4.2459249	8	0.969840084642356	0.996
5	32	40	-0.623032286411303	3.3705390	7	0.969840084642356	0.996
6	40	48	-0.332807126418982	2.1811380	6	0.969840084642356	0.996
					5	0.969840084642356	0.996
					4	0.961668397077522	0.996
					3	0.913097011785557	0.996
					2	0.969840084642356	0.996
					26	0.9596288035272	0.995
					17	0.95136743525044	0.995
					13	0.939052347905393	0.995
					10	0.92222615028022	0.995

Рисунок 4 – Работа приложения при выборе жадного алгоритма

Результат работы приложения при выборе адаптивного режима представлен ниже.

Number Segment		i	j	A	B	R2	L	mm	R2	s
0		0	53	-0.794577474269221	4.2087631167744	0.969840084642356				
1		1	6	-0.327215569675582	4.00427803433093	0.996527536608937				
2		7	37	-0.983650686013575	4.63561655464408	0.996068916836431				
3		38	39	-0.468085106383702	2.76102127659861	1				
4		40	42	-0.450300508552688	2.67210355986953	0.998819397260889				
5		43	44	-0.588235294117069	3.28105882352692	1				
6		45	46	-0.51999999997016	3.00747999998672	1				
7		47	48	-0.188811188811136	1.5030489510487	1				
8		49	52	-0.377618356561656	2.4123458010444	0.999757104117151				

Рисунок 5 – Работа приложения при выборе адаптивного режима

При выборе каждого из режимов приложение генерирует электронный документ, где сохраняются наиболее важные метрики, необходимые для дальнейшего анализа полученных результатов вычислительных экспериментов. Пример сгенерированного документа представлен ниже.

	A	B	C	D	E	F
1	№	i	j	A	B	R ²
2	0	0	53	-0,794577474	4,208763117	0,969840085
3	1	0	8	-0,357057295	4,013769664	0,982515612
4	2	8	16	-0,855573424	4,479137032	0,996829752
5	3	16	24	-1,105348924	4,916616846	0,999248874
6	4	24	32	-0,8739594	4,24592499	0,994836493
7	5	32	40	-0,623032286	3,370539051	0,992433202
8	6	40	48	-0,332807126	2,181138024	0,984747885
9						
10						
11						
12	Количество сравнений = 0					
13	Количество сложений = 424					
14	Количество вычитаний = 3					
15	Количество умножений = 165					
16	Количество делений = 1					

Рисунок 6 – Пример сгенерированного электронного документа

На описанный программный модуль было получено свидетельство о регистрации программы для ЭВМ №2022665809 (Номер заявки: 2022664687) [92].

В дальнейшем планируется усовершенствование модуля аппроксимации путем вынесения всех вычислительных мощностей на отдельный веб-сервис, где запросы будут обрабатываться посредством поступающих HTTP-запросов. Данный подход поможет обеспечить более высокую гибкость программного модуля и возможность ее интеграции с техническими устройствами различного вида.

3.2 Автоматический синтаксический анализатор данных биологического мониторинга

Для практической апробации разработанной в главе 2.3 гибридной нейросетевой модели прогнозирования временных рядов, основанной на предварительной кластеризации, требуются репрезентативные данные из предметной области. В качестве такой области выбрана персонализированная медицина, а именно задача краткосрочного прогнозирования уровня гликемии у пациентов с сахарным диабетом 1 типа.

В качестве обучающей выборки для исследования эффективности использования нейросетевых алгоритмов для прогнозирования значений уровня гликемии требуется построить многомерный временной ряд, включающий следующие компоненты: количество активных углеводов, количество активного инсулина и текущий уровень гликемии.

Данные для построения временного ряда могут быть получены из журналов записей серверного модуля (лог-файлы). Они представляют собой специальные файлы, в которых протоколируются определённые действия программы на сервере.

Из всех протоколированных действий для извлечения параметров временного ряда требуется разработать программный синтаксический анализатор исходного текста. После обработки выделенных значений данные должны быть помещены в таблицу для дальнейшего обучения и валидации нейронных сетей.

Для поиска искомых параметров в лог-файле был использован один из эффективных методов поиска подстроки в строке «алгоритм Кнута – Морриса – Пратта». Данный метод использует предобработку искомой строки и создаёт на её основе префикс функцию. Суть этой функции заключается в нахождении наибольшей подстроки, одновременно находящейся и в начале, и в конце подстроки. Смысл префикс функции заключается в том, что неверные варианты могут быть заранее отброшены, а в начале работы могут рассматриваться некоторые вспомогательные утверждения, где для произвольного слова рассматриваются все его начала, которые по совместительству являются его концами, и выбирается из них самое длинное. Метод Кнута – Морриса – Пратта использует следующую идею: если префикс строки длиной i длиннее другого символа, то он одновременно и префикс подстроки длиной i_1 . Время работы всей процедуры линейно и есть $O(m)$ несмотря на то, что в ней присутствует вложенный цикл [93].

Для написания программы, реализующей вышеописанный алгоритм, был использован высокоуровневый язык программирования C# в интегрированной среде разработки Visual Studio 2022.

Пример необработанных логгированных данных приведён ниже.

```

{"carbs":0,"nsCarbs":0,"bwCarbs":0,"journalCarbs":0,"mealCOB":0,"currentDeviation":1.33,"maxDeviation":1.38,"minDeviation":0.76,"slopeFromMaxDeviation":-0.009,"slopeFromMinDeviation":0.225,"allDeviations":[1,1,1,1,1],"lastCarbTime":0,"bwFound":false}
{"iob":-0.647,"activity":-0.0022,"basaliob":-0.756,"bolusiob":0.109,"netbasalinsulin":-0.5,"bolusinsulin":2.2,"time":"2019-03-03T21:01:39.000Z","iobWithZeroTemp":{"iob":-0.647,"activity":-0.0022,"basaliob":-0.756,"bolusiob":0.109,"netbasalinsulin":-0.5,"bolusinsulin":2.2,"time":"2019-03-03T21:01:39.000Z"},"lastBolusTime":1551636172000,"lastTemp":{"rate":0.85,"timestamp":"2019-03-03T23:56:52+03:00","started_at":"2019-03-03T20:56:52.000Z","date":1551646612000,"duration":5.83}}
{"delta":2,"glucose":90,"noise":null,"short_avgdelta":1.78,"long_avgdelta":0.71,"date":1551646740000}
Autosens ratio: 1.2; Adjusting basal from 0.55 to 0.65; ISF from 69.1 to 57.6; CR: 5.818
currenttemp: { duration: 26, rate: 0.85, temp: 'absolute' } lastTempAge: 5
m tempModulus: 1 m

```

Рисунок 7 – Фрагмент лог-файла с необработанными данными, поступающий из сенсоров

После обработки данные, которые необходимо для включения в обучающую выборку, помещались в таблицу. Таблица с обработанными измеряемыми данными представлена ниже (здесь Carbs – количество активных углеводов, IOB – количество активного инсулина, Glucose – текущий уровень гликемии в мг/дл). Полученная таблица применялась в вычислительном эксперименте, описанном в главе 4.2.2.

Date	Time	Carbs	IOB	Glucose
2018-03-1	06:30:09	0	-0,08	142
2018-03-1	06:32:03	0	-0,023	140
2018-03-1	06:34:44	0	0,136	140
2018-03-1	06:37:56	0	0,197	140
2018-03-1	06:40:37	0	0,354	140
2018-03-1	06:43:25	0	0,361	140
2018-03-1	06:46:35	0	0,268	140
2018-03-1	06:48:38	0	0,272	138
2018-03-1	06:49:51	0	0,273	138
2018-03-1	06:51:55	0	0,278	136
2018-03-1	06:53:30	0	0,28	136
2018-03-1	06:54:59	0	0,282	136
2018-03-1	06:56:30	0	0,283	136
2018-03-1	06:58:59	0	0,509	134

Рисунок 8 – Таблица с данными для обучения нейронной сети

На описанный программный модуль было получено свидетельство о регистрации программы для ЭВМ №2025661819 (Номер заявки: 2025661093) [94].

3.3 Программный комплекс модели для прогнозирования с использованием кластеризации временных рядов

Для автоматизированной обработки временных рядов, включая их предобработку, обучение модели LSTM и прогнозирование, был разработан API-контроллер на основе технологии ASP.NET Core. Этот контроллер обеспечивает удобный интерфейс для взаимодействия с обработчиком данных, реализованным на Python, и интеграции с клиентскими приложениями. Благодаря такому подходу, разработанное решение легко адаптируется к различным сценариям использования, позволяя отправлять запросы как из веб-интерфейса, так и с мобильных и десктопных устройств.

Структура API-контроллера включает три основных блока [95]:

1. Блок предобработки данных – отвечает за очистку и сглаживание временных рядов, заполнение пропущенных значений, выполнение кластеризации и сохранение обученной модели кластеризации.
2. Блок обучения – осуществляет обучение моделей LSTM для каждого кластера временных рядов по отдельности.
3. Блок прогнозирования – выполняет разбиение входного временного ряда на сегменты, определяет их принадлежность к ранее сформированным кластерам и использует соответствующую обученную модель LSTM для предсказания.

Структура архитектуры программного комплекса представлена ниже.

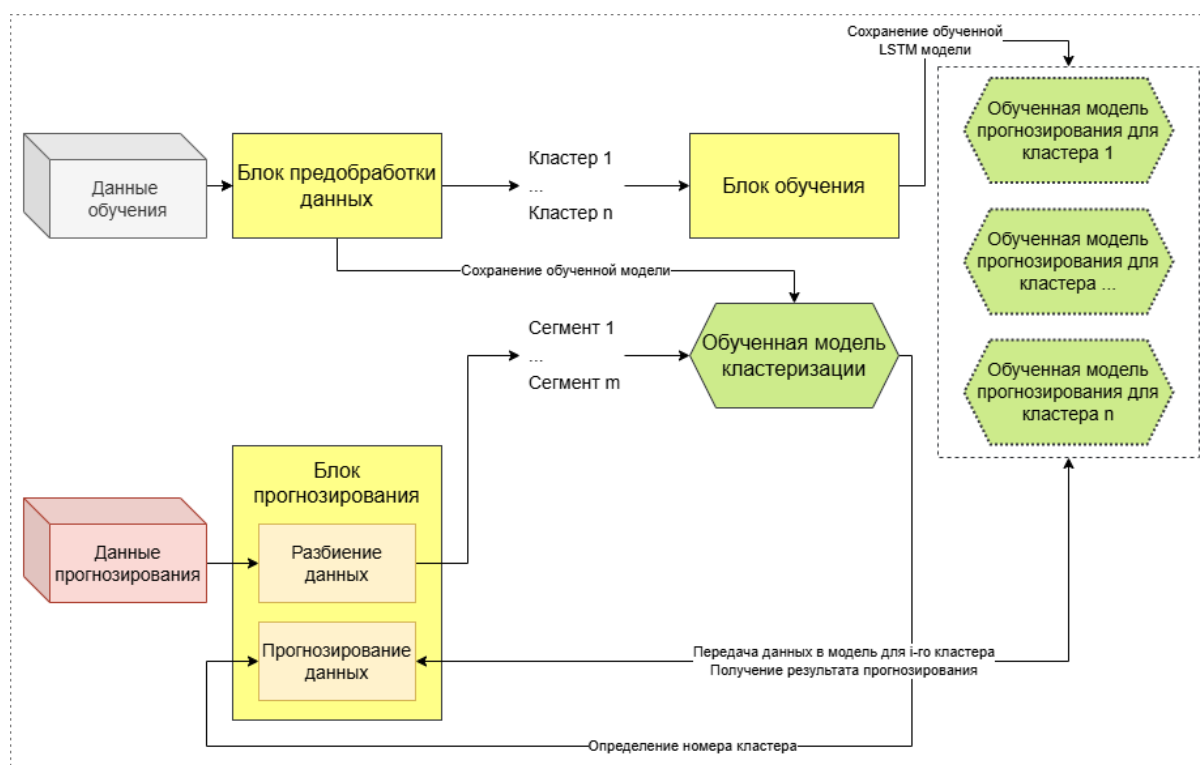


Рисунок 9 – Архитектура программного комплекса

Для разработки серверной части API-контроллера был использован фреймворк ASP.NET Core, обеспечивающий высокую производительность и кроссплатформенность. Основной язык программирования – C#, который позволяет создавать надежные и масштабируемые веб-приложения. Для документирования и тестирования API применена библиотека Swagger,

автоматически генерирующая удобный интерфейс для отправки HTTP-запросов и просмотра их ответов.

Работа с файлами реализована с использованием библиотек IFormFile и System.IO, которые позволяют загружать, сохранять и передавать в обработку Excel-документы с временными рядами. Для взаимодействия с Python-скриптами применяется System.Diagnostics.Process, который обеспечивает вызов внешних процессов, включая выполнение предобработки данных, кластеризации, обучения моделей и прогнозирования.

На описанный программный модуль было получено свидетельство о регистрации программы для ЭВМ №2025661810 (Номер заявки: 2025661090) [96].

Схема алгоритма работы блока предварительной обработки данных представлена на рисунке 10.

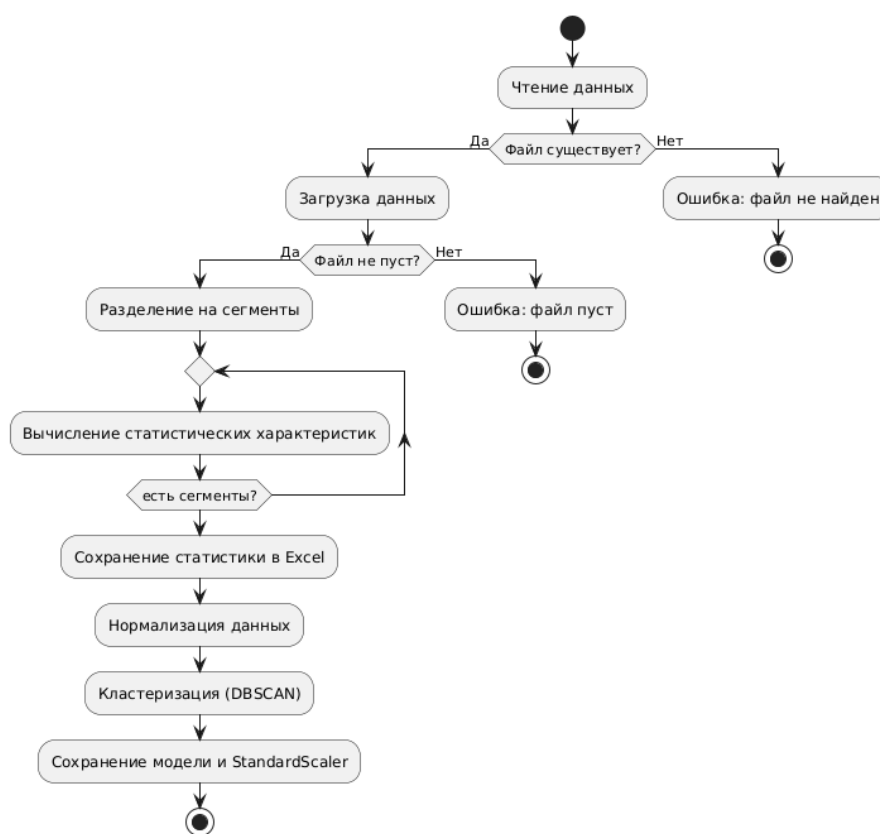


Рисунок 10 – Алгоритм работы блока предобработки

Схема алгоритма работы блока обучения представлена на рисунке 11.



Рисунок 11 – Алгоритм работы блока обучения

Схема алгоритма работы блока прогнозирования представлена рисунке

12.

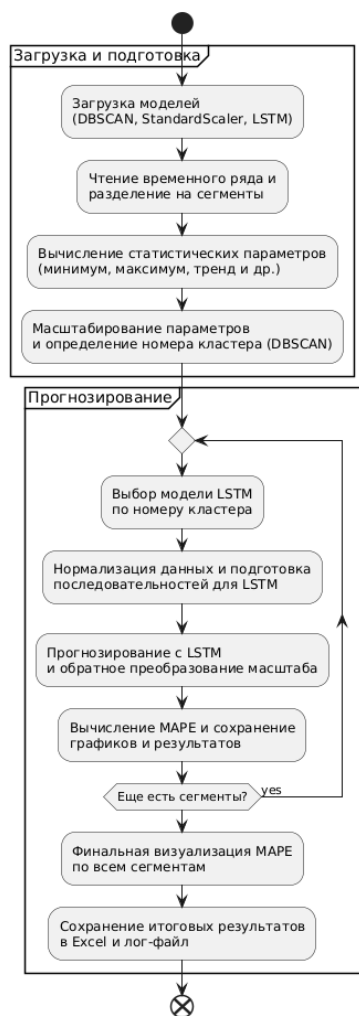


Рисунок 12 – Алгоритм работы блока прогнозирования

3.4 Выводы

В третьей главе описан разработанный программный комплекс, реализующий предложенные методы обработки и прогнозирования временных рядов; система обеспечивает полный цикл работы с данными – от загрузки и предварительной обработки до аппроксимации, кластеризации и построения прогноза.

В состав программного комплекса входит модуль адаптивной аппроксимации, реализующий алгоритм динамического определения сегментов переменной длины. Его использование позволяет уменьшить объем

обрабатываемых данных без существенной потери информации о форме временного ряда и тем самым снижает вычислительные затраты на последующих этапах обработки.

Реализован также модуль автоматического синтаксического анализа данных биологического мониторинга: он выполняет разбор исходных файлов, преобразует данные к единому формату и подготавливает их для дальнейшего использования в вычислительных экспериментах.

Наконец, реализован программный комплекс прогнозирования временных рядов на основе разработанного метода кластеризации и гибридной нейросетевой модели; его архитектура включает блок предобработки данных, блок обучения нейросетевых моделей и блок прогнозирования, что обеспечивает модульность системы и возможность её адаптации к различным типам временных данных.

В результате разработанный программный комплекс обеспечивает практическую реализацию предложенных методов и может использоваться в системах анализа и прогнозирования временных рядов в различных прикладных областях.

4 ВЫЧИСЛИТЕЛЬНЫЕ ЭКСПЕРИМЕНТЫ ПО ПРАКТИЧЕСКОМУ ИСПОЛЬЗОВАНИЮ РАЗРАБОТАННЫХ МЕТОДОВ, МОДЕЛИ И АЛГОРИТМОВ

Разработанные в Главе 2 теоретические методы и реализующий их программный комплекс, описанный в Главе 3, требуют всесторонней экспериментальной проверки для подтверждения их эффективности, корректности и практической применимости.

4.1 Вычислительные эксперименты по применению метода адаптивной кусочно-линейной аппроксимации временных рядов

В данной главе представлены результаты экспериментального исследования эффективности разработанного метода адаптивной кусочно-линейной аппроксимации временных рядов с динамическим выбором шага. Эффективность метода оценивается на трёх практически значимых задачах, различающихся по характеру аппроксимируемых данных, требованиям к точности и областям применения:

1. Аппроксимация траекторий управления гибридным беспилотным летательно-подводным аппаратом (БПЛА) – оценка возможности применения метода для задания сложных угловых траекторий управления беспилотными аппаратами в условиях ограниченных бортовых вычислительных ресурсов.

2. Аппроксимация кривой активности экзогенного инсулина – оценка применимости метода в медицинских носимых устройствах (инсулиновых помпах) для расчёта активного инсулина у пациентов с сахарным диабетом 1 типа.

3. Аппроксимация траектории движения автомобиля – оценка эффективности метода для компактного представления траекторий

транспортных средств в задачах цифрового имитационного моделирования транспортных процессов.

Выбор указанных задач обусловлен их высокой практической значимостью, различной динамикой аппроксимируемых функций (участки разгона, равномерного движения, торможения, резкие изменения угловых координат) и различными требованиями к точности аппроксимации.

В качестве эталона сравнения используется классический подход с фиксированным шагом разбиения, при котором исходная функция аппроксимируется на интервалах минимальной (базовой) длины. Данный подход соответствует традиционному табличному представлению функций и применяется в большинстве практических задач. Сравнение проводится при идентичных требованиях к точности аппроксимации, контролируемых через коэффициент детерминации R^2 .

Количественная оценка эффективности выполняется по следующим метрикам:

- 1) число выделенных сегментов аппроксимации – характеризует компактность представления модели;
- 2) объём хранимых коэффициентов и параметров сегментов – определяет требования к памяти вычислительного устройства;
- 3) количество арифметических операций – характеризует вычислительную сложность построения модели.

Выбор указанных метрик обусловлен необходимостью оценки пригодности разработанного метода для использования на вычислительных устройствах с ограниченными ресурсами (бортовые компьютеры беспилотных аппаратов, носимые медицинские устройства, системы имитационного моделирования).

В разделах 4.1.1–4.1.3 подробно рассматриваются результаты экспериментов для каждой из трёх задач, а в заключительной части раздела приводится сравнительный анализ эффективности предложенного метода по отношению к классическому подходу с постоянным шагом.

4.1.1 Кусочно-линейная аппроксимация сложных траекторий управления с переменным шагом на примере управления беспилотными летательными подводными аппаратами

В качестве примера работы алгоритма рассмотрим кусочно-линейную аппроксимацию целевой траектории движения специфического беспилотного аппарата – летательного подводного аппарата (БПЛА) типа окто-квадрокоптер [97]. БПЛА могут передвигаться в двух средах – по воздуху, как беспилотные летательные аппараты (БПЛА), и под водой, как автономные необитаемые подводные аппараты (АНПА). При этом гибридные БПЛА способны многократно перемещаться через границу «воздух-вода» в обоих направлениях. Одна из самых перспективных конструкций подобных аппаратов представляет собой дрон с восемью винтами: четырьмя подводными и четырьмя воздушными, расположенными одни над другими (двухслойные БПЛА). Такие аппараты получили название окто-квадрокоптеры. Классические октокоптеры также обладают восемью винтами, но все они предназначены для воздушного полета, хотя и могут располагаться в разных плоскостях. Компьютерная симуляция движения одного из существующих прототипов описываемого БПЛА, названного создателями «HyDrone» [98], представлена на рисунке 13.

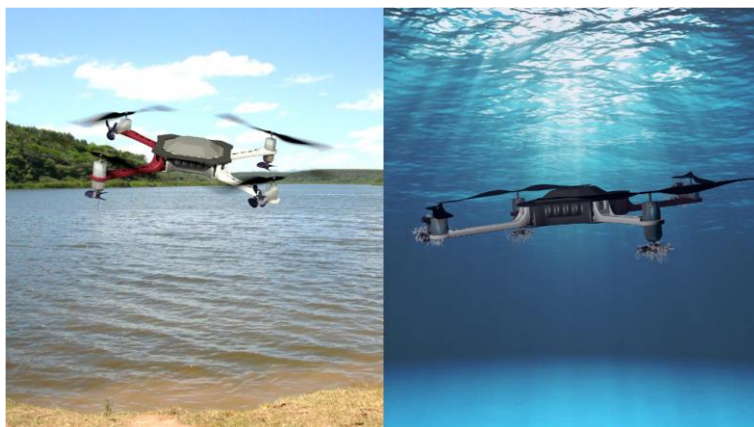


Рисунок 13 – Иллюстрация передвижения БПЛА «HyDrone»: а) при помощи воздушных винтов над водой; б) при помощи подводных винтов под водой

При проектировании систем автоматического управления окто-квадрокоптеров их целевая траектория является одной из самых сложных из-за требований к перемещению в двух средах. Именно поэтому данные траектории были выбраны для проведения вычислительных экспериментов.

Рассмотрим закон управления, описанный в [98]: навигация окто-квадрокоптера в загроможденной среде с переходом из воды в воздух. Целевая траектория перемещения аппарата в трехмерном пространстве представлена на рисунке 14.

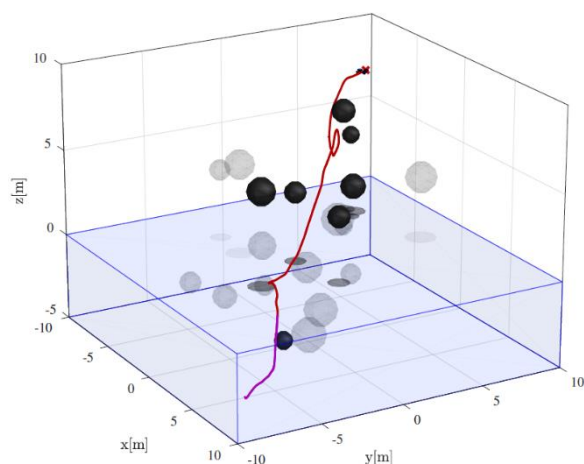


Рисунок 14 – След перемещения БПЛА в трехмерном пространстве по заданной траектории

Целевая траектория по трем координатам описана тремя графиками эмпирических функций (рисунок 15):

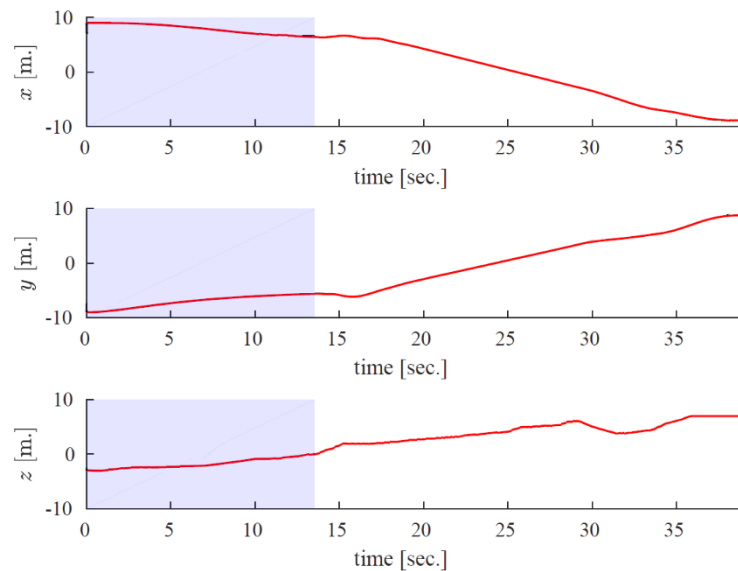


Рисунок 15 – Целевая траектория по пространственным координатам (x, y, z)

Управление осуществлялось посредством изменения скоростей вращения и углов поворотов (тангаж-крен-рысканье) гребных и воздушных винтов. Изменение углов тангажа, крена, и рысканья соответственно для достижения заданной траектории представлено на рисунке 16:

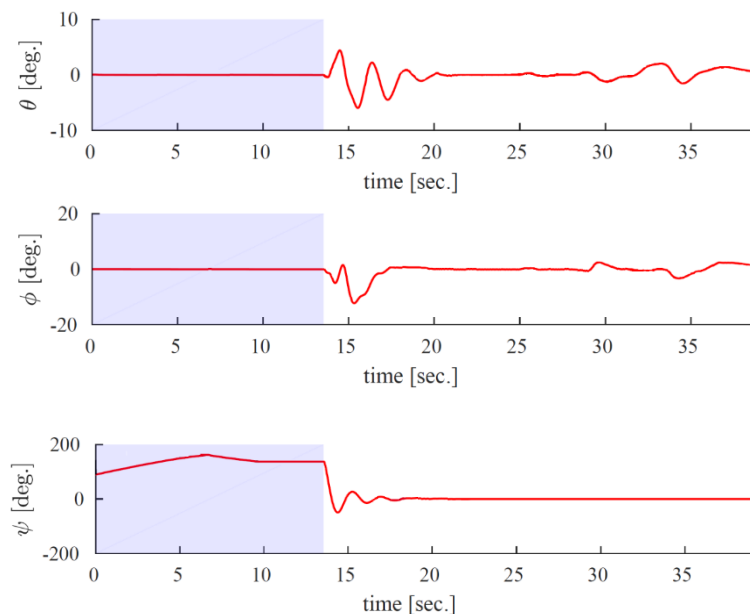


Рисунок 16 – Целевая траектория по угловым координатам (θ - угол тангажа, ϕ - угол крена, ψ - угол рысканья)

Вычислительные эксперименты проведены для трех целевых траекторий по угловым координатам (θ, ϕ, ψ) .

Для проведения вычислительных экспериментов графики угловых траекторий были оцифрованы и оформлены в виде таблиц с минимальным постоянным шагом в 0,5 секунды, что составило 81 контрольных точек. Затем полученные значения были сглажены по методу экспоненциального сглаживания для уменьшения влияния внешнего шума. Полученное табличное представление траектории управления можно принять за базовую аппроксимацию с шагом $h = 0,5$. Фрагменты табличного представления полученных кривых с постоянным шагом $h = 0,5$ представлены в таблицах 8-10.

Таблица 8 – Табличное представление базовой модели управления по углу тангажа Θ (фрагмент)

Время	Целевой угол тангажа	Коэффициенты базовой аппроксимации ($h = 0,5$)		$\hat{y} = kx + b$
		k	b	
0	0	0	0	0
0,5	0	0	0	0
...
14,5	3,592812	-4,43576	67,91139	3,592812
15	1,37493	-12,9394	195,4656	1,37493
...
79,5	0,542765624	-0,094587087	4,27895558	0,542765624
80	0,49547208	0,012386802	0	0,49547208

Таблица 9 – Табличное представление базовой модели управления по углу крена ϕ (фрагмент)

Время	Целевой угол крена	Коэффициенты базовой аппроксимации ($h = 0,5$)		$\hat{y} = kx + b$
		k	b	
0	0	0	0	0
0,5	0	0	0	0
...
14,5	-4,87367	11,79567	-175,911	-4,87367
15	1,024168	-22,4155	337,256	1,024168
...
79,5	1,600445	-1,226	50,02755	1,600445
80	0,987443	0,024686	0	0,987443

Таблица 10 – Табличное представление базовой модели управления по углу рысканья ψ (фрагмент)

Время	Целевой угол рысканья	Коэффициенты базовой аппроксимации ($h = 0,5$)		$\hat{y} = kx + b$
		k	b	
0	110,31	14,45903	110,31	110,31
0,5	117,5395	14,84078	110,1191	117,5395
...
14,5	-29,78	-15,0731	188,7796	-29,78
15	-37,3166	138,4873	-2114,63	-37,3166
79,5	0	0	0	0
80	0	0	0	0

Здесь k , b – коэффициенты линейной модели, \hat{y} – значения, полученные при подстановке узловых точек x_i в полученную аналитическую модель. Способ представления эмпирической функции в табличном виде с минимальным шагом показывает точные значения полученных данных при подстановке в аппроксимирующую функцию, однако он не приемлем для практических вычислений на бортовом компьютере БПЛПА, так как требует больших вычислительных затрат и большого объема памяти для хранения табличных данных. В терминах разработанного алгоритма, базовый подход требует вычислений для 80-ти сегментов каждой пары соседних узловых точек: $S^1(1,2)$, $S^2(2,3)$... $S^{80}(80,81)$.

4.1.1.1 План эксперимента

Для сравнения эффективности предложенного алгоритма с алгоритмом при постоянном шаге будет построено две модели кусочно-линейной аппроксимации для каждой из трех аппроксимируемых функций с одной и той же точностью для двух подходов. Точность ε определяется как значение коэффициента детерминации R^2 линейной модели на каждом выделенном в

процессе работы алгоритмов сегменте. Точность всей модели принимается равной минимальному значению R^2 по всем сегментам.

Для поиска оптимальной длины сегментов при постоянном шаге используется «жадный» алгоритм [102]. На нулевой итерации полагаем, что все точки разбиения образуют единый сегмент. Количество сегментов $k = 1$. На каждой последующей итерации исходная матрица разделяется T на k сегментов равной длины и вычисляется коэффициент детерминации R^2 для каждого полученного сегмента. Если для каждого полученного сегмента $R^2 > \varepsilon$, то разделение считается корректным. В противном случае каждый выделенный ранее сегмент разбивается на два подсегмента, и описанные выше действия повторяются для $k = 2k$.

В качестве количественных критериев сравнения принято количество параметров, которые требуется сохранять в памяти бортового компьютера БПЛА, а также количество операций сравнения и переключений для осуществления управления с заданной точностью.

4.1.1.2 Реализация эксперимента

Для тестирования разработанного метода кусочно-линейной аппроксимации рассмотрено разбиение траекторий управления по тангажу, крену и рысканью, оцифрованные значения которых приведены в таблицах 8-10, с точностью аппроксимации не ниже $\varepsilon = 0,9$. То есть при замене исходных кривых управления кусочно-линейной функцией, коэффициент детерминации R^2 , характеризующий адекватность аппроксимации, не должен быть ниже 0,9.

При допустимой точности кусочно-линейной аппроксимации $\varepsilon = 0,9$ результаты показали следующее.

Для аппроксимации с переменным шагом управления по углу тангажа потребовалось выделить 26 сегментов, длина которых варьировалась от 0,5 сек. до 13,5 сек. В таблице 11 представлен фрагмент результатов аппроксимации с выделенными границами сегментов, их длиной, точностью

аппроксимации и коэффициентами линейной аппроксимации для каждого сегмента.

Таблица 11 – Характеристики кусочно-линейной аппроксимации траектории управления БПЛА по тангажу при точности $\varepsilon = 0,9$ (фрагмент)

Номер сегмента	Начало линейного управления, сек.	Конец линейного управления, сек.	шаг h (сек.)	Коэффициенты аппроксимации на выделенном сегменте		R^2
				k	b	
1	0	13,5	13,5	0	0	1
2	13,5	14	0,5	-0,954	12,879	1
...
17	24	26	2	0,085	-2,177	0,93
...
24	34	35,5	1,5	-2,517	87,818	0,95
25	35,5	37,5	2	1,421	-52,018	0,9
26	37,5	40	2,5	-0,312	13,001	0,927

В силу более гладкого вида траектории управления по крену, для аппроксимации с заявленной точностью 0,9 потребовалось всего 18 сегментов длиной от 0,5 сек. до 13,5 сек. В таблице 12 представлен фрагмент результатов аппроксимации по аналогии с таблицей 11.

Таблица 12 – Характеристики кусочно-линейной аппроксимации траектории управления БПЛА по крену при точности $\varepsilon = 0,9$ (фрагмент)

Номер сегмента	Начало линейного управления, сек.	Конец линейного управления, сек.	шаг h (сек.)	Коэффициенты аппроксимации на выделенном сегменте		R^2
				k	b	
1	0	13,5	13,5	0	0	1
2	13,5	14	0,5	-0,516	6,979	1
...
11	26,5	29	2,5	-0,170	4,554	0,926
...
17	35,5	38	2,5	2,145	-79,154	0,943
18	38	40	2	-0,696	28,866	0,909

Для аппроксимации управления по углу рысканья с заявленной точностью 0,9 потребовалось всего 18 сегментов длиной от 0,5 сек. до 17 сек. Фрагмент результирующих данных представлен в таблице 13.

Таблица 13 – Характеристики кусочно-линейной аппроксимации траектории управления БПЛА по рысканью при точности $\varepsilon = 0,9$ (фрагмент)

Номер сегмента	Начало линейного управления, сек.	Конец линейного управления, сек.	шаг h (сек.)	Коэффициенты аппроксимации на выделенном сегменте		R^2
				k	b	
1	0	6,5	6,5	12,97763	110,31	0,993
2	6,5	13,5	7	-3,88258	219,901	0,900
...
8	15,5	16,5	1	-54,1785	871,693	0,999
...
17	22,5	23	0,5	8,683921	-199,73	1
18	23	40	17	0	0	1

Расхождение графиков реальных и рассчитанных кусочно-линейных траекторий демонстрируют рисунки 17-19.

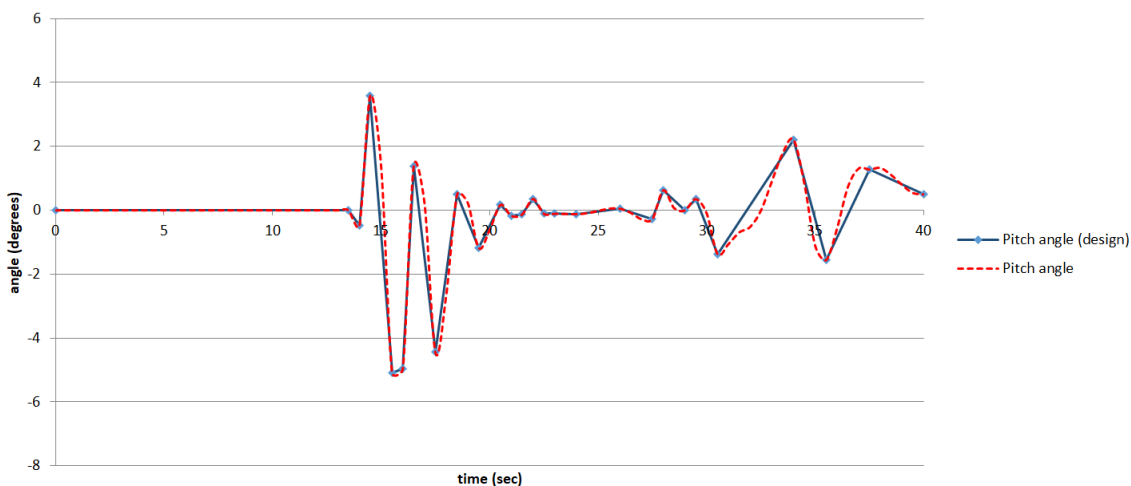


Рисунок 17 – Графики изменения значения угла тангажа (красная пунктирная линия – реальная траектория, синяя ломаная – кусочно-линейная аппроксимация управления)

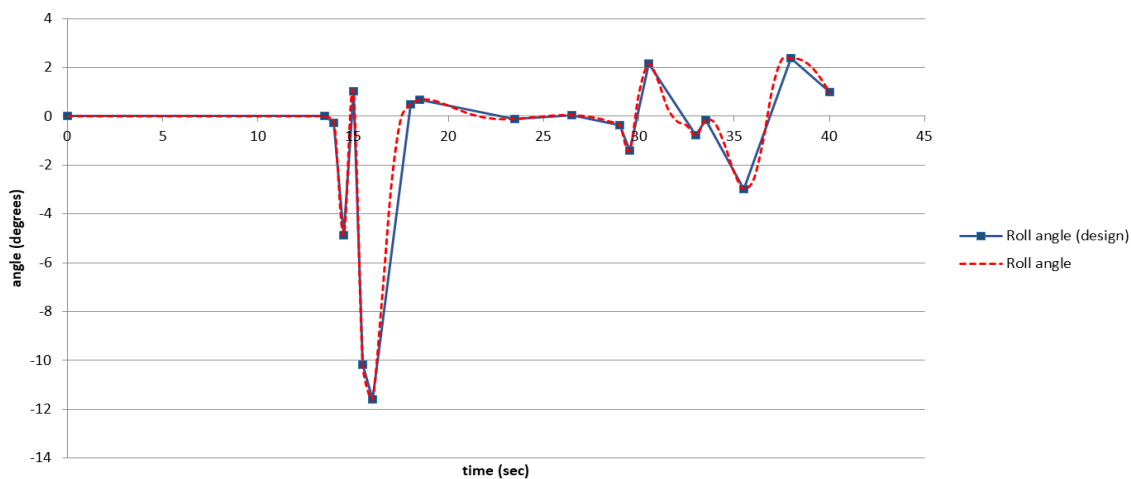


Рисунок 18 – Графики изменения значения угла крена (красная пунктирная линия – реальная траектория, синяя ломаная – кусочно-линейная аппроксимация управления)

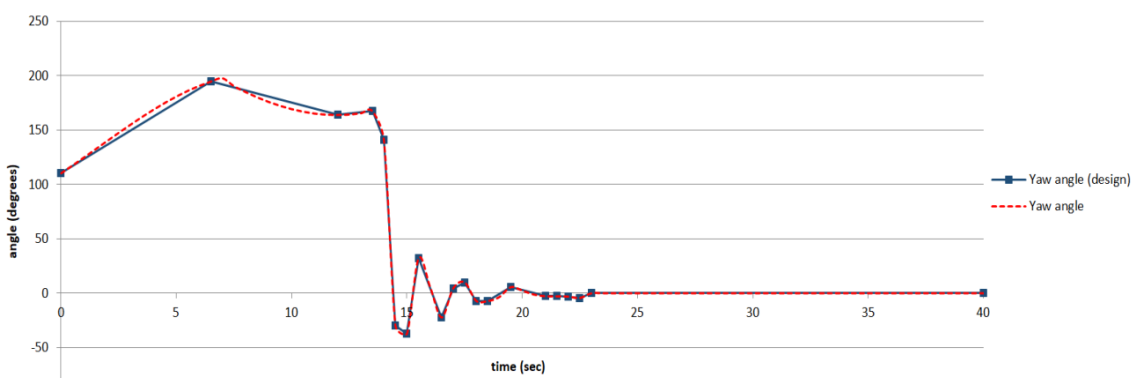


Рисунок 19 – Графики изменения значения угла рысканья (красная пунктирная линия – реальная траектория, синяя ломаная – кусочно-линейная аппроксимация управления)

Абсолютные отклонения траекторий в узловых точках равны нулю. Вне узловых точек отклонения демонстрируют рисунки 20-22.

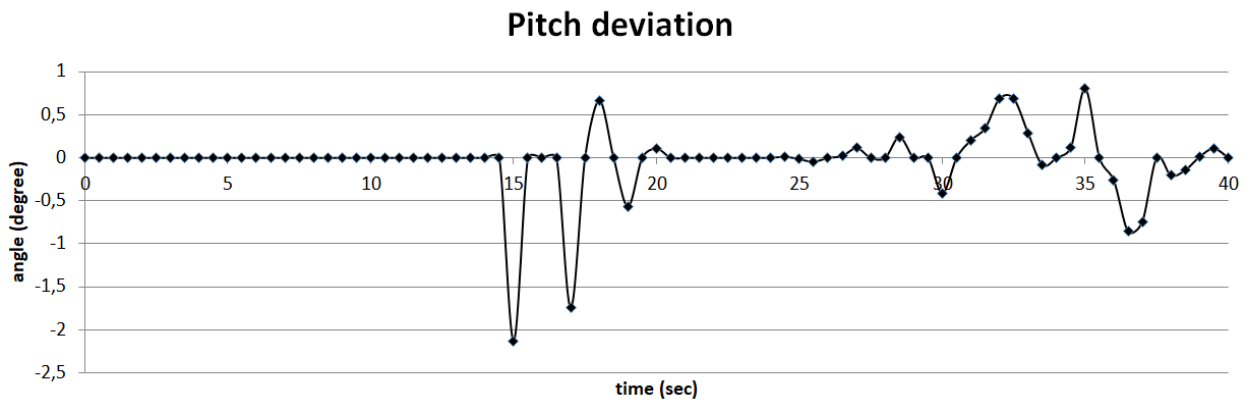


Рисунок 20 – Отклонение по тангажу

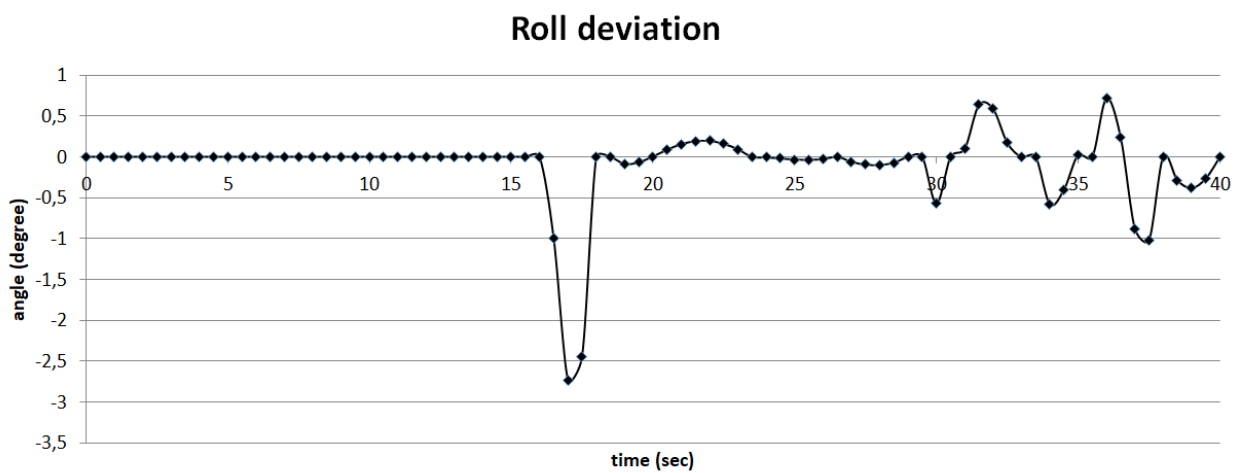


Рисунок 21 – Отклонение по крену

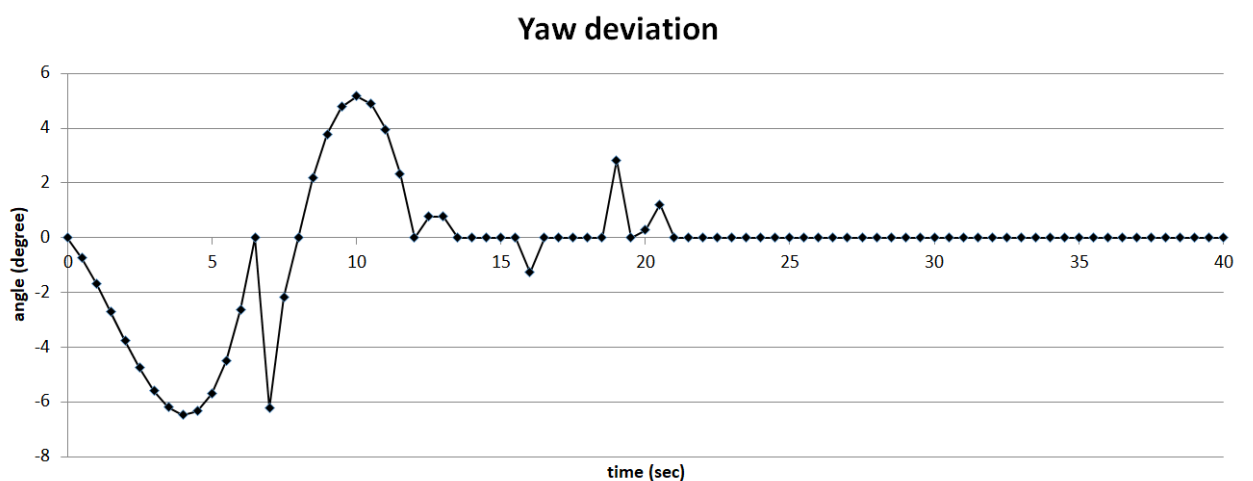


Рисунок 22 – Отклонение по рысканью

Средние относительные отклонения кусочно-линейных траекторий от реальных приведены рисунке 23.

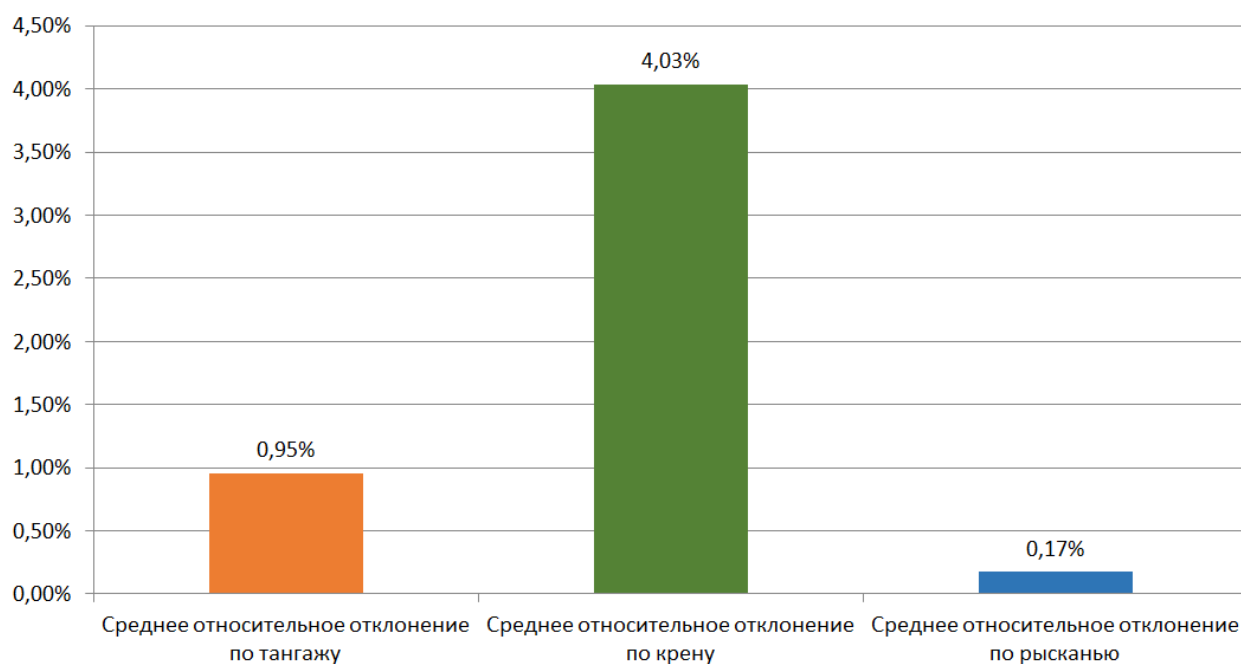


Рисунок 23 – Относительные отклонения аппроксимирующей и реальных траекторий по трем критериям управления

Для практического осуществления управления в памяти бортового микрокомпьютера должны сохраняться координаты границ выделенных сегментов, либо координаты левых границ интервалов с их длинами, а также коэффициенты линейной аппроксимации для каждого из сегментов. Затраты памяти устройства для приведенного выше эксперимента, таким образом, составляют (таблица 14).

Таблица 14 – Количество параметров, хранимых в памяти бортового компьютера БПЛА для осуществления управления с переменным шагом

	Координата левой границы интервалов	Длина интервала	Коэффициенты линейной аппроксимации	Всего
Управление по тангажу	26	26	52	104
Управление по крену	18	18	36	72
Управление по рысканью	18	18	36	72
ИТОГО	62	62	124	248

Количество необходимых операций при осуществлении управления за весь период управления (40 сек.) можно оценить числом сравнений для поиска требуемого сегмента, а также числом переключений управления с одного закона на другой. В случае числа сравнений для каждой точки во времени требуется произвести столько операций, сколько имеется границ интервалов управления. Переключение же управления осуществляется не для любой точки во времени, а только для тех, которые совпадают с началами интервалов, поэтому количество переключений совпадет с количеством интервалов управления (таблица 15).

Таблица 15 – Количество операций для осуществления управления с переменным шагом

	Число сравнений	Число переключений	Всего
Управление по тангажу	2106	26	2132
Управление по крену	1458	18	1476
Управление по рысканью	1458	18	1476
ИТОГО	5022	62	5084

При определении длины постоянного шага аппроксимации, обеспечивающим точность $\varepsilon = 0,9$ на основе жадного алгоритма, не удалось

найти длину шага, сколько-нибудь превышающую базовый шаг оцифровки траекторий управления ни по одному из критериев (тангаж, крен, рысканье).

Следовательно, единственно возможным вариантом остается базовая аппроксимация с шагом $h = 0,5$, коэффициенты которой описаны в таблицах 8-13. При этом количество интервалов управления составляет 80 штук по всем критериям. Тогда общее количество параметров, хранимых в памяти бортового компьютера БПЛПА для практического осуществления управления, составит 723 (таблица 16).

Таблица 16 – Количество параметров, хранимых в памяти бортового компьютера БПЛПА для осуществления управления с постоянным шагом

	Координата левой границы интервалов	Длина интервала	Коэффициенты линейной аппроксимации	Всего
Управление по тангажу	80	1	160	241
Управление по крену	80	1	160	241
Управление по рысканью	80	1	160	241
ИТОГО	240	3	480	723

Количество необходимых операций сравнения и переключения отражает таблица 17.

Таблица 17 – Количество операций для осуществления управления с постоянным шагом

	Число сравнений	Число переключений	Всего
Управление по тангажу	6480	80	6560
Управление по крену	6480	80	6560
Управление по рысканью	6480	80	6560
ИТОГО	19440	240	19680

4.1.1.3 Результаты экспериментов

При одинаковой точности разработанный адаптивный метод динамического определения сегментов переменной длины для кусочно-линейной аппроксимации затрачивает в среднем в 3 раза меньше резервов памяти для хранения параметров управления, и требует почти в 4 раза меньше количества операций, чем метод с постоянным шагом для одних и тех же эталонных кривых.

Сравнительную характеристику двух рассмотренных методов демонстрирует график относительного вклада каждого метода по хранимым параметрам и необходимым операциям (рисунок 24).

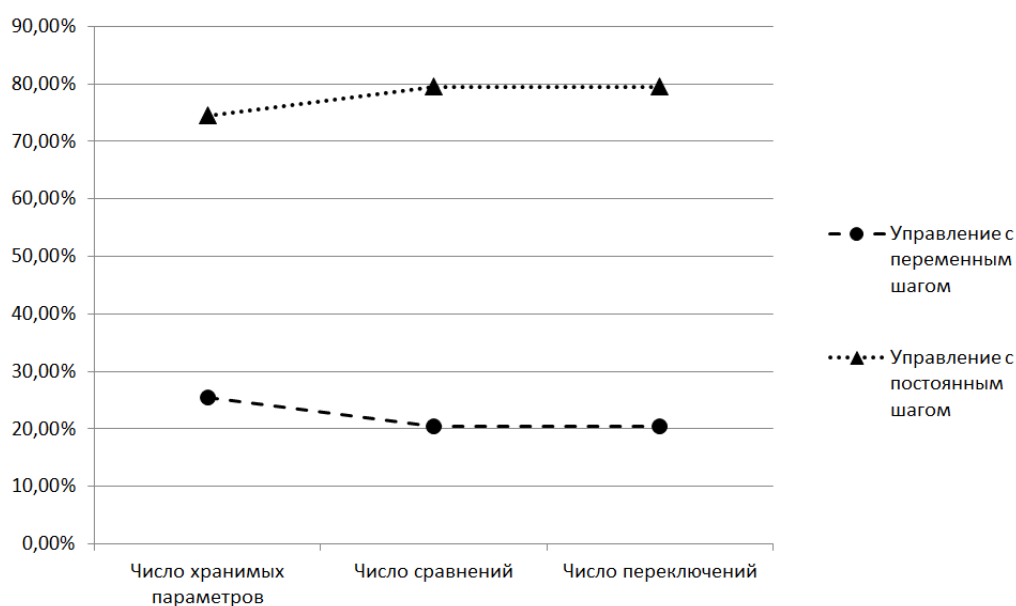


Рисунок 24 – Относительные значения требуемых ресурсов бортового компьютера БПЛА для осуществления управления о тангажу, крену и рысканью для двух подходов к управлению

На основании проведенных вычислительных экспериментов и сравнительного анализа вычислительных затрат, доказана высокая эффективность разработанного адаптивного метода расчета шага переменной длины для кусочно-линейной аппроксимации нелинейной функции сложной

траектории по сравнению с прямыми методами вычисления с постоянным шагом. Метод может применяться для расчета переменного шага при кусочно-линейной аппроксимации нелинейных функций для расчетов на маломощных вычислительных устройствах, в частности, на бортовых компьютерах беспилотных аппаратов различных конструкций для задания целевых траекторий сложной конфигурации. Расчет кусочно-линейно аппроксимации для задания сложных угловых траекторий для беспилотного летательного подводного аппарата показал применимость разработанного алгоритма для целей управления БПЛА.

4.1.2 Кусочно-линейная аппроксимация нелинейной функции, описывающей изменение концентрации экзогенного инсулина в плазме крови

Рассмотрим кусочно-линейная аппроксимация нелинейной функции, отражающей динамику концентрации в плазме крови экзогенного инсулина пациентом с абсолютной инсулиновой недостаточностью (сахарный диабет 1 типа).

Концентрация инсулина, или «активный инсулин» является основополагающим параметром при вычислении потребных доз препарата при интенсивной персонифицированной инсулинотерапии. Персонифицированный алгоритм терапии, в свою очередь, реализуется на носимых малогабаритных устройствах с небольшими вычислительными мощностями. В частности, перспективные разработки «интеллектуальной инсулиновой помпы» со встроенным нейросетевым алгоритмом прогноза гликемии и динамическим расчетом доз, потребовали создания методов расчета активного инсулина на основе нетребовательных к вычислительным ресурсам алгоритмов [99].

Препараты инсулина классифицируются в зависимости от скорости абсорбции и продолжительности действия [100]. Различают следующие группы препаратов инсулина (Таблица 18):

Таблица 18 – Классификация инсулинов по длительности действия

Вид инсулина	Продолжительность действия
Инсулины короткого действия	6-8 часов
Инсулины ультракороткого действия	4-6 часов
Инсулины длительного действия	12-16 часов
Инсулины сверхдлительного действия	20- 42 часа

Для конкретного пациента выбор искусственного инсулина различен, однако общий усредненный вид кривой концентрации постоянен и прилагается к инструкции по конкретному препарату (или группе препаратов) в графическом виде [101]. В данной работе рассматриваются препараты ультракороткого действия, так как именно они используются в инсулиновых помпах при интенсивной инсулинотерапии. Продолжительность действия принята в 6 часов. Кривая активности инсулина представлена ниже.

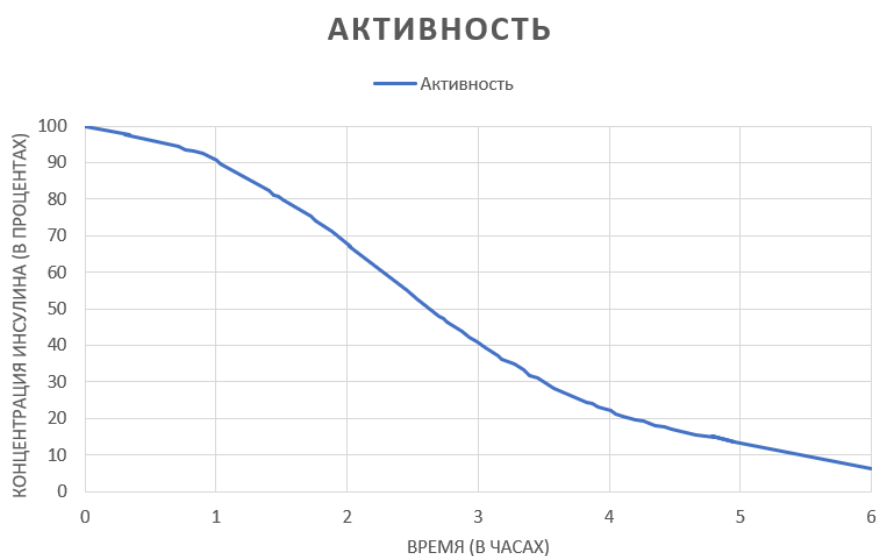


Рисунок 25 – Кривая активности инсулина ультракороткого действия

Для проведения вычислительных экспериментов кривая активности была формализована в виде кусочно-линейной аппроксимации с минимальным постоянным шагом в 6 минут (0,1 часа). Данное представление можно принять за базовую аппроксимацию с шагом $h = 0,1$. Фрагмент табличного представления полученной кривой с постоянным шагом $h = 0,1$ представлено ниже.

Таблица 19 – Табличное представление базовой модели активности инсулина с минимальным постоянным шагом для препарата продолжительностью действия 6 часов

Время	Активность инсулина	k	b	$\hat{y} = kx + b$
0	100	-5,991893182	100	100
0,1	99,40081068	-6,003672386	100,0011779	99,40081068
0,2	98,80044344	-6,40604637	100,0816527	98,80044344
...
5,8	6,293765683	-10,66724731	68,1638001	6,293765683
5,9	5,227040952	-11,35840952	72,24165711	5,227040952
6	4,0912	-40,912	249,5632	4,0912

Здесь k , b – коэффициенты линейной модели, \hat{y} – значения, полученные при подстановке узловых точек x_i в полученную аналитическую модель. Выполненный эксперимент хоть и показывает точные значения полученных данных при подстановке в аппроксимирующую функцию, но требует больших вычислительных затрат. В терминах разработанного алгоритма, базовый подход требует вычислений для 60-ти сегментов каждой пары соседних узловых точек: $S^1(1,2), S^2(2,3) \dots S^{60}(59,60)$.

4.1.2.1 План эксперимента

Для сравнения эффективности предложенного алгоритма с алгоритмом при постоянном шаге, будет построено две модели кусочно-линейной аппроксимации с одной и той же точностью для двух подходов. Точность определяется как значение коэффициента детерминации R^2 линейной модели на каждом выделенном в процессе работы алгоритмов сегменте. Точность всей модели принимается равной минимальному значению R^2 по всем сегментам. В качестве количественных критериев сравнения принято количество потребовавшихся для каждого подхода вычислений.

4.1.2.2 Результаты эксперимента

Эксперимент №1. Для тестирования разработанного метода кусочно-линейной аппроксимации рассмотрена кривая активности инсулина 6-часового действия. Кусочно-линейная аппроксимация строится с допустимой точностью $\varepsilon = 0,996$. В результате вычислений было получено пять сегментов различной длины: $S^1(0,6)$, $S^2(6,10)$, $S^3(10,40)$, $S^4(40,46)$, $S^5(46,60)$. Для каждого сегмента требование по точности построенной модели $R^2 \geq 0,996$ было удовлетворено. Результаты представлены ниже.

Таблица 20 – Результаты вычислительных экспериментов для метода с переменным шагом

m	u	v	A	B	R²
1	0	6	-6,66733	100,07368	0,99622
2	6	10	-12,19437	103,092275	0,99693
3	10	40	-24,69832	116,52314	0,99676
4	40	46	-10,68797	64,64614	0,99645
5	46	60	-7,94256	52,55584	0,99602

Количество произведенных операций для построения линейных моделей на множестве аппроксимирующих сегментов представлено ниже.

Таблица 21 – Представление количества произведенных операций

Операция	Количество операций
Сравнение	144
Сложение	488
Вычитание	3
Умножение	189
Деление	1
ВСЕГО	825

Эксперимент №2. Для построения линейно-кусочной модели с постоянным шагом, обеспечивающую точность $R^2 \geq 0,996$, сначала была определена длина постоянного шага, то есть длина сегментов.

Для поиска подходящей длины сегментов используем «жадный» алгоритм [102]. На каждой итерации будем делить исходную матрицу T на k сегментов равной длины и вычислять коэффициент детерминации R^2 для каждого полученного сегмента. Если для каждого полученного сегмента $R^2 > \varepsilon$, то получаем корректное разделение. В противном случае каждый выделенный ранее сегмент разбивается на два подсегмента, и описанные выше действия повторяются для $k = 2k$.

В результате было выделено 15 сегментов с постоянным шагом $h=0,4$ (с разницей в 4 узловые точки). Далее для каждого сегмента была построена линейная аппроксимация, в результате чего требование по точности построенной модели $R^2 \geq 0,996$ было удовлетворено. Результаты вычислительных экспериментов представлена ниже.

Таблица 22 – Результаты вычислительных экспериментов для метода с постоянным шагом

m	u	v	A	B	R²
1	0	4	-6,12085	100,0084	0,99973
2	4	8	-9,38019	101,2619	0,99626
3	8	12	-15,30108	105,73332	0,99787
4	12	16	-21,29137	112,59827	0,99919
5	16	20	-25,7244	119,41431	0,99979
6	20	24	-27,78595	123,36721	0,99998
7	24	28	-27,32076	122,21019	0,99995
8	28	32	-24,67906	114,89626	0,99974
9	32	36	-20,56334	101,90426	0,99938
10	36	40	-15,87499	85,25817	0,99892
11	40	44	-11,56099	68,23961	0,99862
12	44	48	-8,46034	54,79292	0,99909
13	48	52	-7,15104	48,62531	0,99997
14	52	56	-7,79639	51,998	0,9995
15	56	60	-9,99192	64,21301	0,9991

Вычислительные затраты при использовании «жадного» алгоритма будут значительно масштабнее, чем в разработанном методе с адаптивным шагом. Количество произведенных операций для построения линейных моделей на множестве сегментов постоянной длины представлено ниже.

Таблица 23 – Представление количества произведенных операций для аппроксимации с постоянным шагом

Операция	Количество операций
Сравнение	56
Сложение	5866
Вычитание	314
Умножение	2677
Деление	403
ВСЕГО	9 316

При одинаковой точности разработанный адаптивный метод динамического определения сегментов переменной длины для кусочно-линейной аппроксимации затрачивает в 11 раз меньше операций, чем метод с

постоянным шагом для одной и той же эталонной кривой. Эффективность метода подтверждена.

На основании проведенных вычислительных экспериментов и сравнительного анализа вычислительных затрат, доказана высокая эффективность разработанного адаптивного метода расчета шага переменной длины для кусочно-линейной аппроксимации эмпирической нелинейной функции по сравнению с прямыми методами вычисления с постоянным шагом. Соответственно, метод может применяться для расчета переменного шага при кусочно-линейной аппроксимации нелинейных функций.

4.1.3 Кусочно-линейная аппроксимация траектории движения автомобиля для цифрового имитационного моделирования транспортных процессов

В задачах цифрового имитационного моделирования транспортных систем важную роль играет математическое описание траекторий движения транспортных средств. Траектория движения автомобиля представляет собой нелинейную зависимость координаты от времени, формируемую под воздействием различных факторов, включая режимы разгона и торможения, дорожные условия и алгоритмы управления. На практике такие зависимости, как правило, задаются табличными или экспериментальными данными, полученными в результате измерений или численного моделирования. Для их дальнейшего использования в вычислительных алгоритмах требуется компактное математическое представление, что обуславливает актуальность применения разработанного метода адаптивной кусочно-линейной аппроксимации.

4.1.3.1 План эксперимента

Рассмотрим экспериментальную траекторию движения автомобиля, заданную в виде зависимости координаты от времени.

Траектория задана таблично на интервале времени в виде набора экспериментальных точек.

Необходимо представить исходную траекторию движения автомобиля в виде кусочно-линейной модели вида на последовательности непересекающихся интервалов времени таким образом, чтобы аппроксимирующая функция сохраняла заданную точность, и при этом число сегментов было минимальным. Это обеспечит минимизацию вычислительных затрат при последующем использовании модели в имитационном моделировании.

4.1.3.2 Вычислительный эксперимент

Для проверки эффективности предложенного метода рассмотрим задачу аппроксимации траектории движения автомобиля, полученной в результате моделирования типичного режима движения транспортного средства. В ходе эксперимента траектория движения автомобиля была сформирована в результате численного моделирования движения транспортного средства на интервале времени 6 секунд. Движение автомобиля состоит из следующих фаз [117]:

- **Фаза 1:** разгон (0–2 с);
- **Фаза 2:** равномерное движение (2–4 с);
- **Фаза 3:** торможение (4–6 с).

Для проведения эксперимента функция $s = f(t)$ была представлена таблично с шагом $\Delta t = 0.1$ с с на интервале $t \in [0,6]$, что соответствует 60 узловым точкам траектории. Фрагмент табличного представления траектории представлен ниже.

Таблица 24 – Кусочно-линейная аппроксимация базовой модели траектории движения автомобиля (фрагмент)

Время	Координата	Коэф-ты линейных участков		$\hat{s} = kt + b$
		k	b	
0	100	-5,991893182	100	100
0,1	99,40081068	-6,003672386	100,0011779	99,40081068
0,2	98,80044344	-6,40604637	100,0816527	98,80044344
...
5,8	6,293765683	-10,66724731	68,1638001	6,293765683
5,9	5,227040952	-11,35840952	72,24165711	5,227040952
6	4,0912	-40,912	249,5632	4,0912

Данное представление можно рассматривать как базовую модель траектории движения автомобиля с минимальным постоянным шагом, которая используется в качестве исходных данных для алгоритма аппроксимации.

4.1.3.3 Результаты эксперимента

Для аппроксимации полученной траектории применён разработанный алгоритм динамического определения сегментов переменной длины (описанный в главе 2.1). В качестве критерия точности аппроксимации использовалось значение коэффициента детерминации. В результате выполнения алгоритма исходная траектория движения автомобиля была представлена пятью сегментами различной длины. Каждый сегмент описывается линейной моделью вида, которая аппроксимирует соответствующий участок траектории с требуемой точностью. Полученное разбиение представлено ниже.

Таблица 25 – Участки линейной аппроксимации траектории движения автомобиля при переменном шаге

Сегмент	Границы		Коэф-ты лин.уч.		R^2	Участок траектории
	u	v	k	b		
S1	0	6	-7,007	112,554	0,99452	начальный участок разгона
S2	6	10	-13,897	123,272	0,99125	завершающая фаза разгона
S3	10	40	-23,118	126,117	0,99101	равномерное движение
S4	40	46	-14,123	71,522	0,99847	начало торможения
S5	46	60	-8,685	51,447	0,99912	завершающая фаза торможения

Наибольшая длина сегмента (от 10 до 40) соответствует участку равномерного движения, где траектория практически линейна. Графическое отображение соотношения реальной траектории и её кусочно-линейной аппроксимации, полученной с применением алгоритма переменного шага, представлено на рисунке ниже.

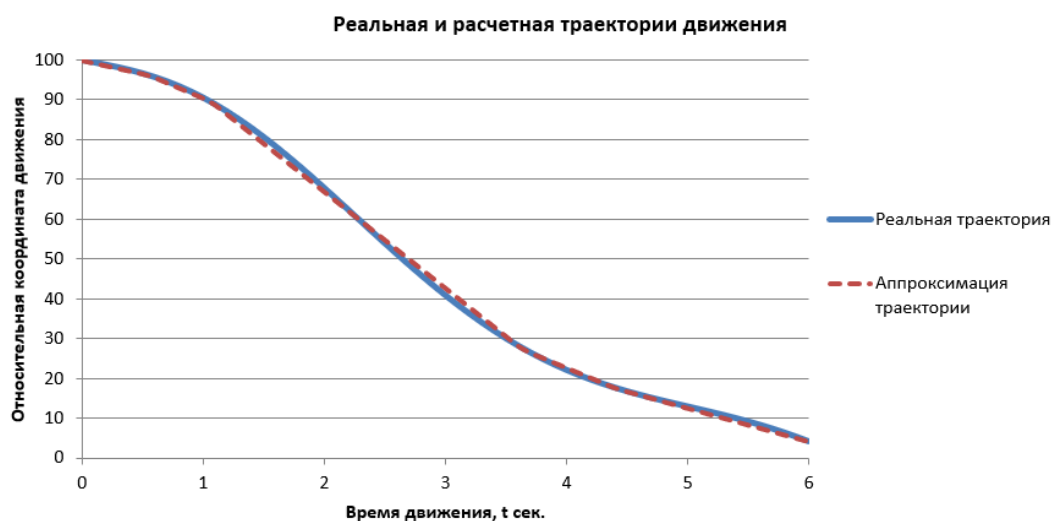


Рисунок 26 – Реальная и аппроксимированная траектории движения автомобиля

Общее количество вычислительных операций, необходимых для построения линейных моделей на всех сегментах, составило 915 операций.

Для сравнения эффективности предложенного метода была построена кусочно-линейная модель траектории с использованием постоянного шага аппроксимации. С помощью жадного алгоритма было установлено, что для обеспечения той же точности необходимо разбить исходную траекторию на 15 сегментов одинаковой длины. Для каждого сегмента была построена линейная аппроксимация методом наименьших квадратов. При этом суммарное количество вычислительных операций составило 8924 операций.

Сравнительные результаты двух экспериментов приведены ниже.

Таблица 26 – Сравнение вычислительных затрат методов кусочно-линейной аппроксимации

Метод	Количество сегментов	Количество операций
Переменный шаг	5	915
Постоянный шаг	15	8924

Полученные результаты показывают, что использование адаптивного шага позволяет существенно уменьшить количество сегментов аппроксимации и снизить вычислительные затраты при последующем использовании модели в задачах имитационного моделирования.

4.2 Вычислительные эксперименты по применению метода адаптивной кластеризации и гибридной модели прогнозирования временных рядов

Для проверки работоспособности и оценки эффективности разработанных в Главе 2 методов адаптивной кластеризации и гибридного LSTM-прогнозирования были проведены вычислительные эксперименты на двух прикладных задачах, различающихся по физической природе, размерности и характеру неоднородности временных рядов.

Первая задача – формирование нейроконтроллера для обеспечения устойчивости беспилотного летательного аппарата с жестким крылом при

выполнении разворота в условиях атмосферной турбулентности. Входными данными служат параметры полета (высота, скорость, масштаб турбулентности) и весовые коэффициенты, отражающие приоритет частных критериев устойчивости. Выходными – оптимальные настройки автопилота (семь управляющих коэффициентов). Неоднородность данных здесь обусловлена резкими изменениями высоты и интенсивности турбулентности, а также сменой приоритетов критериев устойчивости.

Вторая задача – краткосрочное прогнозирование уровня глюкозы в крови пациентов с сахарным диабетом 1 типа на основе многомерного временного ряда, включающего значения гликемии, активного инсулина и активных углеводов. Данные получены с сенсоров инсулиновых помп и характеризуются высокой вариабельностью, наличием шумов и пропусков, а также различными паттернами динамики в зависимости от физиологического состояния пациента (нормогликемия, гипогликемия, гипергликемия).

В ходе экспериментов решались следующие задачи:

- 1) формирование репрезентативных обучающих выборок на основе глобальной оптимизации (для задачи управления БПЛА) и обработки реальных лог-файлов (для задачи прогнозирования гликемии);
- 2) применение разработанного метода кластеризации для выделения однородных групп временных фрагментов;
- 3) обучение специализированных LSTM-моделей для каждого кластера;
- 4) сравнение точности прогнозирования предложенного подхода с альтернативными методами (LSTM без кластеризации, LSTM с кластеризацией на основе евклидова расстояния, LSTM с кластеризацией на основе параметров ARIMA).

4.2.1 Применение для формирования нейроконтроллера, обеспечивающего устойчивость БПЛА в условиях атмосферной турбулентности

Выполнение сложных маневров беспилотными летательными аппаратами, например разворотов, в условиях атмосферной турбулентности остается одной из наиболее сложных задач управления полетом. Порывы ветра и неоднородные воздушные потоки вызывают изменения углов крена, рысканья и атаки, что может привести к отклонению аппарата от заданной траектории и снижению его устойчивости.

Результаты исследований показывают, что существующие системы управления не всегда способны полностью компенсировать влияние таких возмущений. В частности, гибридные методы, которые сочетают в себе активную компенсацию возмущений и управление на основе скользящего режима, позволяют повысить устойчивость летательного аппарата, однако при воздействии случайных факторов, таких как атмосферная турбулентность, их эффективность заметно снижается [103-105]. Более того, турбулентные условия приводят к резкому ухудшению характеристик стабильности, требуя комплексных алгоритмов управления для безопасного выполнения заданных траекторий [106].

4.2.1.1 Стохастическая модель движения БПЛА при развороте в условиях меняющейся высоты, скорости и случайной ветровой турбулентности.

Рассматривается движение БПЛА с жестким крылом при выполнении сложного маневра – разворота – в горизонтальной плоскости в условиях меняющейся высоты, скорости, а также в условиях сильной турбулентности. Динамика полета БПЛА при этом описывается стохастической системой дифференциальных уравнений вида [39-44]:

$$\dot{\beta} = -k_{\beta}(\beta + \beta_W) + \alpha_0 \omega_x + \omega_y + k_{\gamma} \gamma \quad (39)$$

$$\dot{\omega}_x = -l_{\beta}(\beta + \beta_W) - l_{\gamma} \omega_x - l_{\psi} \omega_y - l_e \delta_e \quad (40)$$

$$\dot{\omega}_y = -n_{\beta}(\beta + \beta_W) - n_{\gamma} \omega_x - n_{\psi} \omega_y - n_c \delta_c \quad (41)$$

$$\dot{\gamma} = \omega_x \quad (42)$$

$$\dot{\psi} = \omega_y \quad (43)$$

$$\dot{\rho} = \omega_y - \omega_{y^*} \quad (44)$$

Здесь: β - угол скольжения, град.; γ - угол крена, град.; ψ - угол рысканья, град.; ω_x - угловая скорость по крену, град/с; ω_y - угловая скорость по рысканью, град/с; α_0 - балансировочный угол атаки, град.; δ_e - управляющий сигнал по каналу элеронов; δ_c - управляющий сигнал по каналу руля направления, β_W - ветровая составляющая угла скольжения, $k_{\beta}, k_{\gamma}, l_{\beta}, l_{\gamma}, l_{\psi}, l_e, n_{\beta}, n_{\gamma}, n_{\psi}, n_c$ - аэродинамические коэффициенты БПЛА, ω_{y^*} - заданная целевая угловая скорость разворота, ρ - отклонение угла рыскания от расчетного значения, определяемого заданной угловой скоростью разворота.

Влияние высоты полета H и скорости БПЛА V_0 в модели (39)-(44) неявно выражается в изменении аэродинамических коэффициента k_{β} и k_{γ} согласно формулам:

$$k_{\beta} = \frac{C_z^{\beta}}{\tau}; \quad (45)$$

$$k_{\gamma} = \frac{g}{V_0}. \quad (46)$$

Здесь C_z^{β} - коэффициент боковой силы, $\tau = \frac{2m}{S\rho_0 V_0}$, m - масса самолета; S - площадь крыла самолета; ρ_0 - плотность воздуха на высоте полета H ; V_0 - истинная воздушная скорость БПЛА.

Влияние турбулентности отражается в наличии в системе (39)-(44) стохастической составляющей β_W - ветровой составляющей угла скольжения, определяемой масштабом турбулентности L_{W_x} согласно следующей цепочке соотношений:

$$\beta_W = \frac{180 \times W_x}{\pi \times V_0} \quad (47)$$

где V_0 – воздушная скорость БПЛА; W_x – горизонтальная составляющая турбулентности, которая моделируется случайным гауссовым процессом с нормальной спектральной плотностью распределения:

$$S_{W_x}(\omega) = \frac{2\sigma_{W_x}^2 L_{W_x}}{\pi(1+(\omega L_{W_x})^2)} \quad (48)$$

Здесь $\sigma_{W_x}^2$ – среднеквадратическое отклонение пульсации скорости ветра, L_{W_x} – масштаб турбулентности, ω – частота пульсации. Применение формирующего линейного фильтра приводит к уравнению

$$\dot{\beta}_W = -\frac{V_0}{L_{W_x}} \beta_W + \sigma_{\beta_W} \sqrt{\frac{2V_0}{L_{W_x}}} N(t); \quad (49)$$

где $N(t)$ – белый шум [107].

Задача адаптивного управления состоит в изменении в реальном времени величины управляющих сигналов автопилота по каналу элеронов δ_e и руля направления δ_c , которые формируются в виде линейных обратных связей от текущего состояния БПЛА:

$$\delta_e = i_\beta^e \beta + i_x^e \omega_x - i_y^e (\omega_y - \omega_{y^*}) - q_e \int_0^{t_T} (\omega_y - \omega_{y^*}) dt \quad (50)$$

$$\delta_c = i_\beta^c \beta + i_x^c \omega_x - i_y^c \omega_y \quad (51)$$

здесь $i_\beta^e, i_x^e, i_y^e, q_e, i_\beta^c, i_x^c, i_y^c$ – передаточные числа автопилота. Адаптация управления под конкретные условия полета происходит за счет изменения в реальном времени передаточных чисел автопилота.

Основное требование к процессу движения БПЛА, которое достигается за счет использования управляющего воздействия автопилота, состоит в том, что при выполнении маневра в конечный момент времени БПЛА должен достичь угловой скорости ω_{y^*} с заданной точностью ε . введя дополнительную переменную:

$$f = \int_0^{t_T} (\omega_y - \omega_{y^*})^2 dt$$

для системы (39) - (44) это требование принимает вид неравенства

$$f \leq \varepsilon \quad (52)$$

4.2.1.2 Детерминированная модель движения БПЛА относительно семиинвариантов

Заметим, что система (39)-(44), (52) описывает случайный диффузионный Марковский процесс с плотностью вероятности в форме уравнения Фоккера-Планка [108]:

$$\begin{aligned} \frac{\partial p(t, x)}{\partial t} = & - \sum_{i=1}^8 \frac{\partial}{\partial x_i} [A_i(t, x, a)p(t, x)] \\ & + \frac{1}{2} \sum_{i,p=1}^8 \frac{\partial^2}{\partial x_i \partial x_p} [B_{ip}(t, x, a)p(t, x)] \end{aligned} \quad (53)$$

относительно коэффициентов сноса $A_i(t, x, a)$ и диффузии $B_{ip}(t, x, a)$. Тогда для сведения задачи (39)-(44) при условиях (52) и выражениях (45)-(49) к детерминированной форме с искомыми параметрами $i_\beta^e, i_x^e, i_y^e, q_e, i_\beta^c, i_x^c, i_y^c$, удовлетворяющими (53), воспользуемся разложением (53) в функциональный ряд по семиинвариантам процесса (39)-(44), (52) [109]. При этом будем учитывать, что, так как система (39)-(44), (52) линейна, то она описывает нормальный случайный процесс. Так как для нормального распределения семиинварианты выше второго порядка равны нулю, то разложение будет включать в себя только математическое ожидание, совпадающее с семиинвариантами первого порядка, а также дисперсии и корреляционные моменты, совпадающие с семиинвариантами второго порядка.

Введем следующие обозначения:

$$\left. \begin{aligned} a_0 &= l_\beta + l_e i_\beta^e; a_1 = l_\gamma + l_e i_x^e; a_2 = l_\psi - l_e i_y^e; a_3 = l_e q_e \\ a_4 &= n_\beta + n_c i_\beta^c; a_5 = n_\gamma + n_c i_x^c; a_6 = n_\psi + n_c i_y^c \end{aligned} \right\} \quad (54)$$

Новые переменные $a_0, a_1, a_2, a_3, a_4, a_5, a_6$ позволяют однозначно восстановить передаточные числа. В дальнейшем более удобно рассматривать именно их совокупность $a = (a_0, a_1, a_2, a_3, a_4, a_5, a_6)$ как приведенные управляющие параметры системы.

Введем обозначения для семиинвариантов первого порядка (математических ожиданий) текущих значений переменных системы в каждый момент времени

$$\begin{aligned} Y_1 &= M(\beta); Y_2 = M(\omega_x); Y_3 = M(\omega_y); Y_4 = M(\gamma); Y_5 = M(\psi); Y_6 = M(\rho); \\ Y_7 &= M(\beta_W); Y_8 = M(f). \end{aligned}$$

и для семиинвариантов второго порядка:

$$\begin{aligned} Y_9 &= D(\beta); Y_{10} = D(\omega_x); Y_{11} = D(\omega_y); Y_{12} = D(\gamma); Y_{13} = D(\psi); Y_{14} = D(\rho); \\ Y_{15} &= D(\beta_W); Y_{16} = \int_0^{tT} D(\omega_y - \omega_{y^*})^2 dt; \quad Y_{17} = K(\beta\omega_x) = D_{12}; \quad \dots Y_{44} = \\ K(\beta_W f) &= D_{78}. \end{aligned}$$

Отметим, что уравнения относительно математических ожиданий образуют независимую подсистему, где переменная Y_5 не входит в правые части уравнений для остальных переменных. Тогда от стохастической системы (39)-(44), (52) можно перейти к системе вида:

$$\dot{Y}_1 = -k_\beta Y_1 + a_0 Y_2 + Y_3 + k_\gamma Y_4 \quad (55)$$

$$\dot{Y}_2 = -a_0 Y_1 - a_1 Y_2 - a_2 (Y_3 - \omega_{y^*}) + a_3 Y_6 - l_\psi \omega_{y^*} \quad (56)$$

$$\dot{Y}_3 = -a_4 Y_1 - a_5 Y_2 - a_6 Y_3 \quad (57)$$

$$\dot{Y}_4 = Y_2 \quad (58)$$

$$\dot{Y}_6 = Y_3 - \omega_{y^*} \quad (59)$$

$$\dot{Y}_7 = -\frac{V_0}{L_{W_x}} Y_7 \quad (60)$$

$$Y_8 \leq \varepsilon \quad (61)$$

В силу того, что вход БПЛА в разворот осуществляется из горизонтального прямолинейного полета, система рассматривается при начальных условиях:

$$Y_1(0) = Y_2(0) = Y_3(0) = Y_4(0) = Y_6(0) = 0; Y_7(0) = \frac{V_0}{LW_x}. \quad (62)$$

4.2.1.3 Формулирование многокритериальной задачи адаптивного управления БПЛА в целях обеспечения устойчивости разворота

Стационарные значения фазовых координат системы в устойчивом режиме движения задаются следующей системой:

$$Y_1^* = \frac{-a_6\omega_{y^*}}{a_4} \quad (63)$$

$$Y_2^* = 0; \quad (64)$$

$$Y_3^* = \omega_{y^*}; \quad (65)$$

$$Y_4^* = \frac{-k_\beta a_6 \omega_{y^*}}{a_4 k_\gamma} - \frac{\omega_{y^*}}{k_\gamma} \quad (66)$$

$$Y_6^* = \frac{a_0 a_6 \omega_{y^*}}{a_4 a_3} + \frac{l_\psi \omega_{y^*}}{a_3} \quad (67)$$

$$Y_7^* = 0 \quad (68)$$

Одним из основных требований при выполнении сложных маневров летательного аппарата является его выход на устойчивый режим полета. Это необходимо для предотвращения сваливания или перехода в штопор после завершения маневра. С математической точки зрения данное требование означает, что значения фазовых координат системы к моменту окончания управления должны быть максимально близки к заданным устойчивым значениям, определяемым выражениями (63)-(68).

В общем виде это приводит к минимизации шести отклонений $|Y_i(t_T) - Y_i^*|$, $i=1,2,3,4,6,7$. Это задача многокритериальной оптимизации: каждый критерий показывает, насколько текущая координата отличается от нужной, и тем самым отражает качество управления и устойчивость аппарата. Но на практике ключевыми оказываются всего два критерия – устойчивость по

углу скольжения β и устойчивость по отклонению ρ угла рысканья от расчетного значения. Эксперименты показали, что при устойчивости управления по этим двум критериям устойчивость переходного процесса управления по остальным параметрам, таким как угол крена γ и угловая скорость крена w_x , обеспечивается автоматически. Поэтому целесообразно перейти к задаче лишь с двумя частными критериями:

$$F_1(a) = |Y_1(t_T) - Y_1^*| \Rightarrow \min \quad (69)$$

$$F_2(a) = |Y_6(t_T) - Y_6^*| \Rightarrow \min \quad (70)$$

Для решения используем метод взвешивания критериев, где обобщенный критерий качества управления строится в виде линейной свертки со взвешенными частными критериями:

$$F(a) = \xi_1 F_1(a) + \xi_2 F_2(a) = \xi_1 |Y_1(t_T) - Y_1^*| + \xi_2 |Y_6(t_T) - Y_6^*| \quad (71)$$

Здесь ξ_1, ξ_2 – заданные весовые коэффициенты, отражающие значимость обеспечения устойчивости по отдельным фазовым координатам для достижения общей устойчивости системы. Обобщенный критерий (71) можно рассматривать как взвешенное манхэттенское расстояние пары конечных значений фазовых координат от пары их устойчивых значений в двумерном пространстве.

Весовые критерии сверточного критерия оптимизации могут зависеть от частных условий полета и выполняемого БПЛА полетного задания. В этом случае выбор значений весовых критериев ложится на лицо, принимающее решение.

В результате математическая формулировка задачи адаптивного управления сложными маневрами БПЛА в условиях меняющихся внешних условий и режимов полета выражается соотношениями (55)-(61) при начальных условиях (62), где цель управления задана в виде критерия (71).

4.2.1.4 План эксперимента

В условиях меняющихся характеристик полета, таких как высота и скорость, а также при изменяющемся масштабе турбулентности, для адаптивного обеспечения наилучшего режима полета, выполнения возложенных на БПЛА задач и обеспечения устойчивости необходимо в реальном времени решать оптимизационную задачу (71), (55)-(62) по данным текущих наблюдений за состоянием среды, а также при выбранных для текущей задачи весовых коэффициентах свертки ξ_1, ξ_2 . Однако решать каждый раз обновляющуюся задачу оптимизации в реальном времени специализированными численными методами при скоростях передвижения современных самолетоподобных БПЛА на уровне 100-200 км/час не представляется возможным. Для этих целей предлагается применять предварительно обученный бортовой нейроконтроллер, формирующий квазиоптимальные настройки автопилота в реальном времени как результат обучения на представительной обучающей выборке, содержащей точные решения оптимизационной задачи для достаточно широкого набора возможных внешних условий и весовых коэффициентов.

Соответственно, для осуществления высокоточной адаптивной настройки автопилота в реальном времени необходимо разработать методы для решения двух последовательных задач:

1. Подготовка релевантной обучающей выборки, для чего необходимо решить серию оптимизационных задач (71), (55)-(62) при разных весах ξ_1, ξ_2 локальных целевых параметров БПЛА (69)-(70) и различных внешних условиях полета, таких как высота полета H , воздушная скорость полета V_0 и интенсивность турбулентности L_{W_x} .

2. Конструирование и обучение на основе подготовленной выборки нейроконтроллера, формирующего оптимальные значения настроек автопилота при произвольных целевых параметрах БПЛА, наблюдаемых

параметрах окружающей среды и значениях весовых коэффициентов свертки (80).

Для решения серии оптимизационных задач, необходимых для построения обучающей выборки, был использован численный метод глобальной оптимизации SoFAM. Он относится к классу эволюционных алгоритмов. Подобно другим эволюционным алгоритмам он использует операторы мутации и отбора. Основные идеи метода были сформулированы в [110] и затем получили дальнейшее развитие в [111], что позволило создать высокопроизводительный алгоритм оптимизации. Основное преимущество данного метода по сравнению с другими метаэвристическими алгоритмами состоит в том, что для него обоснована сходимость по вероятности генерируемой последовательности тестовых точек в пространстве поиска к глобальному оптимуму. Другие методы не обеспечивают такую сходимость, что может быть причиной получения ложного локального решения. Производительность метода SoFAM проверялась на классах релевантных тестовых функций. Сравнительный анализ показал, что производительность этого метода не уступает наилучшим известным алгоритмам, а в ряде случаев превосходит их. Метод хорошо зарекомендовал себя в решении прикладных задач, в том числе в выборе наилучших настроек для БПЛА [112].

Однако непосредственное применение метода для решения большого числа оптимизационных задач сталкивается с некоторыми трудностями. Для того, чтобы получить гарантированно наилучшее решение, необходимо сделать полноценный обзор области поиска, что требует проведения большого числа испытаний и соответствующего расчета значений целевой функции. Как показывает опыт, это число исчисляется тысячами испытаний. При значительном количестве различных наборов внешних параметров системы и весовых коэффициентов целевой сверточной функции общее время счета может достигать сотен часов, что представляется крайне обременительным.

Для рационализации применяемого подхода была предложена улучшенная схема расчета. Заметим, что если внешние параметры системы

фиксированы, то целевой сверточный функционал будет зависеть от управляющих параметров a и весовых коэффициентов $\xi = (\xi_1, \xi_2)$:

$$F(\xi, a) = \sum_{i=1}^2 \xi_i F_i(a) \quad (72)$$

При вычислении значения такого функционала основное время тратится на расчет значений частных критериев $F_i(a)$, поскольку для этого требуется проинтегрировать систему (55)-(61) из шести дифференциальных уравнений. Но если эти частные критерии рассчитаны, то легко можно найти значения целевого функционала при всех заданных наборах коэффициентов свертки.

С учетом этого обстоятельства принята следующая схема построения обучающей выборки. При реализации алгоритма поиска оптимального решения (71), (55)-(62), соответствующего конкретному набору значений весовых коэффициентов (ξ_1, ξ_2) :, в каждой точке испытания $a = (a_0, a_1, a_2, a_3, a_4, a_5, a_6)$ рассчитывается не одно значение целевой функции $F(\xi, a)$ для данных весовых коэффициентов, но также значения этой функции для всех других весовых коэффициентов вида (72) при заданных интересующих парах значений (ξ_1, ξ_2) . В результате осуществляется обзор (exploration) области поиска и получение информации о поведении целевой функции не только для одного набора весовых коэффициентов, но для всех нужных наборов весовых коэффициентов сразу.

Затем для каждого набора пар значений весовых коэффициентов (ξ_1, ξ_2) из полученных значений целевых функций (72) выбираются наилучшие. Эти значения используются как хорошие начальные приближения для дальнейшего уточнения полученного результата путем непосредственного интегрирования системы дифференциальных уравнений (55)-(62). Однако в отличие от первоначально решаемой задачи, при проведении уточнения из имеющейся точки хорошего начального приближения уже нет необходимости в подробном обзоре области поиска для каждого набора весовых коэффициентов, поэтому уточнение можно осуществлять с помощью более простого и быстрого в

численной реализации локального метода, например, градиентного спуска. В результате существенно экономится время составления образцов для обучающей выборки, используемой для дальнейшего обучения нейроконтроллера.

Для выбора архитектуры нейросетевого управляющего контроллера сформированный на предыдущем этапе набор данных рассматривался как совокупность временных рядов, где каждый отдельный временной ряд представляет собой фрагмент траектории с плавным изменением высоты, скорости полета и интенсивности турбулентности.

В качестве алгоритма кластеризации временных рядов использован описанный в Главе 2.2 метод адаптивной кластеризации временных рядов. В качестве модели расчета оптимальных параметров управления – представленная в Главе 2.3 гибридная нейросетевая модель прогнозирования временных рядов.

Для выполнения численного исследования эффективности предложенного метода мы моделировали полет среднеразмержного БПЛА с жестким крылом весом 320 кг, размахом крыла 8 м с площадью поверхности крыла 10 м².

4.2.1.5 Реализация эксперимента

На этапе формирования датасета коэффициентов оптимального управления на базе рационализированного метода SoFAM для последующей тренировки нейросетевого контроллера было рассчитано оптимальное управление для автоматического разворота БПЛА, движущегося со скоростью 60-62 м/с при сильном встречном ветре 49-51 м/с для высот от 800 до 1500 м и меняющейся турбулентности по шкале «малая», «средняя», «сильная» European Union Aviation Safety Agency (EASA) [113] от средней - 150 м до сильной - 470 м. Весовые коэффициенты сверточного критерия, обеспечивающего устойчивость управления при развороте, последовательно

варьировались для каждого варианта условий полета от 0.2 до 0.8 с соблюдением условия нормировки.

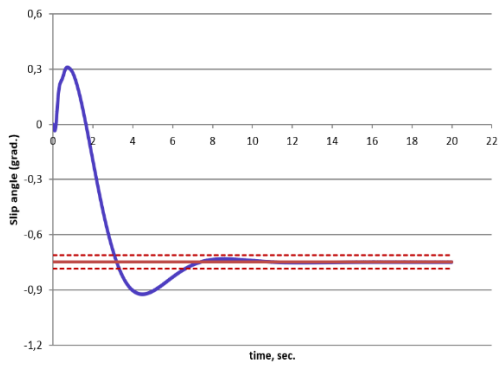
Всего таким образом было сформировано 205 задач оптимального управления вида (71), (55)-(62). Каждая из задач решалась предложенным рационализированным методом глобальной оптимизации SoFAM. Вычисления проводились на персональном компьютере с процессором Intel Core i5, с объемом оперативной памяти 16 ГБ. Общее время расчетов составило 25,5 часов.

Фрагмент полученного в результате датасета с указанием промежуточных характеристик, представлен в таблице 27.

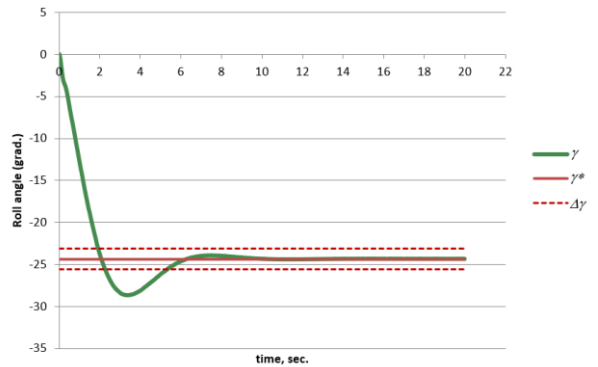
Для всех рассмотренных вариантов полета в результате применения рационализированного алгоритма SoFAM к решению задач вида (80), (64)-(71) было получено управление, обеспечивающее разворот БПЛА с гарантированным выходом на устойчивый стационарный режим за время от 10 до 15 секунд с максимальным перерегулированием не более 30%. Примеры переходных процессов при рассчитанном управлении для трех вариантов условий полета и весовых коэффициентов критериев устойчивости приведены на рисунке 27-29.

Таблица 27 – Набор данных рассчитанных оптимальных управляющих коэффициентов автопилота для различных условий полета и различного сочетания весов важности критериев устойчивости управления (фрагмент)

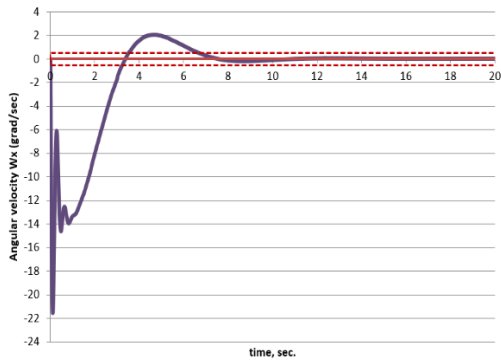
Parameters for calculating inputs						Input parameters									Output parameters						Additional parameters				
UAV weight	Wing area	Flight altitude	Air density at flight altitude	Side force coefficient	Intermediate coefficient	Bank angle coefficient	Slip angle coefficient	Trim angle of attack	Yaw angle coefficient	True airspeed	Turbulence scale	Specified rate of turn	Weight coefficient at F1	Weight coefficient at F6	Parameter for recalculating the control coefficient ϵ_0^y	Parameter for recalculating the control coefficient ϵ_1^y	Parameter for recalculating the control coefficient ϵ_2^y	Parameter for recalculating the control coefficient ϵ_3^y	Parameter for recalculating the control coefficient ϵ_4^y	Parameter for recalculating the control coefficient ϵ_5^y	Parameter for recalculating the control coefficient ϵ_6^y	Criterion of stability by the sliding angle β	Criterion of stability by the deviation ρ of the yaw angle from the calculated value	General convolutional stability criterion	
m	S	H	ρ_0	C_z^β	τ	k_γ	k_β	α_0	L_ψ	V_0	L_{Wx}	ω_y^*	ϵ_1	ϵ_6	a_0	a_1	a_2	a_3	a_4	a_5	a_6	F_1	F_6	F	
320	10	800	1,13366	0,101	0,508597442	0,088288288	0,198585348	0,0680678	1,77647457	111	150	2	0,2	0,8	0,030088755	4	-190,89	149,1849553	20,99999729	1,730417022	7,789936612	9,92257E-08	2,314E-06	1,87105E-06	
320	10	800	1,13366	0,101	0,508597	0,088288	0,198585	0,068068	1,776475	111	170	2	0,8	0,2	0	4,038644	-198,603	107,2536	21	1,542961	7,999991	1,46E-06	3,6E-06	1,89E-06	
320	10	800	1,13366	0,101	0,508597	0,088288	0,198585	0,068068	1,776475	111	210	2	0,4	0,6	0	4	-200	123,4486	21	1,460127	7,55177	6,08E-07	8,35E-08	2,93E-07	
320	10	800	1,13366	0,101	0,508597	0,088288	0,198585	0,068068	1,776475	111	270	2	0,6	0,4	1,71E-06	4	-199,287	176,54	17,76902	1,319231	7,703022	2,06E-08	1,2E-06	4,93E-07	
...
320	10	800	1,13366	0,101	0,508597	0,088288	0,198585	0,068068	1,776475	111	370	2	0,4	0,6	5,27E-07	4	-200	176,5399	9,320154	0,774686	8	3,95E-05	1,02E-05	2,19E-05	
320	10	800	1,13366	0,101	0,508597	0,088288	0,198585	0,068068	1,776475	111	390	2	0,6	0,4	0,004852	4	-200	176,54	8,743626	0,756269	8	0,000263	7,71E-05	0,000189	
320	10	800	1,13366	0,101	0,508597	0,088288	0,198585	0,068068	1,776475	111	450	2	0,4	0,6	0,020229	4,000001	-200	176,019	8,259621	0,7322	8	0,002052	0,00012	0,000893	
320	10	800	1,13366	0,101	0,508597	0,088288	0,198585	0,068068	1,776475	111	470	2	0,2	0,8	1,107643	6,915583	-198,242	170,7837	7	3	4,043928	0,00015	0,000108	0,000116	
...
320	10	1000	1,11166	0,101	0,5186627	0,088288288	0,194731567	0,0680678	1,742	111	150	2	0,2	0,8	0,011031987	4	-200	114,872517	18,60575855	1,482644077	7,887065554	7,22909E-07	3,74536E-06	3,14087E-06	
320	10	1000	1,11166	0,101	0,5186627	0,088288288	0,194731567	0,0680678	1,742	111	170	2	0,4	0,6	0	4	-200	108,8152508	20,78013286	1,522851154	7,192172713	6,67879E-07	6,88136E-07	6,80033E-07	
320	10	1000	1,11166	0,101	0,5186627	0,088288288	0,194731567	0,0680678	1,742	111	190	2	0,6	0,4	0	4,162049479	-199,999644	114,4792041	21	1,477884073	7,566251373	1,25371E-06	7,67583E-07	1,05926E-06	
...
320	10	1000	1,11166	0,101	0,5186627	0,088288288	0,194731567	0,0680678	1,742	111	290	2	0,2	0,8	0,020987382	4,42312039	-155,0715561	172,5863428	17,01635048	1,349963996	7,999993383	1,60235E-06	9,39292E-07	1,0719E-06	
320	10	1000	1,11166	0,101	0,5186627	0,088288288	0,194731567	0,0680678	1,742	111	330	2	0,4	0,6	0	4,038780039	-200	169,5786221	10,740384	0,853659744	7,732122514	6,33311E-06	3,75804E-06	4,78807E-06	
320	10	1000	1,11166	0,101	0,5186627	0,088288288	0,194731567	0,0680678	1,742	111	350	2	0,6	0,4	9,78178E-07	4	-190,8210735	176,54	10,68146807	0,873708134	8	1,15789E-05	6,12953E-06	9,39915E-06	
320	10	1000	1,11166	0,101	0,5186627	0,088288288	0,194731567	0,0680678	1,742	111	390	2	0,6	0,4	0,879898656	5,630763996	-151,9888335	176,54	7	3	2,759708876	6,89493E-05	4,71073E-05	6,02125E-05	
320	10	1000	1,11166	0,101	0,5186627	0,088288288	0,194731567	0,0680678	1,742	111	450	2	0,4	0,6	0,021739272	4	-199,9998647	176,54	8,170877481	0,742350429	8	0,002066773	9,06389E-06	0,000832147	
320	10	1000	1,11166	0,101	0,5186627	0,088288288	0,194731567	0,0680678	1,742	111	470	2	0,8	0,2	0,627033168	5,382177754	-103,6568433	151,793993	7	2,999995452	3,100362729	9,55641E-06	0,000652277	0,0001381	
320	10	1500	1,0581	0,101	0,544916904	0,088288288	0,185349361	0,0680678	1,658070093	111	470	2	0,8	0,2	1,266399448	6,760017455	-174,461691	176,54	7	3	3,026763884	0,000119988	0,000116221	0,000119235	
...
320	10	1500	1,0581	0,101	0,544916904	0,088288288	0,185349361	0,0680678	1,658070093	111	150	2	0,2	0,8	0,04551867	4	-198,7417674	176,54	21	1,840885445	7,67750313	3,97053E-06	2,2083E-06	2,56074E-06	
320	10	1500	1,0581	0,101	0,544916904	0,088288288	0,185349361	0,0680678	1,658070093	111	170	2	0,4	0,6	0	4	-200	106,1184139	20,95147065	1,566929219	7,492343067	2,81393E-06	3,82224E-08	1,14851E-06	
320	10	1500	1,0581	0,101	0,544916904	0,088288288	0,185349361	0,0680678	1,658070093	111	210	2	0,6	0,4	0	4	-200	129,4168862	21	1,435695577	6,267292498	1,27982E-07	1,50198E-06	6,77581E-07	
320	10	1500	1,0581	0,101	0,544916904	0,088288288	0,185349361	0,0680678	1,658070093	111	290	2	0,4	0,6	0	4,075783828	-153,7067789	170,432428	19,47132995	1,462993894	7,714264371	1,53285E-06	2,0503E-06	1,84616E-06	
320	10	1500	1,0581	0,101	0,544916904	0,088288288	0,185349361	0,0680678	1,658070093	111	310	2	0,8	0,2	0	4	-140,8694778	176,54	18,94751456	1,505681048	8	4,16575E-06	2,66916E-06	3,86643E-06	
...
320	10	1500	1,0581	0,101	0,544916904	0,088288288	0,185349361	0,0680678	1,658070093	111	390	2	0,2	0,8	0	4	-200	176,54	0,077943463	0,764940698	8	3,78729E-05	3,38911E-06	1,02859E-05	
320	10	1500	1,0581	0,101	0,544916904	0,088288288	0,185349361	0,0680678	1,658070093	111	430	2	0,2	0,8	0,009870062	4	-197,7063857	175,4345144	8,684890874	0,7582358	8	0,001078298	9,83873E-06	0,000223531	
320	10	1500	1,0581	0,101	0,544916904	0,088288288	0,185349361	0,0680678	1,658070093	111	450	2	0,6	0,4	0,008627345	4	-200	176,5399248	8,26068167	0,73866311	8	0,00149334	0,000143296	0,000953322	
320	10	1500	1,0581	0,101	0,544916904	0,088288288	0,185349361	0,0680678	1,658070093	111	470	2	0,8	0,2	1,266399448	6,760017455	-174,461691	176,54	7	3	3,026763884	0,000119988	0,000116221	0,000119235	



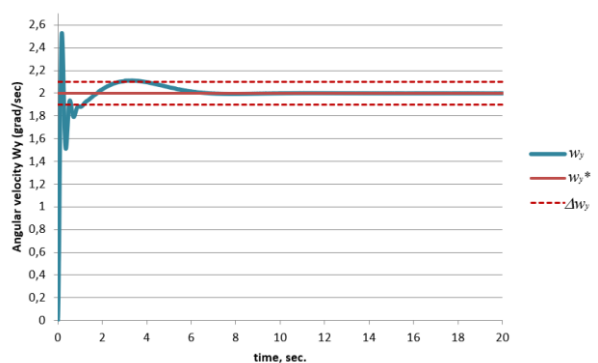
a)



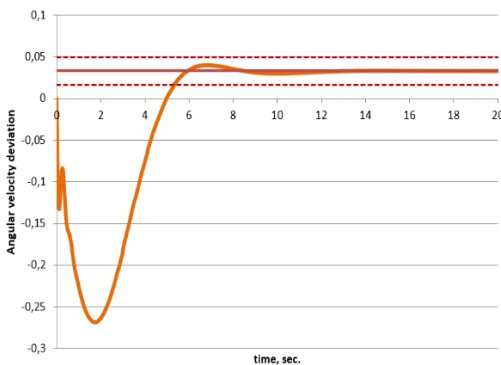
b)



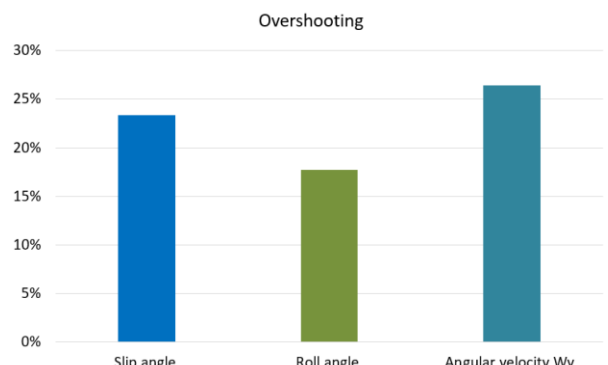
c)



d)

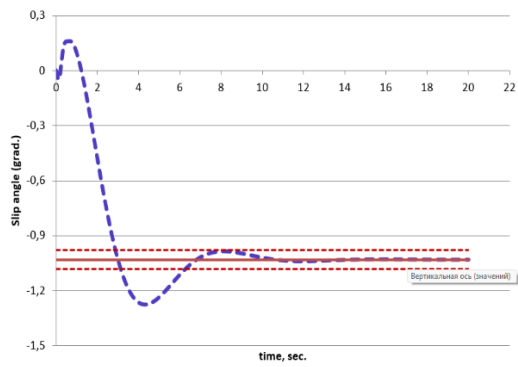


e)

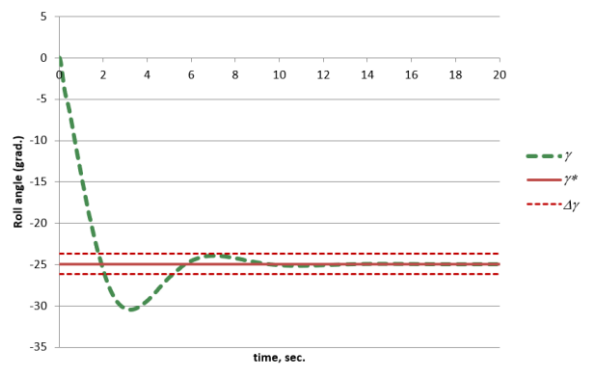


f)

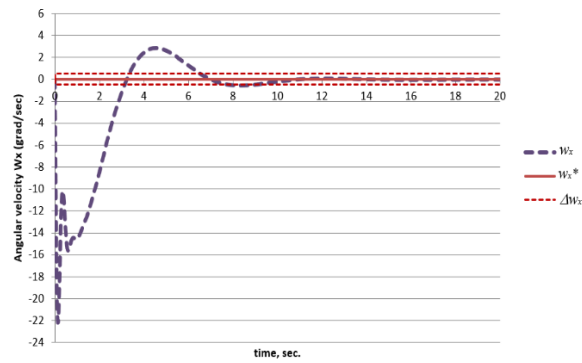
Рисунок 27 – Изменение переменных состояния БПЛА при развороте в условиях: $H=800$ м, $L_{W_x}=170$ м, $\xi_1=0.4$, $\xi_2=0.6$: а) по углу скольжения β ; б) по углу крена γ ; в) по угловой скорости w_x по крену; д) по угловой скорости w_y по рысканью; е) по отклонению ρ угла рысканья от расчетного значения; (здесь $(\cdot)^*$ - значение, гарантирующее устойчивость; $\Delta(\cdot)$ - 5% допуск отклонения от $(\cdot)^*$) ф) перерегулирование системы



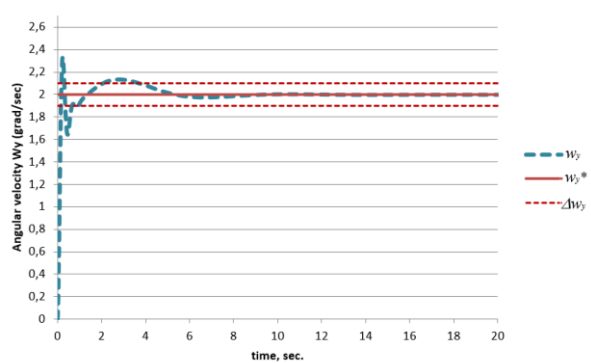
a)



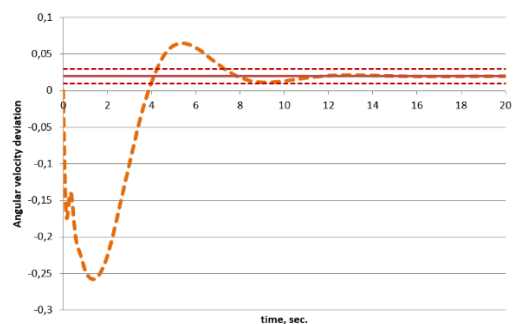
b)



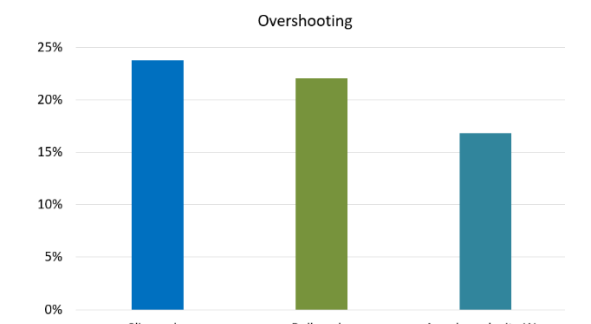
c)



d)

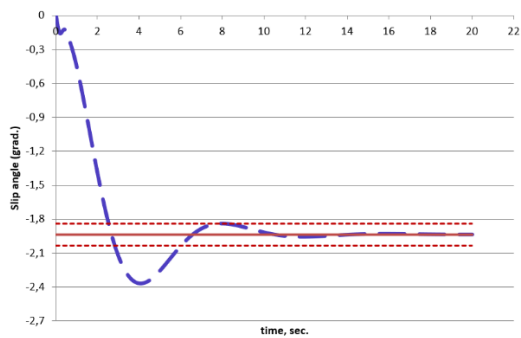


e)

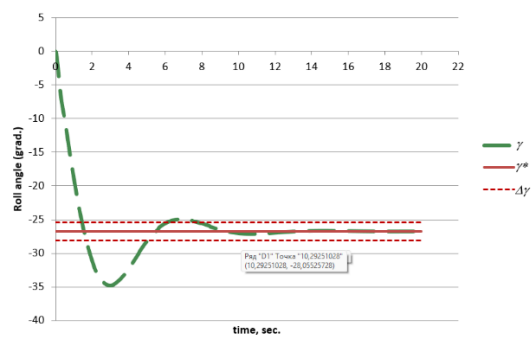


f)

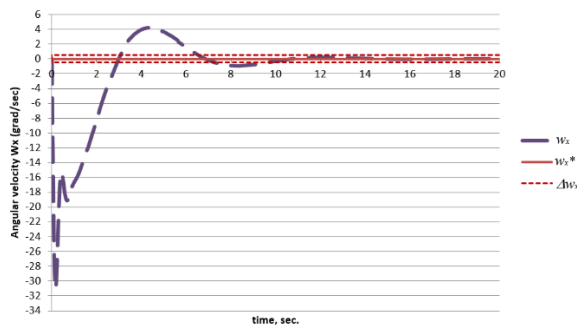
Рисунок 28 – Изменение переменных состояния БПЛА при развороте в условиях: $H=1000$ м, $L_{W_x}=310$ м, $\xi_1=0.2$, $\xi_2=0.8$: а) по углу скольжения β ; б) по углу крена γ ; в) по угловой скорости w_x по крену; д) по угловой скорости w_y по рысканью; е) по отклонению ρ угла рысканья от расчетного значения; (здесь $(\cdot)^*$ - значение, гарантирующее устойчивость; $\Delta(\cdot)$ - 5% допуск отклонения от $(\cdot)^*$) ф) перерегулирование системы



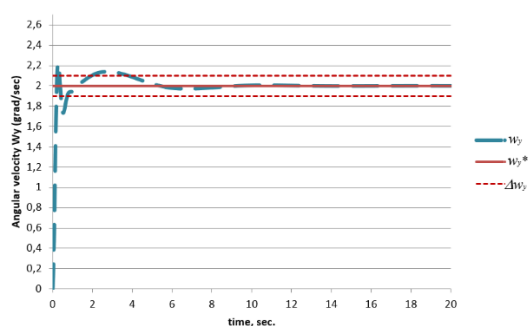
a)



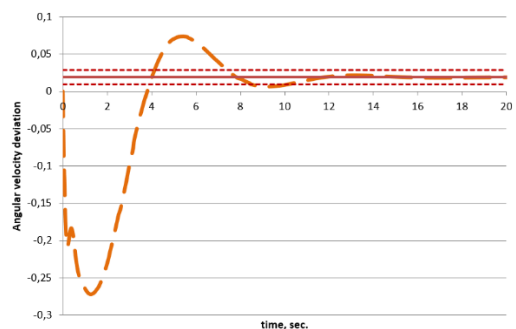
b)



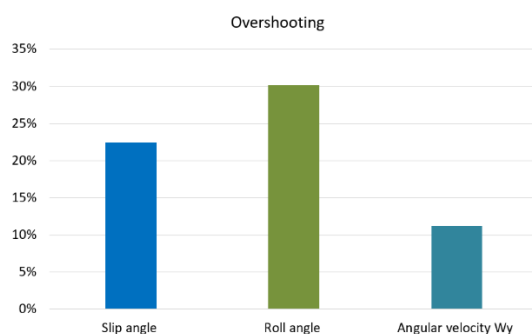
c)



d)



e)



f)

Рисунок 29 – Изменение переменных состояния БПЛА при развороте в условиях: $H=1500$ м, $L_{W_x}=450$ м, $\xi_1=0.8$, $\xi_2=0.2$: а) по углу скольжения β ; б) по углу крена γ ; в) по угловой скорости w_x по крену; д) по угловой скорости w_y по рысканью; е) по отклонению ρ угла рысканья от расчетного значения; (здесь $(\cdot)^*$ - значение, гарантирующее устойчивость; $\Delta(\cdot)$ - 5% допуск отклонения от $(\cdot)^*$) ф) перерегулирование системы

В результате был получен релевантный датасет с параметрами оптимального управления, обеспечивающими гарантированно устойчивый

разворот БПЛА с жестким крылом в различных условиях полета и при меняющейся турбулентности.

Для формирования и обучения многопоточного нейросетевого контроллера с блоком кластеризации временных рядов был применен предложенный алгоритм MTC-NNC на сформированном датасете со следующими параметрами:

- Количество временных фрагментов: 15;
- LSTM Sequence Length: 10;
- Smoothing Window Size: 3;
- Target Variables: 7;
- Разделение датасета на тренировочное и обучающее множество: 85% и 15% соответственно.

В результате данные были разделены на два кластера, содержащих 80% и 20% фрагментов исходной временной последовательности датасета соответственно. Соответственно было обучено 2 LSTM-ветви нейроконтроллера.

Результатирующие значения отклонений расчетов двухпоточного нейроконтроллера от реальных данных на тестовом подмножестве по метрике MAPE (Mean Absolute Percentage Error) [114] приведены на рисунке 30-31.

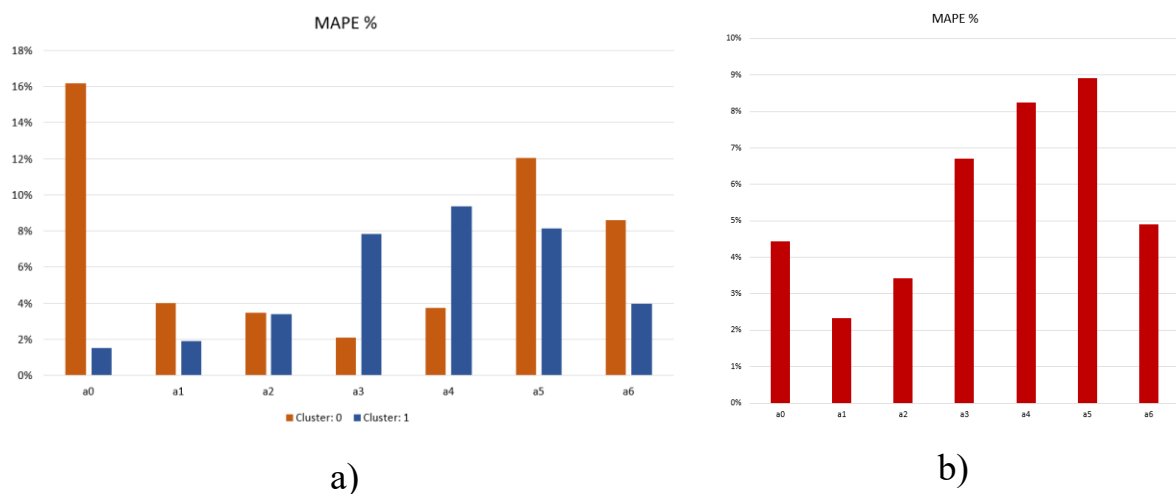
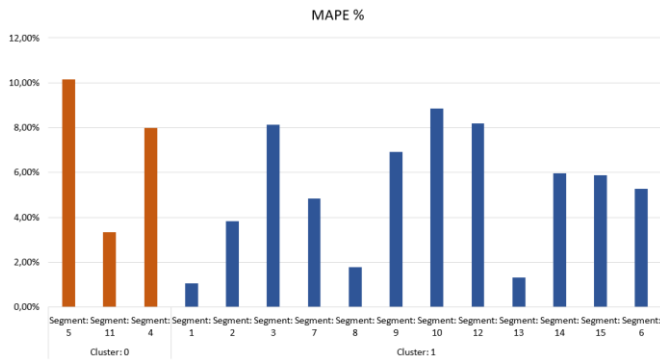
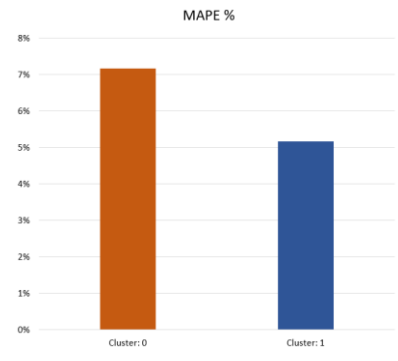


Рисунок 30 – Средние значения ошибок MAPE по управляющим коэффициентам $a_0 - a_6$: а) по каждому из выделенных кластеров; б) по всему датасету в целом



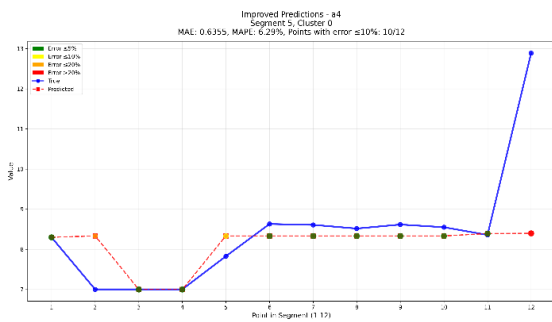
a)



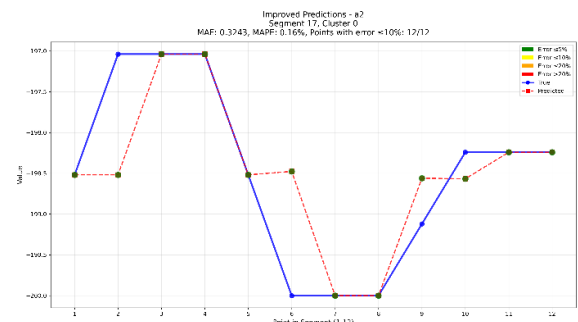
b)

Рисунок 31 – Средние значения ошибок МАРЕ по выделенным фрагментам временного ряда: а) по каждому временному фрагменту; б) по совокупности временных фрагментов внутри каждого кластера

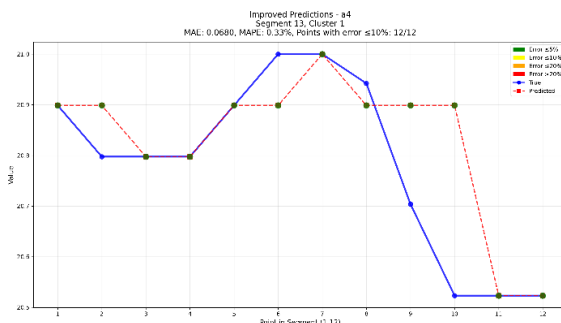
На рисунке 32 в качестве примера приведены графики отклонений реальных и рассчитанных значений управляющего параметра a_4 , имеющего ошибку МАРЕ в среднем несколько выше, чем другие управляющие параметры, как внутри каждого из двух кластеров, так и по датасету.



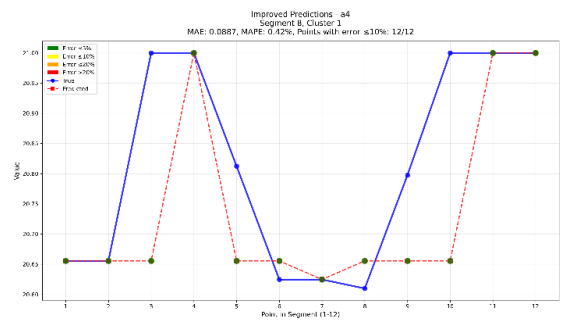
a)



b)



c)



d)

Рисунок 32 – Реальные и рассчитанные двухпоточным нейроконтроллером значения управляющего параметра a_4 : а) временной фрагмент 5, кластер 0; б) временной фрагмент 17, кластер 0; с) временной фрагмент 13, кластер 1; д) временной фрагмент 8, кластер 1

Средняя MAPE ошибка двухпоточного нейроконтроллера по всему датасету составила 5,57%. Время расчета набора управляющих параметров обученным многопоточным нейроконтроллером составило около 0,1 сек. на одну задачу на том же оборудовании, которое использовалось для тестирования алгоритма SoFAM.

4.2.1.6 Результаты эксперимента

Для анализа эффективности разработанного метода было проведено сравнение точности расчетов управляющих коэффициентов автопилота с другими методами того же класса, применяемыми для прогнозирования временных рядов с кластеризацией и без нее. Точность вычисляется как процентное соотношение, обратное значению ошибки MAPE. Для сравнения были исследованы подходы, которые согласно последним исследованиям [115-117] демонстрируют наилучшие результаты по прогнозированию временных рядов однородной и неоднородной структуры:

- 1) прогнозирование единым LSTM-контроллером без предварительной кластеризации исходного датасета;
- 2) прогнозирование LSTM-контроллером с предварительной кластеризацией временных рядов на основе евклидова расстояния;
- 3) прогнозирование LSTM-контроллером с кластеризацией временных рядов, основанной на вычислении параметров модели ARIMA.

Результаты тестирования представлены в таблице 28.

Время расчетов для всех типов проанализированных нейроконтроллеров в среднем составило ту же величину в 0,1 сек, что и для предложенного метода.

Полученные результаты подтверждают высокую эффективность предложенного многопоточного нейросетевого контроллера MTC-NNC, основанного на использовании LSTM-сетей с предварительной кластеризацией временных рядов управляющих параметров. В ходе численных экспериментов было показано, что разработанный подход

обеспечивает среднюю ошибку прогнозирования MAPE на уровне 5,57% по всему датасету, что соответствует средней точности 94,43%, существенно превосходящей как единый LSTM-контроллер без кластеризации, так и известные кластер-LSTM-подходы (таблица 28).

Таблица 28 – Полученная в результате экспериментов точность расчетов управляющих параметров автопилота различными типами нейроконтроллеров

Модель нейроконтроллера	Средняя точность
LSTM без предварительной кластеризации	81,72%
LSTM с кластеризацией временных рядов на основе евклидова расстояния	83,46%
LSTM с кластеризацией, основанной на вычислении параметров модели ARIMA	91,69%
Многопоточный LSTM-контроллер (предложенный) MTC-NNC	94,43%

Результаты, отражённые на рисунках 29–30, подтверждают, что предварительное разделение временных фрагментов на кластеры заметно уменьшает ошибки прогноза в рамках каждой группы. Так, в основном кластере, на который приходится примерно 80 % фрагментов, показатели MAPE по ключевым управляющим коэффициентам стабильно держатся ниже среднего значения по всему набору данных. Во втором кластере (порядка 20 % фрагментов), где фиксируются более выраженные изменения в режимах полёта и управления, погрешности прогноза закономерно выше, но всё ещё укладываются в допустимые границы и не угрожают устойчивости системы управления. Отсюда можно сделать вывод, что разбиение рядов по структурным особенностям выполнено корректно, а LSTM-ветви нейроконтроллера действительно адаптированы под специфику своих кластеров.

Сравнение с LSTM-контроллером без предварительной кластеризации демонстрирует прирост точности более чем на 12,7% (94,43% против 81,72%).

Использование одной модель для структурно разнородных временных рядов ухудшает качество аппроксимации для редких, но важных режимов – например, при резких колебаниях высоты или усилении турбулентности. Кластеризация, напротив, позволяет декомпозировать общую задачу на ряд более простых подзадач: внутри каждого кластера динамика параметров оказывается более предсказуемой и согласованной.

Дополнительный выигрыш по точности по сравнению с LSTM-подходами, использующими кластеризацию на основе евклидова расстояния и параметров ARIMA-моделей (таблица 28), объясняется более информативным представлением временных фрагментов в виде 16-мерного вектора признаков. В него включены такие метрики, как площадь под кривой, сумма квадратов статистической производной, число пересечений заданных уровней и характеристика тренда. Благодаря этому удаётся детальнее отразить и энергетические, и динамические особенности управляющих сигналов, а это критически важно в задачах адаптивного управления летательным аппаратом, где внезапные переходы между режимами способны серьёзно повлиять на устойчивость полёта.

Результаты, приведённые на рисунке 31, демонстрируют, что даже для управляющего параметра с наибольшей средней ошибкой MAPE нейроконтроллер, работающий в многопоточном режиме, достоверно передаёт форму временных зависимостей и корректно воспроизводит ключевые переходные процессы. Отсюда следует, что погрешности прогноза носят преимущественно локальный характер и не формируют систематического отклонения в управляющих воздействиях

Хотя подготовка обучающего набора данных с помощью оптимизационного алгоритма SoFAM требует существенных вычислительных ресурсов (примерно 25,5 часа на 205 задач оптимизации), сам обученный нейроконтроллер справляется с расчётом оптимальных настроек автопилота очень быстро — около 0,1 секунды на задачу. Это открывает возможность для применения предложенного решения в системах адаптивного управления

БПЛА в режиме реального времени, в том числе при скоростях полёта в диапазоне 100–200 км/ч.

4.2.2 Применение для прогнозирования уровня гликемии пациентов с сахарным диабетом

Рассмотрим применение метода из главы 2.3 в задаче краткосрочного прогнозирования уровня гликемии у пациентов с сахарным диабетом 1 типа. Динамика изменения уровня глюкозы в крови характеризуется высокой вариабельностью, наличием шумов и сложными нелинейными зависимостями от множества факторов, таких как питание, введенный инсулин и физическая активность. Традиционные методы прогнозирования зачастую не справляются с такой гетерогенностью данных, что обуславливает актуальность применения других подходов, способных адаптироваться к различным физиологическим состояниям пациента [118].

В данной главе представлены результаты применения метода кластеризации временных рядов на основе параметрического описания (разработанного в Главе 2) к задаче прогнозирования гликемии. Основная цель эксперимента – оценить, насколько предварительное выделение групп схожих паттернов в данных мониторинга пациентов позволяет повысить точность прогнозирования уровня глюкозы с использованием специализированных LSTM-моделей. Исследование проводится на реальных данных, полученных с сенсоров инсулиновых помп, что позволяет проверить работоспособность предложенного подхода в условиях, максимально приближенных к клинической практике.

4.2.2.1 План эксперимента

Для экспериментальной проверки эффективности предложенного в Главе 2 комбинированного подхода к прогнозированию временных рядов на

основе параметрического описания и кластеризации был разработан план вычислительного эксперимента. Целью эксперимента являлась оценка точности прогнозирования уровня гликемии (SGV) с использованием специализированных LSTM-моделей, обученных на выделенных кластерах, в сравнении с классическими подходами.

Эксперимент проводился в несколько последовательных этапов, соответствующих логике разработанного метода.

Этап 1. Формирование и предварительная обработка исходных данных.

На данном этапе из лог-файлов инсулиновых помп извлекаются синхронизированные временные ряды трех ключевых параметров: уровня глюкозы (SGV), активного инсулина (IOB Sum) и активных углеводов (COB). Исходный массив данных делится на две независимые выборки:

- обучающая выборка: данные за первые 8 дней измерений, используемые для кластеризации и обучения нейросетевых моделей;
- тестовая выборка: данные за последующие дни, используемые для финальной оценки точности прогнозирования разработанным методом.

Этап 2. Параметризация и кластеризация временных рядов обучающей выборки.

Этот этап реализуется в следующей последовательности:

1. каждый временной ряд из обучающей выборки разделяется на сегменты длительностью 12 часов для анализа локальных паттернов;
2. для каждого 12-часового сегмента рассчитывается вектор из 15 параметров, детально описанных в главе 2.3.2 (минимум, максимум, среднее, дисперсия, пики, тренд и др.);
3. вычисленные векторы признаков нормализуются для корректной работы алгоритма кластеризации;
4. к нормализованным данным применяется алгоритм DBSCAN.

Цель – автоматическое разделение всех сегментов обучающей выборки на группы (кластеры) со схожими динамическими характеристиками.

Этап 3. Обучение специализированных моделей прогнозирования.

Для каждого кластера, полученного на предыдущем этапе, выполняется:

- 1) из каждого кластера случайным образом выбирается один репрезентативный временной ряд;
- 2) на выбранных данных обучается отдельная модель LSTM. Входными данными для модели служат значения SGV, IOB Sum и COB. Цель обучения – минимизация ошибки прогноза уровня глюкозы;
- 3) точность каждой обученной модели оценивается на ее обучающем ряде с использованием метрики MAPE (Mean Absolute Percentage Error).

Этап 4. Прогнозирование на тестовой выборке и оценка результатов.

Заключительный этап эксперимента включает:

- 1) каждый временной ряд из тестовой выборки (данные за сутки) параметризуется по тому же принципу (деление на сегменты, вычисление 15 признаков) и с помощью сохраненной модели DBSCAN относится к одному из ранее сформированных кластеров;
- 2) для прогнозирования уровня глюкозы на тестовых данных используется LSTM-модель того кластера, к которому был отнесен данный ряд;
- 3) рассчитывается средняя точность прогноза (MAPE) по всей тестовой выборке;
- 4) полученные результаты сравниваются с точностью трех альтернативных подходов:
 - единая LSTM-модель, обученная на всех данных без кластеризации;
 - LSTM с предварительной кластеризацией на основе евклидова расстояния;
 - LSTM с предварительной кластеризацией, основанной на параметрах модели ARIMA.

Данный план позволяет провести полный цикл экспериментальной проверки – от подготовки данных до сравнительного анализа, – и подтвердить эффективность предложенного комбинированного подхода.

4.2.2.2 Реализация эксперимента

Для проведения вычислительных экспериментов использовались данные, полученные с сенсоров инсулиновых помп, которые фиксируют текущий уровень глюкозы в крови (SGV), количество активного инсулина (IOB Sum) и потребление углеводов (COB).

Данные для анализа были извлечены из логов сенсоров инсулиновых помп, которые включают временные метки, значения измерений и другие параметры. Для обработки данных использовались алгоритмы, основанные на регулярных выражениях, которые позволили точно извлечь нужные показатели. Эти данные затем были структурированы в табличный формат для дальнейшего использования в программном комплексе модели для прогнозирования. Фрагмент сформированного смешанного временного ряда, используемого в исследовании, приведен ниже.

Обработанные данные сначала делились на более мелкие временные ряды длительностью 12 часов. Для каждого такого временного ряда определялись динамические и статические характеристики, описанные в главе 2.2.2.

Таблица 29 – Фрагмент сформированного временного ряда

SGV	IOB Sum	COB
83,71	6,15	9,91
83,72	6,03	9,79
83,92	2,54	9,57
84,17	2,42	9,27
84,61	2,16	8,98
84,95	2,06	8,81
85,55	2,06	8,79
86,23	1,94	8,82
86,37	1,65	8,75

В результате обработки обучающей выборки данных, сформированной на основе измерений, выполненных в течение восьми дней, было выделено 16 временных рядов. Для каждого временного ряда были рассчитаны ключевые параметры. Пример рассчитанных параметров представлен ниже.

Таблица 30 – Параметры временных рядов

Ряд	Min	Max	Среднее	Медиана	Дисперсия	Площадь	...	Тренд
1	76	134,8	93,4971	87	268,598	93471,5	...	-0,04052
2	97	134	115,051	111,1	128,165	114904,5	...	-0,02207
3	84	135	110,856	113	312,149	110720,5	...	-0,05264
4	84	132	109,243	110	269,784	109129,5	...	-0,02338
5	55	160	112,751	119,5	1190,361	112664	...	-0,0854
6	85	163	135,128	142	616,3681	134903	...	-0,05
7	88	160	126,172	127	270,434	126040	...	0,05591
8	160	220	186,266	180,1	343,444	186047,5	...	0,05917
9	67	187	112,154	111	1063,401	112211	...	-0,01012
10	102	156	125,127	127,1	214,733	125057,5	...	-0,0197
11	137	176	158,835	157,4	116,403	158638	...	0,01230
12	145	175	159,736	162	75,564	159593,5	...	0,02604
13	135	176	146,844	142	121,029	146726,5	...	-0,02375
14	100	137	115,315	113	164,926	115262	...	0,00556
15	105	156	125,608	127	217,192	125542	...	0,02208
16	133	157	143,348	143	57,191	77587	...	-0,03487

Данные параметров временных рядов затем подвергались кластеризации с использованием алгоритма DBSCAN, который позволяет выделять группы на основе плотности. Полученные кластеры отражали различные состояния пациента, такие как нормальный уровень глюкозы, гипогликемия или гипергликемия. Данные распределились по 4 кластерам. Пример распределения записей по кластерам представлен ниже.

Таблица 31 – Параметры временных рядов, распределенные по кластерам

Ряд	Min	Max	Среднее	Медиана	Дисперсия	Площадь	...	Тренд	Кластер
1	76	134,8	93,4971	87	268,598	93471,5	...	-0,04052	2
2	97	134	115,051	111,1	128,165	114904,5	...	-0,02207	2
3	84	135	110,856	113	312,149	110720,5	...	-0,05264	2
4	84	132	109,243	110	269,784	109129,5	...	-0,02338	2
5	55	160	112,751	119,5	1190,361	112664	...	-0,0854	0
6	85	163	135,128	142	616,3681	134903	...	-0,05	0
7	88	160	126,172	127	270,434	126040	...	0,05591	2
8	160	220	186,266	180,1	343,444	186047,5	...	0,05917	3
9	67	187	112,154	111	1063,401	112211	...	-0,01012	0
10	102	156	125,127	127,1	214,733	125057,5	...	-0,0197	2
11	137	176	158,835	157,4	116,403	158638	...	0,01230	1
12	145	175	159,736	162	75,564	159593,5	...	0,02604	1
13	135	176	146,844	142	121,029	146726,5	...	-0,02375	1
14	100	137	115,315	113	164,926	115262	...	0,00556	2
15	105	156	125,608	127	217,192	125542	...	0,02208	2
16	133	157	143,348	143	57,191	77587	...	-0,03487	1

В результате кластеризации временные ряды распределились по четырем кластерам. Кластер 0 включает ряды 5, 9. Кластер 1 состоит из рядов 11, 12, 13, 16. Кластер 2 включает ряды 1, 2, 3, 4, 7, 10, 14, 15. Кластер 3 включает ряд 8.

Для каждого кластера формировали свою отдельную модель. Номер кластера на вход сети не подавался. Метка кластера выполняла только организационную функцию: с её помощью разделяли данные и обучали узкоспециализированные модели под особенности каждой группы.

В качестве основы брали LSTM-сеть – она хорошо улавливает зависимости во времени. Для обучения из каждого кластера случайным образом брали один временной ряд: его данные шли на тренировку модели. Такой приём помогал точнее прогнозировать уровень глюкозы (SGV) в разных состояниях пациента.

На вход модели подавали три параметра: SGV, IOB Sum и COB. За счёт этого модель учитывала не только текущее значение глюкозы, но и динамику её изменения – с учётом влияния инсулина и углеводов.

Для оценки точности прогнозов использована средняя абсолютная процентная ошибка MAPE [114]:

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| * 100 \quad (73)$$

где y_i – фактическое значение на i -м шаге, \hat{y}_i – предсказанное значение на i -м шаге, n – количество данных для анализа (количество временных шагов после обработки данных).

Точность определялась как величина, обратная ошибке:

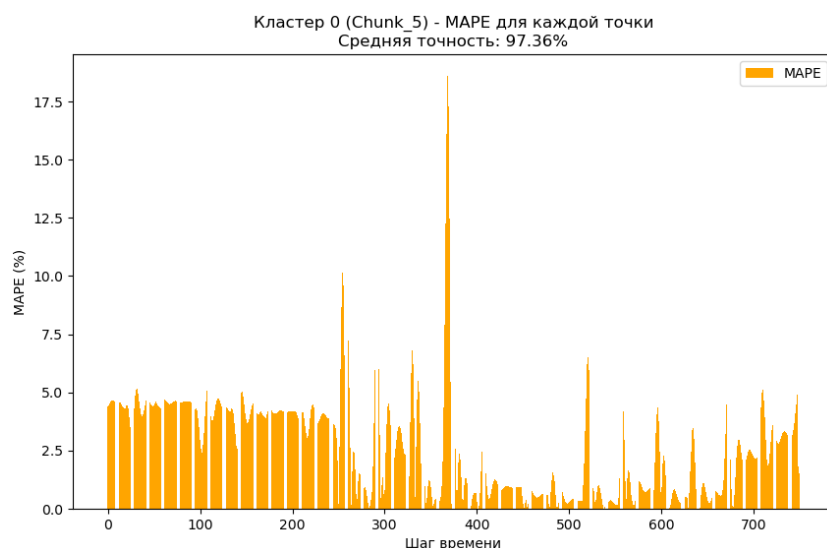
$$\varepsilon = 100 - MAPE \quad (74)$$

4.2.2.3 Результаты экспериментов

Для обучения LSTM-модели кластера 0 использовался временной ряд 5. Обучение показало среднюю точность 97,36%. Результаты обучения представлены на рисунке 33.



а)



б)

Рисунок 33 – Результаты обучения для кластера 0:

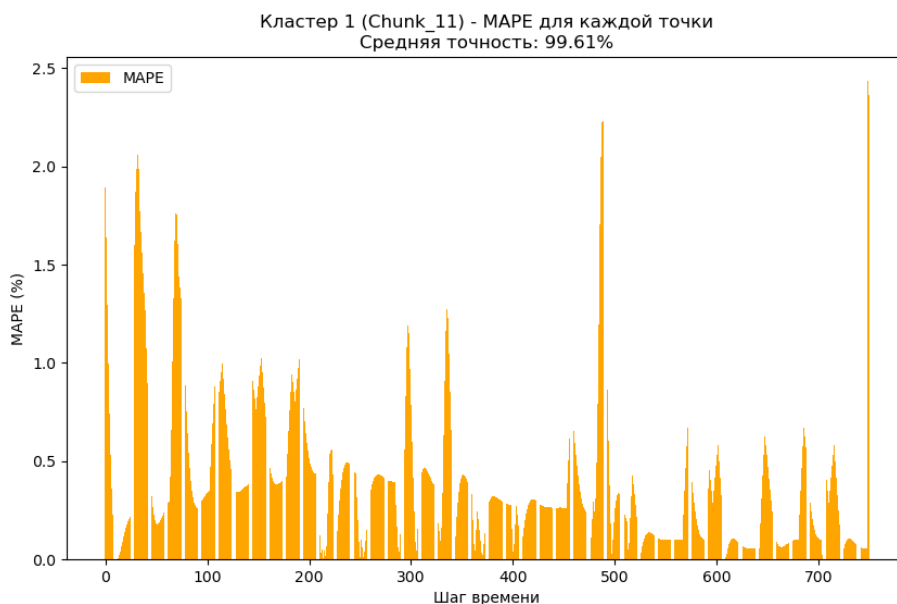
а) отклонение прогнозных значений от реальных данных;

б) пошаговый уровень ошибки прогноза

Для обучения LSTM-модели кластера 1 использовался временной ряд 11. Обучение показало среднюю точность 99,61%. Результаты обучения представлены на рисунке 34.



а)



б)

Рисунок 34 – Результаты обучения для кластера 1:

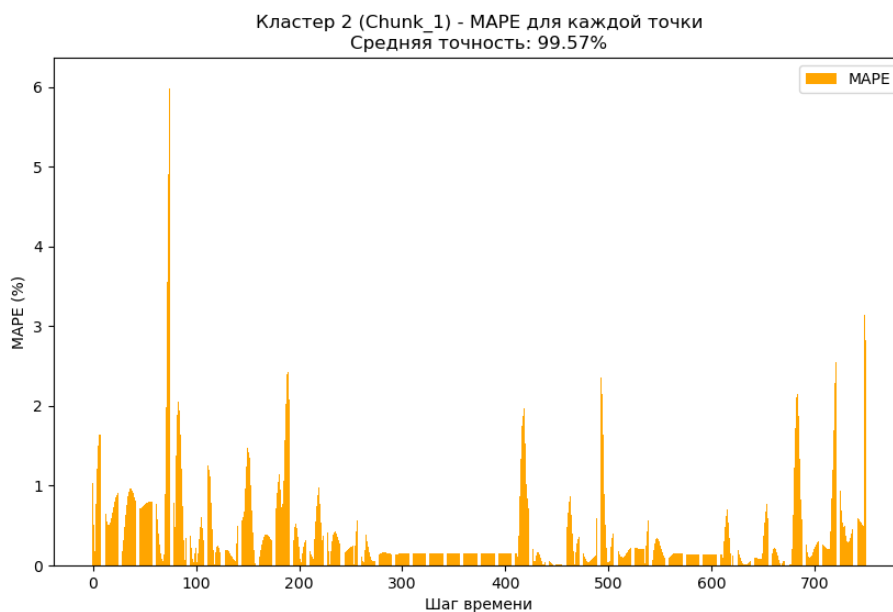
а) отклонение прогнозных значений от реальных;

б) пошаговый уровень ошибки прогноза

Для обучения LSTM-модели кластера 2 использовался временной ряд 1. Обучение показало среднюю точность 99,57%. Результаты обучения представлены на рисунке 35.



а)



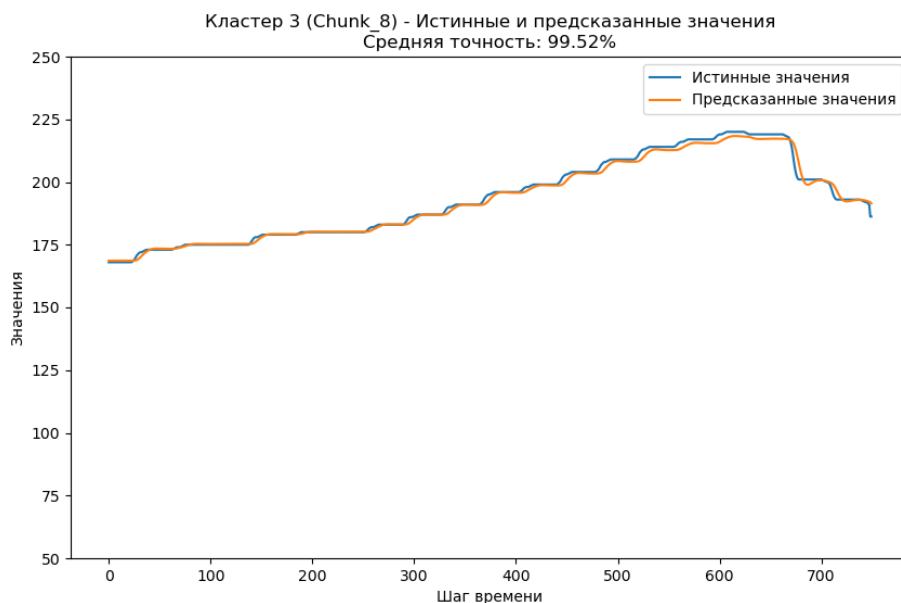
б)

Рисунок 35 – Результаты обучения для кластера 2:

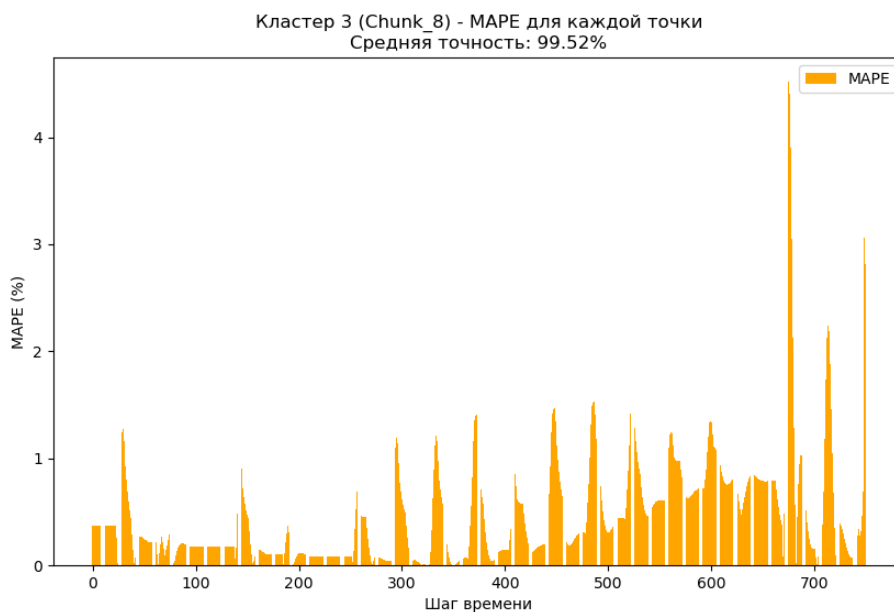
а) отклонение прогнозных значений от реальных;

б) пошаговый уровень ошибки прогноза

Для обучения LSTM-модели кластера 3 использовался временной ряд 8. Обучение показало среднюю точность 99,52%. Результаты обучения представлены на рисунке 36.



а)



б)

Рисунок 36 – Результаты обучения для кластера 3:

а) отклонение прогнозных значений от реальных;

б) пошаговый уровень ошибки прогноза

Процесс прогнозирования для новых данных включал несколько этапов. На первом этапе новый временной ряд, представляющий данные за сутки, был представлен в том же формате, что и обучающие данные.

Далее для нового временного ряда определялся номер кластера с помощью обученной модели кластеризации DBSCAN, на основе тех же ключевых параметров, которые использовались при подготовке данных для кластеризации. После этого выбиралась соответствующая модель LSTM, которая была заранее обучена на данных данного кластера, и выполнялось прогнозирование уровня глюкозы (SGV) на заданный промежуток времени.

Выполним расчет для одного входного временного ряда. Для этого разделим тестовый временной ряд на 15 равных частей и выберем 3-й ряд. Этот 3-й временной ряд был отнесен к кластеру 3, и для его прогнозирования была использована LSTM-модель, обученная на данных, принадлежащих кластеру 3. Результат прогнозирования представлен на рисунке ниже.

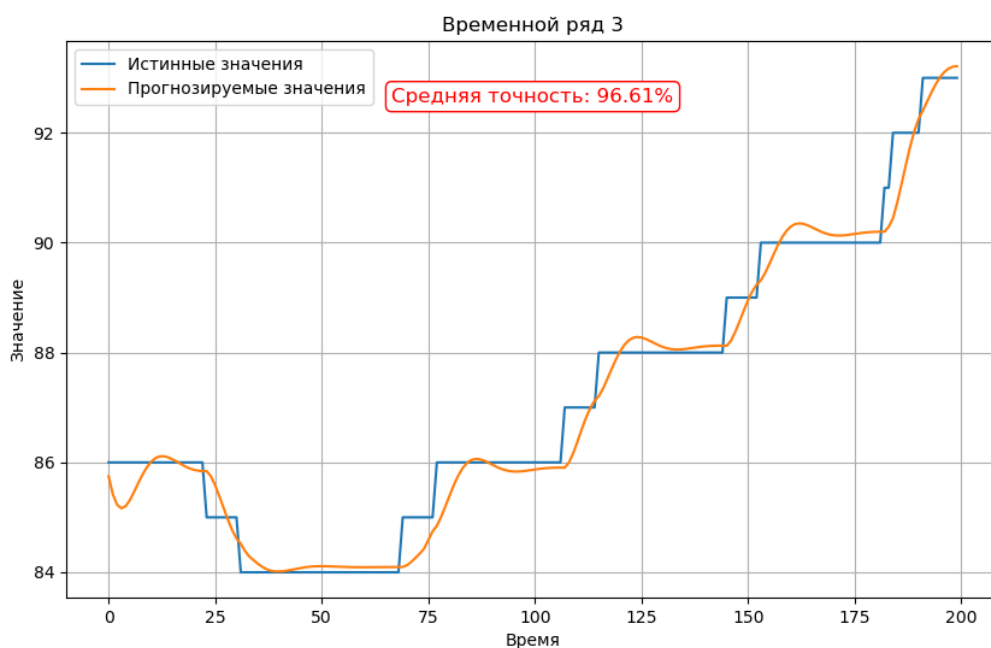


Рисунок 37 – Результаты прогнозирования для нового временного ряда

Обобщенные результаты прогнозирования представлены в таблице 32.

Таблица 32 – Результаты прогнозирования

Новый входной ряд	Кластер, к которому был определен ряд	Средняя точность прогноза
1	1	95,95533635%
2	2	95,63109678%
3	3	96,61094806%
4	2	95,81722369%
5	0	91,02030059%
6	0	92,40097836%
7	2	97,33380423%
8	2	91,63187927%
9	0	92,682793%
10	2	96,87877342%
11	1	99,30361861%
12	1	96,95888013%
13	2	96,73396784%
14	2	96,07275373%
15	3	95,8094856%

Для подтверждения преимущества разработанного метода в рамках данного исследования было проведено сравнение эффективности

прогнозирования с другими методами, применимыми для кластеризации временных рядов. Сравнение предложенного метода осуществлялось по показателю точности, вычисляемому согласно формуле (74), с использованием следующих подходов:

- 1) обучение LSTM без предварительной кластеризации;
- 2) обучение LSTM с кластеризацией временных рядов на основе евклидова расстояния;
- 3) обучение LSTM с кластеризацией, основанной на вычислении параметров модели ARIMA.

Результаты тестирования трех методов, а также предложенного комбинированного метода представлены ниже.

Таблица 33 – Сравнение средней точки существующих методов

Модель	Средняя точность
LSTM без предварительной кластеризации	81,72%
LSTM с кластеризацией временных рядов на основе евклидова расстояния	83,46%
LSTM с кластеризацией, основанной на вычислении параметров модели ARIMA	91,69%
LSTM с кластеризацией на основе комбинированного подхода	95,23%

Рисунок ниже демонстрирует графическое сравнение эффективности рассматриваемых методов.



Рисунок 38 – Сравнение эффективности прогнозирования

Анализ итогов показывает, что кластеризация в связке с параметрическим описанием временных рядов реально помогает делать прогнозы точнее. Ошибка прогноза у моделей, которые работали с преобразованными данными, оказалась на 14 % ниже, чем у тех, что строили прогноз прямо по исходным рядам, без какой-либо предварительной обработки.

Это как раз подтверждает изначальную идею: когда данные переводят в набор ключевых параметров, из них уходит лишний шум, а на первый план выходят именно те характеристики, которые по-настоящему влияют на динамику показателя. В итоге модель лучше улавливает суть процесса – и прогноз получается надёжнее.

4.3 Выводы

В четвёртой главе представлены итоги экспериментальной проверки методов аппроксимации, кластеризации и прогнозирования временных рядов. Тестирование проводили на синтетических данных и на реальных примерах из

разных областей – это дало возможность посмотреть, как предложенные решения ведут себя в неодинаковых условиях.

Метод адаптивной кусочно-линейной аппроксимации на практике показал, что за счёт динамического подбора длины сегментов удаётся заметно уменьшить объём обрабатываемых данных, при этом точность почти не страдает. Метод проверялся на двух прикладных задачах: при аппроксимации траекторий управления гибридными БПЛА и при моделировании динамики концентрации экзогенного инсулина в плазме крови — в обоих случаях подход сработал корректно.

Ещё одна серия экспериментов посвящена кластеризации временных рядов и прогнозированию с помощью гибридной LSTM-модели. На базе этих наработок был собран нейросетевой контроллер, который рассчитывает управляющие параметры автопилота БПЛА в условиях атмосферных возмущений. Результаты подтвердили пользу предварительной кластеризации: она помогает учитывать специфику разных режимов движения и в итоге делает систему управления устойчивее.

Подход также испытан на данных непрерывного мониторинга глюкозы у пациентов с сахарным диабетом. Эксперименты показали, что, если сначала разбить временные ряды на кластеры, а потом для каждого кластера обучить свою LSTM-модель, прогноз получается точнее, чем при использовании одной универсальной модели. К тому же такой подход спокойнее реагирует на шумы и пропуски в данных.

Эксперименты подтвердили, что разработанные методы и программный комплекс действительно работают. Они не только повышают точность прогнозов, но и позволяют сократить вычислительные затраты. Это делает предложенные решения вполне пригодными для систем реального времени и для интеллектуальных инструментов поддержки принятия решений.

ЗАКЛЮЧЕНИЕ

В диссертационной работе решена научная задача разработки методов и моделей адаптивного прогнозирования неоднородных временных рядов в условиях ограниченных вычислительных ресурсов. В ходе исследования проведен анализ существующих методов кластеризации и прогнозирования временных рядов и разработан комплекс алгоритмов адаптивной обработки данных, включающий метод кусочно-линейной аппроксимации с динамическим определением сегментов переменной длины, метод кластеризации временных рядов на основе статистических и динамических характеристик, а также гибридную нейросетевую модель прогнозирования на основе архитектуры LSTM. Предложенные методы реализованы в виде программного комплекса, обеспечивающего полный цикл обработки временных данных: предобработку, структуризацию, кластеризацию и прогнозирование.

Эффективность разработанных методов подтверждена результатами вычислительных экспериментов на реальных и синтетических временных рядах из разных предметных областей, включая задачи управления беспилотными аппаратами и прогнозирования физиологических показателей.

Разработанное в диссертации алгоритмическое обеспечение включает в себя следующие компоненты:

1. метод и алгоритм адаптивной кусочно-линейной аппроксимации временных рядов с динамическим определением сегментов переменной длины, который обеспечивает снижение размерности данных при сохранении заданной точности аппроксимации и позволяет уменьшить вычислительные затраты обработки временных рядов;

2. метод и алгоритм кластеризации временных рядов, основанный на вычислении статистических и динамических характеристик, позволяющий формировать однородные группы временных рядов с использованием алгоритма DBSCAN;

3. гибридная нейросетевая модель прогнозирования временных рядов, основанная на интеграции методов кластеризации и рекуррентных нейронных сетей типа LSTM, которая выполняет обучение специализированных нейросетевых моделей для различных кластеров временных рядов, что позволяет учитывать особенности их динамики и повышать устойчивость прогнозирования к шумам и пропускам данных

Разработанные программные модули спроектированы согласно оригинальной программной архитектуре, реализующей предложенные методы и алгоритмы, и применены на практике.

В качестве практической апробации выполнен расчет управляющих параметров автопилота беспилотного летательного аппарата в условиях атмосферных возмущений, а также прогнозирование уровня глюкозы в крови пациентов с сахарным диабетом на основе данных непрерывного мониторинга. Вычислительные эксперименты показали, что применение разработанного программного комплекса позволяет повысить точность прогнозирования временных рядов до 95,23%, что превышает точность базовой LSTM-модели без кластеризации (81,72%) и известных кластерных подходов (83,46–91,69%), то есть примерно в 1,16 раза; использование метода адаптивной кусочно-линейной аппроксимации снижает при этом вычислительные затраты обработки данных в 11 раз при сохранении заданной точности аппроксимации. Разработанные программные модули могут применяться в системах анализа временных данных, интеллектуальных системах управления и биомедицинских информационных системах, функционирующих в условиях ограниченных вычислительных ресурсов.

Рекомендуется применение разработанных методов и программных средств при создании интеллектуальных систем анализа и прогнозирования временных рядов различной природы.

Перспективы дальнейшей разработки темы связаны с развитием методов адаптивной кластеризации временных рядов, расширением архитектур нейросетевых моделей прогнозирования, а также применением

разработанных методов для анализа больших потоков данных и интеграцией алгоритмов в распределенные вычислительные системы.

СПИСОК ЛИТЕРАТУРЫ

1. Box, G. E. P. Time Series Analysis: Forecasting and Control / G. E. P. Box, G. M. Jenkins, G. C. Reinsel. – 4th ed. – Hoboken : John Wiley & Sons, 2008. – 746 p.
2. Kalman, R. E. A new approach to linear filtering and prediction problems / R. E. Kalman // Journal of Basic Engineering. – 1960. – Vol. 82, No. 1. – P. 35-45.
3. Hyndman, R. J. Forecasting: Principles and Practice / R. J. Hyndman, G. Athanasopoulos. – 3rd ed. – [S. l.] : OTexts, 2021
4. Breiman, L. Random forests / L. Breiman // Machine Learning. – 2001. – Vol. 45, No. 1. – P. 5-32.
5. Friedman, J. H. Stochastic gradient boosting / J. H. Friedman // Computational Statistics & Data Analysis. – 2002. – Vol. 38, No. 4. – P. 367-378.
6. Kushnir, M. Ya. Hybrid model of self-organizing map and adaptive neuro fuzzy inference system in stock indexes forecasting / M. Ya. Kushnir, K. A. Tokarieva // Bukovynian Mathematical Journal. – 2021. – Vol. 9, No. 2. – P. 70-80.
7. Hochreiter, S. Long short-term memory / S. Hochreiter, J. Schmidhuber // Neural Computation. – 1997. – Vol. 9, No. 8. – P. 1735-1780.
8. LeCun, Y. Gradient-based learning applied to document recognition / Y. LeCun, L. Bottou, Y. Bengio, P. Haffner // Proceedings of the IEEE. – 1998. – Vol. 86, No. 11. – P. 2278-2324.
9. Rabiner, L. R. A tutorial on hidden Markov models and selected applications in speech recognition / L. R. Rabiner // Proceedings of the IEEE. – 1989. – Vol. 77, No. 2. – P. 257-286.
10. Zhu, T. Deep learning for diabetes: A systematic review / T. Zhu, K. Li, P. Herrero, P. Georgiou // IEEE Journal of Biomedical and Health Informatics. – 2023. – Vol. 27, No. 6. – P. 2784-2798.
11. Bunescu, R. Blood glucose level prediction using physiological models and support vector machines / R. Bunescu, C. Struble, C. Marling // HEALTHINF

2013 : Proceedings of the International Conference on Health Informatics. – 2013. – P. 135-144.

12. Григорьев, А. А. Методы и алгоритмы обработки данных : учебное пособие / А. А. Григорьев. – 2-е изд., перераб. и доп. – Москва : ИНФРА-М, 2024. – 383 с. – ISBN 978-5-16-019247-1.

13. Алимова, И. С. Сравнительный анализ мер сходства, основанных на преобразовании скользящих аппроксимаций, в задачах классификации временных рядов / И. С. Алимова, В. Д. Соловьев, И. З. Батыршин // Труды Института системного программирования РАН. – 2016. – Т. 28, № 6. – С. 207–222. – DOI: 10.15514/ISPRAS-2016-28(6)-15.

14. Faloutsos, C. Fast subsequence matching in time-series databases / C. Faloutsos, M. Ranganathan, Y. Manolopoulos // Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data. – New York: ACM, 1994. – P. 419-429. – DOI: 10.1145/191843.191925.

15. Berndt, D. J. Using Dynamic Time Warping to Find Patterns in Time Series / D. J. Berndt, J. Clifford // KDD Workshop. – 1994. – Vol. 10, No. 16. – P. 359-370.

16. Box, G. E. P. Time Series Analysis: Forecasting and Control / G. E. P. Box, G. M. Jenkins, G. C. Reinsel. – 4th ed. – Hoboken: John Wiley & Sons, 2008. – 746 p. – DOI: 10.1002/9781118619193.

17. Holder, C. A Review and Evaluation of Elastic Distance Functions for Time Series Clustering / C. Holder, M. Middlehurst, A. Bagnall // Knowledge and Information Systems. – 2023. – Vol. 65, No. 2. – P. 765-809.

18. Бурнаев, Е. В. Меры близости на основе вейвлет коэффициентов для сравнения статистических и расчетных временных рядов / Е. В. Бурнаев, Н. Н. Оленев // Вычислительный центр им. А.А. Дородницына РАН. – Москва, 2006.

19. Болдырев, М. А. Адаптация алгоритма динамической трансформации временной шкалы для задачи поиска расстояния между двумя

временными рядами с периодами низкой изменчивости значений / М. А. Болдырев, А. В. Зубков // Инженерный вестник Дона. – 2025. – № 3.

20. Цымблер, М. Л. Параллельный алгоритм восстановления сенсорных данных в режиме реального времени для многоядерного процессора / М. Л. Цымблер, А. Н. Полуянов, Я. А. Краева // Вестник Южно-Уральского государственного университета. Серия: Вычислительная математика и информатика. – 2022. – Т. 11, № 3. – С. 69-90.

21. Стрижов, В. В. Кластеризация пучков временных рядов путем динамической свертки свободной переменной / В. В. Стрижов // XV-ая конференция. – Москва : Вычислительный центр РАН.

22. Sakoe, H. Dynamic programming algorithm optimization for spoken word recognition / H. Sakoe, S. Chiba // IEEE Transactions on Acoustics, Speech, and Signal Processing. – 1978. – Vol. 26, No. 1. – P. 43-49.

23. Алексеева, В. А. Анализ временных рядов : учебное пособие для студентов, обучающихся по направлению 01.03.04 / В. А. Алексеева ; Министерство науки и высшего образования Российской Федерации, Ульяновский государственный технический университет. – Ульяновск : УлГТУ, 2020. – 147 с. – ISBN 978-5-9795-2091-9.

24. Гуляев, А. В. Метрики оценки зависимости двух временных рядов / А. В. Гуляев, С. В. Пивнева // Сборник трудов IX Международной научной конференции «Конвергентные когнитивно-информационные технологии» (Convergent-2024). – Москва : МГУ имени М.В. Ломоносова, 2024.

25. Ламехов, В. А. Построение краткосрочного прогноза количества вагонов на станции и путях необщего пользования / В. А. Ламехов, Е. К. Коровяковский // Автоматика на транспорте. – 2024. – Т. 10, № 2. – С. 199-216.

26. Кириллов, Д. С. Методика вычисления коэффициента корреляции для нестационарных временных рядов / Д. С. Кириллов, Л. В. Ключкова, Ю. Н. Орлов, В. Ф. Тишкин // Журнал Средневолжского математического общества. – 2013. – Т. 15, № 1. – С. 8-15.

27. Котенко, А. П. Оценка коэффициента корреляции временных рядов на основе транспортной задачи линейного программирования / А. П. Котенко, М. Б. Букаренко // Проблемы экономики современных промышленных комплексов. Финансирование и кредитование в экономике России: методологические и практические аспекты : сборник статей VI-й Всероссийской научно-практической конференции. – Самара : СГАУ, 2013. – Вып. 9.

28. Королев, В. Ю. Асимптотические и аналитические свойства смешанных вероятностных моделей и их применение к анализу сложных систем / В. Ю. Королев, И. Г. Шевцова, О. В. Шестаков // Вестник Московского университета. Серия 15. Вычислительная математика и кибернетика. – 2024. – № 4. – С. 102-148. – DOI: 10.55959/MSU/0137-0782-15-2024-47-4-102-148.

29. Борисова, А. Д. Выявление функциональных состояний оператора по вариабельности сердечного ритма с помощью скрытых марковских моделей / А. Д. Борисова, Н. Ю. Золотых, М. Д. Абрамова // Труды Института системного программирования РАН. – 2022. – Т. 34, № 2. – С. 87-102. – DOI: 10.15514/ISPRAS-2022-34(2)-7.

30. Башкова, К. Поиск моделей временных рядов в виде стохастических дифференциальных уравнений / К. Башкова, А. А. Хватов // Доклады Российской академии наук. Математика, информатика, процессы управления. – 2025. – Т. 527. – С. 301–310. – DOI: 10.7868/S2686954325070264.

31. Рыжков, Д. В. О методах кластеризации динамических временных рядов / Д. В. Рыжков // Сборник научных трудов XVII Всероссийской научной конференции молодых ученых «Наука. Технологии. Инновации» (Новосибирск, 04–08 декабря 2023 г.) : в 11 ч. – Новосибирск : Изд-во НГТУ, 2023. – Ч. 2. – С. 178-182. – ISBN 978-5-7782-5130-4.

32. Бериков, В. Б. Метод кластеризации разнородных временных рядов / В. Б. Бериков, И. А. Пестунов, М. К. Герасимов // Вычислительные технологии. – 2015. – Т. 20, № 2. – С. 20-28. – Zbl 1325.62160.

33. Булашова, Х. А. Применение ARIMA-модели в прогнозировании объема грузооборота в Республике Казахстан / Х. А. Булашова // Международный журнал гуманитарных и естественных наук. – 2020. – № 7-2 (46). – С. 103-107.
34. Егоров, А. Н. Сравнительный анализ прогнозирования почасовой энергетической нагрузки на основе ARIMA и LSTM / А. Н. Егоров, Ю. А. Орлова // Экономика и математические методы. – 2024. – Т. 60, № 1. – С. 99-112.
35. Сафиуллин, Н. А. Сравнительный анализ моделей прогнозирования остаточного ресурса нефтегазового оборудования / Н. А. Сафиуллин, К. А. Однолько // Вестник технологического университета. – 2025. – Т. 28, № 4. – С. 54-64. – DOI: 10.55421/1998-7072_2025_28_4_54.
36. Юмагулова, Д. Р. Анализ временных рядов и прогнозирование добычи нефти с помощью Python / Д. Р. Юмагулова // Студенческий вестник. – 2023. – № 46-5 (285). – С. 28-29.
37. Муродов, Н. Ш. Прогнозирование временного ряда на основе модели ARIMA / Н. Ш. Муродов // Информационные технологии и математическое моделирование в управлении сложными системами. – 2023. – № 1 (17). – С. 26-35.
38. Коршунов, Г. И. Моделирование и прогнозирование аварийности на угольных шахтах на основе скрытых марковских цепей / Г. И. Коршунов, С. Л. Баскаков // Горный информационно-аналитический бюллетень (научно-технический журнал). – 2020. – № 3. – С. 95-105.
39. Дорофеев, Е. А. Марковские модели и рекуррентные нейронные сети в задаче прогнозирования динамики геофизических данных / Е. А. Дорофеев, В. В. Бахтин // Известия высших учебных заведений. Геодезия и аэрофотосъемка. – 2023. – Т. 67, № 2. – С. 144-155.
40. Бабешко, Л. О. Байесовский подход в эконометрике и его реализация в программной среде R / Л. О. Бабешко // Современные проблемы науки и образования. – 2022. – № 2. – С. 1-18.

41. Фокин, Н. Д. Науकाстинг и прогнозирование основных российских макроэкономических показателей с помощью MFVBVAR-модели / Н. Д. Фокин // Экономическая политика. – 2022. – Т. 17, № 3. – С. 110-133.
42. Пыко, Н. С. Непараметрические байесовские сети как инструмент комплексирования данных мультимасштабного анализа временных рядов и дистанционного зондирования / Н. С. Пыко, Д. В. Тишин, П. Ю. Искандиров, А. М. Гафуров, Б. М. Усманов, М. И. Богачев // Известия высших учебных заведений России. Радиоэлектроника. – 2023. – Т. 26, № 3. – С. 32-47.
43. Бабешко, Л. О. Методы Монте-Карло по схеме Марковской цепи в байесовской эконометрике / Л. О. Бабешко // Экономика и математические методы. – 2021. – Т. 57, № 4. – С. 112-124.
44. Масленников, Д. А. Обзор методов кластеризации временных рядов на основе глубокого обучения / Д. А. Масленников, А. А. Гуляев // Научно-технический вестник информационных технологий, механики и оптики. – 2023. – Т. 23, № 4. – С. 712-724.
45. Петров, И. В. Кластеризация многомерных временных рядов с использованием глубоких нейросетевых архитектур / И. В. Петров, С. В. Иванов // Вестник Южно-Уральского государственного университета. Серия: Вычислительная математика и информатика. – 2024. – Т. 13, № 1. – С. 45-58. – DOI: 10.14529/cmse240104.
46. Трофимов, А. Г. Сравнительный анализ алгоритмов кластеризации временных рядов на основе глубокого обучения / А. Г. Трофимов, Е. С. Николаева // Труды Института системного программирования РАН. – 2024. – Т. 36, № 2. – С. 123-138.
47. Пашшоев, Б. Анализ нейросетевых моделей для прогнозирования временных рядов / Б. Пашшоев, Д. А. Петрусевич // Russian Technological Journal. – 2024. – Т. 12, № 4. – С. 78-92.
48. Хуснутдинов, А. О. Глубокое обучение для анализа многомерных временных рядов: систематизация типов данных, задач, архитектур и подходов

/ А. О. Хуснутдинов, В. И. Хабаров, В. С. Карманов // Системы анализа и обработки данных. – 2025. – № 3(99). – С. 113-136.

49. Paparrizos, J. Bridging the Gap: A Decade Review of Time-Series Clustering Methods / J. Paparrizos, J. Yang, I. F. Ilyas, M. J. Franklin, T. Palpanas, A. A. H. A. Qahtan // arXiv preprint arXiv:2412.20582. – 2024.

50. Ay, M. CNN-LSTM and clustering-based spatial-temporal demand forecasting for on-demand ride services / M. Ay, S. Kulluk, L. Özbakır, B. Gülmez, G. Öztürk, S. Özer // Neural Computing and Applications. – 2022. – Vol. 34. – P. 22071-22086.

51. Najafgholizadeh, A. Imaging Time Series for Deep Embedded Clustering: a Cryptocurrency Regime Detection Use Case / A. Najafgholizadeh, H. R. Soltanalizadeh, A. M. E. S. M. S. M. S. A. A. // 2022 8th International Conference on Web Research (ICWR). – Tehran : IEEE, 2022.

52. Wang, J. Jointly learning deep features, topological structure and connectivity for time series clustering / J. Wang, H. Duan, C. Zhang, F. Nie // Applied Intelligence. – 2024. – Vol. 54. – P. 1271-1288.

53. Zhao, L. A deep clustering algorithm based on variational autoencoder and graph convolutional networks / L. Zhao, J. Zhao, Y. Liu, Y. Zhang // Neurocomputing. – 2023. – Vol. 520. – P. 1-12.

54. S.A. Yarushev, & A.N. Averkin (2016). Review of studies on time series forecasting based on hybrid methods, neural networks and multiple regression. Программные продукты и системы, (1 (113)), 75-82. doi: 10.15827/0236-235X.113.075-082

55. Галанов, К. В. Кластеризация для валидации моделей прогнозирования временных рядов / К. В. Галанов, А. В. Куприянов // Информационные технологии и нанотехнологии (ИТНТ-2024) : Сборник трудов по материалам X Международной конференции и молодежной школы. В 6-ти томах, Самара, 20–24 мая 2024 года. – Самара: Самарский национальный исследовательский университет им. акад. С.П. Королева, 2024. – С. 42952.

56. Дуброва, Т. А. Статистические методы прогнозирования : учебное пособие для вузов / Т. А. Дуброва. – Москва : ЮНИТИ, 2003. – 208 с.
57. Филимонов, А. Б. Экспоненциальное сглаживание временных рядов с полиномиальным трендом / А. Б. Филимонов // Вестник Российского университета дружбы народов. Серия: Инженерные исследования. – 2001. – № 1. – С. 17-21.
58. Wang, Y. Deep learning-based time series forecasting / Y. Wang, J. Wang, Z. Cao, G. Barroso // Artificial Intelligence Review. – 2025. – Vol. 58, No. 1. – Art. 23. – DOI: 10.1007/s10462-024-10989-8.
59. Ahmed, M. J. A survey on graph neural networks, machine learning and deep learning techniques for time series applications in industry / M. J. Ahmed, A. Mozo, A. Karamchandani // PeerJ Computer Science. – 2025. – Vol. 11. – Art. e3097. – DOI: 10.7717/peerj-cs.3097.
60. Li, W. Deep Learning Models for Time Series Forecasting: A Review / W. Li, K. L. E. Law // IEEE Access. – 2024. – Vol. 12. – P. 92306-92327. – DOI: 10.1109/ACCESS.2024.3422528.
61. Wang, Y. Neural Network Approaches to Temporal Pattern Recognition: Applications in Demand Forecasting and Predictive Analytics / Y. Wang, S. Qiu, Z. Chen // Journal of Banking and Financial Dynamics. – 2025. – Vol. 9, No. 11. – P. 19-32.
62. Рысин, Н. А. Разработка высоконагруженного решения прогнозирования временных рядов / Н. А. Рысин // Хроноэкономика. – 2020. – № 4(25). – С. 82-87.
63. Bukina, T. V. Regional Inflation Forecasting: Econometric Models Versus Machine Learning Methods? / T. V. Bukina, D. V. Kashin // HSE Economic Journal. – 2024. – Vol. 28, No. 1. – P. 81-107. – DOI: 10.17323/1813-8691-2024-28-1-81-107.
64. Kock, A. B. Penalized Time Series Regression / A. B. Kock, M. Medeiros, G. Vasconcelos // Macroeconomic Forecasting in the Era of Big Data. – Cham : Springer, 2020. – P. 193-228.

65. Babii, A. Machine Learning Time Series Regressions With an Application to Nowcasting / A. Babii, E. Ghysels, J. Striaukas // Journal of Business & Economic Statistics. – 2022. – Vol. 40. – P. 1094-1106.

66. Красоткина, О. В. Методы регуляризации в задаче восстановления нестационарной регрессионной зависимости / О. В. Красоткина, В. В. Моттль // Труды Тульского государственного университета. – 2024.

67. Breiman, L. Classification and Regression Trees / L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone. – Belmont : Wadsworth International Group, 1984. – 368 p.

68. Темурбекова, Г. К. Эффективные модели для предсказания временных рядов / Г. К. Темурбекова, Ш. А. Бобоев // XIII Ломоносовские чтения : материалы международной научно-практической конференции, посвященной 115-летию академика Бободжона Гафурова (28-29 апреля 2023 года) / под редакцией Ф. А. Косимзода. – Душанбе : Филиал МГУ имени М.В. Ломоносова в городе Душанбе, 2023. – Ч. III: Естественные науки. – С. 113-120.

69. Бучацкая, В. В. Web-сервис для решения задачи прогнозирования / В. В. Бучацкая, П. Ю. Бучацкий, В. Е. Лобанов // КиберЛенинка. – 2024. – URL: <https://cyberleninka.ru/article/n/web-servis-dlya-resheniya-zadachi-prognozirovaniya>.

70. Ozdemir, O. Forecasting performance of machine learning, time series, and hybrid methods for low- and high-frequency time series / O. Ozdemir, C. Yozgatligil // Statistica Neerlandica. – 2024. – Vol. 78, No. 2. – P. 441-474.

71. Smola, A. A tutorial on support vector regression / A. Smola, B. Schölkopf // Statistics and Computing. – 2004. – Vol. 14, No. 3. – P. 199-222.

72. Vapnik, V. The Nature of Statistical Learning Theory / V. Vapnik. – 2nd ed. – New York : Springer, 2000. – 314 p.

73. Гуляев, А. В. Гибридная модель глубокого обучения для прогнозирования значений временных рядов / А. В. Гуляев // Нейрокомпьютеры: разработка, применение. – 2025. – Т. 27, № 3. – С. 62-72.

74. Салтанаева, Е. А. Сравнение традиционных методов машинного обучения и глубокого обучения / Е. А. Салтанаева, А. А. Шакиров, А. Р. Гимаева // Научно-технический вестник Поволжья. – 2023. – № 12. – С. 379-381.
75. Шапалин, В. Г. Сравнение структуры, эффективности и скорости работы полносвязных, сверточных и рекуррентных нейронных сетей / В. Г. Шапалин, Д. В. Николаенко // Научный результат. Информационные технологии. – 2024. – Т. 9, № 4. – С. 21-35.
76. Hochreiter, S. Untersuchungen zu dynamischen neuronalen Netzen : Diplomarbeit / S. Hochreiter ; Technische Universität München. – München, 1991.
77. Bengio, Y. Learning long-term dependencies with gradient descent is difficult / Y. Bengio, P. Simard, P. Frasconi // IEEE Transactions on Neural Networks. – 1994. – Vol. 5, No. 2. – P. 157-166. – DOI: 10.1109/72.279181.
78. Zhang, A. Dive into Deep Learning / A. Zhang, Z. C. Lipton, M. Li, A. J. Smola. – Cambridge : Cambridge University Press, 2024. – 574 p.
79. Al-Masri, A. N. Long Short-Term Memory Networks: A Comprehensive Survey / A. N. Al-Masri, T. Guesmi, B. M. Alshammari, A. Alzamil, S. Alshammari, S. Albogami // AI. – 2025. – Vol. 6, No. 9. – P. 215.
80. Nature Scientific Reports. Table 2: Comparison between the three methods (ANN, SVM, LSTM) used in EHAD // Scientific Reports. – 2025.
81. Lazcano, A. Back to Basics: The Power of the Multilayer Perceptron in Financial Time Series Forecasting / A. Lazcano, M. A. Jaramillo-Morán, J. E. Sandubete // Mathematics. – 2024. – Vol. 12, No. 12. – P. 1920.
82. Hornik, K. Multilayer feedforward networks are universal approximators / K. Hornik, M. Stinchcombe, H. White // Neural Networks. – 1989. – Vol. 2, No. 5. – P. 359-366. – DOI: 10.1016/0893-6080(89)90020-8.
83. Xie, R. The Analysis of Time Series Forecasting Based on MLP Models / R. Xie // Transactions on Computer Science and Intelligent Systems Research. – 2024. – Vol. 5. – DOI: 10.62051/qmrw8m76.

84. Chen, S. A. TSMixer: An All-MLP Architecture for Time Series Forecasting / S. A. Chen, C. L. Li, N. Yoder, S. O. Arik, T. Pfister // arXiv preprint arXiv:2303.06053. – 2023.

85. Sohn J., Robertazzi T.G., Luryi S., “Optimizing computing costs using divisible load analysis”, IEEE Transactions on Parallel and Distributed Systems, 9:3 (1998), 225–234.

86. Мингалиев, З. З. Метод расчета переменного шага для кусочно-линейной аппроксимации эмпирической нелинейной функции с заданной точностью / З. З. Мингалиев, С. В. Новикова, Г. В. Моисеев // Вестник Тверского государственного университета. Серия: Прикладная математика. – 2022. – № 3. – С. 35-48.

87. Мингалиев, З. З. Метод адаптивного прогнозирования данных, зависящих от времени, на основе LSTM-моделей с использованием кластеризации временных рядов / З. З. Мингалиев // Вестник Технологического университета. – 2025. – Т. 28, № 2. – С. 70-78.

88. Старовойтов, В. В. Нормализация данных в машинном обучении / В. В. Старовойтов, Ю. И. Голуб // Информатика. – 2021. – Т. 18, № 3. – С. 83-96.

89. Мингалиев, З. З. Адаптивное прогнозирование временных рядов на основе кластеризации и нейросетевых моделей LSTM / З. З. Мингалиев, С. В. Новикова // Цифровые системы и модели: теория и практика проектирования, разработки и использования : Материалы международной научно-практической конференции, Казань, 10–11 апреля 2025 года. – Казань: Казанский государственный энергетический университет, 2025. – С. 1635-1639.

90. Мингалиев, З. З. Программный комплекс для адаптивного прогнозирования временных рядов / З. З. Мингалиев // Профессиональные коммуникации в научной среде - фактор обеспечения качества исследований : сборник материалов XV Всероссийской научно-практической конференции,

Альметьевск, 16 апреля 2025 года. – Санкт-Петербург: ООО Издательский дом "Сциентиа", 2025. – С. 623-626.

91. Мингалиев, З. З. Программная реализация адаптивного модуля аппроксимации / З. З. Мингалиев // XXVI Туполевские чтения (школа молодых ученых) : Материалы Международной молодёжной научной конференции. Сборник докладов, Казань, 09–10 ноября 2023 года. – Казань: ИП Сагиев А.Р., 2023. – С. 3138-3141.

92. Свидетельство о государственной регистрации программы для ЭВМ № 2022665809 Российская Федерация. Адаптивный модуль аппроксимации: № 2022664687: заявл. 08.08.2022: опубл. 22.08.2022 / З. З. Мингалиев; заявитель федеральное государственное бюджетное образовательное учреждение высшего образования «Казанский национальный исследовательский технический университет им. А.Н. Туполева – КАИ».

93. Кормен, Т. Алгоритмы: построение и анализ / Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн; перевод с английского. – 3-е изд. – Москва: Вильямс, 2013. – 1328 с.

94. Свидетельство о государственной регистрации программы для ЭВМ № 2025661819 Российская Федерация. Программа синтаксического анализа больших данных биологического мониторинга: заявл. 13.05.2025: опубл. 13.05.2025 / З. З. Мингалиев; заявитель Федеральное государственное бюджетное образовательное учреждение высшего образования «Казанский национальный исследовательский технический университет им. А.Н. Туполева-КАИ».

95. Мингалиев, З. З. Программный комплекс для прогнозирования смешанных временных рядов с использованием кластеризации / З. З. Мингалиев // Вестник Технологического университета. – 2025. – Т. 28, № 3. – С. 103-110.

96. Свидетельство о государственной регистрации программы для ЭВМ № 2025661810 Российская Федерация. Программа кластеризации и адаптивного прогнозирования временных рядов на основе LSTM : заявл.

13.05.2025 : опублик. 13.05.2025 / З. З. Мингалиев ; заявитель Федеральное государственное бюджетное образовательное учреждение высшего образования «Казанский национальный исследовательский технический университет им. А.Н. Туполева-КАИ».

97. Qiu Suming, Cui Weicheng. An Overview on Aquatic Unmanned Aerial Vehicles. *Ann Rev Resear.* 2019; 5(3): 555663.

98. Pinheiro, P.M., Neto, A.A., Grando, R.B. et al. Trajectory Planning for Hybrid Unmanned Aerial Underwater Vehicles with Smooth Media Transition. *J Intell Robot Syst* 104, 46 (2022).

99. Новикова, С. В. Нейросетевое прогнозирование гликемии у пациентов с сахарным диабетом на основе смешанных временных рядов с перспективой применения в составе интеллектуальной инсулиновой помпы / С. В. Новикова, З. З. Мингалиев. // *Современные информационные технологии и ИТ-образование.* – 2021. – Т. 17, № 1. – С. 90-98.

100. Types of Insulin [Электронный ресурс]. URL: <https://www.cdc.gov/diabetes/basics/type-1-types-of-insulin.html>.

101. Toffanin, Ch. Dynamic insulin on board: incorporation of circadian insulin sensitivity variation / Ch. Toffanin [и др.]. – DOI: 10.1177/193229681300700415 // *Journal of Diabetes Science and Technology.* – 2013. – Vol. 7, issue 4. – Pp. 928-940.

102. Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford (2001). "16 Greedy Algorithms". *Introduction To Algorithms.* MIT Press. pp. 370. ISBN 978-0-262-03293-3.

103. Wu B, Guo Y, Zheng J, Li Z, Gong J, Hui N, Han X. Research on the Stability of UAV Attitude Under Hybrid Control Integrating Active Disturbance Rejection Control and Super-Twisting Sliding Mode Control. *Applied Sciences.* 2025; 15(9):5124.

104. N. P. Nguyen, N. X. Mung, H. L. N. N. Thanh, T. T. Huynh, N. T. Lam and S. K. Hong, "Adaptive Sliding Mode Control for Attitude and Altitude System

of a Quadcopter UAV via Neural Network," in IEEE Access, vol. 9, pp. 40076-40085, 2021.

105. Hoang Van Truong, Singh Ansu, Phung Manh Duong, Ha Quang. (2017). Adaptive Second-order Sliding Mode Control of UAVs for Civil Applications. 2017 Proceedings of the 34rd ISARC, Taipei, Taiwan, Pages 816-822,

106. Sattar A, Wang L, Hoshu AA, Ansari S, Karar H-e, Mohamed A. Automatic Tuning and Turbulence Mitigation for Fixed-Wing UAV with Segmented Control Surfaces. Drones. 2022; 6(10):302. <https://doi.org/10.3390/drones6100302>.

107. William A. Gardner, Introduction to Random Processes with Applications to Signals and Systems, - 2nd ed., McGraw-Hill, Inc., New York, 1990, p.560.

108. Qian, H., Ge, H. (2013). Stochastic Processes, Fokker-Planck Equation. In: Dubitzky, W., Wolkenhauer, O., Cho, KH., Yokota, H. (eds) Encyclopedia of Systems Biology. Springer, New York, NY. https://doi.org/10.1007/978-1-4419-9863-7_279

109. Rodnishchev, N. E. Optimal control of nonlinear stochastic systems under constraints: An approximate determination method / N. E. Rodnishchev // Automation and Remote Control. – 2001. – Vol. 62, No. 3. – P. 401-408. – DOI: 10.1023/A:1002854110800.

110. A. Y. Morozov, O. A. Kuzenkov, and S. K. Sandhu, "Global optimisation in hilbert spaces using the survival of the fittest algorithm," Commun. Nonlinear Sci. Numer. Simul., vol. 103, p. 106007, 2021.

111. O. A. Kuzenkov, A. Y. Morozov, and S. A. Nalchajyan, "Revisiting 'survival of the fittest' principle in global stochastic optimisation: Incorporating anisotropic mutations," Commun. Nonlinear Sci. Numer. Simul., vol. 130, p. 107768, 2024.

112. Liguo TAN, Yongcheng XIONG, Changqing HU, Jianfeng LI, Oleg KUZENKOV, Samvel NALCHAJYAN, Parametric control of UAV U-turns in turbulent wind conditions based on global optimization, Chinese Journal of Aeronautics, 2025, 103746.

113. European Union Aviation Safety Agency (EARforCS-AWOIssue 2):
URL: <https://www.easa.europa.eu/en/document-library/easy-access-rules/easy-access-rules-all-weather-operations-cs-awo>.
114. Ummul Khair et al 2017 J. Phys.: Conf. Ser. 930 012002.
115. Kasun Bandara, Christoph Bergmeir, Slawek Smyl Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach. *Expert Systems with Applications*, Vol. 140, 2020, 112896, <https://doi.org/10.1016/j.eswa.2019.112896>.
116. S. Siami-Namini, N. Tavakoli and A. Siami Namin, "A Comparison of ARIMA and LSTM in Forecasting Time Series," 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, USA, 2018, pp. 1394-1401, doi: 10.1109/ICMLA.2018.00227.
117. Prater, R., Hanne, T., & Dornberger, R. (2024). Generalized Performance of LSTM in Time-Series Forecasting. *Applied Artificial Intelligence*, 38(1). <https://doi.org/10.1080/08839514.2024.2377510>.
118. Puchkin, Nikita & Timofeev, Aleksandr & Spokoiny, Vladimir. (2020). Manifold-based time series forecasting. 10.48550/arXiv.2012.08244.

ПРИЛОЖЕНИЯ

УТВЕРЖДАЮ

Директор ГБУ «Безопасность
дорожного движения»,
Президент Академии наук Республики
Татарстан,
д.т.н., профессор

 / Р.Н. Минниханов /

«10» мая 2026 г.

АКТ

об использовании результатов диссертационной работы

МИНГАЛИЕВА ЗАИДА ЗУЛЬФАТОВИЧА

в научные исследования Центра разработки и сопровождения
информационных систем ГБУ «Безопасность дорожного движения»

Настоящий акт подтверждает, что результаты диссертационной работы Мингалиева З.З. использованы при реализации научного направления деятельности Центра разработки и сопровождения информационных систем ГБУ «Безопасность дорожного движения» при разработке научных и практических основ моделирования транспортных потоков. Разработанные Мингалиевым З.З. модели, методы и программные средства аппроксимации и прогнозирования неоднородных временных рядов использованы для описания и прогноза движения транспортных средств по сложным траекториям в различных дорожных условиях.

Начальник центра разработки
и сопровождения информационных
систем ГБУ БДД,
кандидат технических наук



Дагаева Мария Витальевна

Государственное бюджетное учреждение «Безопасность дорожного движения», Адрес: 420059, г. Казань, Оренбургский тракт, 5. Телефон: +7 (843) 210-18-90. E-Mail: Priemnaya.Minnihanov@tatar.ru 420087, сайт организации: <https://gbubdd.tatarstan.ru/>

«УТВЕРЖДАЮ»

И.о. проректора
по образовательной деятельности
и молодежной политике

В.В. Хамматова

2026 г.

АКТ

о внедрении в учебный процесс университета
результатов кандидатской диссертации аспиранта
кафедры прикладной математики и информатики
Мингалиева Заида Зульфатовича

Мы, нижеподписавшиеся, заведующий кафедрой прикладной математики и информатики, канд. техн. наук, доцент Зайдуллин С.С. и директор института компьютерных технологий и защиты информации, канд. техн. наук, доцент Трегубов В.М., составили настоящий акт о том, что полученные аспирантом кафедры прикладной математики и информатики Мингалиевым З.З. результаты кандидатской диссертации внедрены в учебный процесс КНИТУ-КАИ.

Предложенные в диссертации методы, алгоритмы и программный комплекс адаптивного прогнозирования неоднородных временных рядов в условиях ограниченных вычислительных ресурсов используются на кафедре прикладной математики и информатики для подготовки бакалавров по направлению 01.03.02 «Прикладная математика и информатика» при проведении лекционных занятий и лабораторных работ по дисциплинам «Компьютерное моделирование процессов и систем», «Теория принятия решений» и «Теория игр и исследование операций».

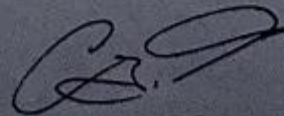
Кроме того, результаты диссертационного исследования Мингалиева З.З. используются на кафедре при выполнении обучающимися выпускных квалификационных, курсовых и научно-исследовательских работ.

/Директор института компьютерных
технологий и защиты информации,
канд. техн. наук, доцент



В.М. Трегубов

Заведующий кафедрой прикладной
математики и информатики,
канд. техн. наук, доцент



С.С. Зайдуллин

УТВЕРЖДАЮ


М.Х.Шакиров

«22» мая 2026 г.



АКТ
об использовании результатов кандидатской диссертации
Мингалиева Заида Зульфатовича

Настоящим подтверждаем, что результаты диссертационного исследования Мингалиева Заида Зульфатовича обладают актуальностью, представляют практический интерес и были использованы для имитации сложных траекторий движения и моделирования управления беспилотных летательных аппаратов в работе научно-производственного центра беспилотных авиационных систем (НПЦ БАС) Татарстана.

МИНИСТЕРСТВО ТРАНСПОРТА
И ДОРОЖНОГО ХОЗЯЙСТВА
РЕСПУБЛИКИ ТАТАРСТАН
Н.Ершова ул., д. 31а, г. Казань, 420061



ТАТАРСТАН РЕСПУБЛИКАСЫ
ТРАНСПОРТ БӨМ ЮЛ
ХУЖАЛЫГЫ МИНИСТРЛЫГЫ
Н.Ершова ур., 31иче а йорт, Казан ш., 420061

Тел.: (843) 291-90-10; E-mail: mindotrans@tatar.ru

№ _____
На № _____ от _____

УТВЕРЖДАЮ
Начальнику Управления
автомобильных дорог министерства
транспорта и дорожного хозяйства
Республики Татарстан
О.А. Казеннов



АКТ
об использовании результатов кандидатской диссертации
Мингалиева Заида Зульфатовича

Дан Мингалиеву Заиду Зульфатовичу в том, что результаты его диссертационной работы использованы в работе Сектора безопасности дорожного движения и контрольно-надзорной деятельности Министерства транспорта и дорожного хозяйства Республики Татарстан в системе поддержки принятия решений для моделирования траектории движения транспортных потоков на дорожных развязках сложных конфигураций на территории Республики Татарстан.

Заведующий сектора
безопасности дорожного движения и
контрольно-надзорной деятельности

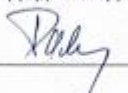
И.В. Ермаков



ГАОУЗ «ДЕТСКАЯ
РЕСПУБЛИКАНСКАЯ
КЛИНИЧЕСКАЯ
БОЛЬНИЦА
МИНИСТЕРСТВА
ЗДРАВООХРАНЕНИЯ
РЕСПУБЛИКИ
ТАТАРСТАН»

УТВЕРЖДАЮ

Заведующая отделением эндокринологии
Детской республиканской
клинической больницы
Министерства здравоохранения
Республики Татарстан
Кандидат медицинских наук, доцент

 /Шайдуллина М.Р./

«01» ~~сентября~~ 2026 г.

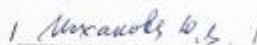
АКТ

об использовании результатов кандидатской диссертации
Мингалиева Заида Зульфатовича

Дан Мингалиеву Заиду Зульфатовичу в том, что результаты его диссертационного исследования использованы в работе отделения эндокринологии в системе поддержки принятия решений для прогнозирования уровня гликемии у больных инсулинозависимым диабетом, получающих помповую инсулинотерапию.

Врач-эндокринолог
Отделения эндокринологии
ДРКБ МЗ РТ



1  1



ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ
ГОСУДАРСТВЕННАЯ РЕГИСТРАЦИЯ ПРОГРАММЫ ДЛЯ ЭВМ

Номер регистрации (свидетельства): 2022665809 Дата регистрации: 22.08.2022 Номер и дата поступления заявки: 2022664687 08.08.2022 Дата публикации и номер бюллетеня: 22.08.2022 Бюл. № 9	Автор(ы): Мингалиев Заид Зульфатович (RU) Правообладатель(и): федеральное государственное бюджетное образовательное учреждение высшего образования «Казанский национальный исследовательский технический университет им. А.Н. Туполева – КАИ» (RU)
--	---

Название программы для ЭВМ:
Адаптивный модуль аппроксимации

Реферат:

Программа предназначена для расчета переменного шага для кусочно-линейной аппроксимации эмпирической нелинейной функции с заданной точностью. Точность определяется как значение коэффициента линейной модели на каждом выделенном в процессе работы программы сегменте. Предполагается использование данной программы для аппроксимации функции, отражающей динамику концентрации в плазме крови экзогенного инсулина, который является основополагающим параметром при вычислении потребных доз препарата при интенсивной персонализированной инсулинотерапии. Программа предоставляет удобный оконный пользовательский интерфейс для ввода данных в формате *xlsx* для дальнейшей обработки и табличного вывода данных о формируемых сегментах (номер сегмента, индекс начального элемента сегмента, индекс конечного элемента сегмента, коэффициенты линейной модели, коэффициент детерминации). Тип ЭВМ: IBM PC-совмест. ПК; ОС: Windows 10 Professional.

Язык программирования: C#
Объем программы для ЭВМ: 1,3 МБ



ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ
ГОСУДАРСТВЕННАЯ РЕГИСТРАЦИЯ ПРОГРАММЫ ДЛЯ ЭВМ

Номер регистрации (свидетельства):
2025661819
Дата регистрации: 13.05.2025
Номер и дата поступления заявки:
2025661093 13.05.2025
Дата публикации и номер бюллетеня:
13.05.2025 Бюл. № 5

Автор(ы):
Мингалиев Заид Зульфатович (RU)
Правообладатель(и):
Федеральное государственное бюджетное
образовательное учреждение высшего
образования «Казанский национальный
исследовательский технический университет
им. А.Н. Туполева-КАИ» (RU)

Название программы для ЭВМ:

Программа синтаксического анализа больших данных биологического мониторинга

Реферат:

Программа предназначена для автоматической обработки и анализа больших объемов данных, полученных с сенсоров уровня глюкозы в крови. Основная функция системы – загрузка файла с данными в формате JSON, извлечение ключевых показателей (SGV, IOB Sum, COB) и формирование структурированного набора данных для последующего анализа или хранения. Алгоритм извлекает массив записей показателей глюкозы и дополняет их сопутствующими параметрами, такими как активный инсулин и углеводы, на основе дополнительных JSON-блоков с данными о состоянии пациента. Встроенный механизм сериализации и десериализации позволяет обрабатывать даже частично структурированные данные с потенциальными пропусками. Итоговый результат сохраняется в формате Excel. Программа позволяет интегрировать разрозненные биомедицинские данные в единый формат.

Язык программирования: C#

Объем программы для ЭВМ: 6 Кб



ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ
ГОСУДАРСТВЕННАЯ РЕГИСТРАЦИЯ ПРОГРАММЫ ДЛЯ ЭВМ

Номер регистрации (свидетельства):
2025661810
Дата регистрации: 13.05.2025
Номер и дата поступления заявки:
2025661090 13.05.2025
Дата публикации и номер бюллетеня:
13.05.2025 Бюл. № 5

Автор(ы):
Мингалиев Заид Зульфатович (RU)
Правообладатель(и):
Федеральное государственное бюджетное
образовательное учреждение высшего
образования «Казанский национальный
исследовательский технический университет
им. А.Н. Туполева-КАИ» (RU)

Название программы для ЭВМ:

Программа кластеризации и адаптивного прогнозирования временных рядов на основе LSTM

Реферат:

Программа предназначена для кластеризации и прогнозирования временных рядов с использованием алгоритма DBSCAN и моделей LSTM. Кластеризация осуществляется не над самими временными рядами, а над их параметризованными представлениями, что позволяет учитывать ключевые статистические и спектральные особенности данных и улучшать точность прогнозирования. Алгоритм DBSCAN позволяет выделить группы схожих данных, не требуя заранее задавать количество кластеров. Для каждого кластера обучается отдельная модель LSTM, что повышает адаптивность и точность прогнозов. Входной временной ряд сначала параметризуется и классифицируется в один из сформированных кластеров, после чего используется соответствующая модель для прогнозирования. Система реализована через API-контроллер на основе ASP.NET Core, интегрированный с Python.

Язык программирования: C++, Python

Объем программы для ЭВМ: 29 Кб



ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ
ГОСУДАРСТВЕННАЯ РЕГИСТРАЦИЯ ПРОГРАММЫ ДЛЯ ЭВМ

Номер регистрации (свидетельства): 2025661805 Дата регистрации: 13.05.2025 Номер и дата поступления заявки: 2025661082 13.05.2025 Дата публикации и номер бюллетеня: 13.05.2025 Бюл. № 5	Автор(ы): Мингалиев Заид Зульфатович (RU) Правообладатель(и): Федеральное государственное бюджетное образовательное учреждение высшего образования «Казанский национальный исследовательский технический университет им. А.Н. Туполева-КАИ» (RU)
--	---

Название программы для ЭВМ:

Программа анализа и прогнозирования смешанных временных рядов

Реферат:

Программа представляет собой интеллектуальную систему для персонализированного прогнозирования уровня глюкозы в крови на основе временных рядов. Комплекс автоматически обрабатывает данные с сенсоров (SGV, IOB, COB), выполняет их очистку, интерполяцию, кластеризацию методом DBSCAN и формирует входные данные для модели прогнозирования на основе нейросетевой модели LSTM. Система адаптируется под индивидуальные особенности пациента, учитывая физиологическое состояние и поведенческие факторы, что обеспечивает высокую точность предсказаний. Комплекс реализован в виде API-контроллера, легко интегрируется в мобильные и веб-приложения и может использоваться как медицинскими специалистами, так и самими пациентами для мониторинга и управления диабетом.

Язык программирования: C#, Python

Объем программы для ЭВМ: 27 Кб