

Школа: Инженерная школа информационных технологий и робототехники
 Направление подготовки: 09.03.04 «Программная инженерия»
 Отделение школы (НОЦ): Отделение информационных технологий

БАКАЛАВРСКАЯ РАБОТА

| Тема работы |
|--|
| Алгоритмическое и программное обеспечение интеллектуального анализа данных о состоянии промышленных манипуляторов |

УДК 004.421:004.415.2:004.896:007.52

Студенты

| Группа | ФИО | Подпись | Дата |
|--------|---------------------------|---------|------|
| 8К71 | Писанкин Андрей Сергеевич | | |
| 8К71 | Чепкасов Артём Юрьевич | | |

Руководитель ВКР

| Должность | ФИО | Ученая степень, звание | Подпись | Дата |
|-----------|---------------------------|------------------------|---------|------|
| Доцент | Савельев Алексей Олегович | к.т.н. | | |

КОНСУЛЬТАНТЫ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

| Должность | ФИО | Ученая степень, звание | Подпись | Дата |
|------------------|-------------------------------|------------------------|---------|------|
| Доцент ОСГН ШБИП | Маланина Вероника Анатольевна | к.э.н. | | |

По разделу «Социальная ответственность»

| Должность | ФИО | Ученая степень, звание | Подпись | Дата |
|--------------------|-----------------------------|------------------------|---------|------|
| Ассистент ООД ШБИП | Черемискина Мария Сергеевна | - | | |

ДОПУСТИТЬ К ЗАЩИТЕ:

| Руководитель ООП | ФИО | Ученая степень, звание | Подпись | Дата |
|------------------|-----------------------------|------------------------|---------|------|
| Доцент ОИТ ИШИТР | Чердынцев Евгений Сергеевич | к.т.н. | | |

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ООП

| Код компетенции | Наименование компетенции |
|---|--|
| Общекультурные (универсальные) компетенции | |
| УК-1 | Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач |
| УК-2 | Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений |
| УК-3 | Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде |
| УК-4 | Способен осуществлять деловую коммуникацию в устной и письменной формах на государственном языке Российской Федерации и иностранном(-ых) языке(-ах) |
| УК-5 | Способен воспринимать межкультурное разнообразие общества в социально-историческом, этическом и философском контекстах |
| УК-6 | Способен управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни |
| УК-7 | Способен поддерживать должный уровень физической подготовленности для обеспечения полноценной социальной и профессиональной деятельности; |
| УК-8 | Способен создавать и поддерживать безопасные условия жизнедеятельности, в том числе при возникновении чрезвычайных ситуаций |
| Общепрофессиональные компетенции | |
| ОПК-1 | Способен применять естественнонаучные и общепрофессиональные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности |
| ОПК-2 | Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности |
| ОПК-3 | Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности |
| ОПК-4 | Готов собирать, обрабатывать, анализировать и систематизировать научно-техническую информацию по тематике исследования, использовать достижения отечественной и зарубежной науки, техники и технологии в своей профессиональной деятельности |
| ОПК-5 | Способен использовать основы экономических знаний при оценке эффективности результатов своей профессиональной деятельности |
| Профессиональные компетенции | |
| ПК-1 | Применять базовые и специальные естественнонаучные и математические знания в области информатики и вычислительной техники, достаточные для комплексной инженерной деятельности. |
| ПК-2 | Применять базовые и специальные знания в области современных информационных технологий для решения инженерных задач. |

| | |
|-------------|---|
| ПК-3 | Ставить и решать задачи комплексного анализа, связанные с созданием аппаратнопрограммных средств информационных и автоматизированных систем, с использованием базовых и специальных знаний, современных аналитических методов и моделей. |
| ПК-4 | Разрабатывать программные и аппаратные средства (системы, устройства, блоки, программы, базы данных и т.п.) в соответствии с техническим заданием и с использованием средств автоматизации проектирования. |
| ПК-5 | Проводить теоретические и экспериментальные исследования, включающие поиск и изучение необходимой научно-технической информации, математическое моделирование, проведение эксперимента, анализ и интерпретацию полученных данных, в области создания аппаратных и программных средств информационных и автоматизированных систем. |
| ПК-6 | Внедрять, эксплуатировать и обслуживать современные программно-аппаратные комплексы, обеспечивать их высокую эффективность, соблюдать правила охраны здоровья, безопасность труда, выполнять требования по защите окружающей среды. |

Школа: Инженерная школа информационных технологий и робототехники
 Направление подготовки: 09.03.04 «Программная инженерия»
 Отделение школы (НОЦ): Отделение информационных технологий

УТВЕРЖДАЮ:
 Руководитель ООП
 _____ Чердынцев Е.С.
 (Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ на выполнение выпускной квалификационной работы

В форме:

| |
|--|
| Бакалаврской работы |
| (бакалаврской работы, дипломного проекта/работы, магистерской диссертации) |

Студентам:

| Группа | ФИО |
|--------|-----------------------------|
| 8K71 | Писанкину Андрею Сергеевичу |
| 8K71 | Чепкасову Артёму Юрьевичу |

Тема работы:

| | |
|---|-----------------------|
| Алгоритмическое и программное обеспечение интеллектуального анализа данных о состоянии промышленных манипуляторов | |
| Утверждена приказом директора (дата, номер) | №32-2/с от 01.02.2021 |

| | |
|--|---------------|
| Срок сдачи студентом выполненной работы: | 16.06.2021 г. |
|--|---------------|

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

| | |
|--|---|
| <p>Исходные данные к работе <i>(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).</i></p> | <p>Работа направлена на создание программного обеспечения интеллектуального анализа данных о состоянии промышленных манипуляторов. Результатом является программное обеспечение для диагностики текущего состояния манипулятора, включая ситуации сбоев и отказов.</p> |
| <p>Перечень подлежащих исследованию, проектированию и разработке вопросов <i>(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).</i></p> | <ol style="list-style-type: none"> 1. Исследование предметной области 2. Организация командной разработки ПО 3. Проектирование системы 4. Реализация системы 5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение 6. Социальная ответственность |

| | |
|---|--|
| Перечень графического материала <i>(с точным указанием обязательных чертежей)</i> | <ol style="list-style-type: none"> 1. Диаграмма вариантов использования в нотации UML 2. Модель базы данных 3. Диаграммы последовательности 4. Рисунки, демонстрирующие результаты 5. Диаграмма Ганта |
|---|--|

| | |
|--|-------------------------------|
| Консультанты по разделам выпускной квалификационной работы <i>(с указанием разделов)</i> | |
| Раздел | Консультант |
| Финансовый менеджмент, ресурсоэффективность и ресурсосбережение | Маланина Вероника Анатольевна |
| Социальная ответственность | Черемискина Мария Сергеевна |

| | |
|---|---------------|
| Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику | 01.03.2021 г. |
|---|---------------|

Задание выдал руководитель:

| Должность | ФИО | Ученая степень, звание | Подпись | Дата |
|------------------|---------------------------|------------------------|---------|------|
| Доцент ОИТ ИШИТР | Савельев Алексей Олегович | к.т.н | | |

Задание приняли к исполнению студенты:

| Группа | ФИО | Подпись | Дата |
|--------|---------------------------|---------|------|
| 8K71 | Писанкин Андрей Сергеевич | | |
| 8K71 | Чепкасов Артём Юрьевич | | |

Школа: Инженерная школа информационных технологий и робототехники
 Направление подготовки: 09.03.04 «Программная инженерия»
 Отделение школы (НОЦ): Отделение информационных технологий
 Период выполнения: весенний семестр 2021 учебного года

Форма представления работы:

Бакалаврская работа

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН выполнения выпускной квалификационной работы

| | |
|---|------------|
| Срок сдачи студентами выполненной работы: | 16.06.2021 |
|---|------------|

| Дата контроля | Название раздела (модуля) / вид работы (исследования) | Максимальный балл раздела (модуля) |
|---------------|---|------------------------------------|
| 04.02.2021 | Раздел 1. Описание предметной области | 15 |
| 25.02.2021 | Раздел 2. Организация командной разработки ПО | 10 |
| 28.04.2021 | Раздел 3. Проектирование системы | 20 |
| 18.05.2021 | Раздел 4. Реализация системы | 35 |
| 20.05.2021 | Раздел 5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение | 10 |
| 25.05.2021 | Раздел 6. Социальная ответственность | 10 |

Составил преподаватель:

| Должность | ФИО | Ученая степень, звание | Подпись | Дата |
|------------------|---------------------------|------------------------|---------|------|
| Доцент ОИТ ИШИТР | Савельев Алексей Олегович | К.Т.Н | | |

СОГЛАСОВАНО:

| Руководитель ООП | ФИО | Ученая степень, звание | Подпись | Дата |
|------------------|-----------------------------|------------------------|---------|------|
| Доцент ОИТ ИШИТР | Чердынцев Евгений Сергеевич | К.Т.Н | | |

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студентам:

| Группа | ФИО |
|--------|-----------------------------|
| 8К71 | Писанкину Андрею Сергеевичу |
| 8К71 | Чепкасову Артёму Юрьевичу |

| Школа | ИШИТР | Отделение школы (НОЦ) | ОИТ |
|---------------------|-------------|---------------------------|--|
| Уровень образования | Бакалавриат | Направление/специальность | 09.03.04 «Программная Инженерия» |

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

| | |
|--|---|
| 1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих | Бюджет проекта не более 393446 руб., в том числе затраты на оплату труда не более 250860 руб. |
| 2. Нормы и нормативы расходования ресурсов | Значение показателя интегральной ресурсоэффективности - не менее 60 баллов из 100. |
| 3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования | Районный коэффициент – 1,3 Коэффициент дополнительной заработной платы – 0,13 Коэффициент отчислений во внебюджетные фонды – 0,302 Коэффициент накладных расходов – 0,16 |

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

| | |
|--|--|
| 1. Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения | 1. Описание потребителей продукта 2. QuaD-анализ 3. SWOT-анализ |
| 2. Планирование и формирование бюджета научных исследований | 1. Описание структуры работ в рамках научного исследования. 2. Определение трудоемкости выполнения работ и разработка графика проведения научного исследования. 3. Подсчет бюджета проекта |
| 3. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования | Оценка сравнительной эффективности исследования |

Перечень графического материала (с точным указанием обязательных чертежей):

| |
|--|
| 1. Оценка конкурентоспособности технических решений |
| 2. Матрица SWOT |
| 3. График проведения и бюджет НИ |
| 4. Оценка ресурсной, финансовой и экономической эффективности НИ |

| | |
|--|------------|
| Дата выдачи задания для раздела по линейному графику | 01.03.2021 |
|--|------------|

Задание выдал консультант:

| Должность | ФИО | Ученая степень, звание | Подпись | Дата |
|---------------------|----------------------------------|---------------------------|---------|------|
| Доцент ОСГН ШБИП | Маланина Вероника Анатольевна | к.э.н. | | |

Задание принял к исполнению студент:

| Группа | ФИО | Подпись | Дата |
|--------|---------------------------|---------|------|
| 8К71 | Писанкин Андрей Сергеевич | | |
| 8К71 | Чепкасов Артём Юрьевич | | |

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»**

Студентам:

| Группа | ФИО |
|--------|-----------------------------|
| 8К71 | Писанкину Андрею Сергеевичу |
| 8К71 | Чепкасову Артёму Юрьевичу |

| Школа | ИШИТР | Отделение (НОЦ) | ОИТ |
|---------------------|-------------|---------------------------|-----------------------------------|
| Уровень образования | Бакалавриат | Направление/специальность | 09.03.04 Программная инженерия |

Тема ВКР:

| Алгоритмическое и программное обеспечение интеллектуального анализа данных о состоянии промышленных манипуляторов | |
|---|--|
| Исходные данные к разделу «Социальная ответственность»: | |
| 1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения | Объект исследования – программное обеспечение интеллектуального анализа данных о состоянии промышленных манипуляторов. Область применения – мониторинг и контроль промышленных роботизированных систем. Рабочее место – рабочий стол в аудитории Кибернетического центра ТПУ с персональным компьютером. |
| Перечень вопросов, подлежащих исследованию, проектированию и разработке: | |
| 1. Правовые и организационные вопросы обеспечения безопасности: <ul style="list-style-type: none"> – специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; – организационные мероприятия при компоновке рабочей зоны. | <ul style="list-style-type: none"> – Рабочее место при выполнении работ сидя регулируется ГОСТом 12.2.032-78 – Организация рабочих мест с электронно-вычислительными машинами регулируется СанПиНом 2.2.2/2.4.1340-03 – Трудовой кодекс Российской Федерации: федер. Закон от 30 дек. 2001 г. № 197-ФЗ Раздел 10 |
| 2. Производственная безопасность: 2.1. Анализ выявленных вредных и опасных факторов 2.2. Обоснование мероприятий по снижению воздействия | <ul style="list-style-type: none"> – Отклонение показателей микроклимата; – Недостаточная освещенность рабочей зоны; – Пониженная световая и цветовая контрастность; – Повышенный уровень шума на рабочем месте; – Повышенный уровень статического электричества – Повышенная запыленность воздуха рабочей зоны; |
| 3. Экологическая безопасность: | Анализ воздействия на литосферу: <ul style="list-style-type: none"> – Утилизация компьютеров, смартфонов, оргтехники и бумаги; Анализ воздействия на гидросферу: |

| | |
|--|--|
| | <ul style="list-style-type: none"> – Производство компьютерной техники и смартфонов; Анализ воздействия на атмосферу: – Выделение вредных веществ в атмосферу при нагреве материнской платы; – Повышенная сухость воздуха при работе компьютера; |
| 4. Безопасность в чрезвычайных ситуациях: | <ul style="list-style-type: none"> – Короткое замыкание проводки – Затопление – Землетрясение – Наиболее типичная ЧС: Пожар; |

| | |
|---|--------------|
| Дата выдачи задания для раздела по линейному графику | 01.03.2021 г |
|---|--------------|

Задание выдал консультант:

| Должность | ФИО | Ученая степень, звание | Подпись | Дата |
|--------------------|-----------------------------|------------------------|---------|------|
| Ассистент ООД ШБИП | Черемискина Мария Сергеевна | - | | |

Задание принял к исполнению студент:

| Группа | ФИО | Подпись | Дата |
|--------|---------------------------|---------|------|
| 8K71 | Писанкин Андрей Сергеевич | | |
| 8K71 | Чепкасов Артём Юрьевич | | |

Реферат

Выпускная квалификационная работа содержит 142 страницы, 50 рисунков, 54 таблицы.

Ключевые слова: промышленные манипуляторы, машинное обучение, анализ данных, веб-сервис, микросервисная архитектура.

Цель работы – создание программного обеспечения прогнозирования изменений температуры приводов промышленного манипулятора в штатном режиме работы, на основе значений физических показателей.

В процессе выполнения работы использовались языки программирования Python и PHP, СУБД PostgreSQL.

В результате работы был реализован веб-сервис интеллектуального анализа данных для прогнозирования температуры.

В первом разделе представлено описание предметной области, формирование проблемы и общие требования к разрабатываемой системе.

Во втором разделе подробно описывается организация командной разработки программного обеспечения в целях оптимизации процесса разработки.

Третий раздел содержит информацию о способе реализации веб-сервиса и его архитектуры, используемых алгоритмах и выборе средств для разработки с приведением сравнения с аналогами.

В четвертом разделе описываются элементы системы и организация непрерывной интеграции и развёртывания программного обеспечения, иллюстрируются результаты работы.

Пятый раздел представляет собой выполненное задание по разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение».

В шестом разделе содержится выполненное задание по разделу «Социальная ответственность».

Оглавление

| | |
|--|----|
| Определения, обозначения, сокращения, нормативные ссылки | 16 |
| Введение..... | 18 |
| Глава 1. Описание предметной области | 20 |
| 1.1 Промышленные манипуляторы | 20 |
| 1.2 Проблемы предметной области..... | 21 |
| 1.3 Постановка задачи | 21 |
| 1.4 Требования к разрабатываемой системе..... | 22 |
| Глава 2. Организация командной разработки ПО | 25 |
| 2.1 Методологии командной разработки ПО | 25 |
| 2.1.1 Kanban | 27 |
| 2.1.2 Scrum | 28 |
| Глава 3. Проектирование системы | 30 |
| 3.1 Машинное обучение | 30 |
| 3.1.1 Описание задачи..... | 30 |
| 3.1.2 Описание исходных данных | 30 |
| 3.1.3 Формирование и преобразование данных | 32 |
| 3.1.4 Обработка отсутствующих и аномальных значений..... | 33 |
| 3.1.5 Преобразование признаков | 44 |
| 3.1.6 Отбор признаков | 45 |
| 3.1.7 Обучение | 48 |
| 3.2 Архитектура системы | 52 |
| 3.3 Проектирование базы данных..... | 58 |
| 3.4 Выбор средств реализации системы | 61 |
| 3.4.1 Серверная часть..... | 61 |

| | |
|--|-----|
| 3.4.2 СУБД | 62 |
| Глава 4. Реализация системы | 66 |
| 4.1 Сервис-прокси API Gateway | 66 |
| 4.2 Микросервис авторизации | 70 |
| 4.3 Микросервис нотификации..... | 78 |
| 4.4 Микросервис анализа данных..... | 81 |
| 4.5 Подготовка к разворачиванию ПО..... | 84 |
| 4.5.1 Контейнеризация..... | 84 |
| 4.5.2 Сценарий разворачивания ПО..... | 86 |
| Глава 5 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение..... | 89 |
| 5.1 Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения | 89 |
| 5.1.1 Потенциальные потребители результатов исследования | 89 |
| 5.1.2 Анализ конкурентных технических решений..... | 91 |
| 5.1.3 QuaD-анализ | 93 |
| 5.1.4 SWOT-анализ..... | 95 |
| 5.2 Определение возможных альтернатив проведения научных исследований | 98 |
| 5.3 Планирование научно-исследовательских работ | 99 |
| 5.3.1 Структура работ в рамках научного исследования | 99 |
| 5.3.2 Определение трудоемкости выполнения работ | 100 |
| 5.3.3 Разработка графика проведения научного исследования..... | 100 |
| 5.4 Бюджет научно-технического исследования (НТИ) | 105 |
| 5.4.1 Расчет материальных затрат НТИ | 105 |

| | |
|---|-----|
| 5.4.2 Основная заработная плата исполнителей темы | 107 |
| 5.4.3 Дополнительная заработная плата исполнителей темы | 108 |
| 5.4.4 Отчисление во внебюджетные фонды | 108 |
| 5.4.5 Накладные расходы | 109 |
| 5.4.6 Формирование бюджета затрат научно-исследовательского проекта | 110 |
| 5.5 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования ... | 110 |
| Вывод по разделу | 112 |
| Глава 6. Социальная ответственность..... | 113 |
| Введение..... | 113 |
| 6.1 Правовые и организационные вопросы обеспечения безопасности ... | 113 |
| 6.2 Производственная безопасность | 114 |
| 6.2.1 Отклонение показателей микроклимата..... | 116 |
| 6.2.2 Недостаточна освещенность рабочей зоны..... | 118 |
| 6.2.3 Повышенная световая и цветовая контрастность..... | 119 |
| 6.2.4 Повышенный уровень шума на рабочем месте | 120 |
| 6.2.5 Повышенный уровень статического электричества..... | 121 |
| 6.2.6 Повышенная запыленность воздуха рабочей зоны | 122 |
| 6.3 Экологическая безопасность..... | 123 |
| 6.4 Безопасность в чрезвычайных ситуациях..... | 123 |
| 6.4.1 Короткое замыкание | 123 |
| 6.4.2 Землетрясение | 124 |
| 6.4.3 Затопление | 124 |
| 6.4.4 Наиболее вероятная ЧС: Пожар | 125 |

| | |
|-------------------------------------|-----|
| Вывод по разделу | 126 |
| Заключение | 127 |
| Список публикаций студентов | 128 |
| Список используемых источников..... | 129 |
| Приложение А | 133 |
| Приложение Б..... | 140 |
| Приложение В..... | 142 |

Определения, обозначения, сокращения, нормативные ссылки

Авторизация – предоставление определённых прав.

Аутентификация – процедура проверки подлинности пользователя.

БД (База Данных) – это хранилище для большого количества структурированных данных, с которыми можно производить определённые действия.

Веб-сервис – идентифицируемый уникальным веб-адресом программная система со стандартизированными интерфейсами.

Машинное обучение – это подраздел искусственного интеллекта (ИИ) и информатики, которая фокусируется на использовании данных и алгоритмов для имитации обучения людей.

Прокси-сервер – сервер, выступающий в роли посредника между клиентом и обрабатывающем его запросы сервером.

СУБД (Система управления базами данных) – совокупность программных средств, обеспечивающих управление и создание баз данных.

Фреймворк – программное обеспечение, позволяющее автоматизировать разработку и тестирование программного продукта.

Эндпоинт – это интерфейс, предоставляемый взаимодействующей стороной или каналом связи.

API (Application Programming Interface) – описание способов, которыми одна компьютерная программа может взаимодействовать с другой программой.

CI/CD (Continuous Integration, Continuous Delivery) – это технология автоматизации тестирования и доставки новых модулей разрабатываемого ПО заинтересованным сторонам: разработчикам, аналитикам, инженерам качества, конечным пользователям.

CSV – текстовый формат, предназначенный для представления табличных данных.

Django – фреймворк для веб-приложений на языке Python.

Docker – платформа, которая позволяет пользователям легко упаковывать, распространять и управлять приложениями в контейнерах.

DRF (Django Rest Framework) – это библиотека, которая работает со стандартными моделями Django.

FastAPI – фреймворк для веб-приложений на языке Python.

JavaScript – прототипно-ориентированный язык программирования.

JSON (JavaScript Object Notation) – стандартный текстовый формат для представления структурированных данных на основе синтаксиса объекта JavaScript.

Lumen – фреймворк для веб-приложений на языке PHP.

ORM – программное решение, позволяющее работать с сущностями базы данных как с объектами.

Python – высокоуровневый язык программирования общего назначения.

PHP – высокоуровневый язык программирования общего назначения.

Postman – программное обеспечение для тестирования веб-сервисов.

Redis – это размещаемое в памяти хранилище ключ-значение.

SMTP (Simple Mail Transfer Protocol) – это приложение, основная цель которого отправлять, получать и/или ретранслировать исходящую почту между отправителями и получателями электронной почты.

URL – унифицированный указатель ресурса, определяющий его конкретное местоположение.

URI – унифицированный идентификатор ресурса.

UUID – это стандарт идентификации, используемый в создании программного обеспечения.

Введение

Промышленные манипуляторы – эффективные инструменты комплексной механизации и автоматизации производственных процессов. Благодаря дистанционному управлению, использование манипуляторов в производстве сводит к минимуму вовлеченность специалистов в монотонную, физически сложную механическую работу, что позволяет повысить уровень качества производимой продукции и увеличить производительность выполняемых работ.

Следующим этапом развития промышленной отрасли – массовое внедрение киберфизических систем в заводские процессы. Увеличение эффективности будет достигаться за счет применения математических и алгоритмических методов анализа информации, собираемой в процессе эксплуатации роботизированных систем. Основная идея данного подхода – извлечение и использование новых знаний из собираемых данных в процессах управления и поддержки принятия решения.

Благодаря появлению технологий облачных вычислений, стала возможна организация информационной инфраструктуры для комплексного мониторинга производственных систем без существенных финансовых затрат, временных и эксплуатационных издержек. Использование систем сбора и анализа данных позволит усовершенствовать оценку текущего состояния оборудования. В настоящее время планирование комплекса сервисных мероприятий на основе эксплуатационной технической документации, является общепринятым подходом, однако он не позволяет избежать непредвиденных поломок. Использование прогнозного подхода в организации технического обслуживания, основываясь на актуальных данных с промышленных манипуляторов, позволит повысить эффективность планирования мероприятий планово-предупредительного ремонта.

Таким образом, автоматизация процесса сбора и анализа данных с целью прогнозирования возникновения аномалий в работе промышленных манипуляторов позволит качественно улучшить производственный процесс.

В качестве способа решения данной проблемы предлагается использовать методы интеллектуального анализа данных для выявления закономерностей между состоянием промышленного манипулятора и изменениями физических показателей приводов.

Апробация реализованного программного обеспечения осуществлена на примере прогнозирования показателей температуры подвижных частей промышленного манипулятора в нормальном режиме работы основываясь на снятых физических метриках, таких как сила тока, напряжение и т.д.

Программный продукт будет представлять собой веб-сервис, что позволит без привязки к текущим вычислительным мощностям отдельного предприятия анализировать данные с манипуляторов и на основе полученного результата формировать план технического обслуживания оборудования.

Целью данной работы является создание программного обеспечения для прогнозирования изменений температуры приводов промышленного манипулятора в штатном режиме работы, на основе значений физических показателей.

Для достижения поставленной цели необходимо решить следующие задачи:

- Анализ предметной области и формирование требований;
- Выбор методологии командной разработки ПО;
- Проектирование архитектуры системы;
- Проектирование базы данных;
- Выбор средств для разработки системы;
- Разработка веб-сервиса;
- Апробация на тестовых данных.

Глава 1. Описание предметной области

1.1 Промышленные манипуляторы

Промышленный манипулятор – это оборудование, позволяющее выполнять сложные пневматические наклоны и повороты в независимости от положения перемещаемого объекта относительно его центра масс. Основное назначение манипуляторов – автоматизация физически сложных процессов.

Промышленные манипуляторы часто используются в случаях если:

- Объект слишком тяжелый, чтобы его можно было переместить вручную;
- Объект необходимо переместить в место, труднодоступное для человека;
- При перемещении объекта вручную человек подвергается риску травмы;
- Необходимо своевременно перемещать большое количество объектов.

Главной особенностью манипуляторов является их способность перемещать объекты за пределы своего центра масс, это особенно востребовано в случае работы с тяжелыми грузами или изделиями нестандартной формы, поскольку другие типы машин не так хорошо работают с объектами, которые не являются относительно легкими и стандартными по форме.

Благодаря своим особенностям манипуляторы используются в самых разных промышленных отраслях для повышения эффективности и безопасности производственных процессов. Некоторые примеры использования манипуляторов в промышленности:

- Строительство;
- Упаковочные линии;
- Электромеханика;
- Сварочные работы.

Критерии выбора промышленного манипулятора:

- Вес товара, который нужно переместить;
- Размер и форма изделия, которое нужно переместить;
- Траектория движения.

Промышленные манипуляторы подходят для работы с изделиями до 2000 фунтов, способны перемещать продукцию нестандартной формы, как в вертикальном, так и в горизонтальном направлении.

1.2 Проблемы предметной области

Отличительной чертой современных промышленных предприятий является высокий уровень автоматизации производственного процесса, однако используемые инструменты не способны полностью исключить человека из всей технологической цепочки. На каждом из этапов, главная задача человека – принятие решений, на основе получаемых данных. В данной работе рассматривается возможность внедрения инструмента поддержки принятия решений в организации планово-предупредительного обслуживания промышленных манипуляторов, с целью повышения экономической эффективности производства. Несвоевременное выявление неисправностей в работе промышленного манипулятора может привести к ухудшению сокращением его эксплуатационного срока или полному выходу из строя, что влечет за собой увеличение финансовых издержек на ремонт дорогостоящего оборудования и нарушение плана графика работ производства.

Рассматривается возможность организации превентивных мер по техническому обслуживанию, основываясь на состоянии промышленного манипулятора.

1.3 Постановка задачи

Исходя из обозначенной проблемы главной назначением системы является техническое диагностирование состояния манипулятора [1]. Техническая диагностика является составной частью технического

обслуживания. Её основной задачей является обеспечение безопасности, функциональной надёжности и эффективности работы технического объекта, а также сокращение затрат на его техническое обслуживание и уменьшение потерь от простоев в результате отказов и преждевременных выводов в ремонт.

Для решения описанной проблемы предлагается создание механизма анализа данных с целью выявления в них аномальных значений. В качестве инструмента предлагается использовать машинное обучение.

Задачей машинного обучения является частичная или полная автоматизация решения сложных профессиональных задач в самых разных областях человеческой деятельности.

Данный подход является актуальным и коммерчески выгодным. По данным PWC [17], машинное обучение в экономике может повысить производительность на 14,3% к 2030 году. Машинное обучение является катализатором роста производительности.

Машинное обучение имеет широкий спектр применений: распознавание речи, жестов и образов, медицинская диагностика и решаемая нами задача технической диагностики. В данном случае машинное обучение используется для прогнозирования будущих значений на основе обрабатываемых текущих и предыдущих данных. В качестве первого этапа предлагается прогнозировать показатели температуры приводов промышленного манипулятора с целью своевременного предотвращения перегрева и, как следствие, выхода из строя.

1.4 Требования к разрабатываемой системе

Основной задачей разрабатываемой системы является осуществление анализа наборов данных с промышленных манипуляторов. Обработка данных требует значительных вычислительных трудозатрат, рост которых пропорционален росту объему обрабатываемых данных. Исходя из основной задачи были определены общие требования к системе:

1. Масштабируемость – должна быть возможность распределять компоненты системы по разным серверам;

2. Отказоустойчивость – вспомогательный функционал системы не должен влиять на обработку данных;
3. Низкая связанность – система должна состоять из низко связанных компонентов, развиваемых независимо друг от друга.

Выделенные требования определяют свойства системы. Для определения основных функций системы был составлен список основных функциональных требований к веб-сервису:

1. Загрузка набора данных и выгрузка результатов интеллектуального анализа;
2. Сохранение и использование обученных моделей;
3. Механизм регистрации и авторизации пользователей;
4. Рассылка уведомлений пользователям.

На рисунке 1 представлена диаграмма вариантов использования. Данная диаграмма описывает представленные в системе роли и все функциональные возможности, сгруппированные по пакетам. На данный момент система содержит 2 роли: гость и авторизованный пользователь, имеющий доступ к основному функционалу системы.

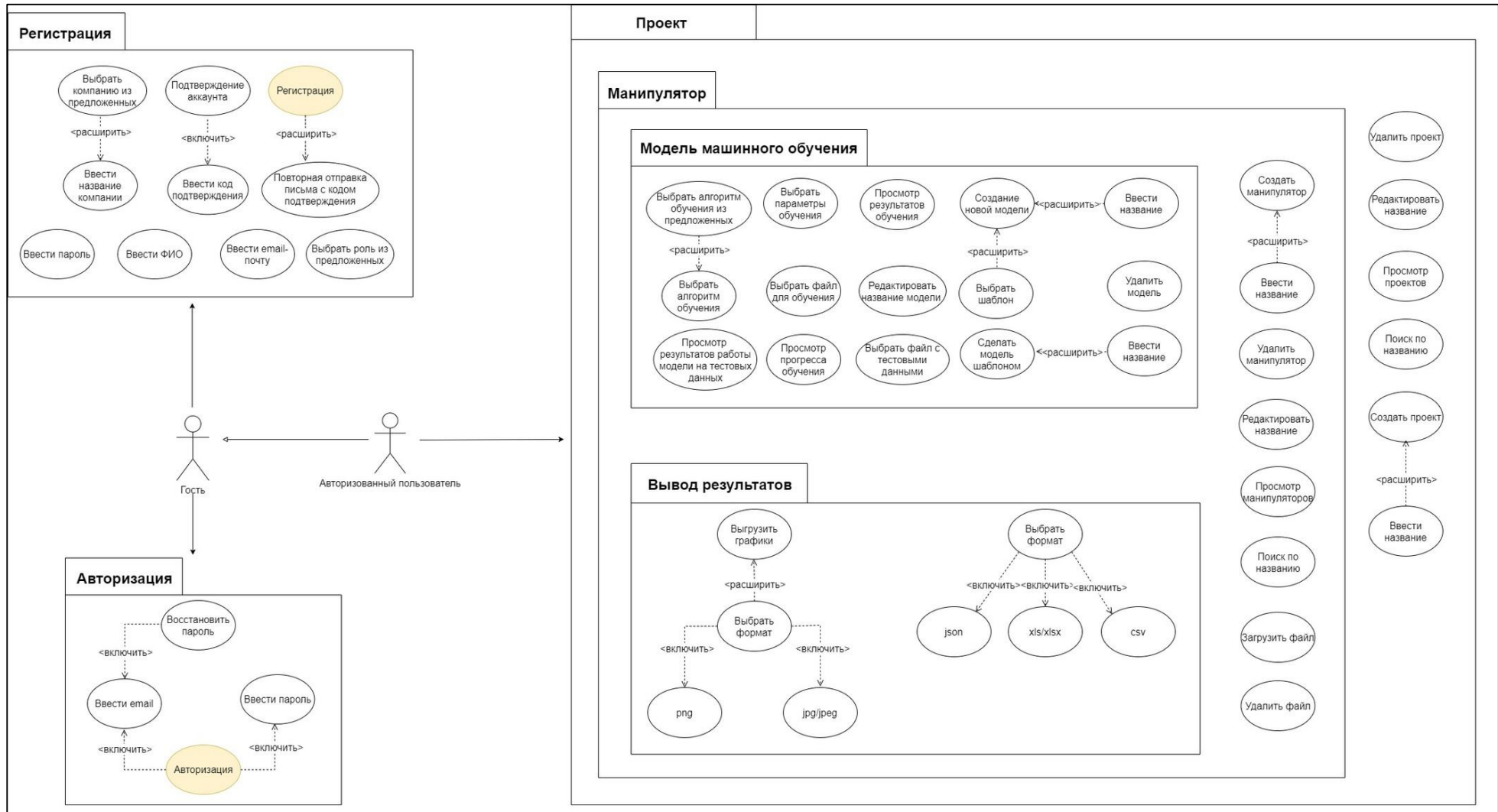


Рисунок 1 – Диаграмма вариантов использования

Глава 2. Организация командной разработки ПО

2.1 Методологии командной разработки ПО

Существует множество идейно различных подходов ведения проектов, которые определяют бизнес процесс разработки. В большинстве случаев, выбор определенной методологии зависит от требований к проекту, поэтому для нас, в условиях строгой ограниченности по времени и людских ресурсам, ключевыми факторами при выборе подхода стали показатели высокой эффективности использования в разработке краткосрочных проектов и наличие возможности модернизировать процесс под нужды небольшой команды. Ключевая проблема – отслеживание прогресса выполнения проекта из-за пересечение выполняемых членами команды функций.

Исходя из вышеперечисленных ограничений, в качестве основного инструмента ведения проекта был выбран фреймворк Scrum в комбинации с Kanban доской. Для решения обозначенной проблемы и оптимизации стандартных Scrum элементов был принят следующий формат работы:

- Спринты, длительностью в 2 недели (повторяемый фиксированный временной интервал, в течение которого создается продукт максимально возможной ценности);
- По результатам спринта проводится ретро-митинг (ретроспектива), на котором обсуждаются итоги спринта и варианты улучшения производительности, если это возможно;
- Так как для малых команд нет необходимости частой синхронизации работы, собрания проводятся не ежедневно, а 2 раза в неделю, при этом длительность собраний увеличена с 15 до 30 минут. Задания находятся в зоне ответственности каждого члена команды, для отслеживания прогресса выполнения достаточно Kanban доски, поддерживаемой в актуальном состоянии.

По результатам собраний составляется протокол. Структура протокола состоит из 3 элементов:

- Раздел «что сделано» – в этом разделе член команды отчитывается о проделанной работе, прикрепляя ссылками на задачи в Kanban-доске;
- Раздел «в процессе» – в данном разделе перечисляются задачи, находящиеся в процессе выполнения;
- Раздел «что будет сделано» – в этом разделе членом команды формируется план работ на следующие дни.

Длительность задачи оценивается исполнителем самостоятельно и вносится в соответствующую задачу на Kanban доске.

На рисунке 2 представлен календарный план, где желтым обозначены собрания, красным – ретро-митинги, а синим – технические митинги, где члены команды рассказывают про непосредственно реализацию и делятся техническими решениями и подходами.

| Февраль 2021г. | | | | | | | |
|----------------|----|----|----|----|----|----|----|
| Sprint-1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| Spring-2 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| | 22 | 23 | 24 | 25 | 26 | 27 | 28 |
| Март 2021г. | | | | | | | |
| Sprint-3 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| Sprint-4 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| | 22 | 23 | 24 | 25 | 26 | 27 | 28 |
| Sprint-5 | 29 | 30 | 31 | | | | |
| Апрель 2021г. | | | | | | | |
| Sprint-5 | | | | 1 | 2 | 3 | 4 |
| | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| Sprint-6 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| | 19 | 20 | 21 | 22 | 23 | 24 | 25 |

Рисунок 2 – Календарный план митингов на 3 месяца

В случае возникновения непредвиденных проблем, которые влияли на работу другого специалиста, организовывается незапланированная встреча в

рамках которой осуществлялся поиск решения или корректируются сроки выполнения работ.

2.1.1 Kanban

Методология управления разработкой Kanban подразумевает активное использование доски заданий, которая содержит как сам список заданий, так и дополнительную информацию о них: срок выполнения задачи, описание, комментарии от разработчиков.

Использование Kanban доски повышает наглядность работы. Все члены команды имеют доступ к данным о прогрессе выполнения работ, поэтому данный подход повышает вероятность выявления проблемных мест в рабочем процессе.

Для упрощения процесса управления, создаваемые задачи были разделены на следующие типы:

- Analysis – аналитика разрабатываемой системы: построение диаграмм, проработка общей архитектуры системы;
- Documenting – составление документации;
- Programming – разработка программного кода;
- Testing – тестирование программного кода;
- Bug – выявленные ошибки в работе, которые необходимо устранить;
- Deployment – развертывание программного обеспечения.

В качестве Kanban доски был использован бесплатный веб-сервис Trello [12]. Используемая Kanban доска представлена на рисунке 48 (Приложение Б).

Типы задач определяются по надписи в квадратных скобках перед началом названия задачи. В название задания также входит его уникальный идентификатор, для упрощения процесса протоколирования встреч.

Столбцы Kanban могут быть адаптированы под специфику проекта. В организованном рабочем процессе использовались следующие типы колонок:

- Для удобства работы:
 - Templates: содержит шаблоны для создания задач;
 - Refs: содержит полезные ссылки проекта;
- Состояния задач:
 - TODO: новая задача;
 - Bugs: найденные неисправности;
 - In progress: задача, которую на текущий момент делает разработчик;
 - Review: проверка результатов выполнения задачи;
 - Done: успешно завершенная задача;
- Спринты: содержат задачи, которые были выполнены за соответствующий спринт.

Пример разбитой по спринтам Kanban доски представлен на рисунке 49 (Приложение Б).

2.1.2 Scrum

Метод управления проектами Scrum подразумевает понятие «Scrum-команда», которая включает в себя три роли:

- Владелец продукта (Product Owner) – член команды, который отвечает за достижение бизнес ценности продукта. Эта роль объединяет в себе функции следующих ролей:
 - Менеджер по разработке – член команды, который отвечает, как за разработку, так и за управление разработкой;
 - Руководитель проекта – член команды, отвечающий за руководство работами в рамках проектной задачи;
 - Менеджера по маркетингу – член команды, который отвечает за выбор целевого рынка и его анализ, с целью увеличения продаж товара.
- Команда создания продукта (Development Team) – самоорганизующаяся команда, которая по завершению

установленных итераций выдает потенциально продаваемый вариант продукта. Внутри команды представлена только одна роль developer. Такой человек обладает всеми необходимыми техническими навыками для осуществления разработки.

- Scrum-мастер (Scrum Master) – отвечает за организацию процесса разработки продукта. Совмещает в себе несколько ролей, таких как:
 - Обслуживающий лидер – помогает понять и внедрить процессы Scrum;
 - Коуч – помогает раскрыть потенциал отдельных людей и команды, помогает в разрешении конфликтов;
 - Ментор – обладает знаниями и может помочь советом.

Ниже приведена таблица ролей в текущей команде.

Таблица 1 – Роли в команде

| № | Роль | Ответственный за выполнение обязанностей |
|----------|------------------|---|
| 1 | Product Owner | Артём |
| 2 | Development Team | Артём, Андрей |
| 3 | Scrum Master | Андрей |

Глава 3. Проектирование системы

3.1 Машинное обучение

3.1.1 Описание задачи

Необходимо предсказать значение температуры промышленного манипулятора, основываясь на физических показателях приводов в процессе эксплуатации (сила тока, скорость вращения и т.д.). Сравнивая предсказанные значения с действительными, определяем отклонение работы манипулятора от его нормального режима.

Для этого необходимо построить предсказательную модель, основываясь на собранных в ходе серии экспериментов данных.

На рисунке 3 представлена схема этапов решения задачи.

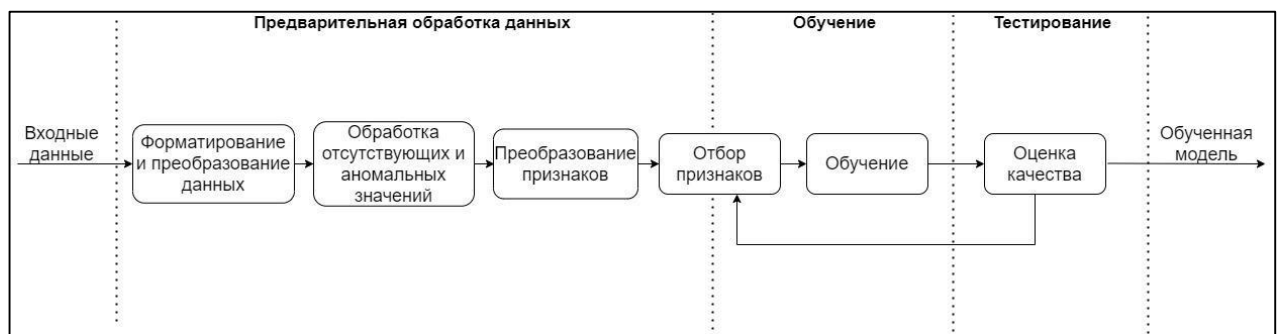


Рисунок 3 – Этапы решения задачи

Подробное описание этапов будет представлено в последующих подразделах.

Требования к модели обучения:

- Точность предсказательной модели не менее 70% методом средней абсолютной ошибки;
- Время обучения модели не более 5 минут.

3.1.2 Описание исходных данных

Исходный набор содержит данные, собранные с 4 различных манипуляторов в ходе проведения 4 независимых серий экспериментов. В таблице 2 представлена общая информация о наборах данных.

Таблица 2 – Описание исследуемых манипуляторов

| Наименование манипулятора | Выполняемая задача | Номер эксперимента | Продолжительность эксперимента |
|------------------------------|--|--------------------|--------------------------------|
| Agilus («FANUC LRM-200iD») | Перемещение пластиковых кубиков внутри роботизированной ячейки | 1 | 04:32:49 |
| | | 2 | 12:56:36 |
| | | 3 | 12:43:52 |
| | | 4 | 09:52:36 |
| Frezer («KUKA KR 120 R2900») | Фрезерование вазы из дерева | 1 | 09:04:30 |
| | | 2 | 12:16:04 |
| | | 3 | 12:38:50 |
| | | 4 | 09:37:23 |
| Palitir («KUKA KR 40-PA») | Перемещение на конвейер с паллеты и обратно коробок размерами 530×360×220 мм | 1 | 07:40:20 |
| | | 2 | 12:19:42 |
| | | 3 | 12:19:42 |
| | | 4 | 09:25:58 |
| Svarka («ABB IRB1200») | Сварочные работы | 1 | 08:43:07 |
| | | 2 | 12:18:59 |
| | | 3 | 12:40:57 |
| | | 4 | 09:45:26 |

Данные по каждому эксперименту содержат одинаковый набор признаков, собранных с датчиков подвижных частей манипуляторов в ходе проведения серии экспериментов. Все измерения производились с интервалом в 1 секунду.

В данной работе рассматривается упрощенный набор, состоящий из данных с основных датчиков, для более наглядного, поэтапного представления применяемого подхода обработки и анализа признаков. В таблице 3 представлено описание исследуемых признаков.

Таблица 3 – Описание исследуемых признаков

| Наименование признака | Единица измерения | Описание |
|-----------------------|-------------------|--|
| Время | - | Время замера показателей датчиков в формате HH:mm:ss |
| Номер привода | - | Номер подвижной части манипулятора |
| Сила тока | мА | Сила тока |
| Температура привода | °С | Температура привода |
| Момент силы | Н*м | Момент силы вращающейся части |
| Угловая скорость | м/с | Угловая скорость вращения привода |
| Линейная скорость | м/с | Линейная скорость вращения привода |

Перед обучением модели, исходный набор данных проходит через следующие этапы предварительной обработки:

- Форматирование и преобразование данных;
- Обработка отсутствующих и аномальных значений;
- Преобразование признаков;
- Отбор признаков.

3.1.3 Формирование и преобразование данных

На данном этапе данные преобразуются в подходящий для проведения статистического анализа и решения поставленной задачи формат. Используемые для прогнозирования временных рядов модели обучения чувствительны к типу входных параметров, поэтому главной целью этого этапа является приведение признаков к единому формату. В рамках решения поставленной задачи категориальные признаки преобразуются в числовой формат.

Исходный набор данных содержит категориальный признак «Номер привода». Известно, что взаимное влияние работы подвижных частей манипулятора минимально, поэтому предлагается разбить данные по каждому манипулятору на N независимых выборок, где N – количество приводов. Все последующие преобразования, построение модели обучения и анализ результатов будет осуществляться для каждого привода отдельно.

Другой нечисловой признак «Время», также преобразуется из строкового формата в числовое значение - количество микросекунд от начала эксперимента. Данный признак введен для контроля проведения измерений и полезен только на этапе предварительной обработки т.к. предсказательная модель не привязывается к моменту проведения замеров относительно начала работы прибора.

Остальные признаки являются числовыми и не требуют обработки на данном этапе.

3.1.4 Обработка отсутствующих и аномальных значений

Наличие отсутствующих значений приводит к невозможности использовать данные для обучения и тестирования. Аномальные значения снижают скорость и точность.

Перед поиском и обработкой аномальных значений заполняются отсутствующие значение т.к. применяемые статистические методы вменения могут привести к возникновению локальных выбросов в обрабатываемом наборе данных.

На рисунке 4 представлена схема обработки отсутствующих значений в исходной выборке.

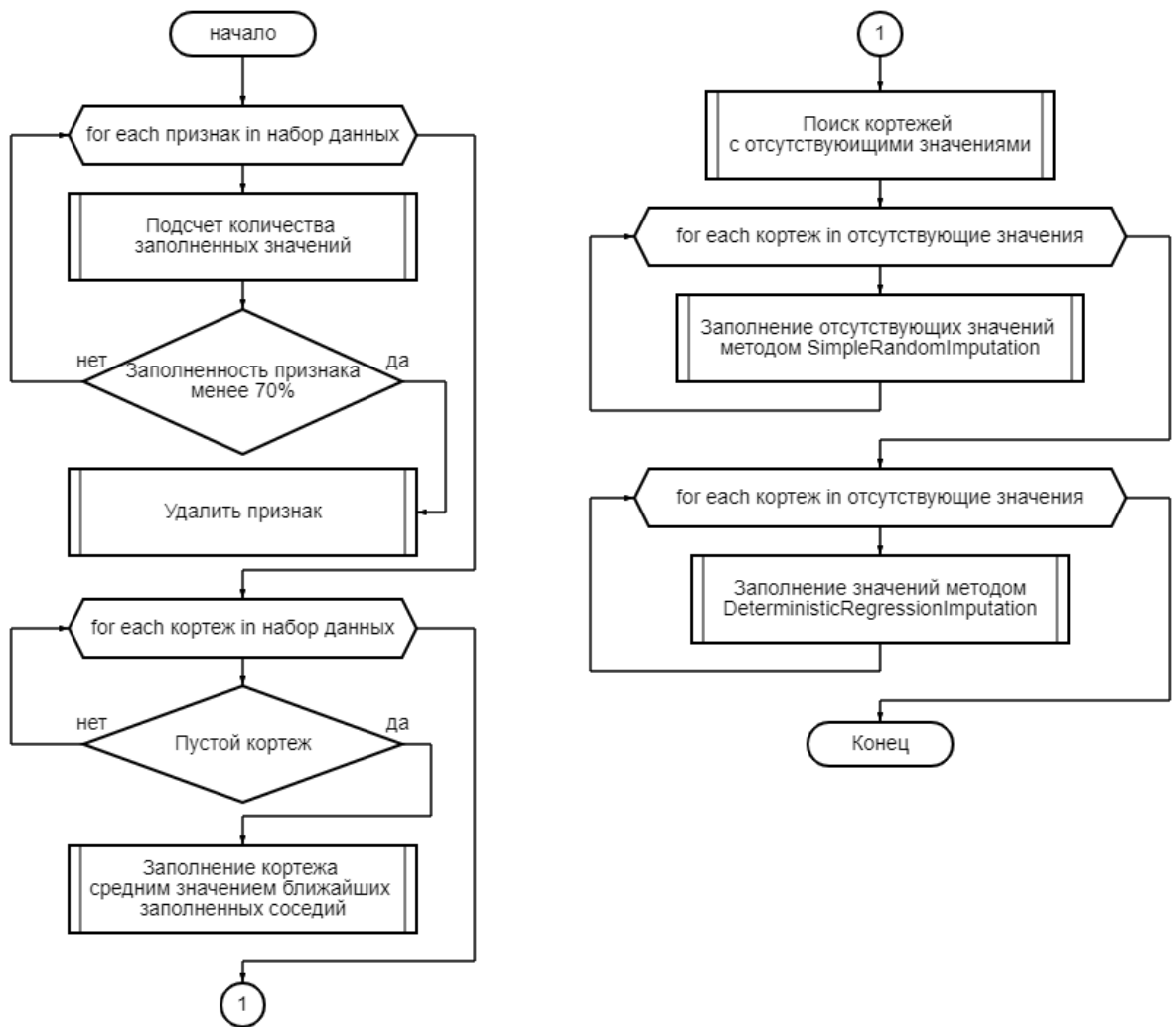


Рисунок 4 – Схема обработки отсутствующих значений

Отсутствующие значения исходного набора данных можно разделить на 2 типа:

- Отсутствие измерения;
- Отсутствие значения.

В рамках решения поставленной задачи применимы 3 подхода:

- Удаление измерения (вместе с отсутствующим значением необходимо удалить показатели других измерительных приборов);
- Удаление признака;
- Заполнение недостающего значения.

Первый подход обладает высокой скоростью, однако в общем случае ухудшает качество модели. Удаление измерений порождает резкие «скачки» в данных, поэтому для обработки записи используется третий подход.

Удаление всего признака из данных применяется в случае, когда отсутствуют слишком много значений, заполнение которых приведет к снижению точности модели. В решаемой задаче, если признак имеет менее 70% значений, то он удаляется.

Для быстрой визуальной оценки, на рисунке 5 представлена матрица распределения и плотности отсутствующих значений для первого привода первой серии экспериментов манипулятора Agilus.

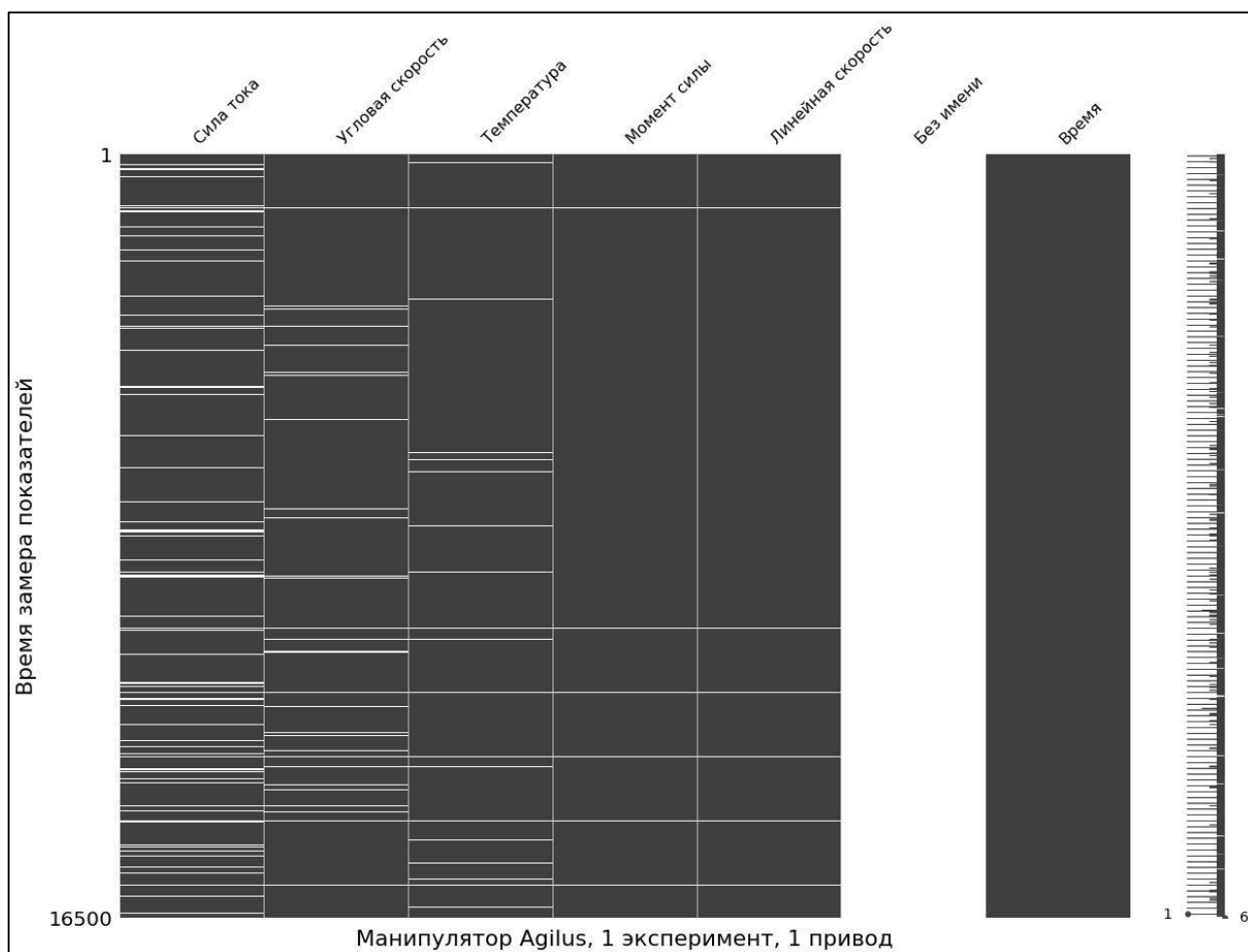


Рисунок 5 – Матрица распределения отсутствующих значений

Все измерительные признаки содержат отсутствующие значения, характер распределения не имеет значения для решаемой задачи.

На рисунке 6 представлена гистограмма, отображающая долю отсутствующих значений в признаке.

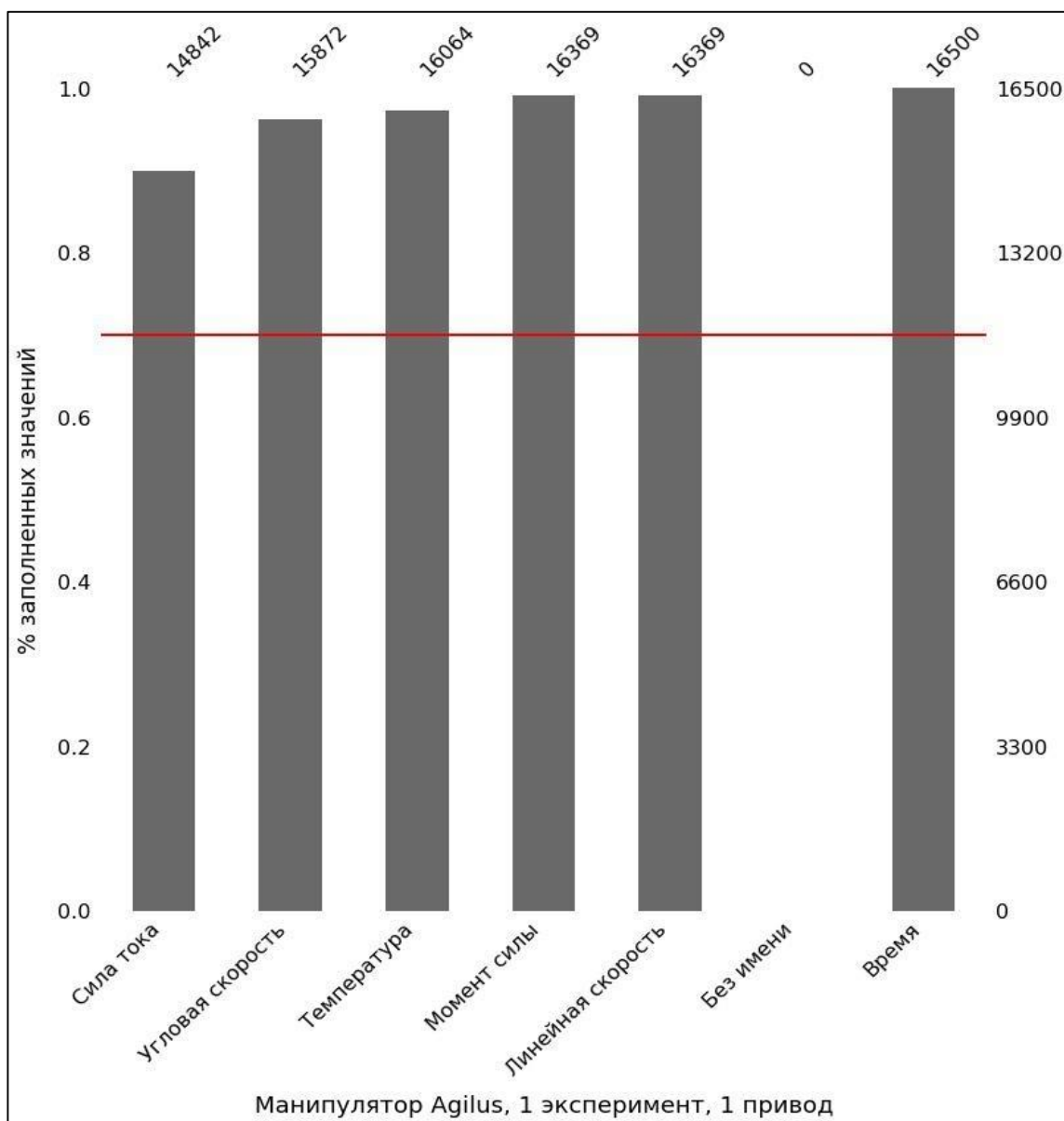


Рисунок 6 – Гистограмма отсутствующих значений

Линия – порог в 70%. Все значения, которые находятся ниже выделенной границы, удаляются.

Для более подробного представления в таблице 4 приведена сводная таблица отсутствующих значений.

Таблица 4 – Таблица отсутствующих значений

| Признак | Количество пустых значений | Процент от общего числа |
|-------------------|-----------------------------------|--------------------------------|
| Без имени | 16500 | 100.0 |
| Сила тока | 1658 | 10.0 |
| Угловая скорость | 628 | 3.8 |
| Температура | 436 | 2.6 |
| Момент силы | 131 | 0.8 |
| Линейная скорость | 131 | 0.8 |

Признак «Без имени» удаляется т.к. не несет в себе полезную для модели информацию. Для заполнения оставшихся отсутствующих значений используются 2 метода:

1. Среднее значение между двумя соседними заполненными измерениями для отсутствующего измерения;
2. Расчет множественной линейной регрессии с детерминированными регрессорами (Deterministic Regression Imputation) для отсутствующего значения.

Второй метод позволяет заполнить пропущенные значения основываясь на данных измерения. Если в рамках одного измерения несколько пропущенных значений, то применить данный метод невозможно, поэтому предлагается предварительно заполнить отсутствующие значения случайно выбранными наблюдаемыми значениями признака (Simple Random Imputation) для первоначального, грубого вменения отсутствующих значений. Далее значения улучшаются при помощи регрессионной модели.

После вменения отсутствующих значений производится поиск и устранение аномальных значений.

На рисунке 7 представлена схема обработки экстремальных значений.

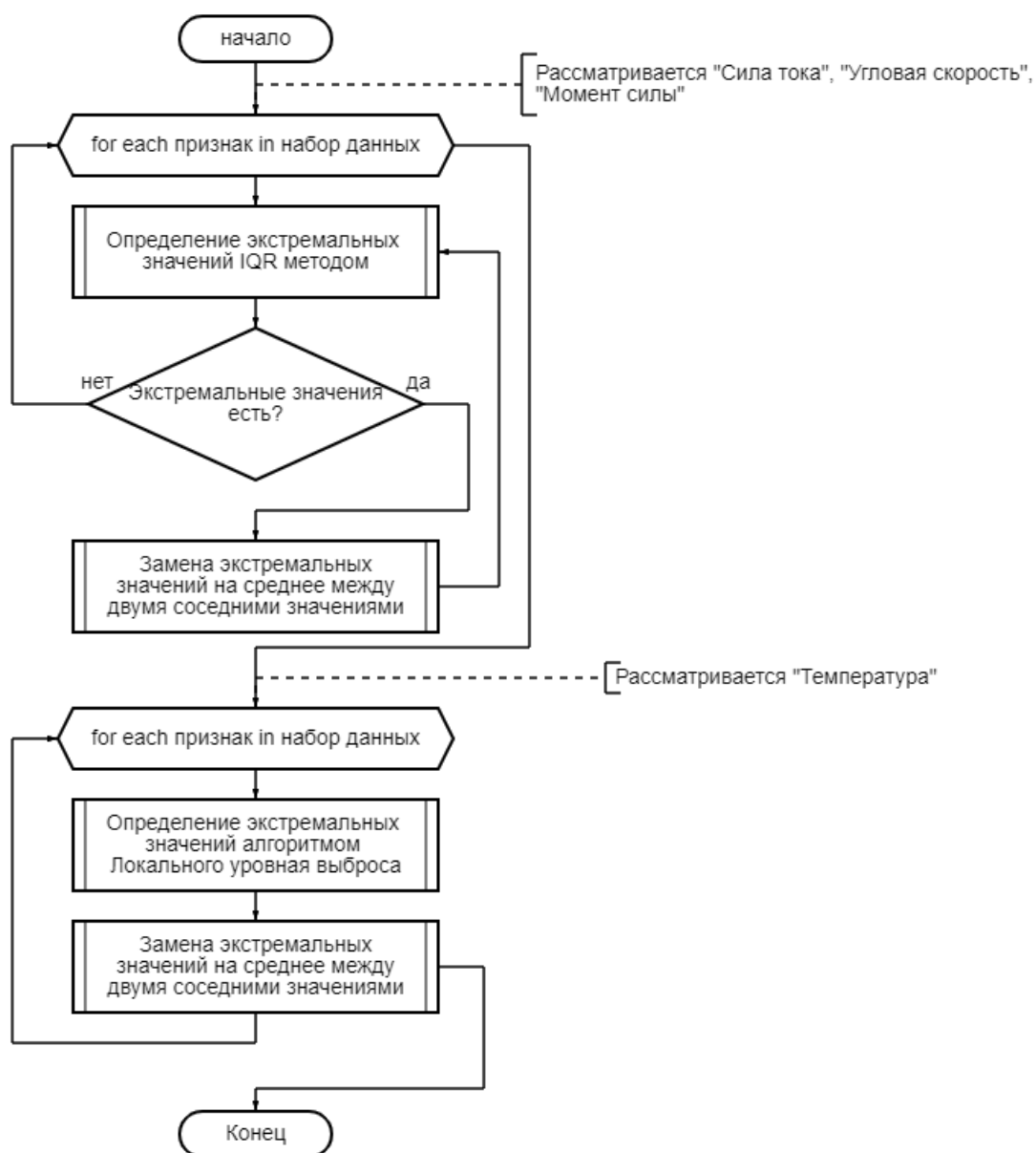


Рисунок 7 – Схема обработки экстремальных значений

Для определения экстремальных выбросов воспользуемся IQR методом. Данный метод позволяет выявить явно выбивающиеся из распределения значения на основе межквартильного размаха.

Редко встречающиеся данные – значения которые лежат за пределами 3 стандартных отклонений от математического ожидания.

Определение верхней и нижней границы осуществляется по формулам 1 и 2.

$$X1 = Q1 - 1,7 * IQR \quad (1)$$

$$X2 = Q3 + 1,7 * IQR \quad (2)$$

где: Q1 – первый квартиль;

Q3 – третий квартиль;

IQR– межквартильный размах (Q3-Q1).

Все значения, находящиеся вне диапазона $[Q1 - 1.7 * IQR, Q3 + 1.7 * IQR]$ будут считаться выбросами.

На рисунке 8 изображена диаграмма размаха для исходных данных (температура и линейная скорость будут рассмотрены отдельно).

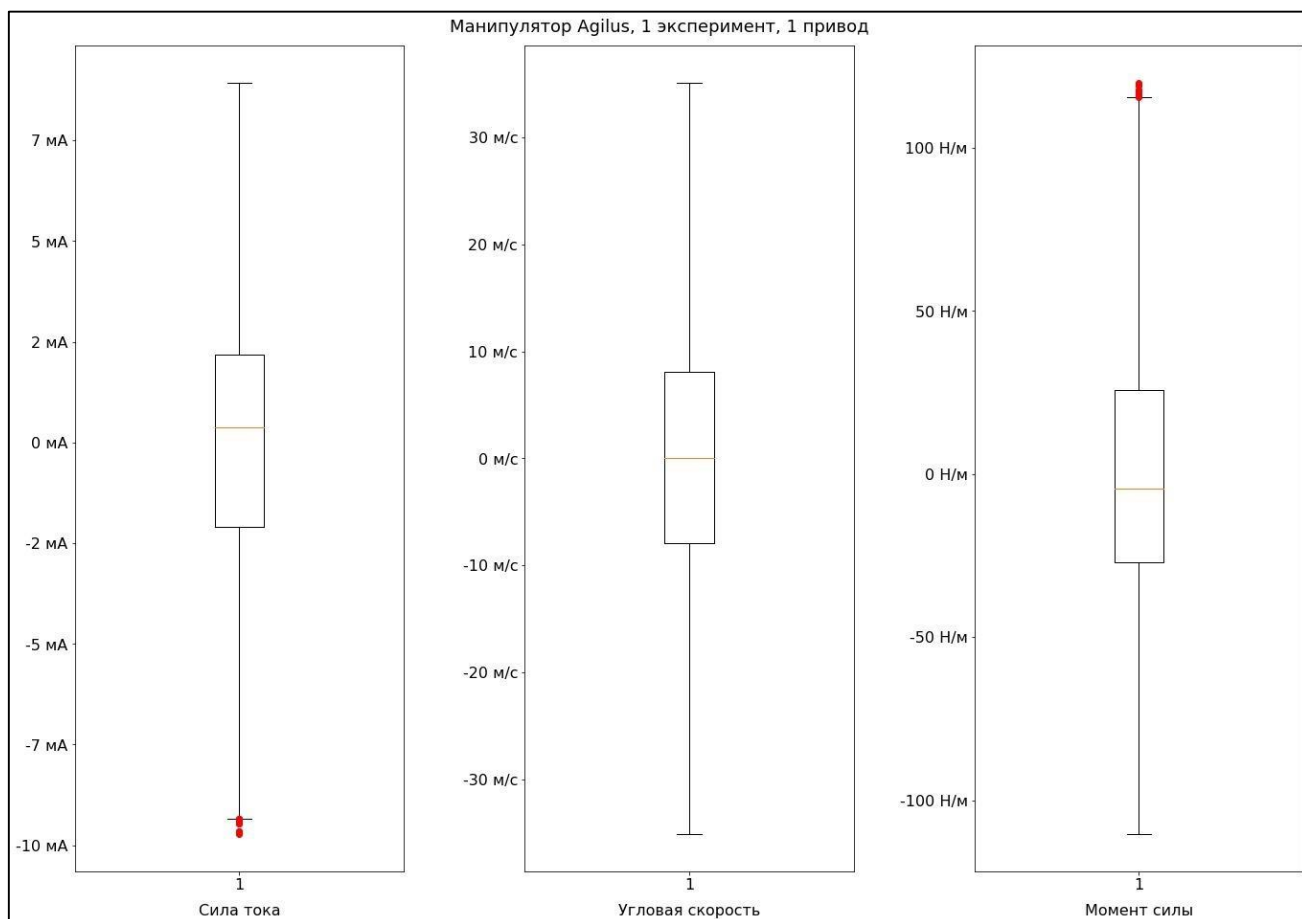


Рисунок 8 – Диаграмма размаха основных исследуемых признаков

Диаграмма отражает наблюдаемые минимумы, максимумы исследуемых величин, медиану и выбросы, которые изображены за краями «усов».

Для сглаживания данных и увеличения точности модели обучения, каждый из выбросов заменяется на среднее двух соседних значений. Данная операция повторяется пока существуют выбросы.

На рисунке 9 изображена диаграмма размаха после устранения экстремальных выбросов.

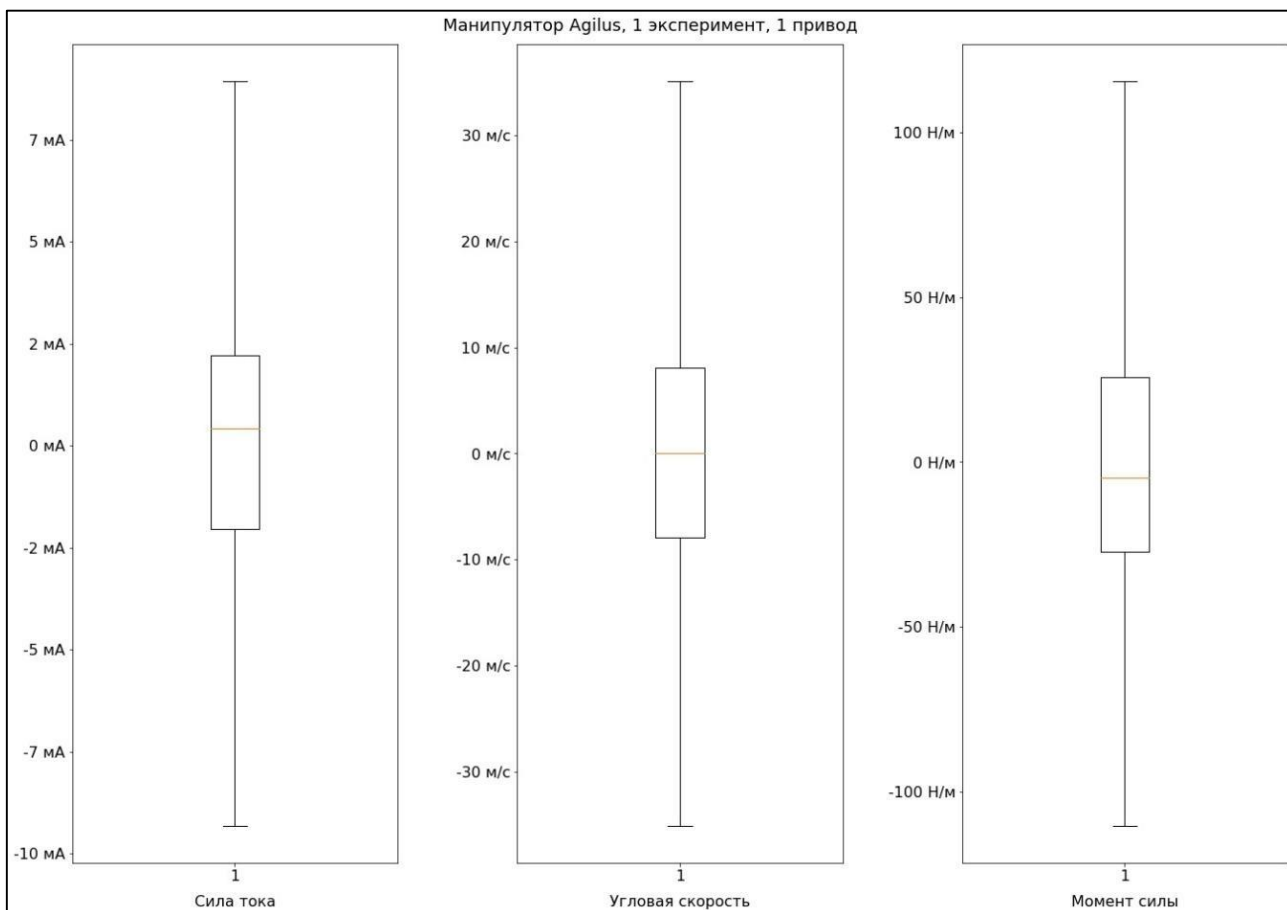


Рисунок 9 – Диаграмма размаха с устраненными аномальными значениями

Используемый метод обладает высокой эффективностью в выявлении экстремальных значений, однако он не позволяет выявлять локальные выбросы, которые вносят значительную погрешность в модель обучения.

Рассмотрим диаграмму размаха для температурных показателей, представленную на рисунке 10.

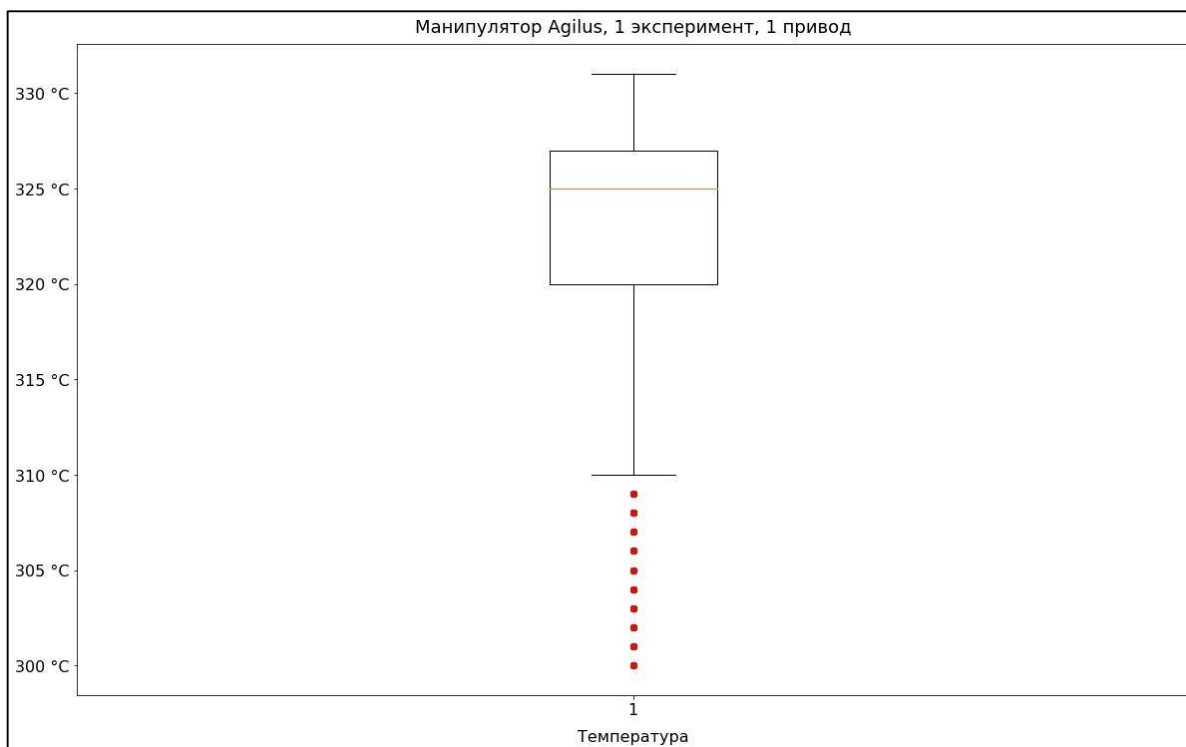


Рисунок 10 – Диаграмма размаха значений температуры

Все выявленные выбросы оказались за нижней границей, что обусловлено характером исследуемой величины. На рисунке 11 представлен график изменения температуры с течением времени работы манипулятора.

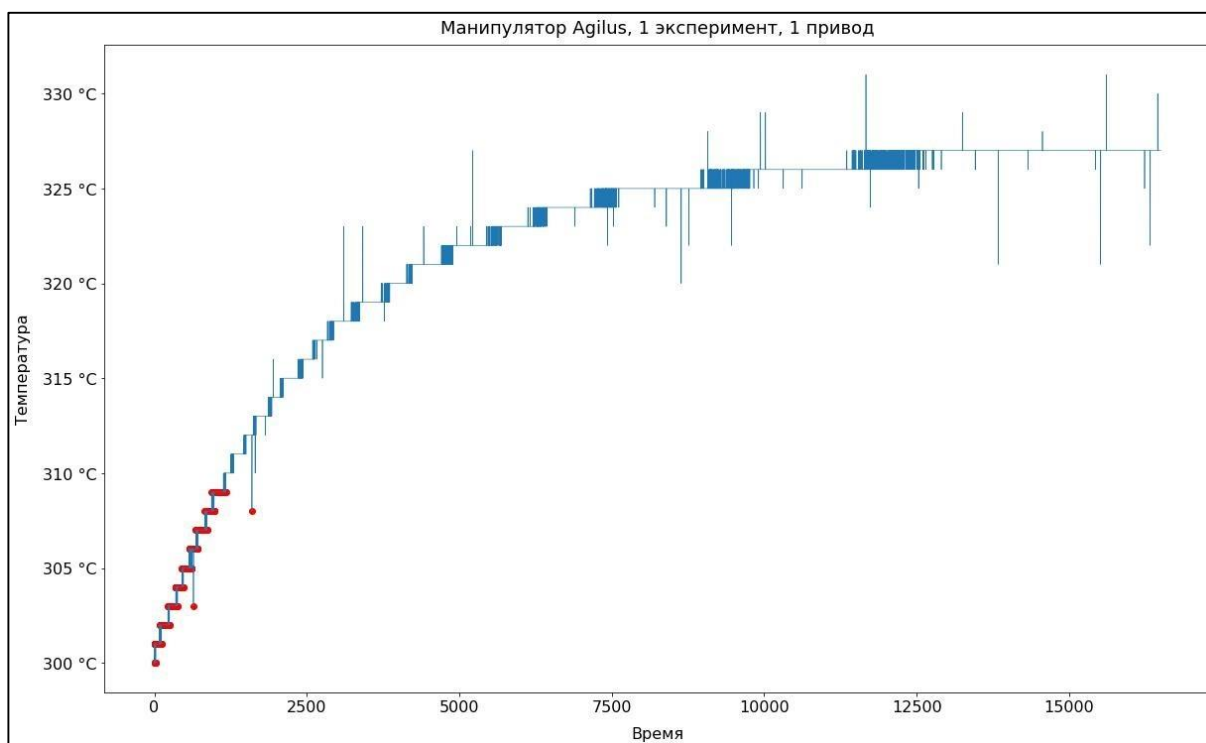


Рисунок 11 – График зависимости температуры от времени

Общая тенденция – рост температуры в процессе работы манипулятора до порогового значения. На графики явно видны локальные выбросы (точечные скачки на несколько градусов), которые могут быть обусловлены неточностью измерительной системы и методами проведения замеров.

Выделенные точки – выбросы, определенные IQR методом. В диапазон экстремальных значений попали все показатели температуры до отметки 310 °С, которые не являются некорректными значениями, а относятся к периоду нагревания манипулятора до рабочих температур. Применение описанного ранее метода устранения экстремальных значений приведет к уменьшению точности модели, при этом локальные выбросы сохранятся.

Для решения этой проблемы предлагается использовать алгоритм «Локального уровня выброса» для подсчета локальной плотности показателей температуры. На рисунке 12 представлен график зависимости температуры от времени с выделенными при помощи алгоритма «Локального уровня выброса» аномальными значениями.

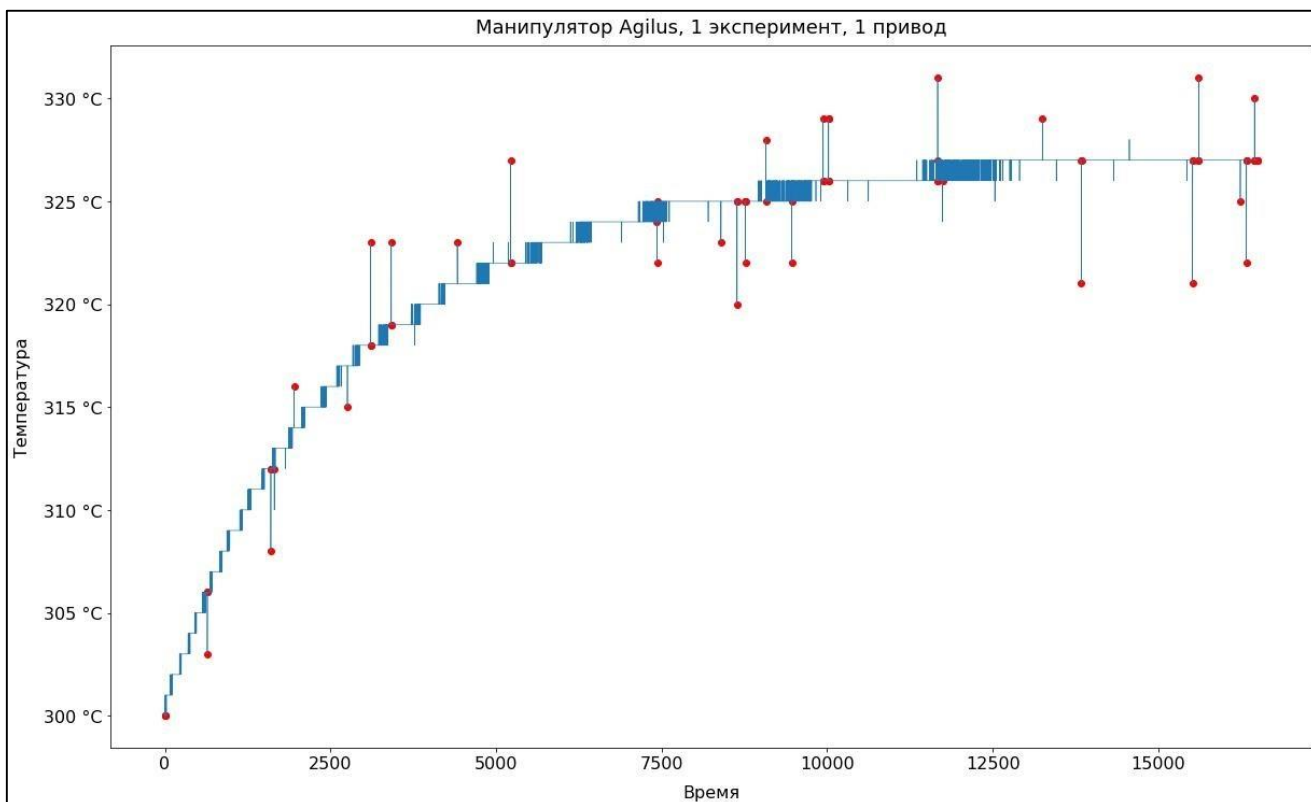


Рисунок 12 – Локальные выбросы температуры

Сравнивая плотность объекта с плотностью 4 ближайших соседей, определяется какие значения являются локальными выбросами.

Из-за низкой точности измерительного прибора (цена деления 1 °С) наблюдаются колебания температуры около истинного значения, это привело к тому, что помимо скачков в несколько градусов, были выделены корректные измерения, расположенные близко к выбросам. Применяемая стратегия сглаживания – среднее значение между двумя ближайшими соседями, позволит заменить выбросы корректными значениями. Применение выбранной стратегии к корректным измерениям не внесет большую погрешность в модель обучения.

На рисунке 13 представлен график температуры после устранения локальных выбросов.

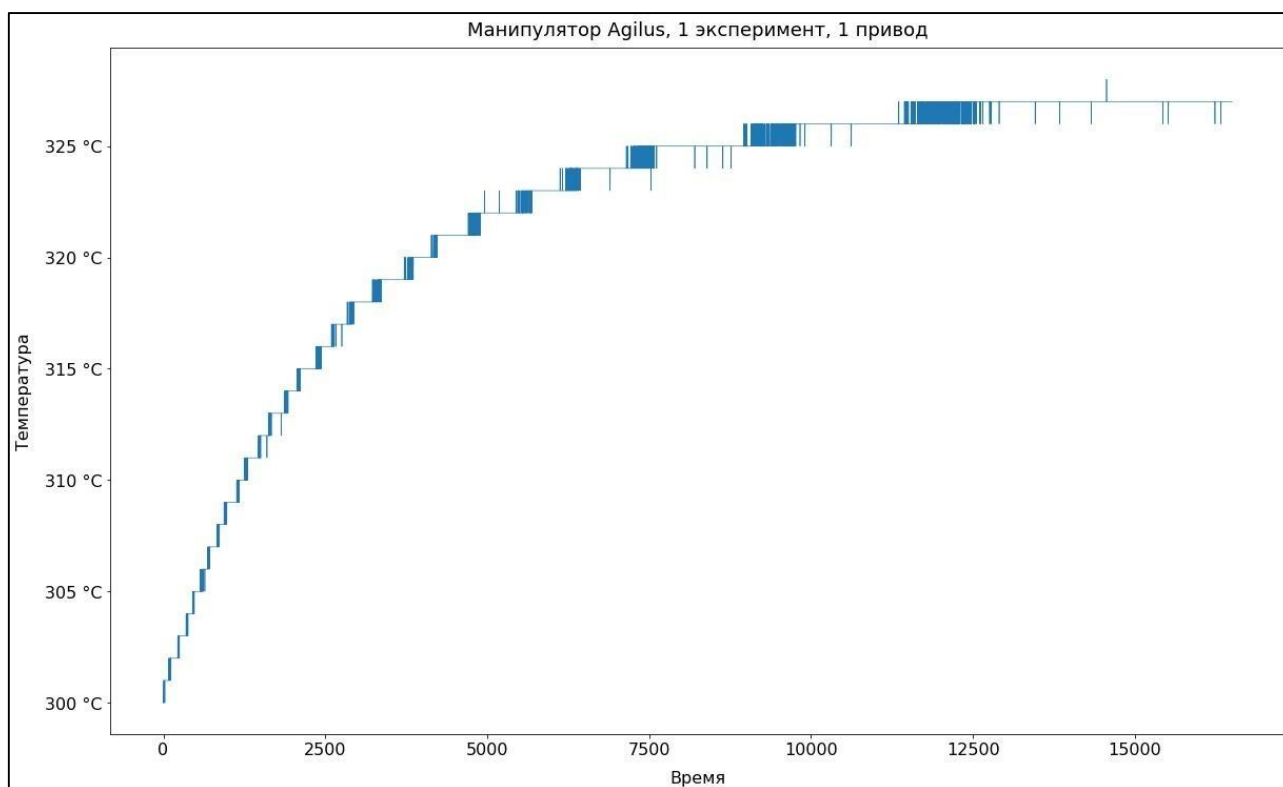


Рисунок 13 – График зависимости температуры от времени

После преобразований показатели температуры находятся в рамках погрешности измерительного прибора.

3.1.5 Преобразование признаков

Цель данного этапа привести данные к подходящему, для использования в модели машинного обучения, виду.

Предсказательные модели требуют стандартизированного представления данных (среднее значение ~ 0 и стандартное отклонение 1). Из-за простоты реализации, в качестве алгоритма стандартизации была выбрана Стандартизованная оценка (z - оценка). Z-Оценка определяется по следующей формуле 3:

$$S = \frac{x - \mu}{\sigma} \quad (3)$$

где: x – измерение;

μ – среднее значение;

σ – среднеквадратичное отклонение.

На рисунке 14 построена диаграмма стандартизированных значений.

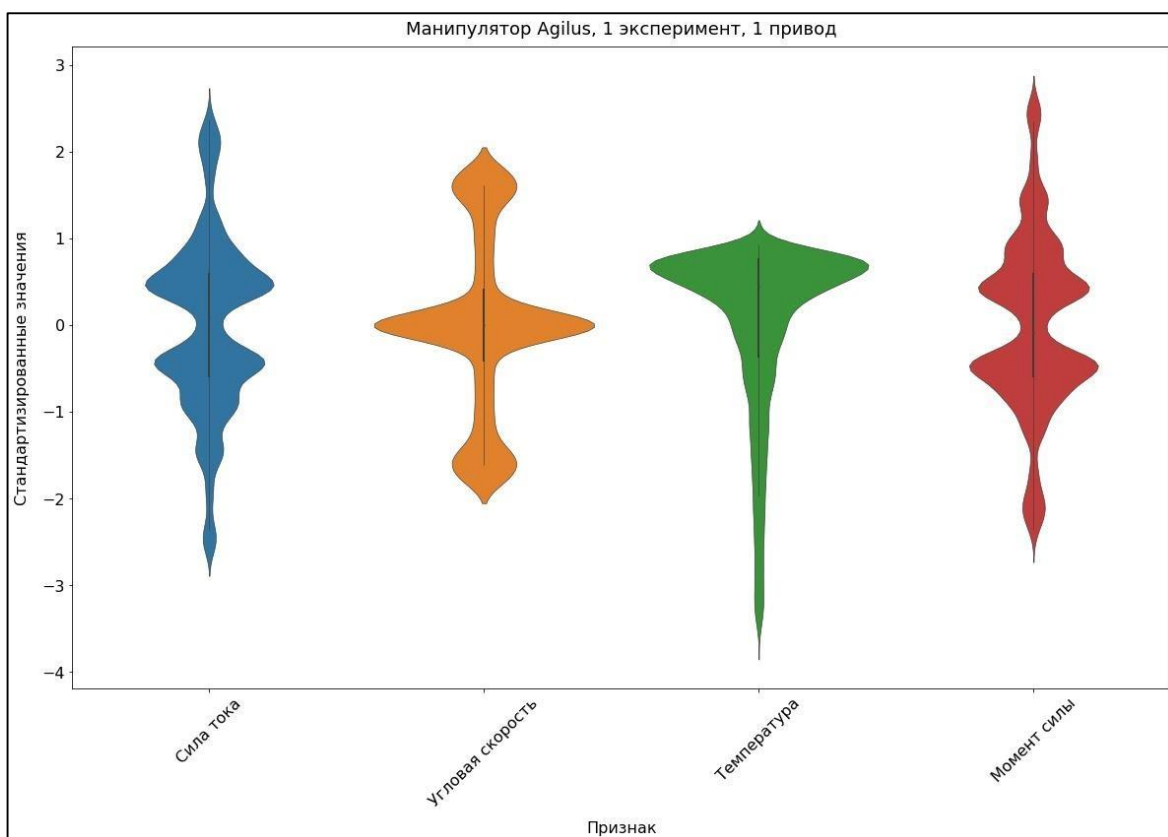


Рисунок 14 – Диаграмма стандартизированных значений

3.1.6 Отбор признаков

Для увеличения точности и скорости работы алгоритма машинного обучения необходимо отобрать набор наиболее полезных признаков. В рамках данной работы используется комбинация 2 подходов: методы фильтрации для предварительного отбора признаков и методы обертки для отбора полезных признаков на основе результатов обучения.

На рисунке 15 представлена схема предварительного отбора признаков методами фильтрации.

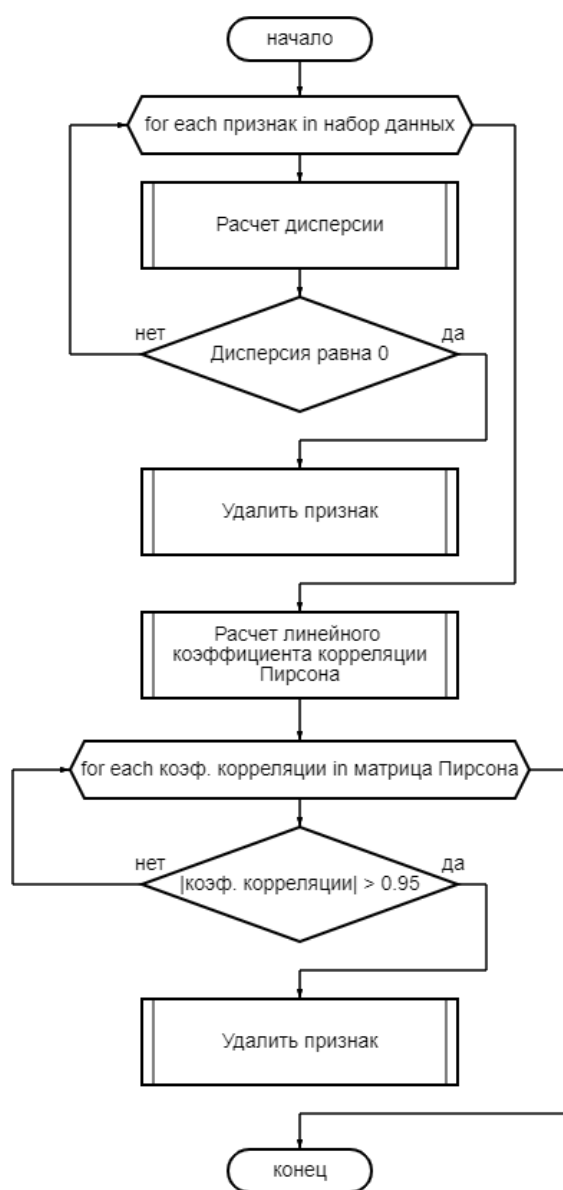


Рисунок 15 – Схема отбора признаков методами фильтрации

Для первоначального отбора признаков рассчитывается величина среднеквадратичного отклонения. В большинстве случаев признаки с нулевой дисперсией являются незначимыми для модели т.к. неизменяемые значения не вносят вклад в модель обучения.

В таблице 5 представлен расчет среднеквадратичного отклонения для числовых признаков. В расчетах не представлено время т.к. этот признак не будет использоваться в модели машинного обучения.

Таблица 5 – среднеквадратичное отклонение числовых признаков.

| Признак | Среднеквадратичное отклонение |
|---------------------|--------------------------------------|
| Сила тока | 3.655080 |
| Температура привода | 6.229197 |
| Момент силы | 44.847265 |
| Угловая скорость | 20.074958 |
| Линейная скорость | 0.000000 |

Линейная скорость имеет среднеквадратичное отклонение 0, следовательно, дисперсия также равна 0, этот признак исключается из набора.

Использование сильно коррелируемых признаков замедляет скорость обучения модели. Если существует четкая зависимость между двумя величинами, то достаточно использовать только одну из них. Для поиска сильно коррелируемых признаков рассчитывается линейный коэффициент корреляции Пирсона.

На рисунке 16 представлена матрица корреляции Пирсона.

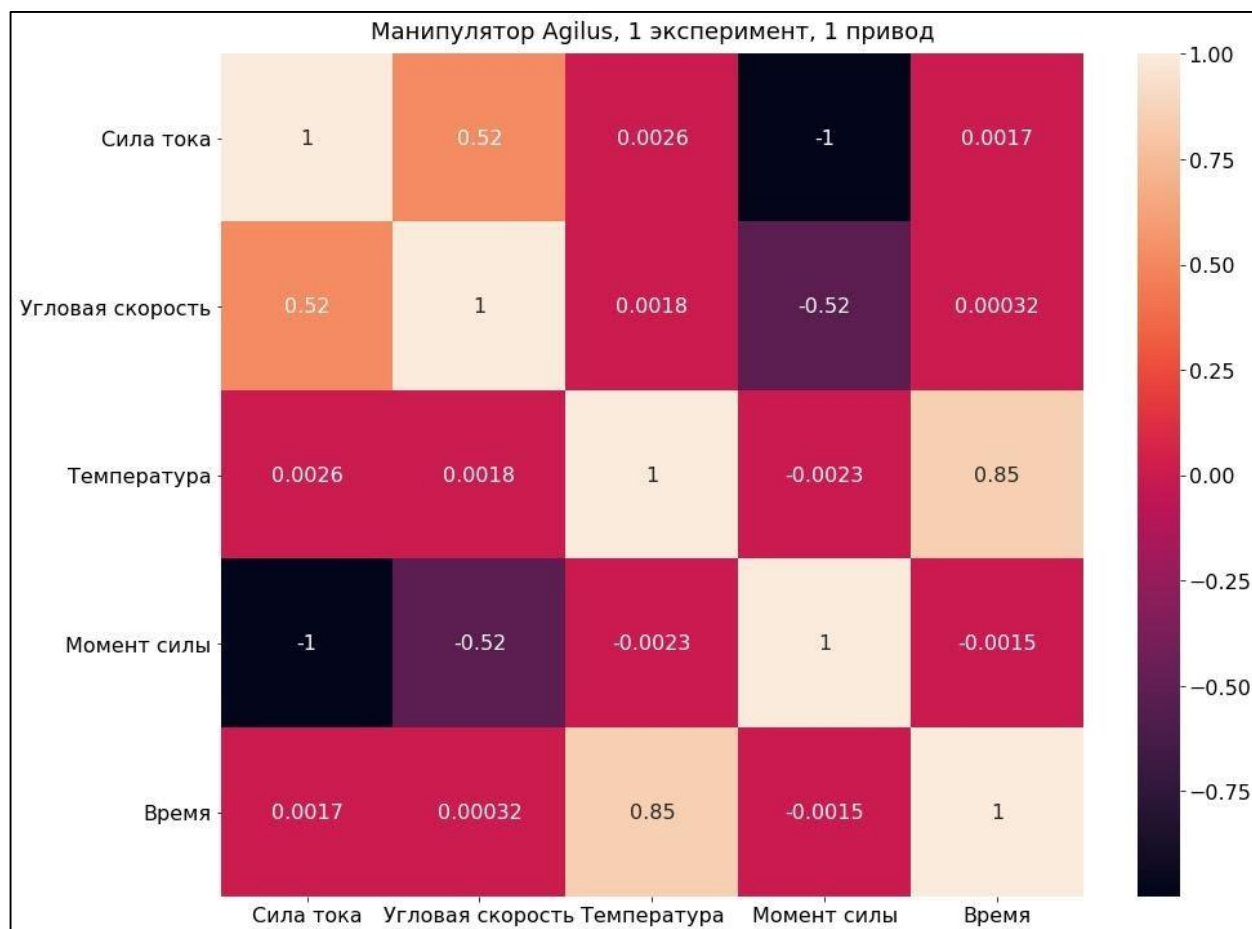


Рисунок 16 – Матрица корреляции Пирсона

Чем ближе модуль значения корреляции к 1, тем сильнее зависимость одного параметра от другого. Количество признаков в исследуемом наборе данных мало, поэтому отбираются только признаки, попадающие в диапазон значений $[-0.95, 0.95]$.

Из матрицы следует, что «Сила тока» и «Момент силы» являются сильно коррелирующими признаками. Использование обоих в обучаемой модели не имеет смысла, поэтому исключаем «Момент силы».

Для дальнейшего отбора признаков применяется метод прямого последовательного отбора признаков (Forward feature selection method). За основу возьмем исследуемый признак – «Температура» и последовательно будем комбинировать его с остальными отобранными признаками для построения модели с самой высокой точностью.

На рисунке 17 представлена схема отбора признаков методом прямого последовательного отбора.

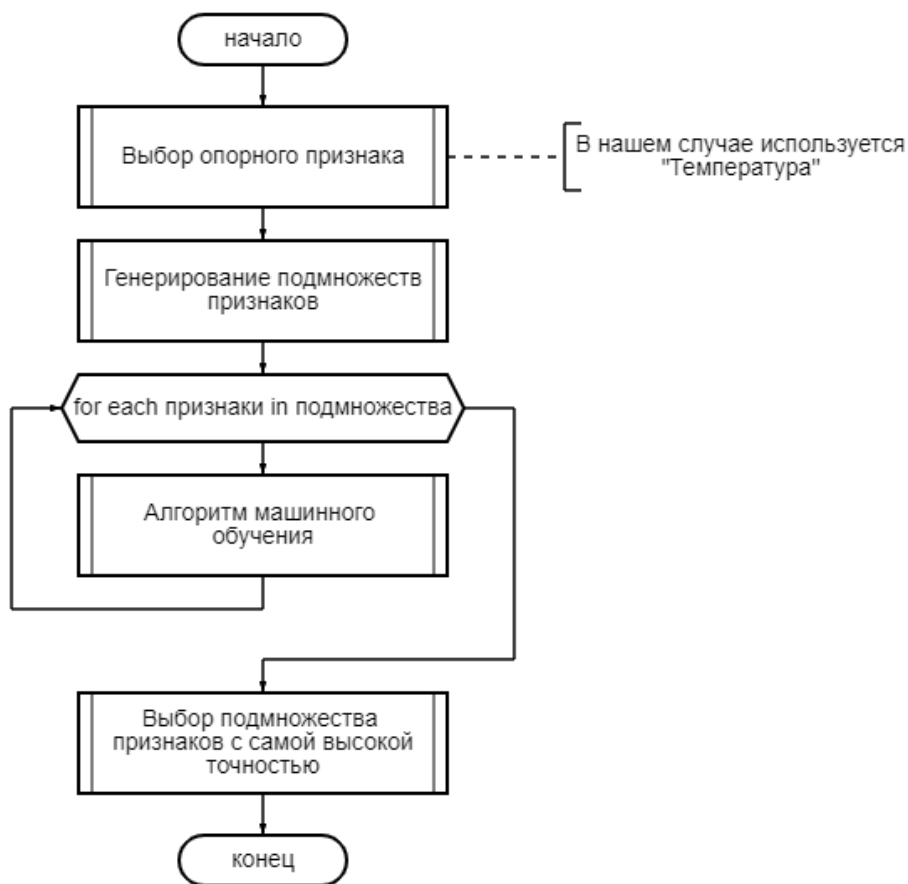


Рисунок 17 – Схема отбора признаков методом обертки

В результате предварительной обработки данных исходный набор был разбит на 6 независимых выборок, заполнены отсутствующие значения, устранены выбросы, проведена стандартизация и отобраны полезные для модели машинного обучения признаки: «Сила тока», «Температура», «Угловая скорость».

3.1.7 Обучение

На данном этапе осуществляется обучение моделей. Для решения поставленной задачи были выбраны следующие алгоритмы:

- Рекуррентная нейронная сеть (RNN) – быстро набирающий популярность инструмент решения задачи прогнозирования временных рядов;

- Сверточная ИНС (CNN) – используется для анализа изображений, однако их способность извлекать общие признаки из набора данных, может быть применена в решении задачи прогнозирования временных рядов. Последовательность наблюдений может быть рассмотрена, как одномерное изображение, которое разлагается на наиболее заметные элементы алгоритмом свертки;
- Многослойный персептрон (MLP);
- Линейная регрессия (далее линейный).

Настройки рекуррентной ИНС:

- Функция активации `relu` – функция возвращающая, переданное значение, если оно положительно, в противном случае возвращает значение ноль;
- Количество слоев – 3;
- Количество узлов – 32;
- Максимальное количество эпох – 25.

Настройки сверточной ИНС:

- Функция активации `relu`;
- Количество слоев – 3 (один слой свертки);
- Размер ядра – 3;
- Количество узлов – 32;
- Максимальное количество эпох – 25.

Настройки многослойного персептрона:

- Функция активации `relu`;
- Количество слоев – 3;
- Количество узлов – 64;
- Максимальное количество эпох – 25.

Набор данных разделяется на 2 части:

- Данные для обучения – 70%;
- Данные для тестирования – 30%.

В таблице 6 представлены подмножества признаков, образованные методом последовательного отбора.

Таблица 6 – Подмножества признаков

| № опыта | Выбранные параметры |
|---------|---------------------|
| 1 | Температура |
| | Угловая скорость |
| 2 | Температура |
| | Сила тока |
| 3 | Температура |
| | Угловая скорость |
| | Сила тока |

Для оценки алгоритмов обучения используется два основных показателя: точность предсказания, время обучения. Для оценки точности определения температуры вводится доверительный интервал. Будем считать, значение верным, если оно попадает в диапазон $[X-1, X+1]$, где X истинный показатель температуры.

На рисунках 18 – 19 представлены графики предсказания температуры Рекуррентной ИНС и плотного алгоритма.

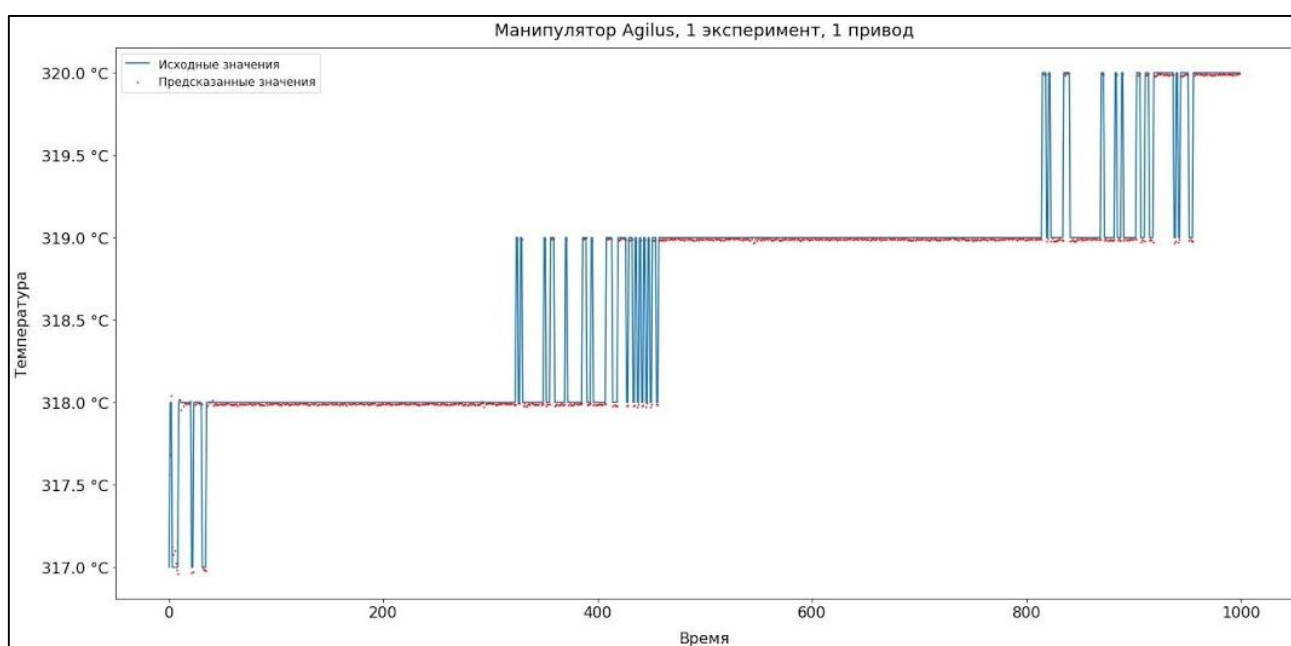


Рисунок 18 – Результат работы рекуррентной ИНС

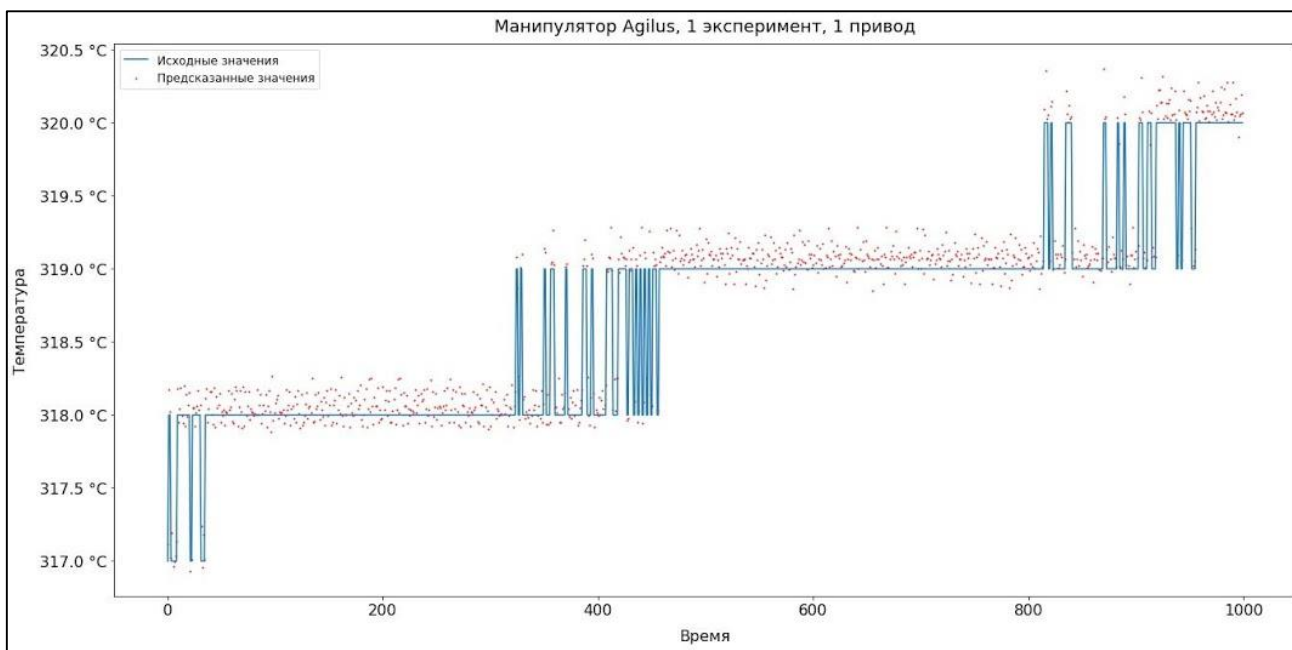


Рисунок 19 – Результат работы плотного алгоритма

В таблице 7 представлены результаты обучения с использованием различных подмножеств признаков.

Таблица 7 – Результаты обучения

| Модель обучения | № опыта | T, сек | Кол-во эпох | Точность, % |
|-------------------------|---------|-----------|-------------|-------------|
| Линейный | 1 | 22.1596 | 11 | 71.2125 |
| | 2 | 6.6132 | 4 | 80.6562 |
| | 3 | 25.47820 | 15 | 71.6437 |
| Многослойный персептрон | 1 | 5.7910 | 3 | 71.2624 |
| | 2 | 14.7603 | 8 | 81.15 |
| | 3 | 9.55662 | 5 | 71.84375 |
| Сверточная ИНС | 1 | 21.2335 | 12 | 68.7312 |
| | 2 | 26.95465 | 15 | 72.6062 |
| | 3 | 29.81979 | 16 | 70.74375 |
| Рекуррентная ИНС | 1 | 712.00319 | 25 | 74.1482 |
| | 2 | 779.93321 | 25 | 85.1512 |
| | 3 | 913.15259 | 25 | 86.01 |

Цветом выделены результаты с наибольшей точностью для каждого метода. Самая высокая точность у Рекуррентной ИНС – 86 % правильных предсказаний для 3 опыта, который включает в себя полный набор признаков,

однако время обучения составило более 5 минут, поэтому данный алгоритм не удовлетворяет заданным ограничениям по времени. Вторым по точности является многослойный персептрон с результатом 81% для 2 опыта – наилучший, удовлетворяющий исходным ограничениям, результат.

3.2 Архитектура системы

Процесс выбор архитектуры разрабатываемой системы состоял из объединения требований к системе в функциональные блоки – компоненты и сравнительного анализа существующих архитектурных подходов организации веб-приложений.

Основополагающие принципы выделения компонентов – разделение ответственности и ограничение контекста. Первый принцип гласит, что каждый компонент должен выполнять только свои функции. Второй – развитие каждого компонента должно происходить независимо от остальных компонентов системы, каждый компонент должен взаимодействовать только со своими хранилищами данных.

На основе описанных в подразделе 1.4 требований к системе, можно выделить 4 основных компонента нашего приложения:

1. Компонент анализа данных и машинного обучения, далее будем называть его Data Analysis;
2. Компонент формирования и отправки уведомлений, далее будем называть его Notification;
3. Компонент управления данными пользователей;
4. Компонент механизма аутентификации/авторизации пользователя и регистрации, далее будем называть его Authentication.

Функционал компонента управления данными пользователя и компонента Authentication сильно связаны. В рамках разрабатываемой системы, компонент управление данными пользователей не содержит в себе большого количества бизнес логики, поэтому, имеет смысл объединить эти компоненты в один.

Ниже представлено описание функционала компонентов системы:

1. Data analysis – загрузка, преобразование данных, обучение модели, формирование результатов;
2. Authentication – регистрация, аутентификации, хранение и обработка пользовательской информации;
3. Notification – формирование и отправка уведомлений, писем.

Каждому из вышеперечисленных компонентов посвящен отдельный подраздел с более подробным описанием.

Помимо выделения основной компонентов разрабатываемой системы, необходимо выбрать архитектуру веб-приложения. Далее будет проведен сравнительный анализ между монолитной, сервис-ориентированной (SOA) и микросервисной архитектурами. Бессерверная архитектура (serverless) и ее компоненты, как часть приложения, не рассматривается т.к. она основывается на BaaS и FaaS облачных вычислениях – управление инфраструктурой лежит на стороне облака.

Концепция монолитной архитектуры заключается в объединении различных компонентов приложения в одну систему на одной платформе. Приложение разрабатывается как единый блок.

На рисунке 20 представлена схема организации монолитной архитектуры для выделенных компонентов разрабатываемой системы.

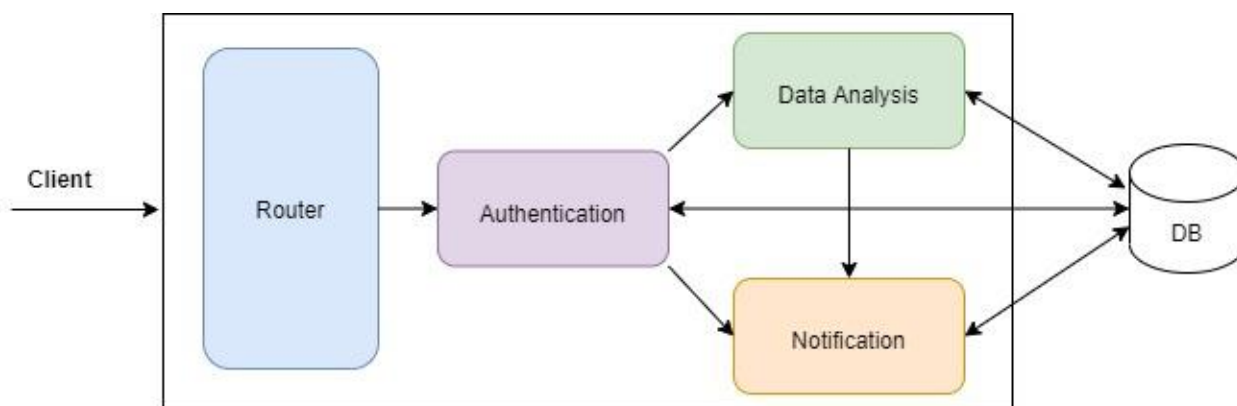


Рисунок 20 – Монолитная архитектура проекта

Сильная связанность компонентов внутри одной системы упрощает процесс разработки и развертывания, отсутствует необходимость организовывать сложные механизмы взаимодействия между компонентами, однако существуют значительные минусы: проблемы горизонтального масштабирования системы, неравномерная нагрузка на определенный компонент системы влияет на работу всей системы в целом, невозможность использовать специфический набор технологий под решаемую задачу.

Вышеописанные проблемы решаются при помощи разбиения единого приложения на независимые службы. Данный подход применяется в SOA и микросервисной архитектуре.

Рассмотрим концепцию SOA построения архитектуры приложения. SOA представляет собой набор самодостаточных сервисов приложения объединенных через брокера сообщений или сервисную шину предприятия (ESB) в единую систему.

На рисунке 21 представлена схема SOA архитектуры для выделенных компонентов разрабатываемой системы.

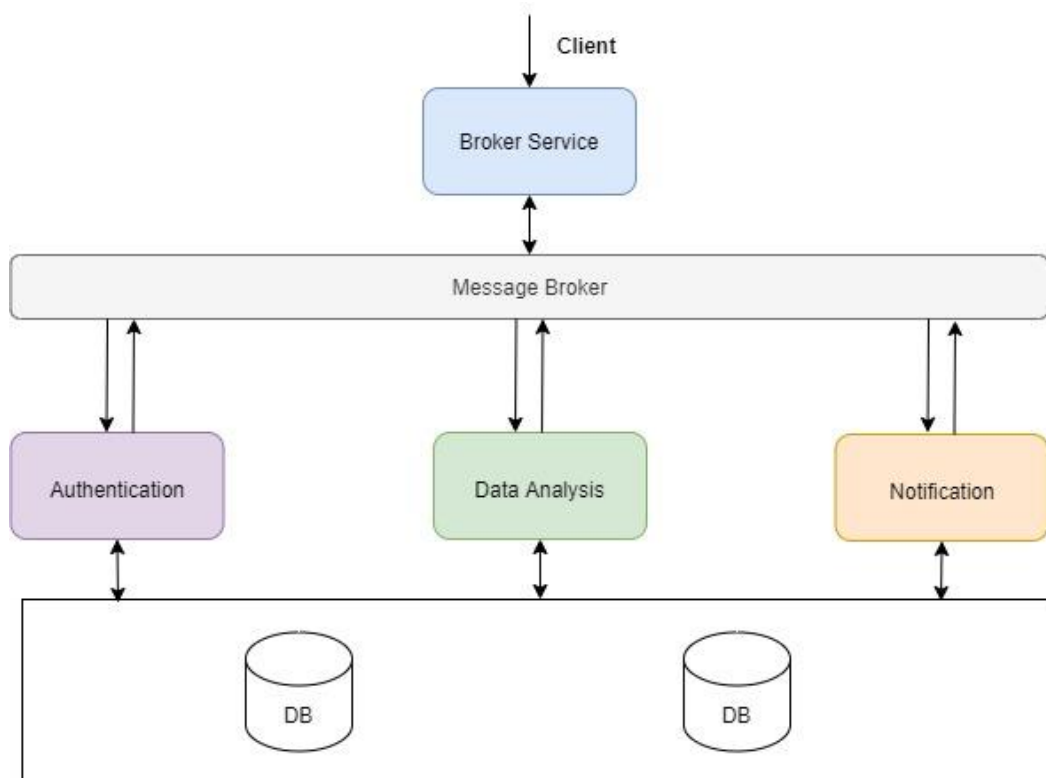


Рисунок 21 – SOA архитектура проекта

Broker service принимает запрос от клиента, передает в брокер сообщений. Каждый из сервисов имеет своего слушателя очереди, в случае появления нового сообщения, он читает его, выполняет требуемую логику и инициализирует новое сообщение, которое переводается другому сервису или обратно broker service.

Использование описанного выше архитектурного подхода решает проблему связанности компонентов, позволяет легко балансировать нагрузку при помощи инициализации копии сервиса, однако система так же имеет единые точки отказа – message broker и broker service.

Микросервисная архитектура является подтипом сервисной архитектуры. Аналогично с SOA приложение разбито на самостоятельные сервисы, однако, теперь они не связаны через единую шину и имеют независимые друг от друга базы данных.

На рисунке 22 представлена схема микросервисов разрабатываемого приложения.

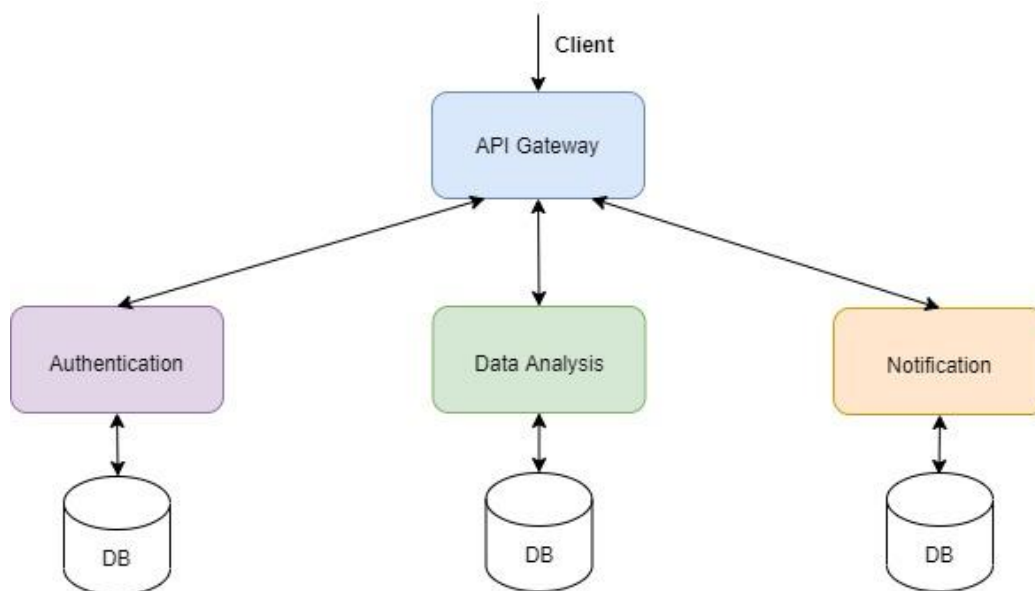


Рисунок 22 – Микросервисная архитектура проекта

В данном случае, общение между микросервисами и клиентом осуществляется через единую точку входа – API Gateway. Подробное описание используемого подхода описывается в разделе 4.1. Сбор данных и опрос микросервисов теперь полностью осуществляется клиентом.

Реализация микросервисов проще, чем SOA за счет отсутствия message broker, однако децентрализованный подход хранения и обработки данных сильно усложняет процесс согласования данных.

В таблице 8 представлены преимущества и недостатки описанных выше концепций.

Таблица 8 – Преимущества и недостатки

| Концепция | Преимущества | Недостатки |
|------------------------|---|--|
| Монолитная | Простота реализации и развертывания. Единая кодовая база приложения. Быстрая связь между программными компонентами системы. Простота обеспечения согласованности данных. | Сложность внедрения новых технологий. Использование одного языка программирования. Низкая гибкость управления и развертывания. Любой компонент системы – единая точка отказа. |
| Сервис-ориентированная | Простота сопровождения. Высокая надежность. Использование различных языков для разработки сервисов. | Высокая сложность реализации. Сложность управления. Увеличенное время отклика. |
| Микросервисная | Высокая надежность. Использование различных языков для разработки сервисов. Простота сопровождения и разворачивания. Возможность горизонтального масштабирования. | Децентрализованное хранение данных. Увеличенное время отклика. Сложность обеспечения безопасности. |

На основании таблицы сравнительного анализа преимуществ и недостатков разных архитектурных решений и ресурсов проекта, была выбрана микросервисная архитектура.

Для решения главной проблемы распределенных систем – децентрализация хранения данных и как следствие невозможность использовать механизм транзакций для обеспечения согласованности данных в случае, если запрос модифицирует данные в нескольких микросервисах одновременно, существует 2 популярных подхода:

1. Распределенные транзакции;
2. Компенсирующие операции.

Первый подход заключается в ручном контроле транзакций баз данных.

На рисунке 23 представлена схема распределенных транзакций микросервисной архитектуры.

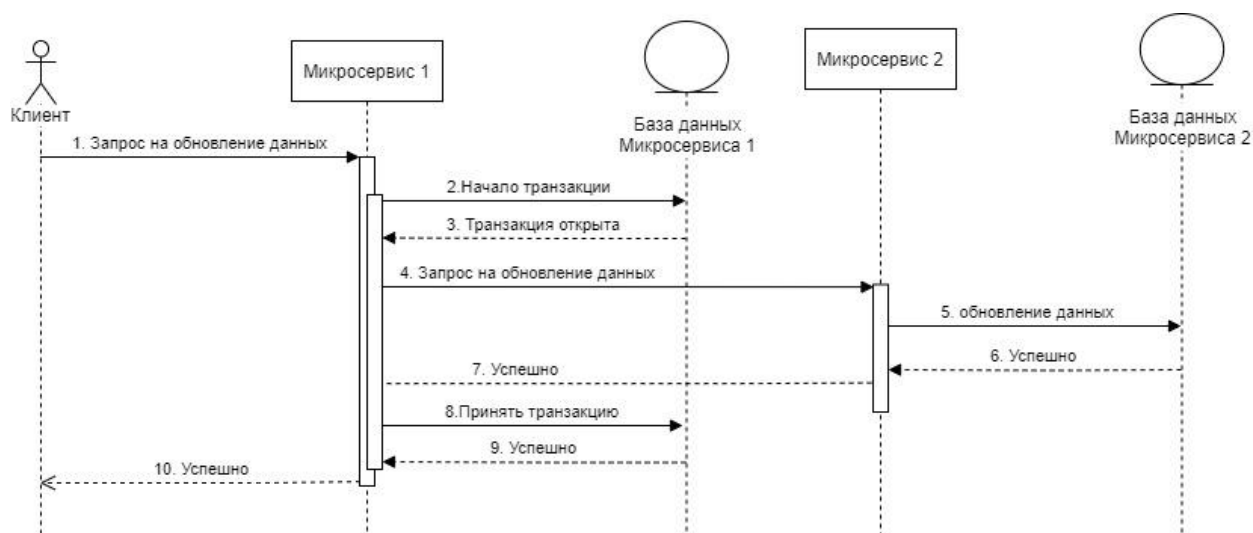


Рисунок 23 – Распределенные транзакции

Данный подход обладает следующими недостатками:

- Необходимо контролировать жизненный цикл открытых транзакций;
- Резкое уменьшение скорости ответа БД с увеличением количества запросов из-за большого количества открытых транзакций;
- С увеличением количества микросервисов участвующих в распределенной транзакции, сложность операций возрастает.

Несомненным преимуществом данного подхода является использование механизмов реляционных баз данных для обеспечения согласованности, нет

необходимости реализовывать это со стороны микросервиса, что уменьшает вероятность ошибки.

Второй подход заключается в делегировании ответственности за поддержание согласованности данных на сторону приложения.

К сущностям добавляется дополнительный атрибут указатель. Все операции модификации в нескольких микросервисах помечают данные специальным указателем типа выполняемой операции (create, update, delete), который указывает приложению, что данная запись находится в процессе обновления и использовать ее нельзя.

На рисунке 24 представлена схема компенсирующих операций.

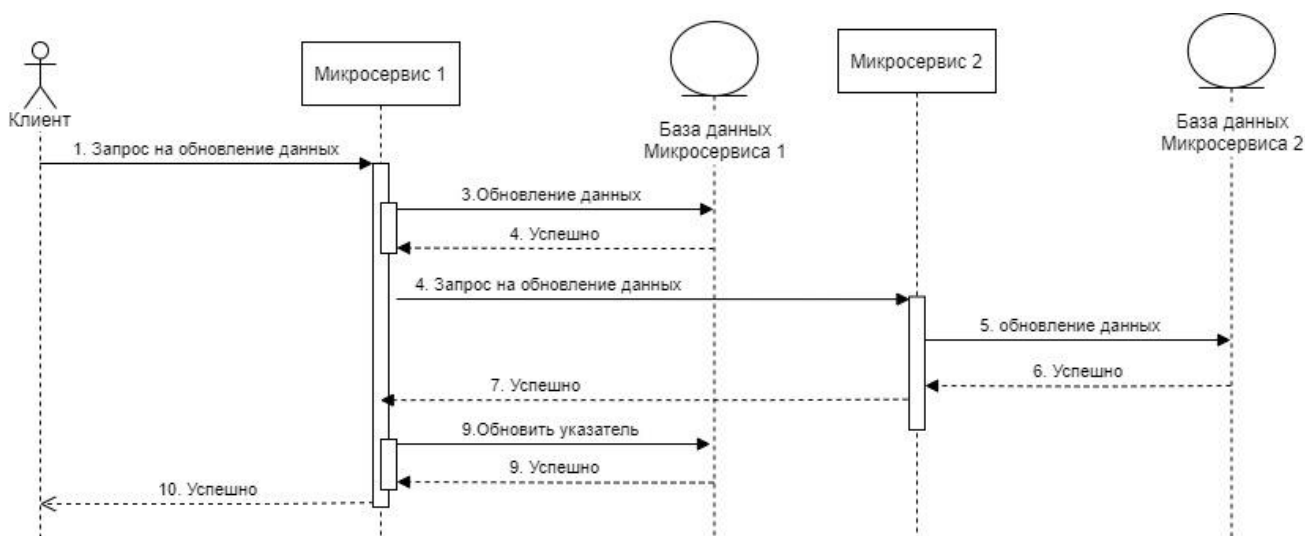


Рисунок 24 – Использование указателей

Для разрабатываемого приложения выбран подход компенсирующих операций, т.к. его проще реализовывать и в системе нет высокой связанности между данными хранимых в разных БД.

В разделе 4.2 представлены подходы решения проблемы сложности обеспечения безопасности.

3.3 Проектирование базы данных

Перед реализацией системы производится декомпозиция предметной области на сущности и определяются взаимосвязи между ними. Для графического представления логической структуры исследуемой области

разрабатывается логическая модель данных. Она позволяет без привязки к специфике конкретной СУБД описать шаблон для создания физической модели данных.

Главная особенность проектируемой схемы данных – неопределенное количество атрибутов в загружаемых данных. Существует несколько подходов хранения таких данных в реляционных БД: EAV модель, денормализованное представление данных, хранение набора данных в агрегированном виде.

EAV модель – классический подход, однако обладающий низкой скоростью из-за необходимости соединения нескольких таблиц.

В данной работе предлагается использовать агрегирование произвольного количества атрибутов в json объект. Сравнение скорости операций выборки данных для EAV модели и JSONB типа данных СУБД PostgreSQL представлены на рисунке 25.

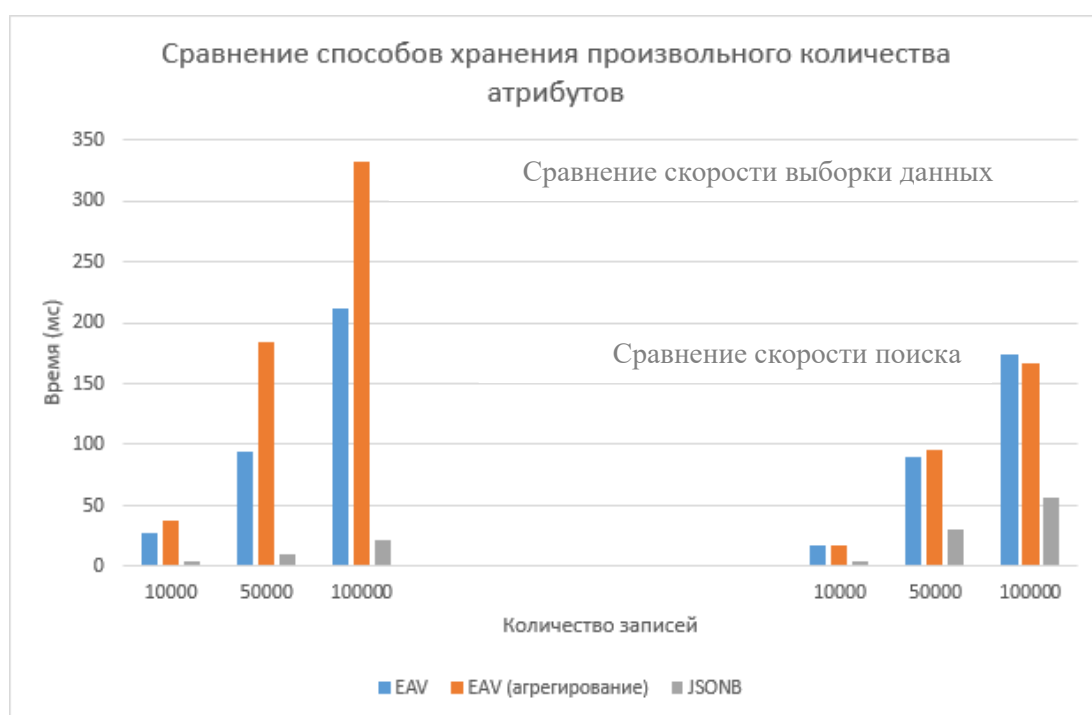


Рисунок 25 – Сравнение EAV и JSONB

Разработанная логическая диаграмма базы данных представлена на рисунке 26. Подробное описание сущностей базы данных представлено в приложении А.

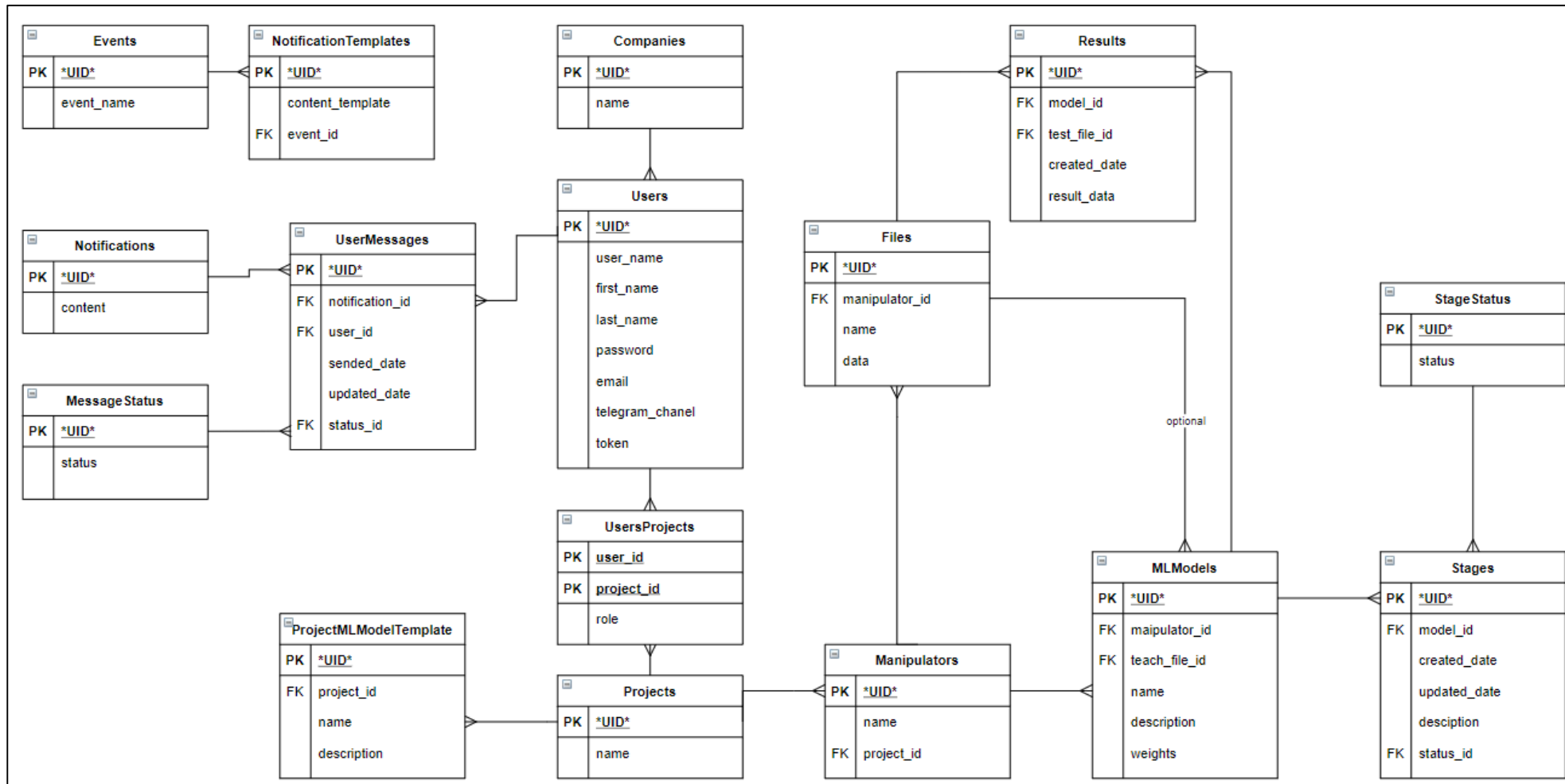


Рисунок 26 – Логическая модель базы данных (нотация Мартина)

3.4 Выбор средств реализации системы

3.4.1 Серверная часть

Разрабатываемая система состоит из нескольких независимых микросервисов реализующих общий интерфейс взаимодействия с клиентом. Таким образом реализация микросервисов системы будет осуществляться с использованием наиболее подходящих, для предполагаемого функционала, инструментов. В таблице 9 приведен список используемых для микросервисов технологий.

Таблица 9 – Выбранные технологии для элементов системы

| Элемент системы | Язык программирования | Фреймворк |
|----------------------------|-----------------------|-----------|
| API Gateway | Python | FastAPI |
| Микросервис авторизации | PHP | Lumen |
| Микросервис уведомлений | Python | FastAPI |
| Микросервис анализа данных | Python | Django |

В качестве языка для реализации большей части функционала был выбран Python, так как он обладает большим количеством применимых в данной работе библиотек, написанных на языке Си. Данные библиотеки совмещают высокую скорость вычисления и синтаксические возможности Python. Также выбор языка Python обуславливается наличием большого количества современных фреймворков для разработки веб-сервисов.

Фреймворк FastAPI обладает рядом преимуществ, которые имеют ценность для микросервисов API Gateway и микросервиса уведомлений:

- Высокая скорость работы, в сравнении с другими основными фреймворками Python, такими как Flask и Django. Это преимущество особенно необходимо микросервису API Gateway, чтобы тот мог быстро перенаправлять запросы к другим сервисам и также быстро отдавать результаты;
- Поддержка асинхронного кода «из коробки»;

- Малые затраты времени на разработку;
- Простота развертывания приложения FastAPI.

Отсутствие миграций базы данных из коробки является проблемой, но не существенной, так как API Gateway не взаимодействует с базой данных напрямую, а микросервис уведомлений взаимодействует с базой данных, обладающих малым количеством сущностей, что позволит быстро написать миграции с использованием сторонних библиотек Python.

Для более «тяжелого» и функционального микросервиса, плотно работающего с большим количеством сущностей базы данных, был выбран Django. Основное преимущество Django для микросервиса анализа данных – наличие многофункциональной ORM «из коробки» и встроенный функционал для миграций базы данных, что позволит существенно упростить и ускорить разработку сервиса.

Основные требования к микросервису Authentication: быстрое действие и высокая доступность. Для быстрой разработки данного микросервиса был выбран язык программирования PHP, микрофреймворк Lumen. Данный инструмент является высоконадежным решением предоставляющий всю необходимую базовую инфраструктуру для реализации API и при этом обладающий высокой скоростью по сравнению с тяжеловесными решениями, такими как Laravel, Yii2, Symfony.

3.4.2 СУБД

Прежде чем перейти к сравнению конкретных СУБД необходимо выбрать тип базы данных: реляционная или нереляционная.

Основным требованием к СУБД является гарантия целостности данных. Дополнительное требование, механизм обеспечения согласованности данных должен соответствовать принципам ACID:

- А – Atomicity (атомарность): гарантия, что каждый запрос, входящий в транзакцию, будет выполнен успешно. В случае если

возникает ошибка хотя бы в одной из запросов, то ни один запрос не будет выполнен;

- C – Consistency (консистентность): гарантия, что все данные корректны и соответствуют со всеми правилами и ограничениями;
- I – Isolation (изолированность): гарантия, что все транзакции будут выполнены изолированно от других. Одна транзакция не может влиять на другую;
- D – Durability (стойкость): гарантия, что после успешного применения транзакции, изменения останутся примененными даже после сбоя системы.

Другим важным требованием является соответствие схемы данных предметной области для использования всех возможностей Django ORM. Соответствие этому требованию позволит сильно увеличить скорость разработки. Последним требованием является поддержка многострочных транзакций.

В таблице 10 представлено сравнение SQL и NoSQL баз данных.

Таблица 10 – Сравнение SQL и NoSQL баз данных

| | SQL Базы данных | NoSQL Базы данных |
|-------------------------------------|---|--|
| Масштабирование | Вертикальное: увеличить мощности одного сервера | Горизонтальное: увеличить количество серверов Также подходит вертикальное |
| Поддержка принципа ACID | Поддерживается | Большинство из NoSQL баз данных не поддерживают транзакции ACID. |
| Операции объединения JOIN | Обычно требуются | Обычно не требуются |
| Сопоставление данных с объектами | Требуется ORM | Ограниченный функционал ORM |

В сравнительной таблице не рассматривается производительность реляционных и нереляционных баз данных, так как этот показатель зависит от конкретной технологии. Исходя из вышеописанных требований и сравнительного анализа была выбрана реляционная база данных.

Следующий этап – выбор СУБД. При выборе СУБД в первую очередь рассматривалось бесплатное ПО, поэтому, такие решения, как Oracle или SQL Server не рассматриваются далее. После анализа существующих решений были отобраны СУБД с открытым исходным кодом – PostgreSQL и MySQL.

На данном этапе нет требований к количеству обрабатываемых транзакций и объему хранимой информации, поэтому при выборе между MySQL и PostgreSQL не учитывалась производительность СУБД, так как скорость операций чтения, записи данных этих двух СУБД отличается незначительно на небольшом и среднем объемах данных. В будущем, скорость обработки транзакций, может быть увеличена при помощи горизонтального масштабирования системы. Обе платформы отлично поддерживают репликацию, и многие облачные провайдеры предлагают управляемые масштабируемые версии любой базы данных.

Однако существует ряд значимых отличий, с которыми необходимо ознакомиться перед финальным выбором СУБД. Таблица со сравнением основных характеристик PostgreSQL и MySQL представлена ниже.

Таблица 11 – Сравнение MySQL и PostgreSQL

| MySQL | PostgreSQL |
|--|--|
| Совместим с ACID только при использовании с механизмами NDB и InnoDB Cluster Storage | Полностью совместим с ACID |
| Не поддерживает пользовательские типы данных | Поддерживает пользовательские типы данных |
| Не поддерживает индексацию составных типов данных | Поддерживает индексацию массивов и JSON-объектов |

Стоит отметить, что PostgreSQL является объектно-реляционной базой данных. Она поддерживает ряд расширенных типов данных, недоступных в MySQL, такие как геометрические/ГИС, типы сетевых адресов, индексруемый JSONB, собственный UUID, временные метки с учетом часовых поясов. Помимо расширенных типов данных в PostgreSQL можно добавить пользовательские типы данных, операторы и индексы.

Несмотря на все вышеперечисленные преимущества, у PostgreSQL все же есть некоторые небольшие недостатки, которые следует учитывать:

- Для PostgreSQL доступно меньшее количество сторонних инструментов в сравнении с MySQL;
- PostgreSQL построен с учетом расширяемости, соответствия стандартам, масштабируемости и целостности данных – в ущерб скорости.

Вышеперечисленные недостатки несущественно влияют на производительность системы, так как главным критерием остается возможность хранения большого объема данных, и высокая расширяемость с возможностью создавать собственные типы данных. Таким образом в качестве СУБД выбрана PostgreSQL.

Схема физической модели базы данных представлена в приложении Б.

Глава 4. Реализация системы

4.1 Сервис-прокси API Gateway

В рамках реализации системы, необходимо разработать общий интерфейс взаимодействия клиентской части с микросервисами. Например, если в системе помимо веб-приложения предусмотрено мобильное приложение как источник запросов к системе, необходимо разработать два разных интерфейса взаимодействия в каждом из разрабатываемых микросервисов. Помимо вышеперечисленных примеров, сервис может предоставлять REST API для сторонних приложений. Общий интерфейс взаимодействия позволит организовывать общения серверной части с различными типами клиентов. Проблему предоставления доступа к разработанным сервисам решает API Gateway.

API Gateway – это инструмент управления API, который находится между клиентом и набором серверных служб. Он является единой точкой входа для всех клиентов. Принцип действия – обратный прокси-сервер, чтобы принимать все вызовы API, агрегировать различные службы, необходимые для их выполнения, и возвращать соответствующий результат.

Одной из реализаций API Gateway является «Backend to Frontends». В этом варианте для каждого типа приложений предоставляется отдельный API Gateway (рисунок 27).

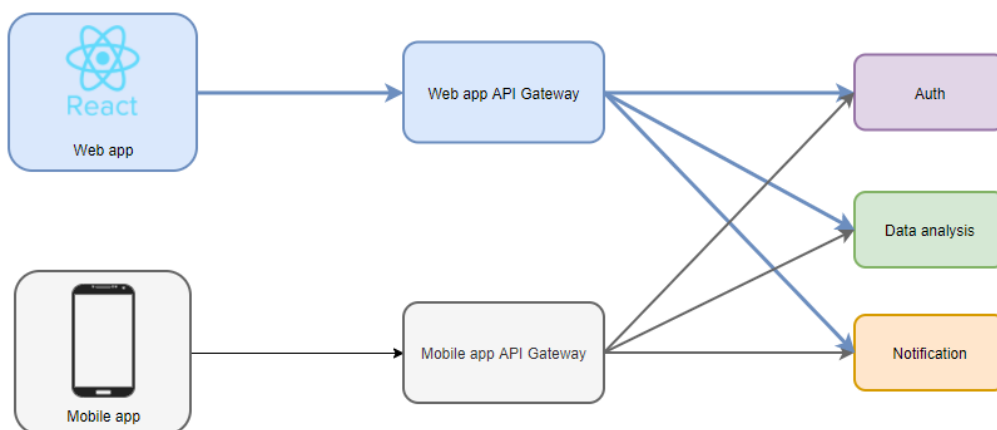


Рисунок 27 – Схема работы паттерна «Backend to Frontends»

В приведенном примере есть 2 типа клиентов: веб-приложение и мобильное приложение. Для них предоставлены свои собственные API Gateway.

Использование шлюза API дает следующие преимущества:

- Изолирует клиентов от разделения приложения на микросервисы;
- Изолирует клиентов от проблемы определения местоположения экземпляров сервисов;
- Предоставляет оптимальный API для каждого типа клиента;
- Уменьшает количество запросов / обходов. Например, API Gateway позволяет клиентам извлекать данные из нескольких служб за один цикл приема-передачи. Меньшее количество запросов также означает меньше накладных расходов и улучшает взаимодействие с пользователем;
- Упрощает клиент, перемещая логику вызова нескольких служб с клиента на API Gateway;
- Переводит из «стандартного» общедоступного веб-дружественного протокола API к любым протоколам, используемым внутри компании.

Шаблон шлюза API имеет ряд недостатков:

- Повышенная сложность – API Gateway – это еще одна движущаяся часть, которую необходимо разработать, развернуть и управлять, и поддерживать;
- Увеличенное время отклика из-за дополнительного сетевого перехода через API Gateway – однако для большинства приложений стоимость дополнительного обхода незначительна.

На рисунках 28 – 29 представлен графический интерфейс для API Gateway, созданный при помощи встроенного в фреймворк FastAPI инструмента Swagger.

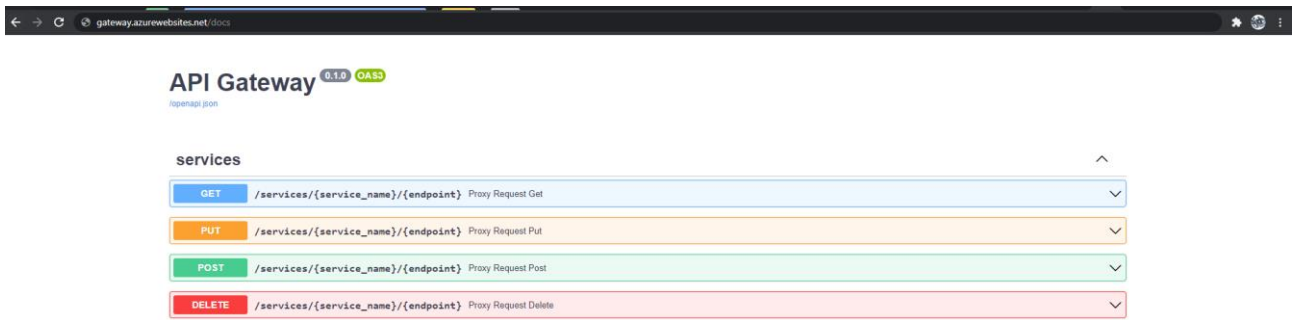


Рисунок 28 – Swagger API Gateway

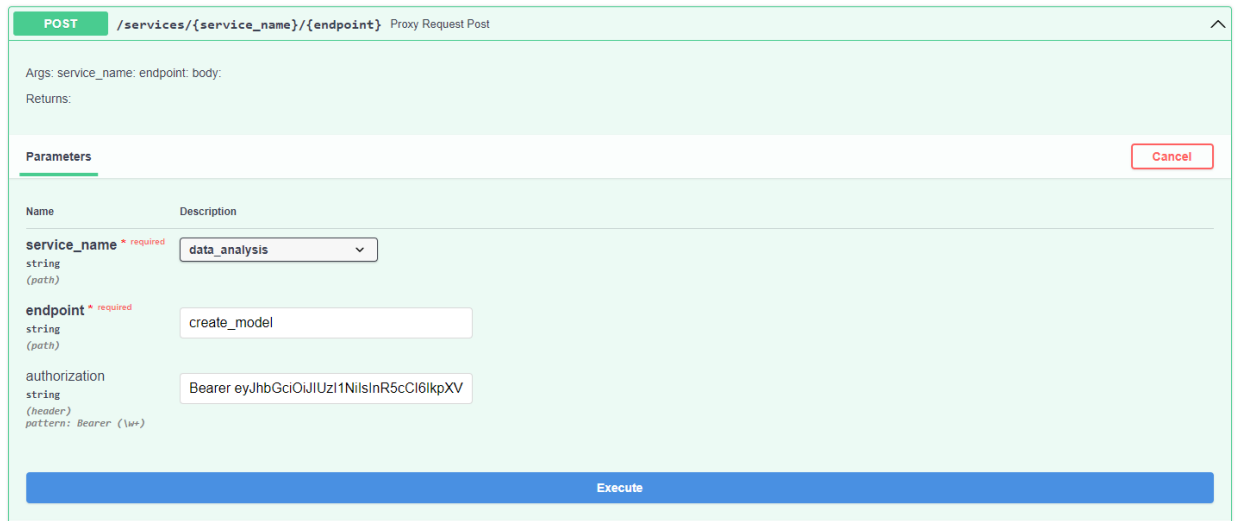


Рисунок 29 – Пример выбора сервиса для запроса

При реализации API Gateway использовался инструмент фреймворка Fast API – dependencies. Пример представлен ниже.

```
router.include_router(
    services_router,
    prefix='/services',
    tags=['services'],
    dependencies=[Depends(verify_auth_header)],
)
```

Строка «dependencies» переводит все запросы к сервисам системы на другой метод для проверки JWT-токена пользователя. После проверки JWT-токена запрос переводится дальше. Фрагмент кода отправки запроса на проверку JWT-токена представлен ниже.

```

def check_token(token: str) -> bool:
    response = await get_response(
        POST,
        settings.auth_token_check_url,
        json={'token': token},
    )
    return response.get('result')

def get_response(method: str, url: str, **kwargs) -> dict:
    async with httpx.AsyncClient() as client:
        headers = {'Content-Type': 'application/json', 'Accept': 'application/json'}
        headers.update(kwargs.pop('headers', {}))
        response = await client.request(method, url, headers=headers, **kwargs)
        try:
            data = response.json()

        except JSONDecodeError:
            raise ResponseJSONError(response.status_code)
        try:
            response.raise_for_status()
        except httpx.HTTPError:
            raise HTTPError(response.status_code, data)
        else:
            return response.json()

```

В качестве примера работы сервиса ниже рассмотрен процесс получения информации о пользователе.

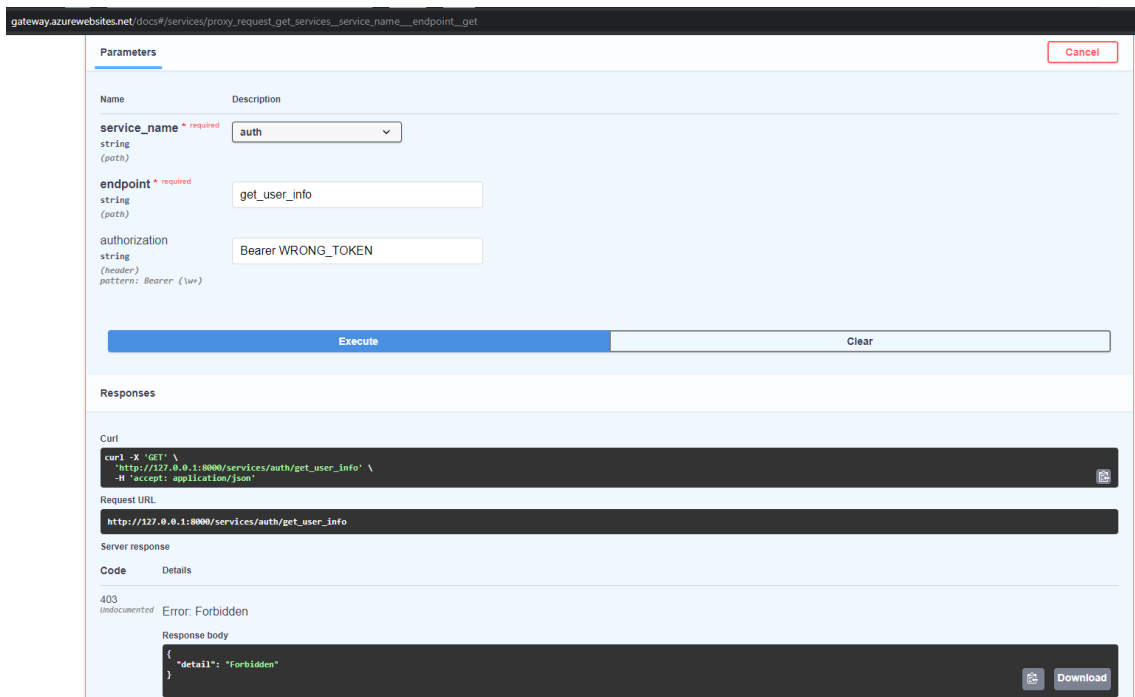


Рисунок 30 – Работа микросервиса при использовании действующего токена

В примере выше использовался JWT-токен с истекшим сроком действия, запрос был прерван на уровне API Gateway. В следующем примере в запросе использовался действующий JWT-токен.

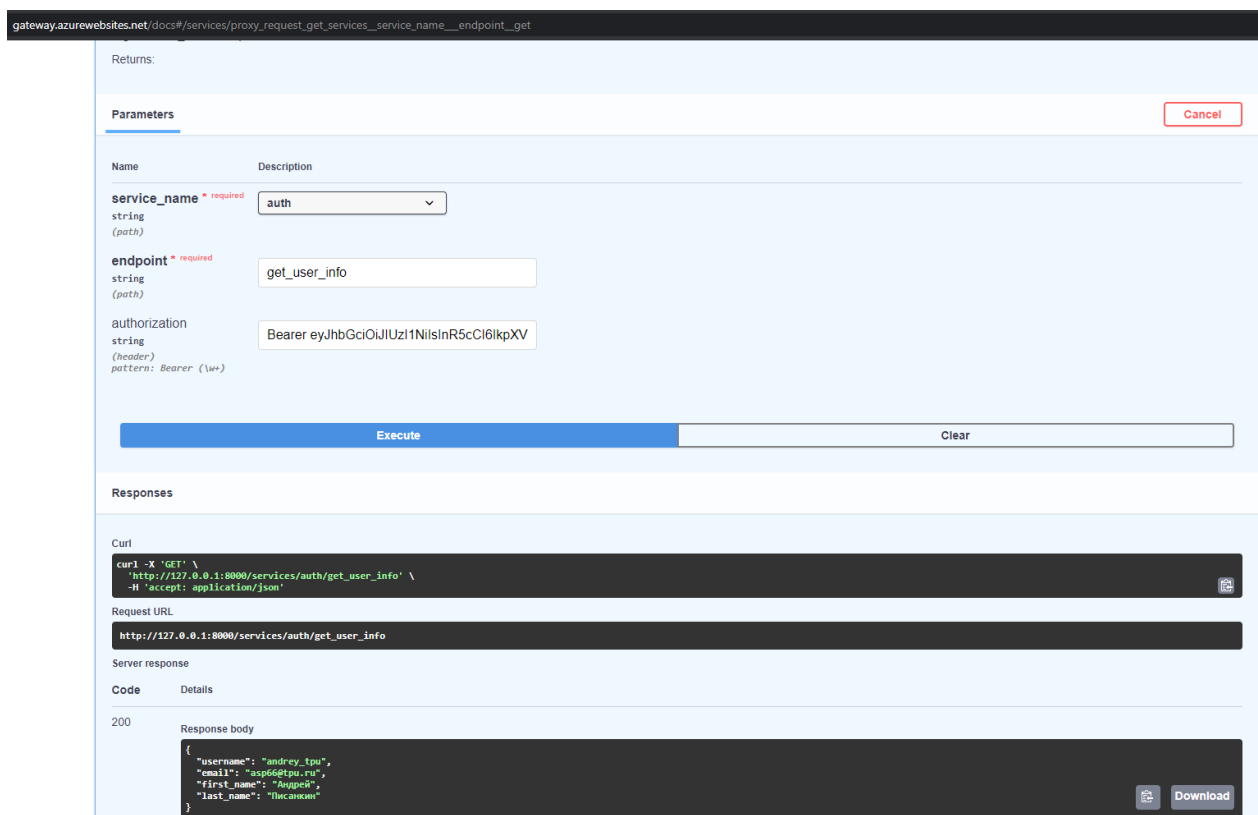


Рисунок 31 – Работа микросервиса при использовании действующего токена

4.2 Микросервис авторизации

Внедрение новых элементов в распределенную систему пропорционально увеличивает сложность управления, появляется проблема обеспечения безопасности каждой предоставляемой точки входа.

В рамках разрабатываемого приложения, основными требованиями к системе аутентификации и авторизации являются:

1. Соответствие выбранной микросервисной парадигме;
2. Высокая отказоустойчивость;
3. Гибкая настройка прав доступа для каждого микросервиса.

Рассмотрим 3 стратегии аутентификации и авторизации пользователя.

Локальная аутентификация и авторизация – каждый микросервис осуществляет аутентификацию и авторизацию пользователя самостоятельно.

На рисунке 32 представлена схема организации сервисов для локальной схемы аутентификации и авторизации пользователей.

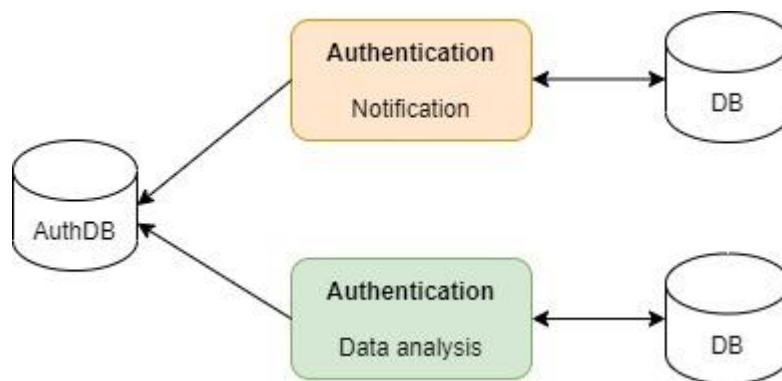


Рисунок 32 – Схема локальной аутентификации и авторизации

Для хранения данных пользователей используется специализированная база данных, на рисунке 32 это AuthDB. Использование данного подхода приводит к образованию единой точки отказа – выход из строя этой базы данных станет причиной неработоспособности всей системы, что недопустимо для распределенных систем и противоречит принципам организации микросервисной архитектуры. С увеличением количества сервисов, количество запросов к указанной базе данных будет пропорционально расти, что закономерно приводит к увеличению времени ответа.

Помимо, описанного выше главного недостатка, появляется необходимость реализации логики аутентификации, авторизации для каждого сервиса отдельно. Это позволяет гибко реализовывать специфическую логику для каждого микросервиса отдельно, однако, в большинстве случаев приводит к необходимости дублирования реализации, особенно, если в реализации системы используются различные языки программирования. Если используется один, то решение может быть оформлено в отдельный пакет.

Глобальная аутентификация и авторизация – существует специализированная служба, ответственная за аутентификацию и авторизацию пользователя.

На рисунке 33 представлена схема организации сервисов для глобальной аутентификации/авторизации пользователей.

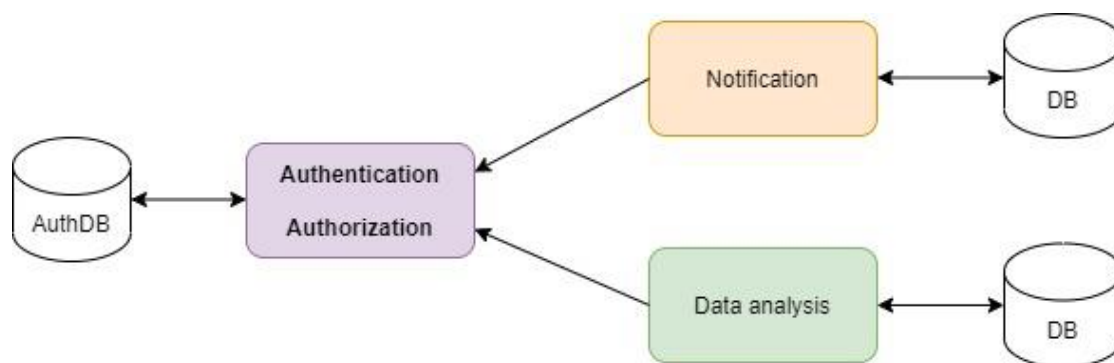


Рисунок 33 – Схема локальной аутентификации и авторизации

Использование специализированного микросервиса частично решает проблему уменьшения скорости обработки запросов с увеличением количества используемых микросервисов благодаря огромному количеству методов балансировки входящих запросов, однако т.к. авторизация происходит на стороне одного микросервиса – при каждом запросе к определенному эндпоинту необходимо сделать дополнительный запрос к микросервису авторизации для проверки прав, это не решает проблему единой точки отказа.

Главное преимущество данного метода – единая точка авторизации и аутентификации, однако данная стратегия не позволяет производить тонкую настройку для каждого микросервиса отдельно.

Глобальная аутентификация, локальная авторизация – специализированная служба аутентифицирует пользователя, авторизация пользователя происходит на стороне микросервиса.

Данный подход идентичен предыдущему, однако авторизация происходит на стороне каждого микросервиса, что позволяет уменьшить связанность между ними.

На рисунке 34 представлена схема глобальной аутентификации, локальной авторизации.

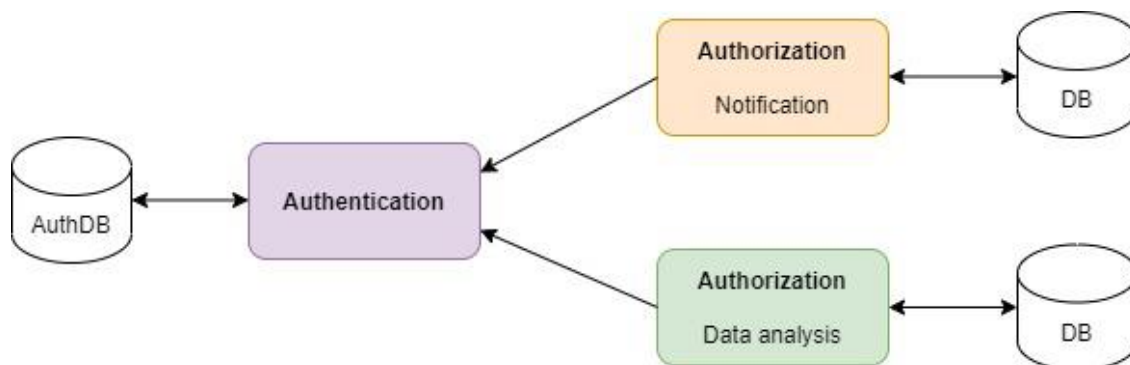


Рисунок 34 – Схема глобальной аутентификации, локальной авторизации

Благодаря разделению процессов авторизации и аутентификации частично решается проблема единой точки отказа, пропадает необходимость делать дополнительный запрос к микросервису аутентификации для проверки прав пользователя.

В таблице 12 представлены преимущества, недостатки описанных выше подходов.

Таблица 12 – Сравнение стратегий аутентификации и авторизации

| | Преимущества | Недостатки |
|---------------------------------------|---|---|
| Локальная аутентификация, авторизация | Гибкая настройка аутентификации, авторизации. Простота реализации. | Появление единой точки отказа распределенной системы. Дублирование функционала модуля аутентификации и авторизации в случае реализации на разных языках. С увеличением количества сервисов падает производительность. |

| | Преимущества | Недостатки |
|--|---|--|
| Глобальная аутентификация, авторизация | Обособленная система авторизации, аутентификации. Возможность горизонтального масштабирования. | Появление единой точки отказа распределенной системы. Время ответа микросервиса зависит от системы авторизации, аутентификации. Отсутствие гибкой настройки системы авторизации. |
| Глобальная аутентификация, локальная авторизация | Гибкая настройка прав доступа. Отсутствие единой точки отказа системы авторизации. Обособленная система аутентификации. Возможность горизонтального масштабирования. | Сложность реализации. Дублирование функционала модуля авторизации в случае реализации на разных языках. Сложность контроля токенов доступа. |

На основе обозначенных требований и таблицы сравнительного анализа разных стратегий была выбрана реализация глобальной аутентификации, локальной авторизации.

Механизм получения доступа к определенному эндпоинту основан на использовании 2 токенов доступа:

1. Access – токен доступа, который используется для авторизации пользователя в остальных микросервисах системы;
2. Refresh – токен, который выдается микросервисом аутентификации после успешной проверки учетных данных пользователя (логин и пароль) для обновления пары access/refresh токенов.

Для обеспечения безопасности, срок жизни Access-токена не более 5 минут, поэтому в случае перехвата, злоумышленник не сможет использовать его продолжительное время.

Refresh токен, время жизни которого несколько дней, используется для обновления пары access/refresh токенов, без необходимости использовать пары логина и пароля.

На рисунке 35 представлена логическая модель микросервиса аутентификации.

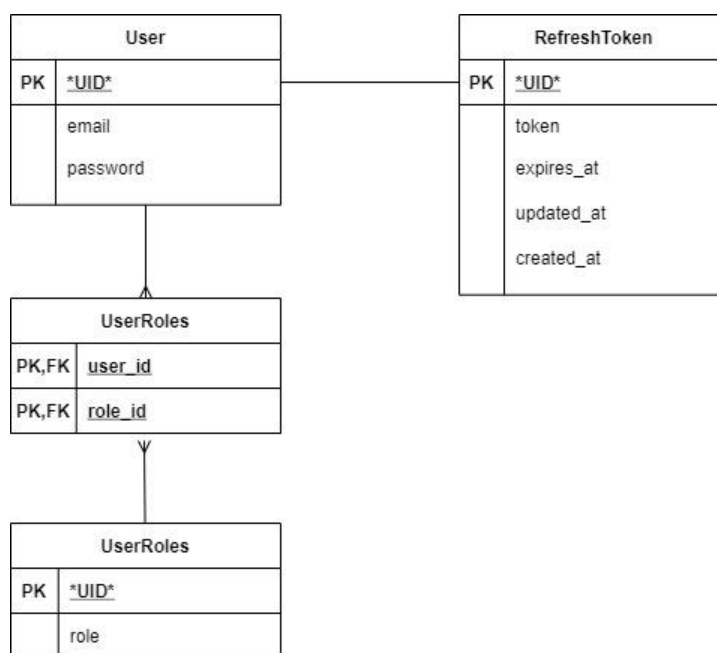


Рисунок 35 – Логическая модель базы данных микросервиса аутентификации

Сущность RefreshToken содержит токены аутентифицированных пользователей, что позволяет в любой момент отозвать их. Благодаря этому, если злоумышленник после кражи refresh-токена попытается сгенерировать новую пару refresh/access токенов, этот механизм позволит пользователю сделать утерянный токен недействительным.

В качестве access-токена используется JWT-токен, который не хранится в базе данных и содержит подписанную информацию о роли пользователя, его UUID и сроке действия токена.

На рисунке 36 представлена схема устройства JWT токена.

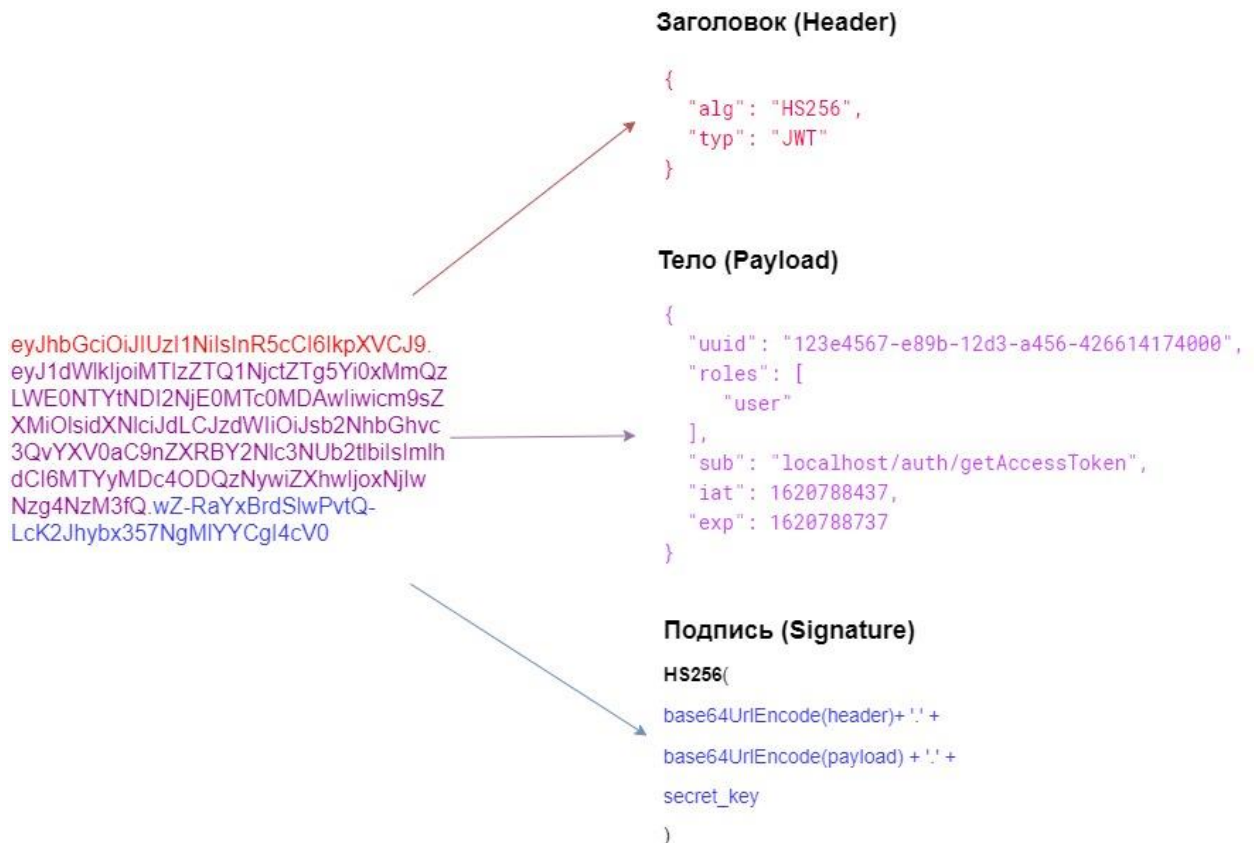


Рисунок 36 – JWT-токен

Токен состоит из 3 частей:

- Заголовок – метаинформация о токене (способ хеширования, тип токена и т.д.);
- Тело – содержит полезную нагрузку: UUID пользователя, его роли, URI эндпоинта генерации токена, время генерации и время жизни;
- Подпись.

Каждый из токенов подписывается секретным ключом, который знает только микросервис аутентификации и микросервис для которого этот токен был сгенерирован. Благодаря механизму подписей, мы можем быть уверены, что информация не была модифицирована третьим лицом. Использование JWT-токенов, которые передают необходимую для авторизации в конкретном микросервисе информацию о пользователе, позволяет свести к минимуму связанность между микросервисами.

После получения токена, проверки подписи и срока действия, каждый микросервис самостоятельно проводит авторизацию пользователя на основе данных в теле токена. В реализованной системе это осуществляется через механизм сопоставления каждой роли набора разрешений на доступ к определенному функционалу системы (RBAC).

На рисунке 37 представлена диаграмма последовательности процесса аутентификации с генерацией access токена. Для упрощения на схеме отсутствует этап обработки запроса API Gateway.

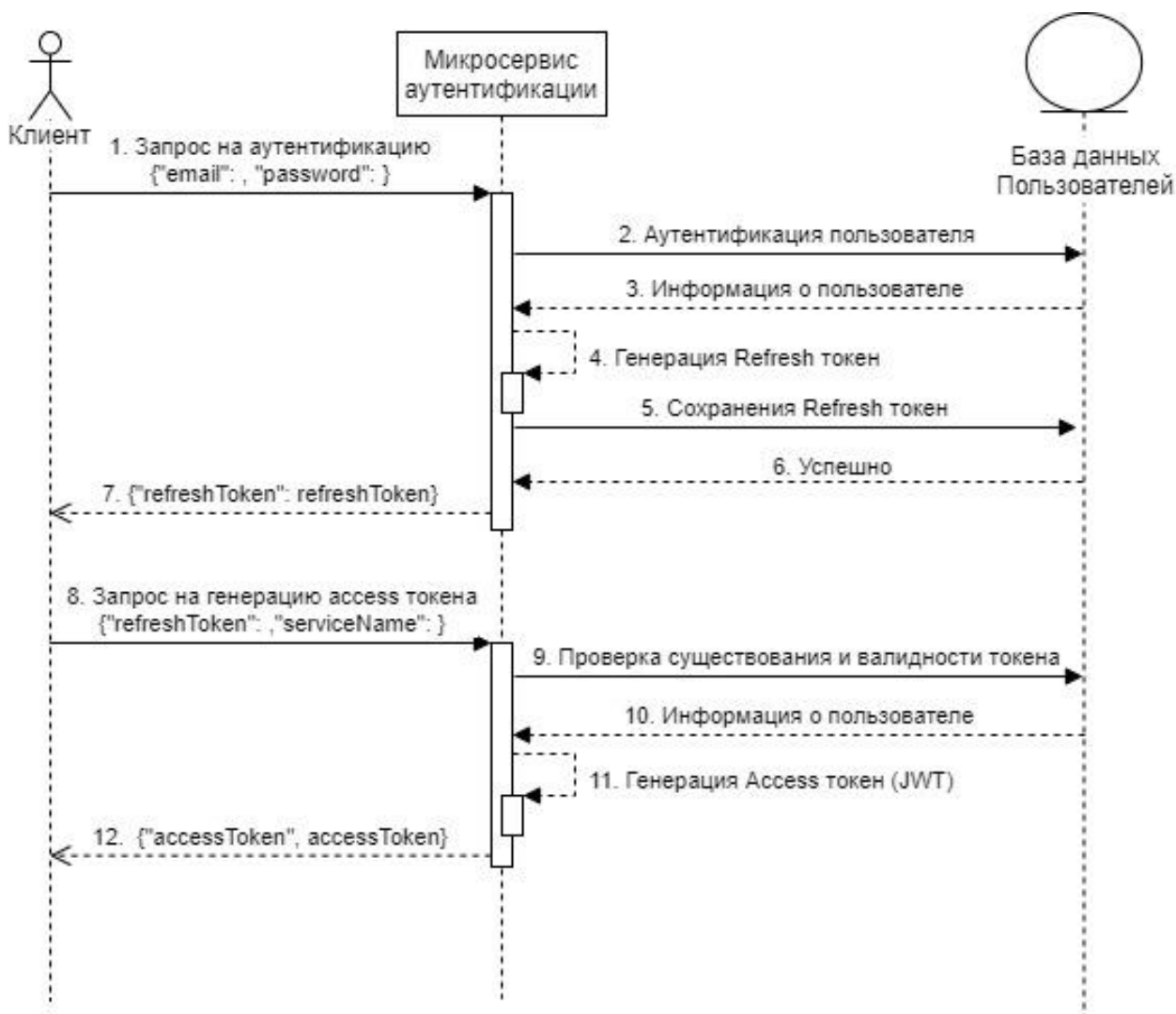


Рисунок 37 – Процесс аутентификации пользователя

На рисунке 38 представлен процесс авторизации пользователя в микросервисе Data Analysis. Для микросервиса Notification процесс аналогичен.



Рисунок 38 – Процесс авторизации пользователя

Представленная схема авторизации не завязана на микросервисе аутентификации, благодаря этому, кратковременные сбои работы этого микросервиса не повлияют на работу большей части системы. Другое преимущество реализованного подхода – строгое разделение пользовательской информации между микросервисами (микросервис не получает и не хранит информацию, которую не должен знать).

4.3 Микросервис нотификации

Цель микросервиса нотификации – отправлять различные уведомления пользователям. Основные виды уведомлений:

- Завершение обучение модели;
- Подтверждение регистрации;
- Восстановление пароля.

Работа микросервиса завязана на работе SMTP-сервере. Когда микросервис отправляет электронное письмо, SMTP-сервер обрабатывает электронную почту, решает на какой сервер отправить сообщение и ретранслирует сообщение на этот сервер.

При разработке данного сервиса использовался SMTP-сервер Gmail: smtp.gmail.com. На рисунке 39 представлена схема взаимодействия с SMTP-сервером.

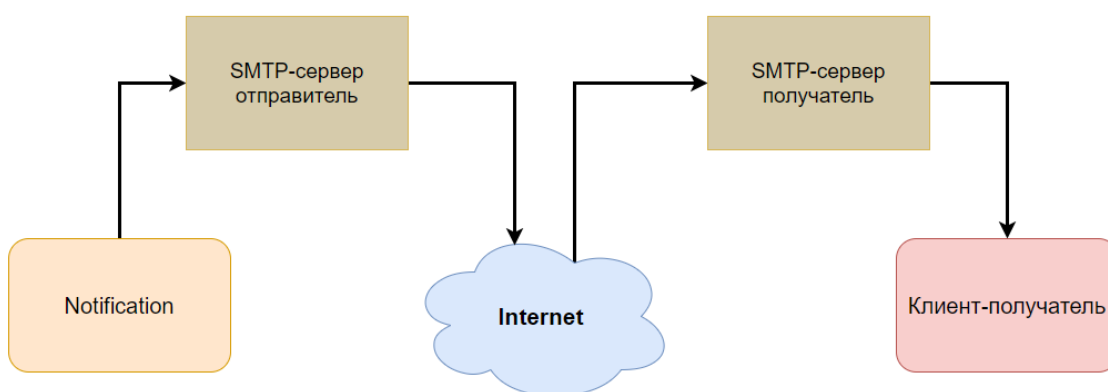


Рисунок 39 – Схема взаимодействия с SMTP-сервером

Чтобы отправить сообщение в начале необходимо инициализировать подключение к SMTP-клиенту для отправки сообщений. Фрагмент кода с использованием библиотеки smtplib на Python представлен ниже.

```
try:
    smtp = SMTP(host=self.__smtp_server, port=self.__smtp_port)
    smtp.ehlo()
    return smtp

except (SMTPServerDisconnected, AssertionError) as ex:
    error_message = f'No connect to SMTP server {self.__smtp_server}:{self.__smtp_port}: {ex}'
    logger.error(error_message)
    raise ServerDisconnectedException(error_message)

except gaierror as ex:
    error_message = f'SMTP Server host error ({self.__smtp_server}): {ex}'
    logger.error(error_message)
    raise SMTPServerException(error_message)

except ConnectionRefusedError as ex:
    error_message = f'Error of the SMTP server (possibly an error with the server port {self.__smtp_port}) {ex}'
    logger.error(error_message)
    raise SMTPConnectionRefusedException(error_message)
```

Ниже приведен пример работы сервиса уведомлений в случае, когда пользователь прошел регистрацию.

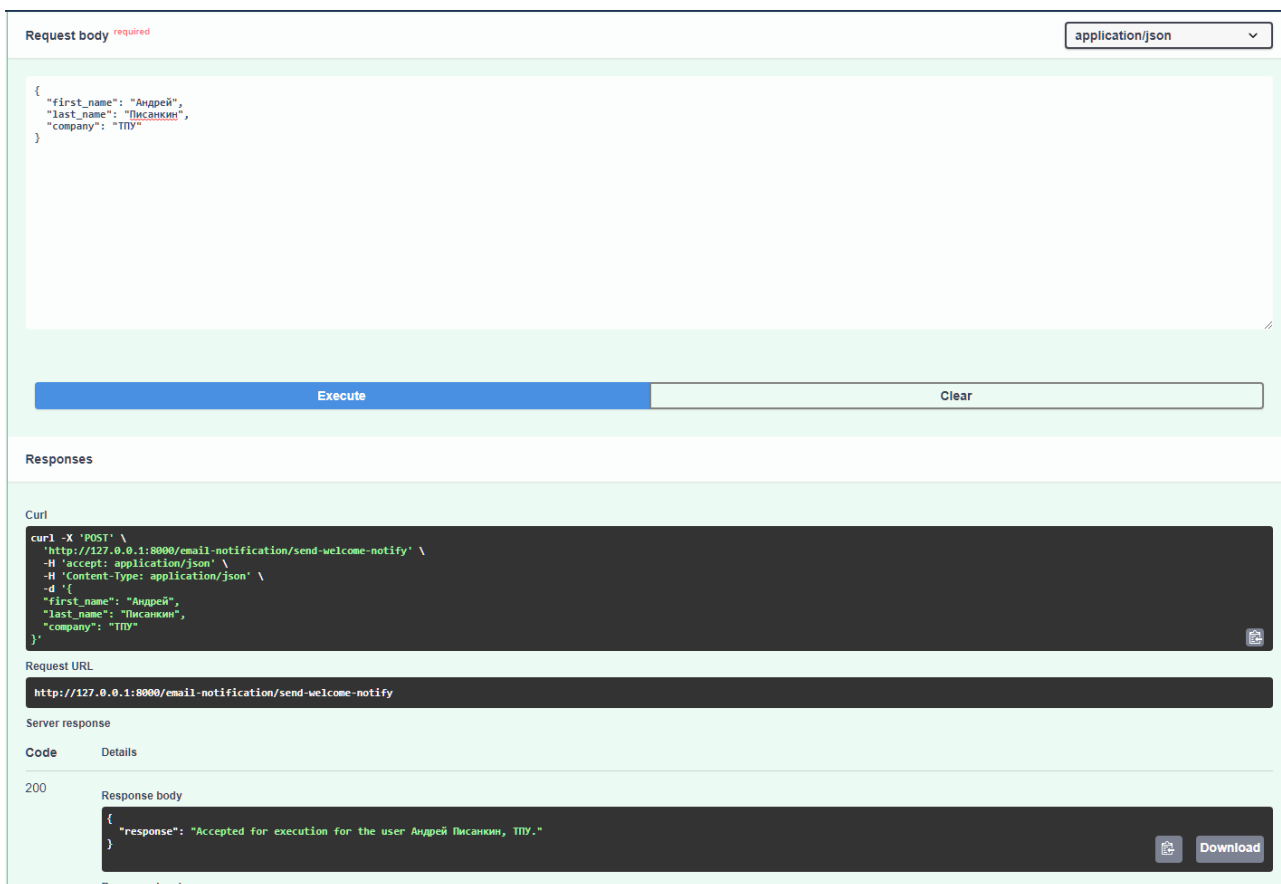


Рисунок 40 – Запрос на отправку уведомления (Swagger микросервиса)

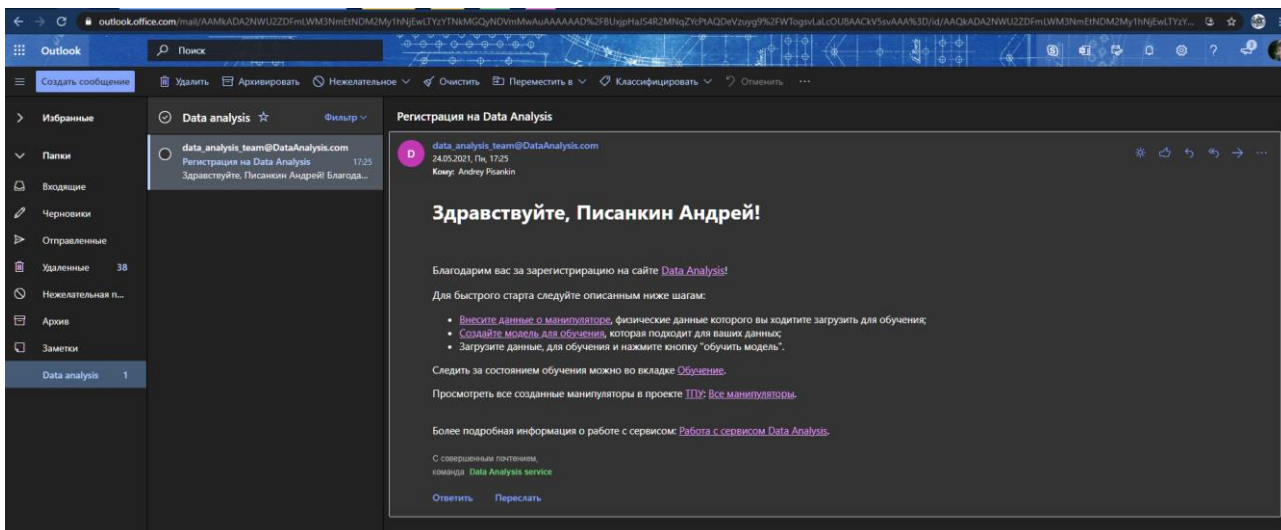


Рисунок 41 – Пример письма о регистрации

Данные для письма сервис получает, обращаясь к другим сервисам через API, затем подставляет полученные значения в шаблон письма.

4.4 Микросервис анализа данных

Следующим этапом после преобразование данных, обучения модели и вывода результатов, был написан веб-сервис на фреймворке Django. Для создания API была использована библиотека Django Rest Framework.

API Django Rest Framework состоит из 3-х слоев:

- Сериализатор: преобразует информацию, хранящуюся в базе данных и определенную с помощью моделей Django, в формат, передается через API;
- Вид (ViewSet): определяет формат который передается через API;
- Маршрутизатор: определяет URL-адреса, которые будут предоставлять доступ к каждому представлению.

Пользователь загружает файл в формате csv или xls/xlsx. Пример загрузки данных через программу Postman представлен на рисунке 42.

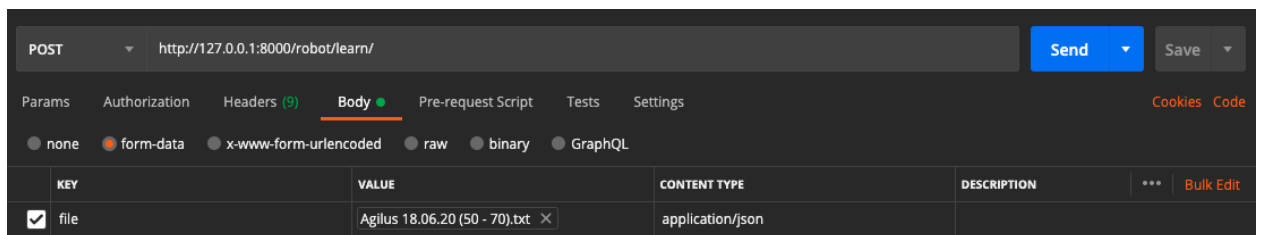


Рисунок 42 – Загрузка данных через Postman

Данные обрабатываются и преобразуются в формат json. Листинг кода обработки запроса представлен ниже.

```
class RobotLearnView(APIView, RobotHandler):
    parser_class = (FileUploadParser,)
    def post(self, request, *args, **kwargs):
        file = request.FILES['file']
        df = pd.read_csv(file)
        df_json = df.to_json()
        file_data = QueryDict("", mutable=True)
        file_data.update({'robot_data': df_json})
        file_serializer = RobotSerializer(data=file_data)
        if file_serializer.is_valid():
            # file_serializer.save()
            # print(df.loc[df['_NumMotor_'] == '_1_'])
            do_learn(robot_data=df.loc[df['_NumMotor_'] == '_1_'])
            return Response('Success', status=status.HTTP_201_CREATED)
        return Response(file_serializer.errors, status=status.HTTP_400_BAD_REQUEST)
```

Пример формата данных при запросе информации о добавенных данных представлен ниже.



Рисунок 43 – Пример формата данных

Листинг кода обработки запроса на получение подробной информации о данных представлен ниже.

```
class RobotDetail(APIView, RobotHandler):

    def get(self, request, pk, format=None):
        robot = self.get_robot(pk)
        serializer = RobotSerializer(robot)
        return Response(serializer.data)

    def put(self, request, pk, format=None):
        robot = self.get_robot(pk)
        serializer = RobotSerializer(robot, data=request.data)
        if serializer.is_valid():
            serializer.save()
            return Response(serializer.data)
        return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)

    def delete(self, request, pk, format=None):
        robot = self.get_robot(pk)
        robot.delete()
        return Response(status=status.HTTP_204_NO_CONTENT)
```

Входные данные обрабатываются по алгоритмам, описанным в пункте 3.1. Пример реализации алгоритмов в коде представлен ниже.

```

def standartization(train_df, val_df, test_df):
    train_mean = train_df.mean()
    train_std = train_df.std()

    n_train_df = (train_df - train_mean) / train_std
    n_val_df = (val_df - train_mean) / train_std
    n_test_df = (test_df - train_mean) / train_std

    return n_train_df, n_val_df, n_test_df, train_std, train_mean

def data_splitting(df):
    column_indices = {name: i for i, name in enumerate(df.columns)}

    n = len(df)
    train_df = df[0:int(n * 0.7)]
    val_df = df[int(n * 0.7):int(n * 0.9)]
    test_df = df[int(n * 0.9):]

    num_features = df.shape[1]

    return train_df, val_df, test_df, column_indices

def remove_columns(df_data, columns_to_remove: list):
    # Удаление ненужных для обучения колонок
    df = df_data.drop(columns_to_remove, axis=1)

    for col in df.columns:
        if 'Unnamed' in col:
            df = df.drop(col, axis=1)

    return df

```

Обработанные данные подаются на вход модели для обучения и тестирования.

```

def compile_and_fit(model, window, patience=2, max_epochs=20):
    early_stopping = tf.keras.callbacks.EarlyStopping(monitor='val_loss',patience=patience, mode='min')

    time_callback = TimeHistory()
    model.compile(
        loss=tf.losses.MeanSquaredError(),
        optimizer=tf.optimizers.Adam(),
        metrics=[tf.metrics.MeanAbsoluteError()],
    )

    history = model.fit(
        window.train,
        epochs=max_epochs,
        validation_data=window.val,
        callbacks=[early_stopping, time_callback],
    )

```

```

times = time_callback.times
studying_time = 0

for time in times:
    studying_time += time

logger.info('Studying time- ', studying_time)
return history

def build_linear(single_step_window, val_performance, performance):
    linear = tf.keras.Sequential([
        tf.keras.layers.Dense(units=1)
    ])

    history = compile_and_fit(linear, single_step_window)
    val_performance['Linear'] = linear.evaluate(single_step_window.val)
    performance['Linear'] = linear.evaluate(single_step_window.test, verbose=0)
    return linear

```

Результат работы – обученные модели и графики данных описанные в пункте 3.2.7.

4.5 Подготовка к развертыванию ПО

4.5.1 Контейнеризация

Использование контейнеров вместо виртуальных машин является одним из самых быстроразвивающихся подходов в новейшей истории индустрии программного обеспечения. Самая популярная на текущий момент технология.

Docker упрощает создание, развертывание и запуск приложений в контейнерах, контейнеры позволяют разработчику упаковать приложение со всеми необходимыми частями, такими как библиотеки и другие зависимости, и управлять этим как одним пакетом. Использование данного подхода позволяет запускать упакованное приложение на любой платформе, поддерживающей Docker независимо от индивидуальных настроек среды. Данная технология гарантирует, что приложение будет работать везде одинаково.

Образ Docker контейнера создается на основе Dockerfile, который представляет собой последовательность команд создания и конфигурации.

Пример листинга файла Docker для микросервиса уведомлений представлен ниже.

```
FROM python:3.8

ARG CI_COMMIT_SHORT_SHA=unknown
ARG CI_COMMIT_REF_SLUG=unknown

ENV CI_COMMIT_SHORT_SHA=${CI_COMMIT_SHORT_SHA}
ENV CI_COMMIT_REF_SLUG=${CI_COMMIT_REF_SLUG}

COPY ./opt/Notification
WORKDIR /opt/Notification

RUN python -m pip --no-cache-dir install --upgrade pip setuptools
RUN python -m pip install -r requirements.txt

ENV PYTHONUNBUFFERED 1
EXPOSE 8000

CMD exec uvicorn --host 0.0.0.0 --port 8000 app.main:app
```

Хорошей практикой управления и развертывания является создание отдельного docker контейнера для каждой системы приложения. Например, согласно идеологии docker, веб-сервер и база данных должны работать в отдельных контейнерах. Сборка, запуск и подключение контейнеров из отдельных файлов Docker сложны и могут занять много времени.

Для упрощения процесса сборки и управления контейнерами используется утилита оркестрации - Docker Compose. Основой этой утилиты является Docker Compose файл, который описывает инструкцию поднятия docker образов на языке разметки YAML. Каждый из контейнеров запускается изолированно, но при необходимости может взаимодействовать с другими контейнерами.

Пример используемого Docker Compose файла для контейнера с базой данных PostgreSQL представлен ниже.

```
db:
  image: postgres:12.2-alpine
  container_name: db_importrepo
  ports:
    - ${POSTGRES_PORT}:5432
  networks:
    - backend
  environment:
    POSTGRES_DB: ${POSTGRES_DB}
    POSTGRES_USER: ${POSTGRES_USER}
    POSTGRES_PASSWORD: ${POSTGRES_PASSWORD}
  volumes:
    - db-data:/var/lib/postgresql/data
    - db-data-dump:/docker-entrypoint-initdb.d/
  restart: unless-stopped
  <<: *logging-rules
```

Использование Docker Compose позволяет обеспечить:

- Развертывание с одним хостом – это означает, что вы можете запускать все на одном оборудовании;
- Быстрая и простая настройка благодаря скриптам YAML;
- Сокращает время, необходимое для выполнения задач за счет удобства использования;
- Обеспечивает безопасность – все контейнеры изолированы друг от друга.

4.5.2 Сценарий развертывания ПО

Если возникает необходимость выполнить обновление разрабатываемого ПО во время цикла выпуска продукта, за счет настройки CI/CD можно легко внести необходимые изменения в контейнеры Docker, протестировать их и внедрить изменения в существующие контейнеры.

При подготовке к развертыванию ПО основной уклон был сделан на систему CI/CD от Azure DevOps. Была использована бесплатная пробная версия для проверки работоспособности ПО. Для настройки CI/CD в системе Azure DevOps необходимо в корень репозитория поместить файл `azure-pipeline.yml`, в котором описываются шаги по развёртыванию приложения.

Примеры отображения запусков CI/CD для сервиса нотификации представлены на рисунке 44.

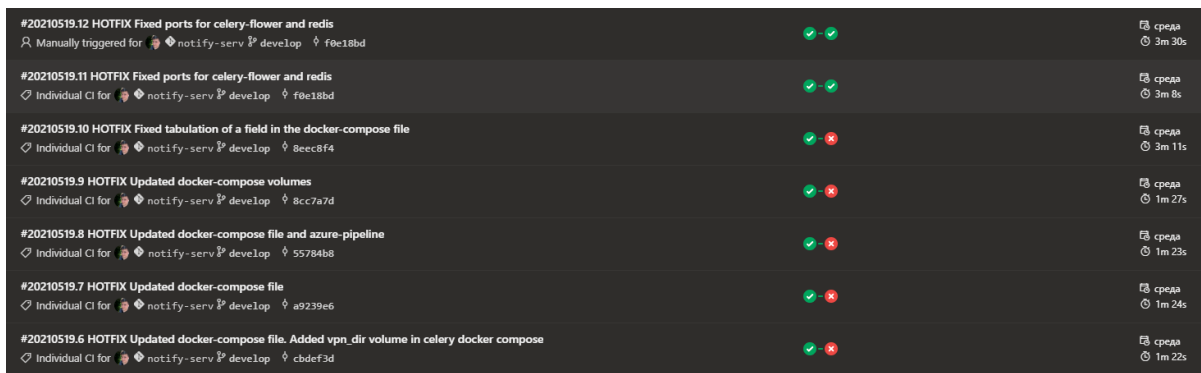


Рисунок 44 – Отображения запусков CI/CD в AzureDevOps

Процесс развертывания ПО разделён на 2 стадии (рисунок 45):

- Стадия тестирования кода: в случае работы с языком программирования Python, использование библиотеки pylint для проверки корректности написания программного кода;
- Стадия поставки кода на виртуальную машину: доставка кода на удаленную машину, установка необходимых зависимостей и запуск Docker Compose.

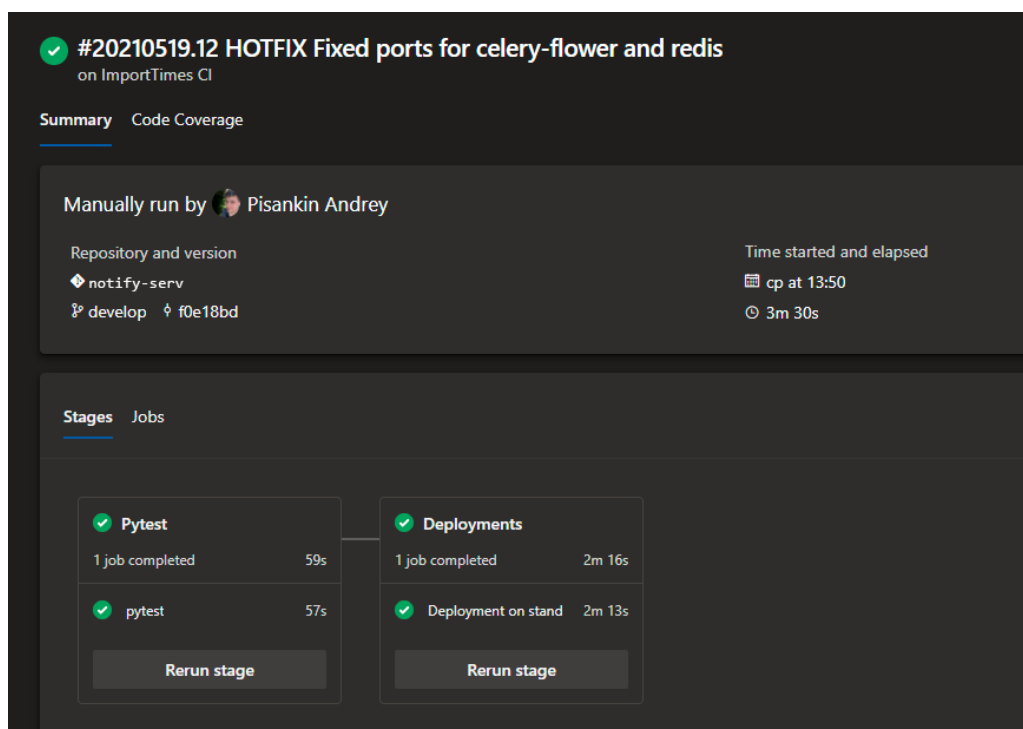


Рисунок 45 – Стадии CI/CD

Фрагмент файла azure-pipeline.yml для настройки первой стадии CI/CD

представлен ниже.

```
stages:
- stage: test
  displayName: Pytest
  dependsOn: []
  jobs:
  - job: pytest
    displayName: pytest
    pool: Linux
    steps:
    - checkout: self
      clean: true
    - script: |
      python3.8 -m pip --no-cache-dir install --upgrade pip setuptools 2>&1
      if [ $? -ne 0 ]
      then exit 1
      fi
      python3.8 -m pip install -r requirements.txt 2>&1
      if [ $? -ne 0 ]
      then exit 1
      fi
      pylint $(app_name) || pylint-exit -efail $? 2>&1
      if [ $? -ne 0 ]
      then exit 1
      fi
      pylint $(celery_app_name) || pylint-exit -efail $? 2>&1
      if [ $? -ne 0 ]
      then exit 1
      fi
    continueOnError: false
```

Использование этого метода обеспечивает оперативность поставки нового функционала продукта и автоматизацию первичного тестирования, что позволяет оптимизировать затраты ресурсов на решение задач.

Глава 5 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

5.1 Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения

5.1.1 Потенциальные потребители результатов исследования

В представленной работе рассматривается разработка веб-сервиса анализа данных с промышленных манипуляторов при помощи методов машинного обучения. Идея продукта – предоставлять коммерческим организациям, использующим роботизированные системы, инструмент мониторинга и анализа состояния промышленных манипуляторов не требующий значительных капиталовложений в организацию вычислительной инфраструктуры. Разработанное ПО, в первую очередь, направленно на небольшие компании, которые не используют существующие дорогостоящие инструменты обработки данных с манипуляторов, предоставляемые компаниями производителями, и облачные сервисы непрерывного сбора и анализа данных в режиме реального времени.

Система хранит и обрабатывает срезы данных, предоставляемые пользователями интервально; загружаемые данные сравниваются с «эталонными» показателями работы манипулятора. На основании произведенного сравнения компания составляет стратегию планово-предупредительного ремонта. Описанная модель позволяет эффективнее использовать ресурсы компании, сокращает издержки производства, связанные с организацией технического обслуживания.

На этапе проверки гипотезы, плата за использования сервиса взиматься не будет, это позволит собрать необходимую обратную связь от конечного потребителя, далее предполагается комиссионная модель монетизации –

пользователь получает в распоряжение бесплатные, ограниченные ресурсы системы (процессорное время, объем хранимой информации), а за увеличение предоставляемых ресурсов взимается дополнительная плата. Целевые потребители – российские компании.

Проведем сегментирование рынка по критериям: вид клиента, модель предоставления вычислительных ресурсов. В таблице 14 представлена карта сегментирования рынка систем интеллектуального анализа и мониторинга данных с промышленных манипуляторов.

Таблица 13 – Карта сегментирования рынка систем интеллектуального анализа и мониторинга данных с промышленных манипуляторов.

| | Модель предоставления вычислительных ресурсов | | | |
|-----------------------|--|----------------------------|--------------------------|--------------------------------------|
| | Облако | Разделяемый хостинг | Выделенный сервер | Виртуальный выделенный сервер |
| Вид приложения | IXON [13], AVEA [14], SDC [15] | | | |
| | | | SDC | |
| | IXON | | | |

В таблице 14 представлены компании предоставляющие схожие разрабатываемому ПО услуги. Самый популярный вид приложения – веб клиент с облачной моделью предоставления вычислительных ресурсов. Популярность данного решения обусловлена спецификой решаемой задачи (анализ данных требует больших вычислительных мощностей и большого объема хранилища) и моделью монетизации предоставляемой услуги (пользователь платит только за используемые ресурсы).

Несмотря на высокую конкуренцию, из-за специфики решаемой задачи, разрабатываемое ПО представлено в виде веб приложения, однако мы отказались от использования облачных вычислений в пользу виртуального выделенного сервера.

Занимаемый сегмент рынка – веб-приложение на виртуально выделенном сервере.

5.1.2 Анализ конкурентных технических решений

Для оценки анализа конкурентных решений используем оценочную карту для следующих вариантов:

1. Веб-приложение – виртуально выделенный сервер (предлагаемый нами вариант);
2. Веб-приложение – облако;
3. Desktopное приложение – выделенный сервер.

При составлении не учитывается вариант «мобильное приложение – облако» т.к. предлагаемый нами подход не подразумевает использование мобильных устройств (в карте сегментирования рынка, компания использует гибридный вариант – мобильное приложение как дополнение к основному сервису).

В таблице 14 представлена оценочная карта для сравнения выбранных нами конкурентных решений.

Таблица 14 – Оценочная карта для сравнения конкурентных технических решений

| Критерий оценки | Вес критерия | Баллы | | | Конкурентоспособность | | |
|--|--------------|----------------|-----------------|-----------------|-----------------------|-----------------|-----------------|
| | | Б _ф | Б _{к1} | Б _{к2} | К _ф | К _{к1} | К _{к2} |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Технические критерии оценки ресурсоэффективности | | | | | | | |
| Потребность в изменениях процессов производственного цикла | 0,2 | 5 | 4 | 2 | 1 | 0,8 | 0,4 |
| Трудоемкость внедрения | 0,14 | 4 | 4 | 1 | 0,56 | 0,56 | 0,14 |
| Потребность в дополнительных ресурсах | 0,16 | 5 | 4 | 2 | 0,8 | 0,64 | 0,32 |
| Длительность внедрения системы | 0,10 | 4 | 4 | 2 | 0,4 | 0,4 | 0,2 |
| Простота эксплуатации | 0,07 | 4 | 4 | 2 | 0,28 | 0,28 | 0,14 |
| Экономические критерии оценки эффективности | | | | | | | |
| Конкурентоспособность продукта на российском рынке | 0,10 | 4 | 4 | 4 | 0,4 | 0,4 | 0,4 |
| Стоимость внедрения | 0,15 | 5 | 4 | 2 | 0,75 | 0,6 | 0,3 |

Основываясь на данных из таблицы 14, конкурентоспособность нашей разработки составила 4,43 ед. – наилучший показатель по сравнению с аналогами. Вариант 3 «Десктопное приложение – выделенный сервер» имеет наименьший показатель т.к. организация инфраструктуры локально требует огромных капиталовложений и наличия специалистов для организации работ. Подобная модель распространения ПО используется при работе с конфиденциальными данными или на крупных производствах, где организация собственного решения в перспективе выгоднее, чем зависимость от предоставляемого, стороннего сервиса. Предлагаемый нами вариант развития продукта оказался выгоднее, чем облачное веб-приложение благодаря более низкой сложности внедрения (процессы компании остаются неизменными) и стоимости по сравнению с облаком (в стоимость облачных решений закладывается расходы на организацию инфраструктуры и масштабирование используемых ресурсов). С другой стороны, наш вариант подходит только малым компаниям, которым не требуется анализ состояния большого количества устройств в режиме реального времени.

5.1.3 QuaD-анализ

В таблице 15 представлена оценка перспективности разрабатываемой системы при помощи инструмента измерения характеристик – QuaD. Вычисленное средневзвешенное значение показателя качества и перспективности разрабатываемой системы равна 81,95, следовательно, создание предлагаемой системы является перспективной.

Таблица 15 – QuaD анализ

| Критерии оценки | Вес критерия | Баллы | Максимальный балл | Относительное значение | Средневзвешенное значение (*100) |
|---|--------------|------------|-------------------|------------------------|----------------------------------|
| 1 | 2 | 3 | 4 | 5 | 6 |
| Показатели оценки качества разработки | | | | | |
| Потребность в ресурсах памяти | 0,05 | 70 | 100 | 0,7 | 3,5 |
| Функциональность | 0,07 | 85 | 100 | 0,85 | 5,95 |
| Простота эксплуатации | 0,12 | 90 | 100 | 0,9 | 10,8 |
| Скорость обработки данных | 0,15 | 90 | 100 | 0,9 | 13,5 |
| Отказоустойчивость | 0,15 | 70 | 100 | 0,7 | 10,5 |
| Точность вычислительных операций | 0,2 | 90 | 100 | 0,9 | 18 |
| Длительность внедрения | 0,07 | 65 | 100 | 0,65 | 4,55 |
| Показатели оценки коммерческого потенциала разработки | | | | | |
| Срок выхода на рынок | 0,05 | 70 | 100 | 0,7 | 3,5 |
| Техническая поддержка продукта | 0,05 | 80 | 100 | 0,8 | 4 |
| Стоимость продукта | 0,09 | 85 | 100 | 0,85 | 7,65 |
| Итого | 1 | 795 | 1000 | 7,95 | 81,95 |

5.1.4 SWOT-анализ

В таблице 16 представлена SWOT матрица (анализ сильных, слабых сторон проекта и возможных угроз).

Таблица 16 – Матрица SWOT

| | Сильные стороны научно-исследовательского проекта: С1. Простота использования С2. Низкая стоимость по сравнению с аналогами С3. Удобство использования С4. Отсутствие привязки к вычислительным мощностям организации | Слабые стороны научно-исследовательского проекта: Сл1. Отсутствие апробации алгоритмов на данных с различных манипуляторов Сл2. Отсутствие проверки гипотезы, целесообразности использования предлагаемого подхода в реальных условиях Сл3. Узкий круг предоставляемых услуг |
|--|--|--|
| Возможности: В1. Ускорение вычислений за счет появления более эффективных инструментов В2. Уменьшение стоимости оборудования за счет появления более мощных аналогов В3. Повышение эффективности использования ресурсов за счет появления новых алгоритмов | | |
| Угрозы: У1. Появление аналогичных услуг у прямых конкурентов У2. Падение серверов из за высоких нагрузок | | |

Для выявления соответствий сильных и слабых сторон проекта, для оценки целесообразности внесения стратегических изменений в разрабатываемую систему была построена интерактивная матрица. В таблице 17 представлена интерактивная матрица сильных сторон и возможностей проекта.

Таблица 17 – Интерактивная матрица

| Сильные стороны проекта | | | | | |
|--------------------------------|-----------|-----------|-----------|-----------|-----------|
| | | C1 | C2 | C3 | C4 |
| Возможности проекта | B1 | - | + | + | - |
| | B2 | - | + | - | - |
| | B3 | - | + | + | - |

На основе анализа интерактивной матрицы можно сделать вывод, что B2, B3 имеют единую природу – качественное улучшение проекта за счет оптимизации используемых ресурсов.

В таблице 18 представлена итоговая матрица SWOT.

Таблица 18 – Матрица SWOT

| | | |
|---|---|---|
| | <p>Сильные стороны научно-исследовательского проекта:</p> <p>С1. Простота использования</p> <p>С2. Низкая стоимость по сравнению с аналогами</p> <p>С3. Удобство использования</p> <p>С4. Отсутствие привязки к вычислительным мощностям организации</p> | <p>Слабые стороны научно-исследовательского проекта:</p> <p>Сл1. Отсутствие апробации алгоритмов на данных с различных манипуляторов</p> <p>Сл2. Отсутствие проверки гипотезы, целесообразности использования предлагаемого подхода в реальных условиях</p> <p>Сл3. Узкий круг предоставляемых услуг</p> |
| <p>Возможности:</p> <p>В1. Ускорение вычислений за счет появления более эффективных инструментов</p> <p>В2. Уменьшение стоимости оборудования за счет появления более мощных аналогов</p> <p>В3. Повышение эффективности использования ресурсов за счет появления новых алгоритмов</p> | <p>Продукт имеет преимущества по сравнению с прямыми конкурентами, существуют возможности качественной оптимизации для улучшения пользовательского опыта</p> | <p>Для удовлетворения потребностей большего круга лиц необходимо увеличить количество распознаваемых и обрабатываемых выборок данных</p> |
| <p>Угрозы:</p> <p>У1. Появление аналогичных услуг у прямых конкурентов</p> <p>У2. Падение серверов из за высоких нагрузок</p> | <p>При возрастании спроса на продукт необходимо горизонтальное масштабирование</p> | <p>В перспективе необходимо расширить функциональные возможности продукта для сохранения конкурентоспособности</p> |

5.2 Определение возможных альтернатив проведения научных исследований

Для представления возможных альтернатив развития разрабатываемой системы возможные варианты характеристик представлены в виде морфологической матрицы в таблице 19 (цветом указаны выбранные варианты).

Таблица 19 – Морфологическая матрица

| | 1 | 2 | 3 |
|--|-------------------------|----------------|--------------|
| А. Общая архитектура веб приложения | Монолитная | Микросервисная | Серверная |
| Б. Протокол взаимодействия клиента и сервера | Websocket | HTTP | HTTPS |
| В. Хранение временных данных | Redis | MongoDB | KeyDB |
| Г. Хранение данных | Oracle | PostgreSQL | MySQL |
| Д. Аутентификация пользователей | SAML | Basic HTTPS | Kerberos |
| Е. Алгоритм машинного обучения | Многословный перцептрон | Линейный | Рекуррентный |
| Ж. Метод обработки входных признаков | Фильтрация | Обертка | Гибридный |

5.3 Планирование научно-исследовательских работ

5.3.1 Структура работ в рамках научного исследования

В рамках выполнения данной научной-исследовательской работы в качестве исполнителей были задействованы два студента. Общая таблица с этапами работ и распределенными исполнителями представлена ниже.

Таблица 20 – Материальные затраты

| № | Этапы работы | Исполнители работы |
|----------|--|---|
| 1 | Выбор научного руководителя бакалаврской работы | Чепкасов А.Ю. Писанкин А.С. |
| 2 | Составление и утверждение темы бакалаврской работы | Савельев А.О. Чепкасов А.Ю. Писанкин А.С. |
| 3 | Постановка целей и задач | Савельев А.О. Чепкасов А.Ю. Писанкин А.С. |
| 4 | Разработка календарного плана | Савельев А.О. |
| 5 | Анализ предметной области | Чепкасов А.Ю. Писанкин А.С. |
| 6 | Составление и утверждение технического задания | Савельев А.О. Чепкасов А.Ю. Писанкин А.С. |
| 7 | Проектирование системы | Чепкасов А.Ю. Писанкин А.С. |
| 8 | Разработка веб-сервиса | Чепкасов А.Ю. Писанкин А.С. |
| 9 | Согласование выполненной работы с научным руководителем | Савельев А.О. Чепкасов А.Ю. Писанкин А.С. |
| 10 | Заполнение разделов фин. Менеджмент и соц. Ответственность | Чепкасов А.Ю. Писанкин А.С. |
| 11 | Составление и согласование итогового варианта ВКР | Чепкасов А.Ю. Писанкин А.С. |

5.3.2 Определение трудоемкости выполнения работ

Для определения основной части стоимости разработки необходимо определить трудоёмкость планируемых работ для каждого из участников научного исследования. Для определения трудоёмкости работ, необходимо оценить минимальное и максимальное затраченное на работу время. Произведём расчёт ожидаемой трудоёмкости с помощью формулы (4).

$$t_{ож\ i} = \frac{3t_{min\ i} + 2t_{max\ i}}{5} \quad (4)$$

где $t_{ож\ i}$ – ожидаемая трудоемкость выполнения i -ой работы чел./дн.;

$t_{min\ i}$ – минимально возможная трудоемкость выполнения заданной i -ой работы, чел./дн.;

$t_{max\ i}$ – максимально возможная трудоемкость выполнения заданной i -ой работы (пессимистическая оценка: в предположении наиболее неблагоприятного стечения обстоятельств), чел.-дн.

5.3.3 Разработка графика проведения научного исследования

Согласно производственному календарю (для 6-дневной рабочей недели) в 2021 году 365 календарных дней, 299 рабочих дней, 66 выходных/праздничных дней. Таким образом, коэффициент календарности на 2021 год равен:

$$k_{кал} = \frac{T_{кал}}{T_{кал} - T_{вых} - T_{пр}} = \frac{365}{365 - 66} = 1.22$$

После расчета коэффициента календарности можно составить таблицу временных показателей проведения научного исследования и диаграмму Ганта.

Таблица 21 – Временные показатели проведения научного исследования

| Наименование работы | Исполнители работы | Трудоемкость работ, чел-дни | | | Длительность работ, дни | |
|---|--------------------|-----------------------------|------------------|-----------------|-------------------------|----------------|
| | | t _{min} | t _{max} | t _{ож} | Т _р | Т _к |
| Выбор научного руководителя ВКР | Чепкасов А.Ю. | 1 | 2 | 1,4 | 1,4 | 2 |
| | Писанкин А.С. | 1 | 2 | 1,4 | 1,4 | 2 |
| Составление и утверждение темы бакалаврской работы | Савельев А.О. | 2 | 4 | 2,8 | 2,8 | 4 |
| | Чепкасов А.Ю. | 2 | 4 | 2,8 | 2,8 | 4 |
| | Писанкин А.С. | 2 | 4 | 2,8 | 2,8 | 4 |
| Постановка целей и задач | Чепкасов А.Ю. | 5 | 8 | 6,2 | 6,2 | 8 |
| | Писанкин А.С. | 5 | 8 | 6,2 | 6,2 | 8 |
| Разработка календарного плана | Савельев А.О. | 3 | 5 | 3,8 | 3,8 | 5 |
| Анализ предметной области | Чепкасов А.Ю. | 5 | 8 | 6,2 | 3,1 | 4 |
| | Писанкин А.С. | | | | 3,1 | 4 |
| Составление и утверждение технического задания | Савельев А.О. | 1 | 1 | 1 | 1 | 1 |
| | Чепкасов А.Ю. | 2 | 4 | 2,8 | 2,8 | 4 |
| | Писанкин А.С. | 2 | 4 | 2,8 | 2,8 | 4 |
| Проектирование системы | Чепкасов А.Ю. | 8 | 10 | 8,8 | 4,4 | 6 |
| | Писанкин А.С. | | | | 4,4 | 6 |
| Разработка веб-сервиса | Чепкасов А.Ю. | 72 | 84 | 76,8 | 38,4 | 47 |
| | Писанкин А.С. | | | | 38,4 | 47 |
| Согласование выполненной работы с научным руководителем | Савельев А.О. | 3 | 5 | 3,8 | 3,8 | 5 |
| | Чепкасов А.Ю. | 1 | 2 | 1,4 | 1,4 | 2 |
| | Писанкин А.С. | 1 | 2 | 1,4 | 1,4 | 2 |

| Наименование работ | Исполнители работ | Трудоемкость работ, чел-дни | | | Длительность работ, дни | |
|--|-------------------|-----------------------------|------|-----|-------------------------|----|
| | | tmin | tmax | тож | Тр | Тк |
| Заполнение разделов фин. Менеджмент и соц. Ответственность | Чепкасов А.Ю. | 5 | 7 | 5,8 | 2,9 | 4 |
| | Писанкин А.С. | | | | 2,9 | 4 |
| Составление и согласование итогового варианта ВКР | Чепкасов А.Ю. | 6 | 10 | 7,6 | 3,8 | 5 |
| | Писанкин А.С. | | | | 3,8 | 5 |

Для визуализации продолжительности этапов разработки была построена диаграмма Ганта, представленная на рисунке 46 – 47.

Согласно диаграмме Ганта даты выполнения работ: с 01.02.2021 г. по 02.06.2021 г.

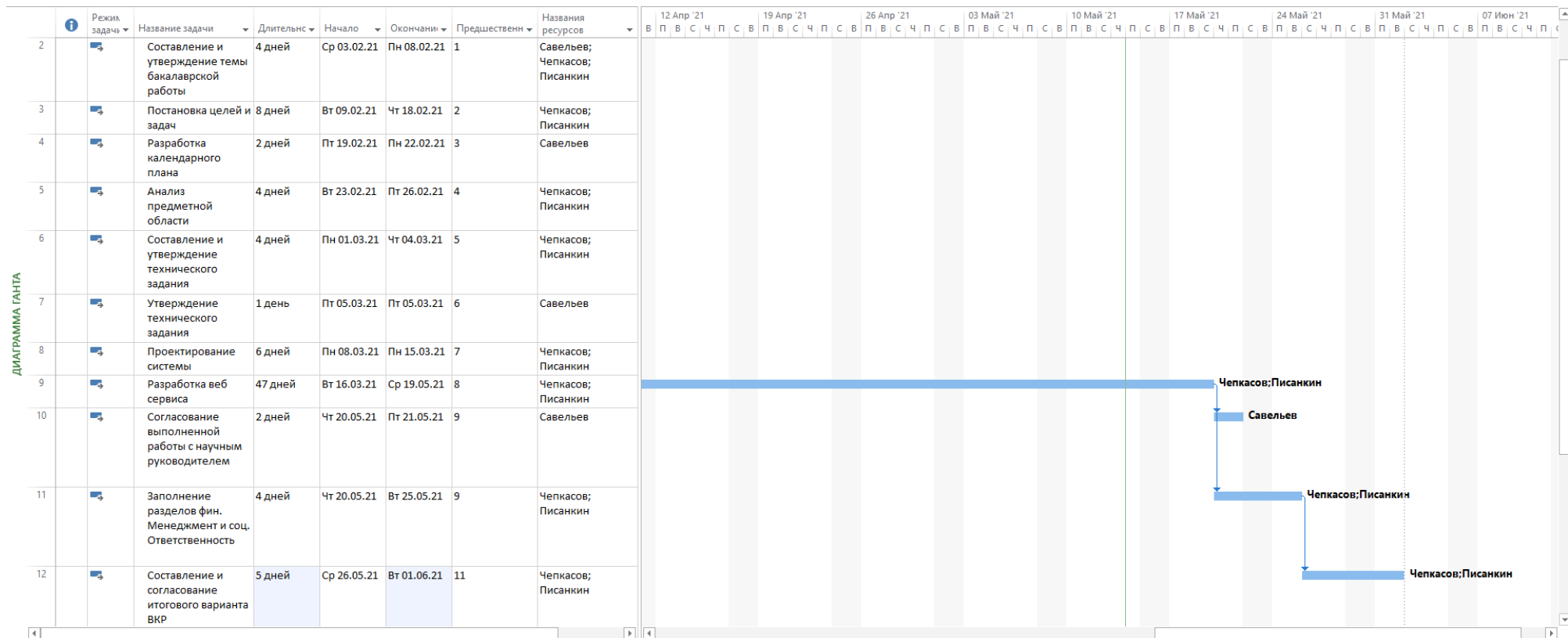


Рисунок 47 – Диаграмма Ганта (часть 2)

5.4 Бюджет научно-технического исследования (НТИ)

5.4.1 Расчет материальных затрат НТИ

Так как все прочие принадлежности, необходимые для работы, имелись у исполнителей, при расчете материальных затрат были учтены только компьютеры и компьютерная периферия, необходимая для разработки ПО. Также были учтены расходы на специализированное ПО. Итоговая таблица с затратами на оборудование для разработки ПО представлена ниже.

Таблица 22 – Материальные затраты

| Наименование оборудования | Количество единиц оборудования | | Цена за 1 ед. оборудования | | Общая стоимость оборудования, руб | |
|---------------------------|--------------------------------|-------|----------------------------|-------|-----------------------------------|-------|
| | Ст. 1 | Ст. 2 | Ст. 1 | Ст. 2 | Ст. 1 | Ст. 2 |
| Персональный компьютер | 1 шт. | 1 шт. | 43000 | 43000 | 43000 | 43000 |
| Монитор | 1 шт. | 1 шт. | 9500 | 9500 | 9500 | 9500 |
| Клавиатура | 1 шт. | 1 шт. | 650 | 650 | 650 | 650 |
| Компьютерная мышь | 1 шт. | 1 шт. | 580 | 580 | 580 | 580 |
| PyCharm Professional | - | 1 шт. | - | 14742 | - | 14742 |
| PhpStorm Professional | 1 шт. | - | 14742 | - | 14742 | - |
| Итого: | | | | | 136944 | |

Для проекта по разработке ПО амортизационные отчисления включают амортизацию используемого оборудования за время эксплуатации вплоть до момента окончания работ. Расчет амортизационных вычислений проводится по следующей формуле:

$$C_{AM} = \frac{H_A * Ц_{ОБ}}{F_D} * t_{рф} * n \quad (5)$$

где H_A – годовая норма амортизации;

$Ц_{ОБ}$ – цена оборудования;

F_d – действительный годовой фонд рабочего времени;

$t_{рф}$ – время работы вычислительной техники;

n – число задействованных единиц оборудования.

Для определения N_A необходимо определить срок амортизации оборудования C_A . Согласно постановлению правительства РФ «О классификации основных средств, включенных в амортизационные группы» C_A для персонального компьютера и периферии, составляет 3 года, следовательно, N_A равна обратному сроку амортизации оборудования, то есть 0,33. Для программного обеспечения C_A , равна 5 годам и N_A соответственно равен 0,2. Полный расчет затрат на амортизационные отчисления представлен в таблице 23.

Таблица 23 – Затраты на амортизационные отчисления

| Наименование оборудования | Норма амортиз. Оборуд., N_A | Стоим. Оборуд., Цоб, руб. | Факт. р/вр. Оборуд., $t_{рф}$, ч | Действ. Год. Фонд р/вр., F_d , ч. | Аморт. Отчисл., $C_{ам}$, руб. |
|---------------------------|-------------------------------|---------------------------|-----------------------------------|-------------------------------------|---------------------------------|
| Персональный компьютер | 0,33 | 43000 | 600 | 1976 | 8617,4 |
| Монитор | 0,33 | 9500 | 600 | 1976 | 1903,84 |
| Клавиатура | 0,33 | 650 | 600 | 1976 | 130,26 |
| Компьютерная мышь | 0,33 | 580 | 600 | 1976 | 116,23 |
| PyCharm Professional | 0,2 | 14742 | 600 | 1976 | 895,26 |
| PhpStorm Professional | 0,2 | 14742 | 600 | 1976 | 895,26 |
| Итого: | | | | | 12558,25 |

5.4.2 Основная заработная плата исполнителей темы

Основная зарплата – это часть, которая выплачивается работнику гарантированно, не зависимо от результатов труда. Рассчитанный баланс времени представлен в таблице ниже.

Таблица 24 – Баланс рабочего времени

| Показатели рабочего времени | Научный руководитель | Студент |
|--|----------------------|---------|
| Календарное число дней | 366 | 366 |
| Количество нерабочих дней (выходные дни/праздничные дни) | 66 | 66 |
| Потери рабочего времени (отпуск/невыходы по болезни) | 56 | 56 |
| Действительный годовой фонд рабочего времени | 243 | 243 |

Планирование основной заработной платы производилось из расчета того, что основная заработная плата научного руководителя составляет 1500 руб/день, студента – 700 руб/день.

Расчет основной заработной платы представлен в таблице 25.

Таблица 25 – Расчет основной заработной платы

| Исполнители | $Z_{гс}$, руб. | $k_{пр}$ | k_d | k_p | Z_m , руб | $Z_{дн}$, руб | T_p , раб. Дн | $Z_{осн}$, руб |
|------------------------------------|-----------------|----------|-------|-------|-------------|----------------|-----------------|-----------------|
| Чепкасов А.Ю. | 16153 | - | - | 1,3 | 21000 | 700 | 135 | 94 500 |
| Писанкин А.С. | 16153 | - | - | 1,3 | 21000 | 700 | 135 | 94 500 |
| Савельев А.О. | 34615 | - | - | 1,3 | 45000 | 1500 | 22 | 33 000 |
| Итого $Z_{осн}$: | | | | | | | | 222000 |

5.4.3 Дополнительная заработная плата исполнителей темы

Расчет дополнительной заработной платы осуществляется по формуле:

$$З_{\text{доп}} = k_{\text{доп}} * З_{\text{осн}} \quad (6)$$

где $З_{\text{доп}}$ – затраты по дополнительной заработной плате исполнителей темы;

$k_{\text{доп}}$ – коэффициент дополнительной заработной платы (на стадии проектирования принимается равным 0,12 – 0,15);

$З_{\text{осн}}$ – затраты по основной заработной плате исполнителей темы.

Ниже представлена таблица расчетов дополнительной заработной платы для работников.

Таблица 26 – Расчет дополнительной заработной платы

| Исполнители | Основная заработная плата | Набавочный коэффициент | Дополнительная заработная плата |
|---------------|---------------------------|------------------------|---------------------------------|
| Чепкасов А.Ю. | 94500 | 0,13 | 12285 |
| Писанкин А.С. | 94500 | | 12285 |
| Савельев А.О. | 33000 | | 4290 |
| Итого: | | | 28860 |

Таким образом сумма основной и дополнительной заработной платы составила 250860 рублей.

5.4.4 Отчисление во внебюджетные фонды

В данной статье расходов учитываются обязательные отчисления в фонд социального страхования, пенсионный фонд и фонд медицинского страхования. Размер отчислений зависит от размера заработной платы и вычисляется по следующей формуле:

$$З_{\text{внеб}} = k_{\text{внеб}} * (З_{\text{осн}} + З_{\text{доп}}) \quad (7)$$

где $З_{\text{внеб}}$ – размер отчислений на уплату во внебюджетные фонды;

$k_{\text{внеб}}$ – коэффициент отчислений во внебюджетные фонды;

$Z_{\text{осн}}$ – основная заработная плата работника;

$Z_{\text{доп}}$ – дополнительная заработная плата работника.

В таблице 15 представлен результат расчёта отчислений во внебюджетные фонды.

Таблица 27 – Расчет отчислений во внебюджетные фонды

| Исполнители | Основная заработная плата | Дополнительная заработная плата | Коэффициент отчислений во внебюджетные фонды | Сумма отчислений во внебюджетные фонды |
|---------------|---------------------------|---------------------------------|--|--|
| Чепкасов А.Ю. | 94500 | 12285 | 0,302 | 32249 |
| Писанкин А.С. | 94500 | 12285 | | 32249 |
| Савельев А.О. | 33000 | 4290 | | 11262 |
| Итого: | | | | 75760 |

Исходя из данных, представленных в таблице, суммарный размер отчислений во внебюджетные фонды составил 75760 рублей.

5.4.5 Накладные расходы

Данная статья расходов учитывает прочие затраты организации, которые не попали в предыдущие пункты. Величина данной статьи расходов определяется по формуле:

$$Z_{\text{накл}} = 0,16 * (Z_{\text{м}} + Z_{\text{осн}} + Z_{\text{доп}} + Z_{\text{внеб}} + Z_{\text{ам}}) \quad (8)$$

Таким образом, накладные расходы составляют:

$$Z_{\text{накл}} = 0,16 * (12558 + 222000 + 28860 + 75760) = 54268,48 \text{ руб.}$$

Для данной научно-исследовательской работы накладные расходы составляют примерно 54268 рублей.

5.4.6 Формирование бюджета затрат научно-исследовательского проекта

Все рассчитанные выше статьи бюджета формируют основу бюджета научно-исследовательской работы, которая является нижней границей реальных затрат. Результат вычислений всех статей расхода и их сумма представлены в таблице 28.

Таблица 28 – Расчет бюджета проекта

| Наименование статьи | Сумма, руб | Примечание |
|--|------------|--------------------|
| Амортизация основных средств | 12558 | Пункт 4.3.1 |
| Затраты на основную заработную плату | 222000 | Пункт 4.3.2 |
| Затраты на дополнительную заработную плату | 28860 | Пункт 4.3.3 |
| Страховые взносы | 75760 | Пункт 4.3.4 |
| Накладные расходы | 54268 | Пункт 4.3.5 |
| Общий бюджет | 393446 | Сумма всех пунктов |

5.5 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования

В данном разделе будет произведено определение эффективности научного исследования на основе расчета интегрального показателя. Его нахождение связано с определением двух средневзвешенных величин: финансовой эффективности и ресурсоэффективности.

Интегральный финансовый показатель разработки определяется как:

$$I_{\text{финр}}^{\text{исп.}i} = \frac{\Phi_{pi}}{\Phi_{\text{max}}}, \quad (9)$$

где $I_{\text{финр}}^{\text{исп.}i}$ – интегральный финансовый показатель разработки;

Φ_{pi} – стоимость i -го варианта исполнения;

Φ_{max} – максимальная стоимость исполнения научно-исследовательского проекта (в том числе аналоги).

$$I_{pi} = \sum a_i * b_i \quad (10)$$

где I_{pi} – интегральный показатель ресурсоэффективности;

a_i – весовой коэффициент i -го варианта исполнения разработки;

b_i – бальная оценка i -го варианта исполнения разработки.

Для расчета показателя используются данные, представленные в таблице 29.

Таблица 29 – Расчёт интегральных показателей ресурсоэффективности

| Критерии | Весовой коэффициент параметра | Исп.1 | Исп.2 | Исп.3 |
|----------------------------------|--------------------------------------|--------------|--------------|--------------|
| Потребность в ресурсах памяти | 0,15 | 4 | 5 | 3 |
| Предоставляемые услуги | 0,25 | 5 | 4 | 5 |
| Дружественность интерфейса | 0,05 | 3 | 4 | 3 |
| Критерии | Весовой коэффициент параметра | Исп.1 | Исп.2 | Исп.3 |
| Точность вычислительных операций | 0,30 | 5 | 5 | 5 |
| Скорость работы продукта | 0,25 | 5 | 5 | 5 |
| I_p | | 4,65 | 4,35 | 4,1 |

Сравнение интегрального показателя эффективности вариантов исполнения позволит определить сравнительную эффективность проекта и выбрать наиболее целесообразный вариант из предложенных.

Проведенный расчет данного показателя, а также сравнение вариантов исполнения по нему представлены в таблице 30.

Таблица 30 – Эффективность разработки

| Показатели | Исп.1 | Исп.2 | Исп.3 |
|---|--------------|--------------|--------------|
| Интегральный финансовый показатель разработки | 0.9 | 0.95 | 0,85 |
| Интегральный показатель ресурсоэффективности разработки | 4,65 | 4,35 | 4,1 |
| Интегральный показатель эффективности | 5.16 | 4.13 | 4.82 |
| Сравнительная эффективность вариантов исполнения | 1 | 0,85 | 0,93 |

Вывод по разделу

В результате выполнения работы по разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение» была рассмотрена финансовая сторона проекта по разработке ПО, выполняемого в рамках научно-исследовательской работы. Были определены сильные и слабые стороны в сравнении с конкурентами, а также эффективность разрабатываемого проекта. Для данного проекта был просчитан бюджет, составляющий 393446 рублей. Срок выполнения работы с 01.02.2021 по 02.06.2021 по 6-дневному рабочему графику.

Глава 6. Социальная ответственность

Введение

Современные приложения всё чаще переходят на облачные платформы. Это позволяет обеспечить доступ к функциям приложения почти в любое время, необходимо лишь наличие Интернет-соединения и браузера на пользовательском девайсе. Такие облачные технологии позволяют без привязки к вычислительным мощностям девайсов пользователя предоставлять весь функционал. Научно-исследовательская работа заключается в разработке программного обеспечения интеллектуального анализа данных о состоянии промышленных манипуляторов. Целевой аудиторией данного программного обеспечения в первую очередь являются предприятия, использующие промышленные манипуляторы. Соответственно для целевой аудитории будет реализован функционал, который отвечает их потребностям, а именно: регистрация, настройка проекта и приглашение туда пользователей, настройка модели машинного обучения, загрузка данных о физических показателях манипулятора, выбор параметров для анализа данных, просмотр и скачивание результатов анализа данных.

Программное обеспечение разрабатывалось во время прохождения производственной практики в ТПУ, ОИТ ИШИТР в 418 аудитории кибернетического центра ТПУ.

6.1 Правовые и организационные вопросы обеспечения безопасности

Разработка программного обеспечения происходит за компьютерным столом, следовательно, рабочее место должно удовлетворять требованиям ГОСТ 12.2.032-78 «Система стандартов безопасности труда (ССБТ). Рабочее место при выполнении работ сидя» [18] РД 153-34.0-03.298-2001 «Типовая инструкция по охране труда для пользователей персональными электронно-вычислительными машинами (ПЭВМ) в электроэнергетике» [19]. Требования к нормам труда

(продолжительность рабочего дня, перерывы в течении рабочего дня, перерывы на обед) регламентируется ТК РФ «Рабочее время» [25].

При разработке программного обеспечения интеллектуального анализа данных о состоянии промышленных манипуляторов было предоставлено рабочее место, где были соблюдены все требования по организации труда с ЭВМ.

6.2 Производственная безопасность

При разработке программного обеспечения разработчики подвергаются воздействию различных вредных и опасных факторов, которые представлены в таблице 31, вместе с соответствующими нормативными документами и этапами работ, во время которых разработчики могут столкнуться с их влиянием.

Таблица 31 – Возможные опасные и вредные факторы

| Факторы (ГОСТ 12.0.003-2015) | Этапы работ | | | | Нормативные документы |
|---|-------------|--------------|----------|--------------|--|
| | Разработка | Тестирование | Введение | Эксплуатация | |
| Отклонение показателей микроклимата | + | + | + | + | СанПиН 2.2.4.548-96 «Гигиенические требования к микроклимату производственных помещений» [20] |
| Недостаточная освещенность рабочей зоны | + | + | + | + | СП 52.13330.2016 «Естественное и искусственное освещение» [21] |

| Факторы (ГОСТ 12.0.003-2015) | Этапы работ | | | | Нормативные документы |
|---|-------------|--------------|----------|--------------|--|
| | Разработка | Тестирование | Введение | Эксплуатация | |
| Повышенная световая и цветовая контрастность | + | + | + | + | СП 52.13330.2016 «Естественное и искусственное освещение» [21] |
| Повышенный уровень шума на рабочем месте | + | + | + | + | СН 2.2.4/2.1.8.562-96 «Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории жилой застройки» [22] |
| Повышенный уровень статического электричества | + | + | + | + | ГОСТ Р 53734.1-2014 «Электростатические явления» [23] |
| Повышенная запыленность воздуха рабочей зоны | + | + | + | + | ГОСТ 12.1.005-88 «Общие санитарно-гигиенические требования к воздуху рабочей зоны» [24] |

Исходя из данной таблицы можно сделать вывод, что на разработчиков программного обеспечения в ходе их деятельности воздействуют только физические и психологические факторы, в то время как химические и биологические факторы отсутствуют.

6.2.1 Отклонение показателей микроклимата

Отклонение показателей микроклимата на рабочем месте от комфортных непосредственно влияет на здоровье работников. Повышение скорости движения воздуха и понижение температуры может привести к переохлаждению организма путем усиления теплообмена и процесса теплоотдачи при испарении пота. Недостаточная влажность в свою очередь ведет к интенсивному испарению влаги со слизистых оболочек. Это может привести к пересыханию, растрескиванию и затем к заражению болезнетворными микробами. При разработке программного обеспечения используются персональные компьютеры, которые могут непосредственно влиять на микроклимат путем снижения относительной влажности и повышению температуры в рабочем помещении.

Общие требования к микроклимату производственных помещений регламентируются СанПиН 2.2.4.548-96 «Гигиенические требования к микроклимату производственных помещений». Санитарные нормы регулируют оптимальные и допустимые значения показателей в рабочей зоне, соответствующие физиологическим потребностям организма человека, для создания комфортных и безопасных условий труда.

Работа, выполняемая командой разработки программного обеспечения, по энергозатратам относится к категории Ia (производится сидя, сопровождается незначительными физическими усилиями). В таблицах 32 и 33 представлены оптимальные и допустимые значения показателей микроклимата на рабочих местах для данной категории.

Таблица 32 – Оптимальные величины показателей микроклимата на рабочих местах

| Период года | Температура воздуха, °С | Температура поверхностей, °С | Относительная влажность воздуха, % | Скорость движения воздуха, м/с |
|-------------|-------------------------|------------------------------|------------------------------------|--------------------------------|
| Холодный | 22-24 | 21-25 | 60-40 | 0,1 |
| Теплый | 23-25 | 22-26 | 60-40 | 0,1 |

Таблица 33 – Допустимые величины показателей микроклимата на рабочих местах

| Период года | Температура воздуха, °С | | Температура поверхностей, °С | Относительная влажность воздуха, % | Скорость движения воздуха, м/с | |
|-------------|-----------------------------------|-----------------------------------|------------------------------|------------------------------------|---|---|
| | диапазон ниже оптимальных величин | диапазон выше оптимальных величин | | | для диапазона температур воздуха ниже оптимальных величин, не более | для диапазона температур воздуха выше оптимальных величин, не более |
| Холодный | 20,0 | 24,1 | 19,0 | 15 | 0,1 | 0,1 |
| | - | - | - | - | | |
| Теплый | 21,9 | 25,0 | 26,0 | 75 | 0,1 | 0,2 |
| | 21,0 | 25,1 | 20,0 | 15 | | |
| | - | - | - | - | 0,1 | 0,2 |
| | 22,9 | 28,0 | 29,0 | 75 | | |

Для минимизации воздействия отклонений показателей микроклимата в рабочем помещении необходимо использовать кондиционеры, обогреватели и увлажнители воздуха, которые помогают привести эти показатели к оптимальным значениям.

6.2.2 Недостаточна освещенность рабочей зоны

Недостаточная освещенность рабочей зоны является вредным производственным фактором, приводящим к быстрому утомлению и снижению работоспособности человека на предприятии. При недостаточной освещенности помещение человек быстрее устает, снижается внимание и концентрация, что приводит к возникновению опасных производственных ситуаций. Продолжительная работа в условиях низкой освещенности приводит к ухудшению зрения.

Нормы естественного, искусственного и совместного освещения регламентируются СП 52.13330.2016 «Естественное и искусственное освещение». Разработка программного обеспечения относится к категории работ высокой точности – Б (наименьший или эквивалентный объект различения 0,30 – 0,50 мм), подразряд 1 (относительная продолжительность зрительной работы при направлении зрения на рабочую поверхность не менее 70%).

В таблице 34 представлены требования к освещению рабочего помещения для разряда Б1.

Таблица 34 – Требования к освещению рабочего помещения

| Искусственное освещение | | | | Естественное освещение | |
|---|---------------------------------|---|---|---|---------|
| Освещенность на рабочей поверхности от системы общего освещения, лк | Цилиндрическая освещенность, лк | Объединенный показатель дискомфорта, не более | Коэффициент пульсации и освещенности, Кп, %, не более | Коэффициент естественной освещенности, %, при | |
| | | | | верхнем или комбинированном | боковом |
| 300 | 100 | 21 | 15 | 3 | 1 |

Яркий свет в зоне периферийного зрения заметно увеличивает глазное напряжение. Для снижения влияния вредного фактора недостаточной освещенности необходимо, чтобы уровень естественного освещения рабочего пространства приблизительно совпадал с яркостью дисплея. Проблему недостаточной освещенности помещения можно решить при помощи установки дополнительных осветительных приборов, расширения световых проемов.

6.2.3 Повышенная световая и цветовая контрастность

Отклонение светового и цветового контраста на рабочем месте приводит к быстрому утомлению и снижению уровня работоспособности человека на предприятии. Продолжительное воздействие этого вредного фактора приводит к возникновению проблем со зрением. Нормы светового и цветового контраста регламентируются СП 52.13330.2016 «Естественное и искусственное

освещение». Для работы за компьютером (категория работ Б1) нормы контраста представлены в таблице 35.

Таблица 35 – Требования к освещению рабочего помещения

| Характеристика зрительной работы | Контраст объекта с фоном | Характеристика фона |
|---|---------------------------------|----------------------------|
| Высокой точности | Малый Средний | Средний Темный |

Для изменения светового и цветового контраста необходимо отрегулировать уровень естественной и искусственной освещенности рабочего помещения или заменить текущее оборудование (мониторы) на более качественные, которые позволят сгладить контраст.

6.2.4 Повышенный уровень шума на рабочем месте

Превышения уровня шума на рабочем месте создает психологический и физический стресс, снижающий производительность, концентрацию, внимание, повышает утомляемость. Повышение уровня шума на рабочем месте возможно из-за фона, создаваемого работой персональных компьютеров, наличия центральной системы вентиляции и кондиционирования воздуха.

Предельно допустимые показатели уровня звука, звукового давления регламентируются СН 2.2.4/2.1.8.562-96 «Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории жилой застройки»

Для команды разработчиков программного обеспечения, эти показатели представлены в таблице 36.

Таблица 36 – Предельно допустимые уровни звукового давления, уровни звука и эквивалентные уровни звука для инженера-программиста

| Вид трудовой деятельности, рабочее место | Уровни звукового давления, дБ в октавных полосах со среднегеометрическими частотами, Гц | | | | | | |
|--|--|----|-----|-----|-----|------|------|
| | 31,5 | 63 | 125 | 250 | 500 | 1000 | 2000 |
| Творческая деятельность, руководящая работа с повышенными требованиями, научная деятельность, конструирование и проектирование, программирование, преподавание и обучение, врачебная деятельность. Рабочие места в помещениях дирекции, проектно-конструкторских бюро, расчетчиков, программистов вычислительных машин, в лабораториях для теоретических работ и обработки данных, приема больных в здравпунктах | 86 | 71 | 61 | 54 | 49 | 45 | 42 |

Для снижения уровня шума в производственном помещении можно использовать защитные звукопоглощающие экраны. Для любого оборудования необходимо регулярно проводить техническое обслуживание, так как загрязнение может увеличить производимый шум.

6.2.5 Повышенный уровень статического электричества

Статическое электричество является опасным производственным фактором, проявление которого может нанести вред здоровью человека (ожоги) или привести пожару и другим чрезвычайным ситуациям.

При работе за компьютером статический заряд может накапливаться если нет хорошего контакта с землей или влажность/ионизация воздуха превышает допустимые нормы. Статический разряд в производственных помещениях рассматриваемого типа при условии соответствии нормам микроклимата и организации работ при воздействии на человека вызывает только дискомфорт.

Допустимые показатели уровня статического электричества на производстве регламентируются ГОСТ Р 53734.1-2014 «Электростатические явления». В таблице ниже представлены уровни восприятия электростатического заряда человеком.

Таблица 37 – Уровни восприятия людьми электростатического заряда и ответной реакции при емкости тела в 200 пФ

| Энергия разряда, мДж | Реакция | Потенциал тела, В |
|----------------------|----------------|-------------------|
| 0,1 | Разряд ощутим | 1000 |
| 0,9 | Четко ощутим | 3000 |
| 6,4 | Неприятный шок | 8000 |

Для уменьшения накапливаемого статического заряда при работе за компьютером необходимо организовать антистатические рабочие места, соблюдать установленную норму влажности воздуха и поддерживать чистоту помещения, поскольку пыль обладает свойствами диэлектрика.

6.2.6 Повышенная запыленность воздуха рабочей зоны

Пыль характеризуется совокупностью свойств, определяющих поведение ее в воздухе, превращение и действие на организм человека.

Вредное воздействие пыли на организм человека зависит от ряда факторов: концентрации в воздухе, химического состава, размеров частиц, дисперсности, твердости, заряженности пылинок. Норма запыленности воздуха регламентируется ГОСТ 12.1.005-88 «Общие санитарно-гигиенические требования к воздуху рабочей зоны».

В офисном помещении, пыль может оказывать на организм раздражающее и аллергическое действия. В таблице 38 представлены предельно допустимые значения концентрации пыли и аэрозолей в воздухе жилых помещений.

Таблица 38 – Предельно допустимые концентрации пыли и аэрозолей в воздухе

| | | | |
|--|------------------------|-------------------------|-------------------------|
| Взвешенные частицы PM2.5 | 0,16 мг/м ³ | 0,035 мг/м ³ | 0,025 мг/м ³ |
| Взвешенные частицы PM10 | 0,3 мг/м ³ | 0,06 мг/м ³ | 0,04 мг/м ³ |
| Взвешенные частицы (общая пыль) | 0,5 мг/м ³ | 0,15 мг/м ³ | – |
| Сажа (углерод) | 0,15 мг/м ³ | 0,05 мг/м ³ | – |

Для снижения уровня содержащейся в воздухе пыли необходимо организовать систему вентиляции воздуха помещения и производить регулярную уборку помещений.

6.3 Экологическая безопасность

Программное обеспечение не оказывает влияния на окружающую среду, так как разрабатывается и используется внутри персональных компьютеров, однако использование самого компьютера может оказать влияние на окружающую среду. Так при нагревании материнской платы и корпуса монитора происходит выброс в воздух вредных вещества, а при работе самого компьютера воздух вокруг него становится слабо ионизированным. Всё это приводит к повышенной сухости воздуха на рабочем месте воздух сухим со специфическим запахом.

При производстве компьютеров и их комплектующих используются материалы, которые могут нанести вред окружающей среде, поэтому и неправильная утилизация компьютерной техники может стать причиной загрязнения литосферы. Утилизировать компьютер необходимо после извлечения его компонент, их сортировку и отправку на повторное использование. Всё это необходимо делать на специально отведенных полигонах с присутствием квалифицированного персонала.

Соблюдение всех норм при использовании и утилизации компьютерной техники позволит уменьшить вредное воздействие на окружающую среду и предотвращает попадание вредных веществ в организм.

6.4 Безопасность в чрезвычайных ситуациях

6.4.1 Короткое замыкание

Работа с компьютерной техникой подразумевает постоянное использование электрического тока. При несоблюдении правил электробезопасности возможно возникновение таких чрезвычайных ситуаций, как пожар и короткое замыкание проводки.

Возникновение короткого замыкания может быть обусловлено использованием неисправного оборудования. Если в это время человек находится рядом с участком цепи в котором произошло короткое замыкание, он может получить ожоги, которые могут оказаться смертельными. Замыкание может привести к выводу из строя прибора и стать причиной пожара.

Для минимизации негативных последствий короткого замыкания необходимо использовать кабель не распространяющий горение, или помещать его в стальные трубы с определенной толщиной стенки, которая не прожжется в результате возникновения короткого замыкания.

6.4.2 Землетрясение

Землетрясение происходит в результате возникновения подземных толчков и колебаний земной поверхности из-за внезапных смещений и разрывов в земной коре или верхней мантии. Данная чрезвычайная ситуация имеет природный характер и может привести к выходу из строя коммуникации и энергетических объектов, разрушению здания, появлению трещины в грунте и возникновению пожаров. Все перечисленные последствия могут привести к значительным людским потерям.

6.4.3 Затопление

При наличии сильной изношенности водопровода, свищей или негерметичных соединений может произойти затопление помещения. Главная опасность при затоплении помещения, в котором находятся ЭВМ – это возможность воды проводить электрический ток. На самом деле ток проводит не вода, т.е. не молекулы воды, а различные примеси, содержащиеся в ней, в частности ионы различных минеральных солей, которых в сточной воде достаточное количество. Действие электрического тока подразделяется на 4 типа:

- Тепловое: ожоги;
- Механическое: растрескивание костей, разрывы тканей;

- Химическое: электролизу;
- Биологическое: нарушению деятельности нервной системы и органов.

6.4.4 Наиболее вероятная ЧС: Пожар

Причиной возникновения пожар при работе с ЭВМ может служить короткое замыкание проводки, вследствие неисправности прибора, сильный перегрев ЭВМ в результате использования ЭВМ в режиме повышенной нагрузки.

Для предотвращения возникновения пожара, необходима периодическая, своевременная диагностика по обнаружению неисправностей в оборудовании и электрической проводки, соблюдение норм при работе с ЭВМ, обязательное наличие средств пожаротушения и пожарного оборудования в рабочем помещении, готовых к эксплуатации.

Порядок действий при возникновении пожара в случае, если дым и пламя позволяют выйти из помещения (здания) наружу:

1. Уходите от огня, используя основные и запасные пути эвакуации;
2. Отключите электроэнергию в здании;
3. Идите к выходу на четвереньках, так как вредные продукты горения скапливаются на уровне роста человека и выше, закрывая при этом рот и нос подручными средствами защиты;
4. Покинув опасное помещение, не возвращайтесь назад
5. Сообщите о себе должностным лицам.

Если дым и пламя в соседних помещениях не позволяет выйти наружу:

1. Не поддавайтесь панике;
2. Накройтесь полностью мокрым покрывалом (тканью);
3. Проверьте существует ли возможность выйти на крышу или спуститься по пожарной лестнице;
4. Если возможности эвакуироваться нет, то для защиты от тепла и дыма необходимо:

- 4.1. Плотно закрыть входную дверь и щели двери изнутри помещения, используя при этом любую ткань;
 - 4.2. Закрыть окна, форточки и вентиляционные отверстия;
 - 4.3. Если в помещении есть вода, постоянно необходимо смачивать дверь и пол.
5. Если помещение наполнилось дымом, передвигайтесь на четвереньках, прикрыв рот и нос влажной тряпкой (носовым платком, рукавом от рубашки), в сторону окна и находитесь возле окна и привлекайте к себе внимание людей на улице.

Вывод по разделу

В результате работы по разделу «Социальная ответственность» были выявлены основные нормативные акты для обеспечения безопасности жизнедеятельности на рабочем месте, рассмотрены наиболее значимые опасные и вредные факторы, возникающие при разработке программного обеспечения интеллектуального анализа данных о состоянии промышленных манипуляторов, описано влияние процесса разработки данного программного обеспечения на окружающую среду и меры, необходимые для уменьшения влияния вредных и опасных факторов на организм человека и для сокращения негативного влияния процесса разработки программного обеспечения на окружающую среду.

Заключение

В результате выполнения выпускной квалификационной работы был разработан веб-сервис прогнозирования изменения температуры подвижных частей промышленного манипулятора на основе значений физических показателей приводов в штатном режиме работы.

На первом этапе был проведен анализ предметной области, существующих способов формирования плана сервисных работ на предприятиях и аналогов разрабатываемой системы. В результате были сформулированы основные бизнес требования к системе на основании которых были разработаны UML диаграммы для формирования концептуального представления о разрабатываемом продукте, описывающие функционал системы.

Следующим этапом был проведен анализ инструментов разработки и подходов реализации высоконагруженных, распределенных систем. В результате была выбрана микросервисная архитектура. По завершении процесса проектирования и разработки системы было составлено описание результатов проделанных этапов.

Были определены затраты на разработку и эксплуатацию системы с целью определения выгоды реализации собственного решения по сравнению с использованием существующих аналогов.

Апробация разработанного программного продукта на примере прогнозирования температуры приводов промышленного манипулятора продемонстрировала точность прогноза на краткосрочный период равную 86%.

Список публикаций студентов

1. А.С. Писанкин, А.Ю. Чепкасов, А.С. Гончаров. Исследование зависимостей физических характеристик при работе промышленного манипулятора с использованием методов машинного обучения // МСИТ. – 2020. – [С. 68].
2. А.С. Гончаров, А.О. Савельев, А.С. Писанкин, А.Ю. Чепкасов, Джаякоди Д.-Налин К. Прогнозирование перегрева привода промышленного манипулятора с использованием методов интеллектуального анализа данных // Доклады ТУСУР. – 2021. – Т. 24, № 1. – [С. 48].
3. А.С. Писанкин, А.Ю. Чепкасов, студенты, А.С. Гончаров. Прогнозирование перегрева привода промышленного манипулятора с использованием методов интеллектуального анализа данных // Электронные средства и системы управления: материалы докладов XVI Международной научно-практической конференции (18–20 ноября 2020 г.): 2 ч. – [С. 10].

Список используемых источников

1. ГОСТ 12.2.032-78 Техническая диагностика. Термина и определения // Электронный фонд правовой и нормативно-технической документации [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/1200009481> (дата обращения: 01.06.2021).
2. Статья «Feature subset selection Filter–Wrapper based on low quality data» [Электронный ресурс]. – Режим доступа: <https://www.sciencedirect.com/science/article/pii/S0957417413003497?via%3Dihub> (дата обращения: 10.05.2021).
3. Статья «Popular feature selection methods in machine learning» [Электронный ресурс]. – Режим доступа: <https://dataaspirant.com/feature-selection-methods-machine-learning/>
4. Статья «Корреляция и машинное обучение» [Электронный ресурс]. – Режим доступа: <https://ichi.pro/ru/korrelacia-i-masinnoe-obucenie-280242317294252> (дата обращения: 01.06.2021).
5. Манипуляторы KUKA [Электронный ресурс]. – Режим доступа: <https://www.kuka.com/en-de/products>, свободный (дата обращения: 08.05.2021).
6. Статья «Electric motor: predicting Stator Temperature» [Электронный ресурс]. – Режим доступа: <https://www.kaggle.com/ke2207/electric-motor-predicting-stator-temperature/notebook> свободный (дата обращения 01.06.2021).
7. Microservice Architecture [Электронный курс]. – Режим доступа: <https://microservices.io/> (дата обращения 01.06.2021).
8. Искусственный интеллект повсюду [Электронный ресурс]. – Режим доступа: <https://www.pwc.com/gx/en/issues/data-and-analytics/artificial-intelligence.html> (дата обращения: 01.06.2021).

9. Deploying Django to Azure [Электронный ресурс]. URL: <https://azure.microsoft.com/ru-ru/resources/videos/azure-friday-python-on-azure-part-2-deploying-django-services-to-azure-web-apps/> (Дата обращения 01.06.2021).
10. Django с Postgres [Электронный курс]. Режим доступа: <https://docs.microsoft.com/ru-ru/azure/app-service/tutorial-python-postgresql-app?tabs=bash%2Cclone> (Дата обращения 01.06.2021).
11. Docker: основные термины и принципы [Электронный ресурс]. – Режим доступа: <https://aws.amazon.com/ru/docker/> (дата обращения: 01.06.2021).
12. Trello – программная для управления проектами небольших групп [Электронный ресурс]. – Режим доступа: <https://trello.com/> (дата обращения 01.06.2021).
13. IXON [Электронный ресурс]. – Режим доступа: <https://www.ixon.cloud/applications> (дата обращения: 01.06.2021).
14. AVEA [Электронный ресурс]. – Режим доступа: <https://www.aveva.com/en/products/predictive-analytics/> (дата обращения: 01.06.2021).
15. SDC [Электронный ресурс]. – Режим доступа: <https://www.sdcautomation.com/about/> (дата обращения: 01.06.2021).
16. Popular feature selection methods in machine learning [Электронный ресурс]. – Режим доступа: <https://dataaspirant.com/feature-selection-methods-machine-learning/> (дата обращения: 01.06.2021).
17. Anomaly Detection: Techniques to detect outliers [Электронный ресурс] – Режим доступа: <https://towardsdatascience.com/anomaly-detection-techniques-to-detect-outliers-fea92047a222> (дата обращения: 01.06.2021).
18. ГОСТ 12.2.032-78 Система стандартов безопасности труда (ССБТ). Рабочее место при выполнении работ сидя. Общие эргономические требования // Электронный фонд правовой и нормативно-технической

- документации [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/1200003913> (дата обращения: 01.06.2021).
19. РД 153-34.0-03.298-2001 Типовая инструкция по охране труда для пользователей персональными электронно-вычислительными машинами (ПЭВМ) в электроэнергетике // Электронный фонд правовой и нормативно-технической документации [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/1200031404> (дата обращения: 01.06.2021).
20. СанПиН 2.2.4.548-96 Гигиенические требования к микроклимату производственных помещений // Электронный фонд правовой и нормативно-технической документации [Электронный ресурс]. 2021. – Режим доступа: <https://docs.cntd.ru/document/901704046> (дата обращения: 01.06.2021).
21. СП 52.13330.2016 «Естественное и искусственное освещение» // Электронный фонд правовой и нормативно-технической документации [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/456054197> (дата обращения: 01.06.2021).
22. СН 2.2.4/2.1.8.562-96 «Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории жилой застройки» // Электронный фонд правовой и нормативно-технической документации [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/901703278> (дата обращения: 01.06.2021).
23. ГОСТ Р 53734.1-2014 «Электростатические явления» // Электронный фонд правовой и нормативно-технической документации [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/1200111323> (дата обращения: 01.06.2021).

24. ГОСТ 12.1.005-88 «Общие санитарно-гигиенические требования к воздуху рабочей зоны»// Электронный фонд правовой и нормативно-технической документации [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/1200003608> (дата обращения: 01.06.2021).
25. Трудовой кодекс (ТК РФ) «Рабочее время»// Электронный фонд правовой и нормативно-технической документации [Электронный ресурс]. – Режим доступа: <http://base.garant.ru/12125268/> (дата обращения: 01.06.2021).

Приложение А

Описание таблицы «NotificationTemplate» представлено в таблице 39.

Таблица 39 – Описание атрибутов таблицы «NotificationTemplate»

| № | Название | Описание | Тип данных | Ограничения |
|---|------------------|---|------------|-------------------------------|
| 1 | UID | Уникальный идентификатор записи шаблона сообщения | Integer | Primary Key |
| 2 | content_template | Шаблон для создания сообщения в соответствии с событием | String | NOT NULL |
| 3 | event_id | Идентификатор события для создания сообщения | Integer | Foreign Key, UNIQUE, NOT NULL |

Описание таблицы «Events» представлено в таблице 40.

Таблица 40 – Описание атрибутов таблицы «Events»

| № | Название | Описание | Тип данных | Ограничения |
|---|----------|---|------------|------------------|
| 1 | UID | Уникальный идентификатор записи об событиях системы | Integer | Primary Key |
| 2 | event | Название события | String | UNIQUE, NOT NULL |

Описание таблицы «Notification» представлено в таблице 41.

Таблица 41 – Описание атрибутов таблицы «Notification»

| № | Название | Описание | Тип данных | Ограничения |
|---|----------|--|------------|-------------|
| 1 | UID | Уникальный идентификатор записи о тексте сообщения пользователю | Integer | Primary Key |
| 2 | content | Заполненный данными шаблон сообщения из таблицы «NotificationTemplate» | String | NOT NULL |

Описание таблицы «Message» представлено в таблице 42.

Таблица 42 – Описание атрибутов таблицы «Message»

| № | Название | Описание | Тип данных | Ограничения |
|---|-------------------|--|------------|-----------------------|
| 1 | UID | Уникальный идентификатор записи о сообщении пользователю | Integer | Primary Key |
| 2 | notification_id | Идентификатор текста сообщения | Integer | Foreign Key, NOT NULL |
| 3 | user_id | Идентификатор получателя сообщений (пользователя) | Integer | Foreign Key, NOT NULL |
| 4 | date | Отправки сообщения | date | NOT NULL |
| 5 | message_status_id | Идентификатор статуса сообщения | Integer | Foreign Key, NOT NULL |

Описание таблицы «MessageStatus» представлено в таблице 43.

Таблица 43 – Описание атрибутов таблицы «MessageStatus»

| № | Название | Описание | Тип данных | Ограничения |
|---|----------|---|------------|------------------|
| 1 | UID | Уникальный идентификатор записи о статусе сообщения | Integer | Primary Key |
| 2 | status | Статус сообщения (получено, прочитано...) | String | UNIQUE, NOT NULL |

Описание таблицы «UserProjects» представлено в таблице 44.

Таблица 44 – Описание атрибутов таблицы «UserProjects»

| № | Название | Описание | Тип данных | Ограничения |
|---|----------|--|------------|--------------------------|
| 1 | user_id | Уникальный идентификатор записи о пользователе системы | Integer | Foreign Key, Primary Key |

| № | Название | Описание | Тип данных | Ограничения |
|---|------------|---|------------|-------------------------------|
| 2 | project_id | Уникальный идентификатор записи о проекте | String | Foreign Key, NOT NULL, UNIQUE |
| 3 | role | Роль пользователя на проекте: администратор, редактор, обычный пользователь | String | NOT NULL, UNIQUE |

Описание таблицы «Users» представлено в таблице 45.

Таблица 45 – Описание атрибутов таблицы «Users»

| № | Название | Описание | Тип данных | Ограничения |
|---|-----------------|---|------------|-----------------------|
| 1 | UID | Уникальный идентификатор записи о пользователе системы | Integer | Primary Key |
| 2 | user_name | Логин пользователя для отображения в системе | String | NOT NULL, UNIQUE |
| 3 | first_name | Имя пользователя для отображения в системе | String | NOT NULL, |
| 4 | last_name | Фамилия пользователя для отображения в системе | String | NOT NULL, |
| 5 | password | Пароль для выхода в личный кабинет | String | NOT NULL, UNIQUE |
| 6 | email | Е-мейл пользователя для входа в личный кабинет и отправки уведомлений | String | NOT NULL, UNIQUE |
| 7 | telegram_chanel | Ссылка на телеграм (в случае отправки уведомлений в телеграм) | String | NOT NULL, UNIQUE |
| 8 | company_id | Идентификатор компании | Integer | Foreign Key, NOT NULL |
| 9 | token | Личный токен доступа | String | UNIQUE |

Описание таблицы «Company» представлено в таблице 46.

Таблица 46 – Описание атрибутов таблицы «Company»

| № | Название | Описание | Тип данных | Ограничения |
|---|----------|--|------------|------------------|
| 1 | UID | Уникальный идентификатор записи о компании | Integer | Primary Key |
| 2 | name | Название компании | String | NOT NULL, UNIQUE |

Описание таблицы «Project» представлено в таблице 47.

Таблица 47 – Описание атрибутов таблицы «Project»

| № | Название | Описание | Тип данных | Ограничения |
|---|----------|---|------------|-----------------------|
| 1 | UID | Уникальный идентификатор записи о проекте | Integer | Primary Key |
| 2 | name | Название проекта | String | NOT NULL |
| 3 | owner_id | Идентификатор владельца проекта | Integer | Foreign Key, NOT NULL |

Описание таблицы «ProjectModelTemplate» представлено в таблице 48.

Таблица 48 – Описание атрибутов таблицы «ProjectModelTemplate»

| № | Название | Описание | Тип данных | Ограничения |
|---|-------------|--|------------|-----------------------|
| 1 | UID | Уникальный идентификатор записи о шаблоне для создания модели машинного обучения | Integer | Primary Key |
| 2 | project_id | Идентификатор проекта, которому принадлежит эта запись | Integer | Foreign Key, NOT NULL |
| 3 | name | Название шаблона | String | NOT NULL |
| 4 | description | Описание шаблона | String | - |

Описание таблицы «Manipulator» представлено в таблице 49.

Таблица 49 – Описание атрибутов таблицы «Manipulator»

| № | Название | Описание | Тип данных | Ограничения |
|---|------------|---|------------|-----------------------|
| 1 | UID | Уникальный идентификатор записи о промышленном манипуляторе | Integer | Primary Key |
| 2 | name | Название манипулятора | String | NOT NULL |
| 3 | project_id | Идентификатор проекта, которому принадлежит эта запись | Integer | Foreign Key, NOT NULL |

Описание таблицы «ML_Model» представлено в таблице 50.

Таблица 50 – Описание атрибутов таблицы «ML_Model»

| № | Название | Описание | Тип данных | Ограничения |
|---|----------------|--|------------|-----------------------|
| 1 | UID | Уникальный идентификатор записи о модели машинного обучения | Integer | Primary Key |
| 2 | manipulator_id | Идентификатор манипулятора, которому принадлежит эта модель машинного обучения | Integer | Foreign Key, NOT NULL |
| 3 | teach_file | Идентификатор файла для обучения | Integer | NOT NULL |
| 4 | name | Имя модели машинного обучения | String | NOT NULL |
| 5 | description | Описание модели | String | NOT NULL |
| 6 | weights | Весы модели (формируются после обучения) | String | NOT NULL |

Описание таблицы «Stage» представлено в таблице 51.

Таблица 51 – Описание атрибутов таблицы «Stage»

| № | Название | Описание | Тип данных | Ограничения |
|---|--------------|---|------------|-----------------------|
| 1 | UID | Уникальный идентификатор записи о состоянии обучения модели | Integer | Primary Key |
| 2 | model_id | Идентификатор манипулятора, которому принадлежит эта запись | Integer | Foreign Key, NOT NULL |
| 3 | created_date | Дата создания записи о состоянии обучения модели | date | NOT NULL |
| 4 | description | Описание процесса обучения | String | NOT NULL |
| 5 | status_id | Идентификатор статуса обучения модули | Integer | Foreign Key, NOT NULL |

Описание таблицы «Status» представлено в таблице 52.

Таблица 52 – Описание атрибутов таблицы «Status»

| № | Название | Описание | Тип данных | Ограничения |
|---|----------|---------------------------------|------------|-------------|
| 1 | UID | Уникальный идентификатор записи | Integer | Primary Key |
| 2 | status | Статус обучения модули | String | NOT NULL |

Описание таблицы «Files» представлено в таблице 53.

Таблица 53 – Описание атрибутов таблицы «Files»

| № | Название | Описание | Тип данных | Ограничения |
|---|----------|---------------------------------|------------|-------------|
| 1 | UID | Уникальный идентификатор записи | Integer | Primary Key |

| № | Название | Описание | Тип данных | Ограничения |
|---|----------------|--|------------|-----------------------|
| 2 | manipulator_id | Идентификатор манипулятора, к которому относится файл | Integer | Foreign Key, NOT NULL |
| 3 | name | Название файла | String | NOT NULL |
| 4 | data | Данные содержащиеся в файле (физические показатели манипулятора) | JSON | NOT NULL |

Описание таблицы «Results» представлено в таблице 54.

Таблица 54 – Описание атрибутов таблицы «Results»

| № | Название | Описание | Тип данных | Ограничения |
|---|--------------|--|------------|-----------------------|
| 1 | UID | Уникальный идентификатор записи | Integer | Primary Key |
| 2 | model_id | Идентификатор модели машинного обучения, на основе которой были сформированы результаты (предсказание температуры) | Integer | Foreign Key, NOT NULL |
| 3 | test_file | Идентификатор файла с тестовыми данными для предсказания температуры | Integer | Foreign Key, NOT NULL |
| 4 | created_date | Дата создания результата | date | NOT NULL |
| 5 | result_data | Результаты работы модели машинного обучения (предсказание температуры) | JSON | NOT NULL |

Приложение Б

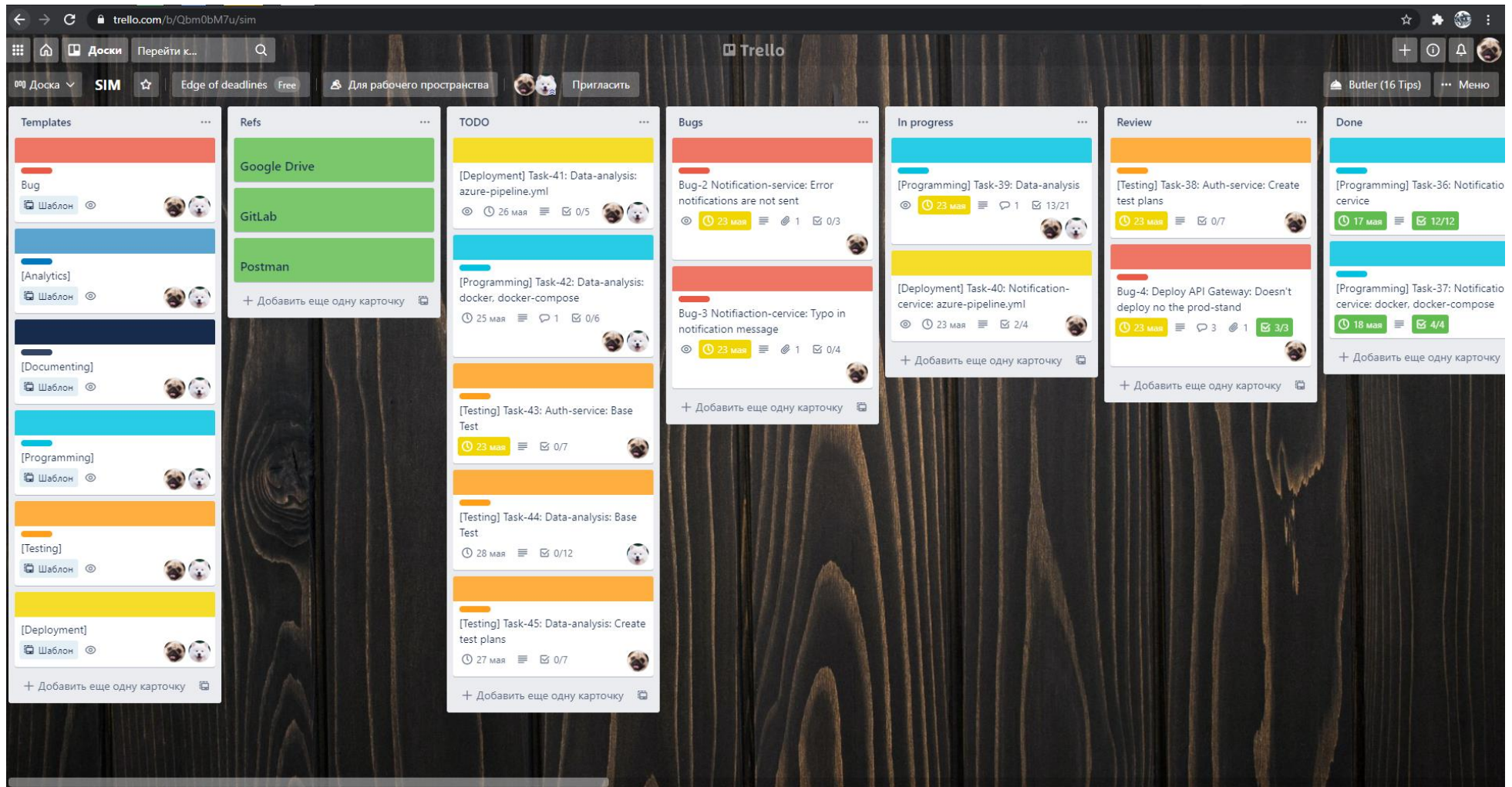


Рисунок 48 – Общий вид Kanban доски

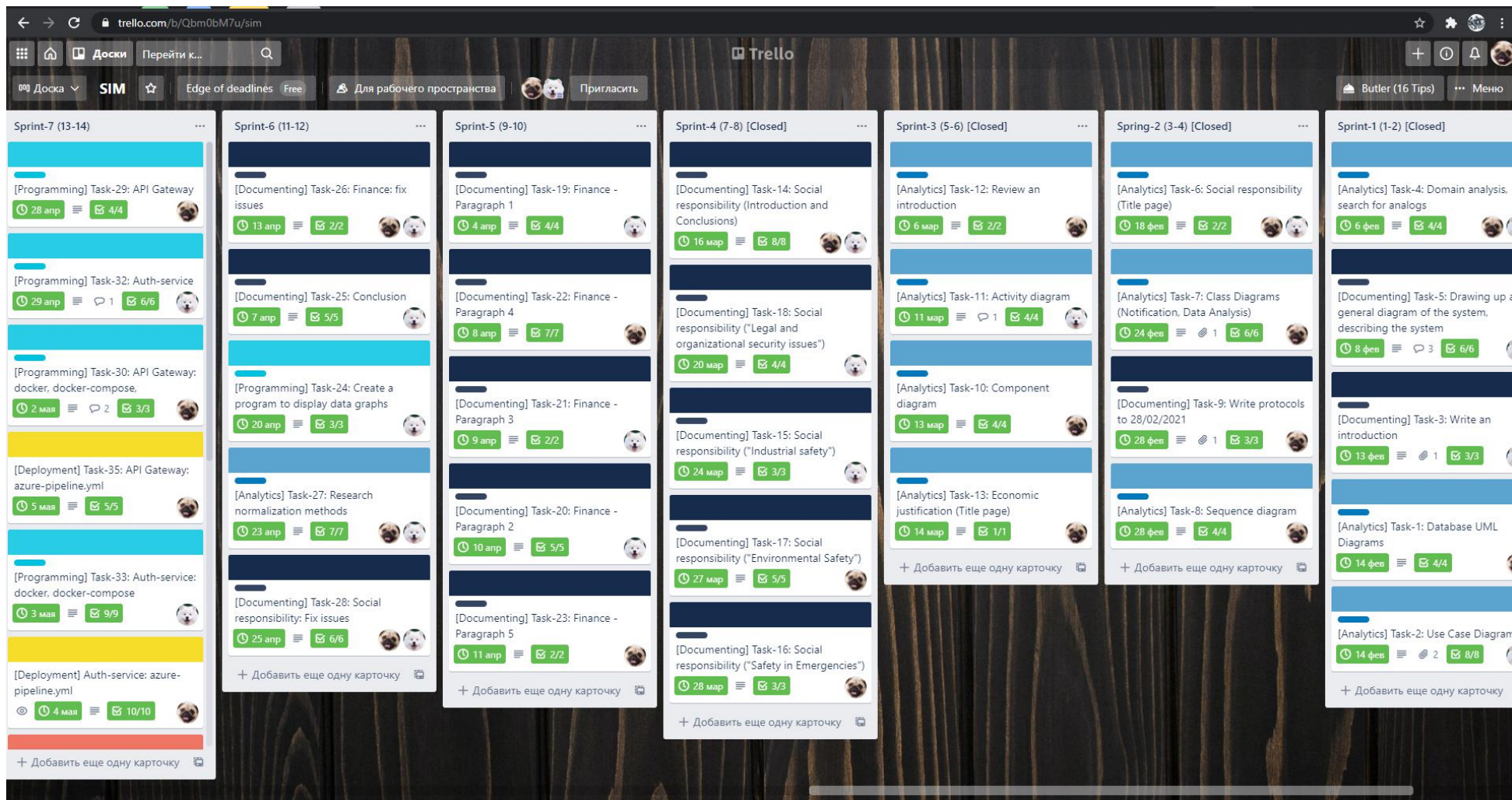


Рисунок 49 – Спринты в Канбан доске

Приложение В

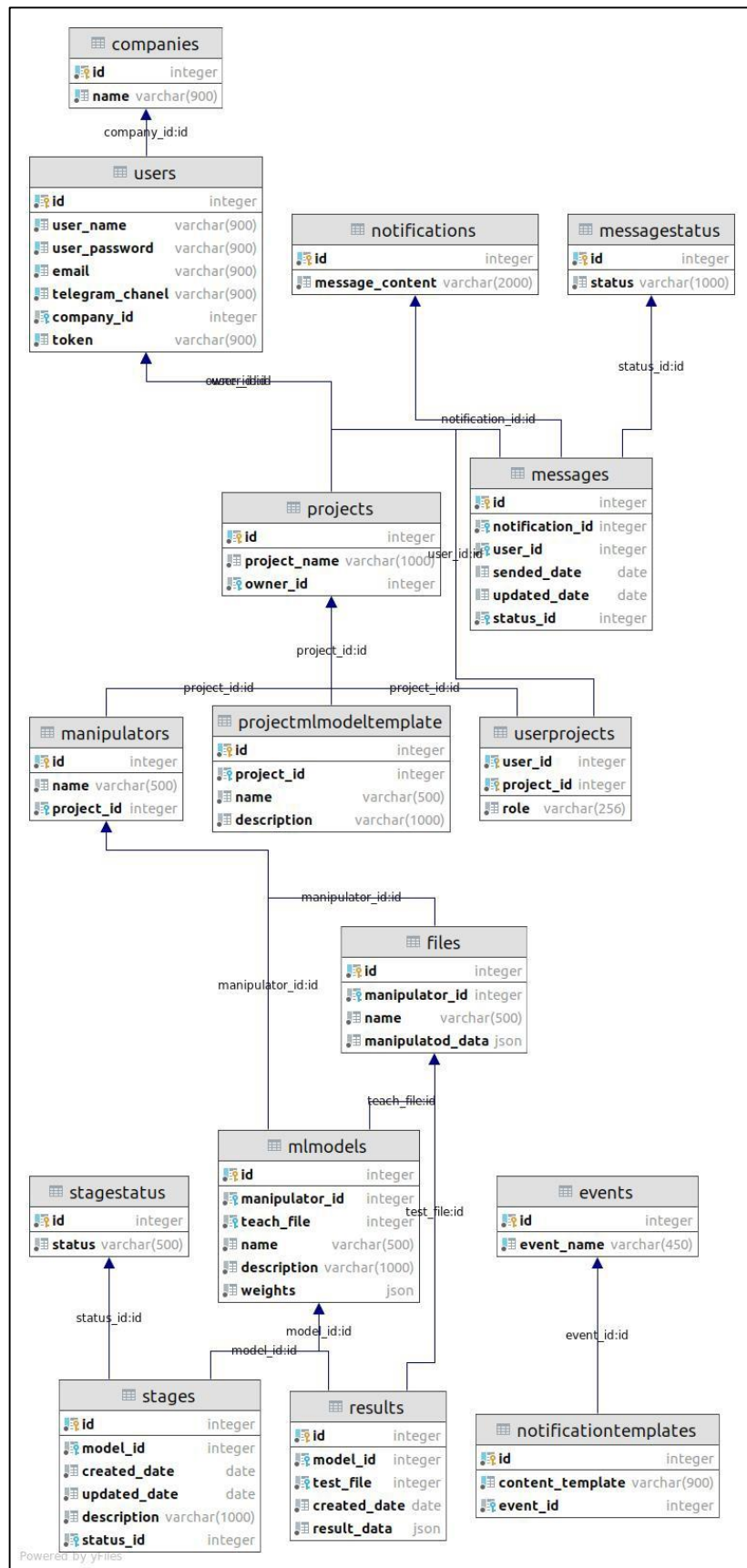


Рисунок 50 – Физическая модель базы данных