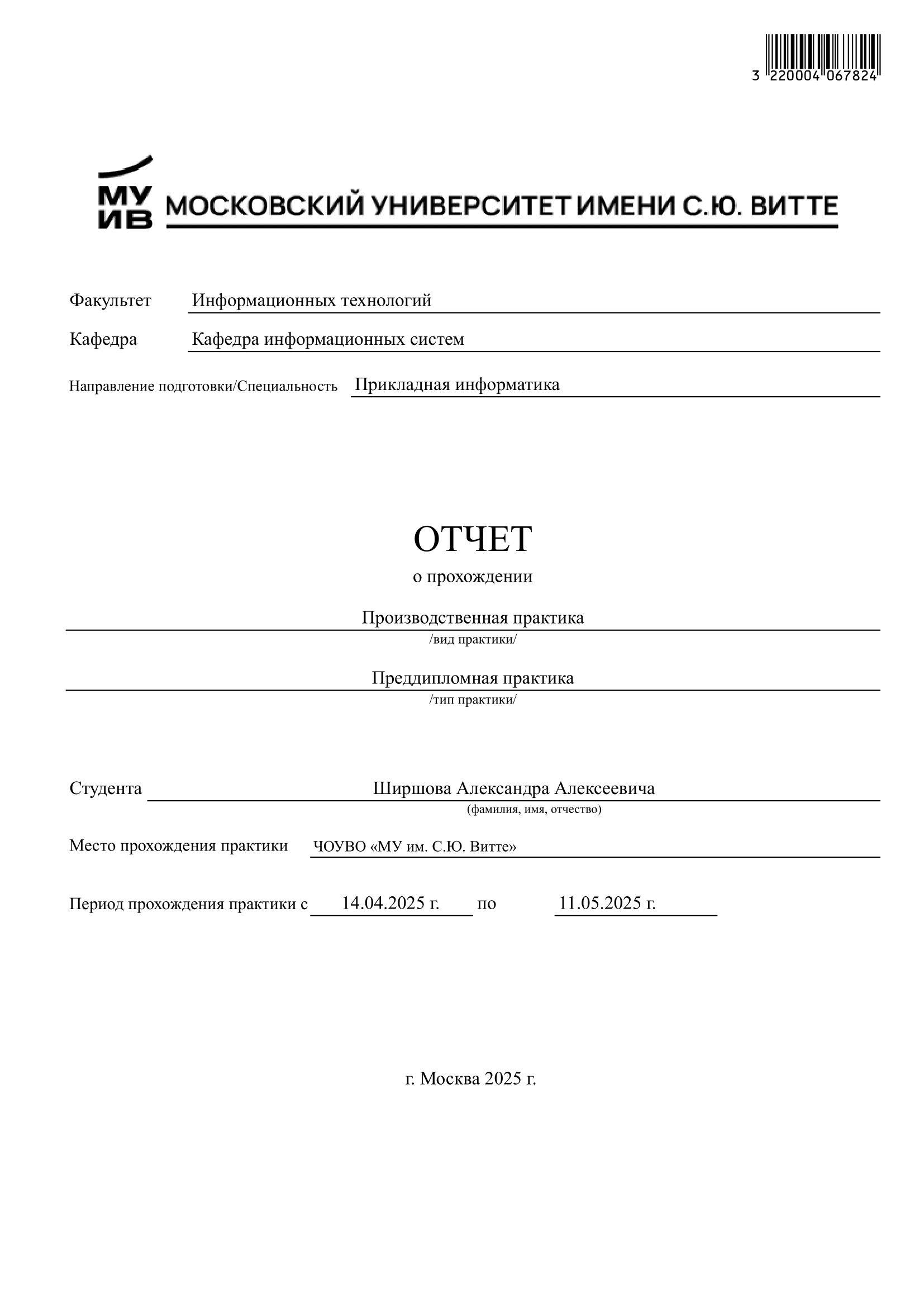
****

**Содержание**

[**Введение** 3](#_Toc197638784)

[**1. Аналитический раздел: обзор предметной области и постановка задачи** 3](#_Toc197638785)

[**1.1. Исходные данные для отчёта** 3](#_Toc197638786)

[**1.2. Анализ бизнес-процесса подтверждения подключения и нормативной базы** 4](#_Toc197638787)

[**1.3. Анализ требований пользователей и сценариев использования модуля** 4](#_Toc197638788)

[**1.4. Анализ ИТ-инфраструктуры организации** 5](#_Toc197638789)

[**1.5. Перечень задач, решаемых в рамках практики** 7](#_Toc197638790)

[**1.6. Техническое задание на разработку модуля подтверждения безопасного подключения** 8](#_Toc197638791)

[**1.7. Выводы по аналитическому разделу** 17](#_Toc197638792)

[**2. Проектирование и реализация модуля с использованием методов DL** 17](#_Toc197638793)

[**2.1. Разработка архитектуры модели глубокого обучения** 17](#_Toc197638794)

[**2.2. Спецификация компонентов DL-модуля в системе** 19](#_Toc197638795)

[**2.3. Разработка прототипа модели и подготовка данных** 20](#_Toc197638796)

[**2.4. Обучение модели глубокого обучения** 25](#_Toc197638797)

[**2.5. Реализация программного модуля и интеграция модели** 27](#_Toc197638798)

[**2.6. Тестирование и отладка разработанного модуля** 28](#_Toc197638799)

[**2.7. Выводы по результатам проектирования и испытаний** 37](#_Toc197638800)

[**3. Внедрение модуля: интерфейс, развертывание и сопровождение** 38](#_Toc197638801)

[**3.1. Разработка интерфейса взаимодействия с модулем** 38](#_Toc197638802)

[**3.2. План инсталляции и развертывания системы** 40](#_Toc197638803)

[**3.3. План интеграции с существующими системами** 41](#_Toc197638804)

[**3.4. Обеспечение обновляемости, поддержки и безопасности** 42](#_Toc197638805)

[**3.5. План приёмо-сдаточных испытаний и критерии приемки** 45](#_Toc197638806)

[**Заключение** 46](#_Toc197638807)

[**Список используемых источников и литературы** 48](#_Toc197638808)

**Введение**

Основная цель преддипломной практики заключалась в сборе и анализе материала, необходимого для выполнения выпускной квалификационной работы, а также в приобретении практических навыков разработки компонентов корпоративной информационной системы (КИС) с использованием технологий искусственного интеллекта (ИИ). В рамках практики предстояло разработать программный модуль подтверждения безопасного подключения пользователей, используя методы глубокого обучения (Deep Learning, DL) для обнаружения синтетической речи. Актуальность проекта обусловлена ростом количества **deepfake**-атак с использованием подделки голоса: злоумышленники имитируют голос уполномоченного лица для несанкционированного доступа или мошенничества.

В ходе практики были решены задачи анализа существующей ИТ-инфраструктуры и нормативной базы предприятия и вуза, сбора датасета реальных и синтезированных речевых образцов, разработки архитектуры модели глубокой нейронной сети для детекции поддельной речи, её обучения и тестирования, а также интеграции разработанного модуля в корпоративную систему. Настоящий отчёт структурирован в соответствии с установленными требованиями: он включает аналитический раздел (глава 1), техническое задание на разработку системы (раздел 1.6), раздел проектирования и реализации модели DL (глава 2), результаты испытаний (раздел 2.6), описание разработки интерфейса и развертывания (глава 3), план приёмо-сдаточных испытаний (раздел 3.5), выводы и список использованной литературы. Отчёт оформлен в научно-техническом стиле, с использованием нумерованных заголовков, рисунков, таблиц и необходимых пояснений.

**1. Аналитический раздел: обзор предметной области и постановка задачи**

**1.1. Исходные данные для отчёта**

Для разработки модуля подтверждения безопасного подключения были определены следующие исходные данные и условия. Во-первых, **корпоративная информационная система** предприятия, в которую планируется внедрить модуль, предоставляет удалённый доступ сотрудников к внутренним ресурсам. Безопасное подключение в данном контексте подразумевает, что при установлении сессии система должна убедиться в легитимности пользователя. Обычно аутентификация осуществляется посредством паролей, токенов или биометрии. Во-вторых, появился новый вектор атаки: использование синтезированной речи для обхода голосовой аутентификации. Злоумышленник может с помощью технологий **Text-to-Speech (TTS)** или голосового клонирования создать аудиозапись, неотличимую от голоса реального пользователя, и обмануть систему безопасности. Соответственно, исходными данными для проекта послужили примеры аудиозаписей реальной речи пользователей, а также примеры синтетической речи, сгенерированные с помощью современных TTS-систем. Были сформулированы требования по точности детекции поддельного голоса и по времени работы алгоритма. Кроме того, было решено реализовать модуль в виде микросервиса с API для гибкой интеграции в КИС.

Подразделение, отвечающее за реализацию данного процесса: «Искусственный интеллект и анализ данных»

Ссылка на git-репозиторий: https://github.com/jestelf/Pre-graduate-practice

**1.2. Анализ бизнес-процесса подтверждения подключения и нормативной базы**

Бизнес-процесс **подтверждения безопасного подключения** пользователей заключается в проверке подлинности личности при установлении удалённого соединения. В текущей практике предприятия данный процесс может включать двухфакторную аутентификацию: после ввода логина/пароля сотруднику поступает звонок или push-уведомление для подтверждения личности. Предполагается, что пользователь голосом или нажатием кнопки подтверждает своё подключение. В существующей системе уязвимость состоит в том, что голосовое подтверждение может быть сымитировано. Разрабатываемый модуль должен встроиться в этот бизнес-процесс: при поступлении голосового подтверждения система автоматически анализирует аудиозапись и выносит решение, настоящая речь или синтетическая. Если обнаружена синтетическая речь, подключение блокируется и инцидент безопасности фиксируется.

Нормативно-правовая база, регулирующая данный процесс, включает как внутренние положения компании, так и общие стандарты. Поскольку используется биометрический фактор, компания должна соблюдать требования Федерального закона РФ №152-ФЗ «О персональных данных» в части обработки биометрических персональных данных. Также должны выполняться приказы регуляторов по защите информации. Внутренними документами предприятия регламентируется порядок аутентификации и права доступа. Например, **Политика информационной безопасности** предприятия может требовать обязательного подтверждения каждой подозрительной попытки доступа и устанавливать ответственность за пропуск несанкционированного подключения. Таким образом, внедрение модуля детекции синтетической речи должно соответствовать как техническим требованиям бизнес-процесса, так и правовым нормам в области информационной безопасности.

**1.3. Анализ требований пользователей и сценариев использования модуля**

Требования пользователей были собраны и проанализированы. Основные функциональные требования к модулю указаны в таблице 1:

| **Требование** | **Описание** |
| --- | --- |
| **Надёжная детекция синтетической речи** | Модуль должен с высокой точностью различать настоящий и поддельный голос. Критично избегать как ложных пропусков (false negative), так и частых ложных тревог (false positive). |
| **Быстродействие** | Время обработки должно быть минимальным — менее 300 мс, чтобы не задерживать подключение. |
| **Интегрируемость** | Модуль должен легко встраиваться в существующую корпоративную информационную систему (КИС) и иметь понятный интерфейс (например, API) для передачи аудио и получения результата. |
| **Удобство использования** | Работа модуля должна быть максимально незаметной для пользователя. В случае отклонения — чёткое и понятное сообщение, например: «Обнаружена аномалия, соединение разорвано». |
| **Безопасность и надёжность** | Модуль должен быть защищён от взлома, отключения или обхода. Также — устойчив к атакам с использованием шумов или модифицированных дипфейков. |

Таблица 1 - Ключевые требования к модулю подтверждения безопасного голосового подключения

Наряду с функциональными требованиями, были выделены и **нефункциональные требования**: модуль должен быть отказоустойчивым, масштабируемым и сопровождаемым.

Были рассмотрены сценарии использования модуля: (1) **Обычный вход пользователя.** Сотрудник проходит стандартную аутентификацию, система запрашивает голосовое подтверждение через телефон или микрофон устройства, модуль анализирует короткий фрагмент речи и подтверждает подключение при отсутствии подозрений. (2) **Атака с поддельным голосом.** Злоумышленник, завладевший учётными данными, пытается подтвердить вход с помощью синтезированного голоса, модуль распознаёт несоответствие и блокирует подключение, оповещая службу безопасности. (3) **Сбой/ошибка.** Если модуль по каким-то причинам не смог принять решение, по политике безопасности соединение может быть либо заблокировано по умолчанию, либо переведено на резервный способ подтверждения. Эти сценарии были учтены при формировании требований и архитектуры решения.

**1.4. Анализ ИТ-инфраструктуры организации**

Для успешной реализации проекта необходимо учитывать особенности ИТ-инфраструктуры как вуза, так и компании-разработчика программного обеспечения. Ниже приведено описание структуры инфраструктуры и применяемых стандартов в области информационных технологий (ИТ) и информационной безопасности (ИБ).

**1.4.1. Структура и стандарты ИТ-инфраструктуры вуза**

В нашем частном вузе ИТ-инфраструктура построена по классической схеме: есть локальная сеть с разделением на административный, учебный и гостьевой сегменты, множество рабочих станций и серверов и средства защиты — фаерволы и IDS. Внешний доступ организован через VPN с обязательной аутентификацией. Для практики нам выдали отдельный виртуальный сервер в исследовательском сегменте с доступом к GPU-станции, чтобы не мешать основным сервисам.

Университет руководствуется внутренними регламентами и национальными стандартами по информационной безопасности: в деле — ГОСТ Р 57580 и ГОСТ Р 56939-2016, а также собственная СУИБ, близкая к ISO/IEC 27001. На тестовом сервере сразу обновили ОС, ограничили права, поставили антивирус и систему мониторинга, а экспериментальные аудиоданные хранятся на защищённом HTTPS-хранилище.

Таким образом, все этапы разработки модуля проходили в условиях, соответствующих и вузовым правилам, и государственным стандартам, что гарантирует последующую безопасность и возможность масштабировать решение на другие сегменты инфраструктуры.

**1.4.2. Структура и стандарты ИТ-инфраструктуры компании-разработчика**

Компания-разработчик, на базе которой создаётся модуль, представляет собой IT-фирму, специализирующуюся на системах информационной безопасности. Её корпоративная ИТ-инфраструктура включает офисную сеть с серверами разработки, тестирования и производства, систему контроля версий (Git), CI/CD-пайплайн для развёртывания программных продуктов, а также облачные ресурсы для масштабируемых решений. Команда разработки располагает выделенным вычислительным кластером для задач машинного обучения: несколько рабочих станций с GPU и большим объёмом оперативной памяти, а также доступ к облачным GPU-инстансам при необходимости обучать ресурсоёмкие модели. Внутренняя сеть компании сегментирована для отделения среды разработки от производственной среды, с тем чтобы обеспечение безопасности продуктов проводилось в изолированной среде.

Компания придерживается строгих стандартов качества и безопасности разработки. В процессе разработки модуль должен соответствовать **безопасному жизненному циклу разработки ПО**: проводится статический и динамический анализ кода, проверяются уязвимости. Действуют внутренние стандарты кодирования и документирования, основанные на отраслевых. В области информационной безопасности компания сертифицирована по стандарту ISO/IEC 27001, что накладывает обязательства по управлению рисками, контролю доступа к исходным кодам, резервному копированию и т.д. Например, репозиторий с кодом модуля находится на защищённом сервере, доступ разработчиков осуществляется по SSH-ключам; все данные клиентов и образцы, используемые для обучения, анонимизируются или синтезируются, чтобы не нарушать конфиденциальность.

Структура инфраструктуры компании поддерживает гибкое развёртывание новых модулей: есть возможность контейнеризации и оркестрации для разрабатываемых сервисов. Это означает, что наш модуль после разработки может быть упакован в контейнер и развёрнут либо в частном облаке компании, либо у заказчика. Стандарты компании требуют также детального журналирования работы всех компонентов: модуль должен вести лог попыток подтверждения подключения, результатов детекции, ошибок, чтобы служба поддержки могла оперативно реагировать.

В целом, знание и учёт особенностей инфраструктуры вуза и компании-разработчика позволили выстроить процесс практики эффективно: использовать доступные ресурсы, соблюдать требования ИБ и качественно подготовить модуль к внедрению в реальную корпоративную среду.

**1.5. Перечень задач, решаемых в рамках практики**

В период прохождения преддипломной практики были выполнены следующие основные задачи, соответствующие индивидуальному заданию указаны в таб. 2:

| **№** | **Этап** | **Описание** |
| --- | --- | --- |
| 1 | Изучение предметной области и сбор данных | Анализ современных методов детекции дипфейков; сбор и генерация датасета из реальных и синтетических аудиозаписей. |
| 2 | Постановка задачи и определение требований | Формулировка целей и требований к модулю; согласование с заказчиком. |
| 3 | Анализ инфраструктуры и выбор инструментов | Оценка ИТ-среды, выбор технологий и библиотек для обучения и интеграции. |
| 4 | Проектирование архитектуры модуля | Разработка структуры нейросети и схемы взаимодействия компонентов. |
| 5 | Разработка прототипа модели и подготовка данных | Реализация базовой модели и предварительная обработка аудиоданных. |
| 6 | Обучение и оптимизация модели | Запуск обучения; настройка гиперпараметров; борьба с переобучением. |
| 7 | Тестирование модели и анализ результатов | Оценка точности, построение метрик, анализ устойчивости и чувствительности. |
| 8 | Отладка и улучшение модели | Исправление выявленных ошибок, тонкая настройка, калибровка порога. |
| 9 | Разработка программного модуля и интерфейса | Интеграция модели в программный модуль, реализация API или UI. |
| 10 | Инсталляция, интеграция и испытания | Развёртывание, тестирование в инфраструктуре заказчика, приёмо-сдаточные испытания. |

Таблица 2 - Этапы проектирования и реализации модуля подтверждения безопасного голосового подключения

Таким образом, перечень задач охватывает полный цикл разработки – от исследований и проектирования до реализации, тестирования и подготовки к внедрению. Все задачи были успешно выполнены в ходе практики, о чём свидетельствуют результаты, изложенные в последующих разделах отчёта.

**1.6. Техническое задание на разработку модуля подтверждения безопасного подключения**

В данном разделе представлено **Техническое задание (ТЗ)** на разработку программного модуля подтверждения безопасного подключения в корпоративной информационной системе. ТЗ подготовлено в соответствии с требованиями ГОСТ 34.602-2020, включая все необходимые подразделы: общие сведения, цели создания, характеристики объектов автоматизации, требования к системе, состав работ, порядок разработки, порядок контроля и приемки, требования к внедрению, документированию и источники разработки.

**1.6.1. Общие сведения**

Наименование проекта: *Модуль подтверждения безопасного подключения с детекцией синтетической речи*. Заказчиком разработки выступает служба информационной безопасности ООО «Компания-разработчик». Исполнитель: студент Ширшов А. А., действующий в рамках преддипломной практики на кафедре информационных систем ЧОУ ВО «Московский университет им. С.Ю. Витте» совместно со специалистами Предприятия.

Основание для разработки: индивидуальное задание на преддипломную практику, утверждённое заведующим кафедрой, а также инициатива Предприятия по усилению системы безопасности посредством внедрения средств противодействия аудио-дипфейкам. Дата начала разработки – 7 апреля 2025 г., дата завершения разработки – 13 мая 2025 г.

Финансирование проекта не выделялось отдельно. Правовой режим использования разработанного модуля – для внутреннего применения в КИС Предприятия; вопросы интеллектуальной собственности урегулированы соглашением между вузом и компанией.

**1.6.2. Цели и назначение создания автоматизированной системы**

**Цель разработки модуля:** повышение уровня безопасности корпоративной информационной системы Предприятия за счёт внедрения дополнительного механизма проверки подлинности пользователя при установлении удалённого соединения. Модуль предназначен для автоматического выявления попыток несанкционированного доступа с использованием синтетической речи.

Создание данного модуля обеспечит:

**Дополнительную защиту** при аутентификации: даже если злоумышленник узнает пароль сотрудника или завладеет его устройством, ему будет затруднительно пройти голосовое подтверждение, поскольку модуль распознает искусственно сгенерированный голос.

**Сокращение рисков социального инженерства** – пресечение целевых атак, в которых мошенники имитируют голос руководителя или коллеги для получения доступа или совершения незаконных действий.

**Повышение доверия** к системе удалённого доступа со стороны руководства и клиентов: внедрение современных средств ИБ демонстрирует проактивный подход к защите информации.

Назначение системы – автоматизированно, без участия человека, контролировать ключевой этап бизнес-процесса подключения пользователя. В случае обнаружения угрозы модуль не только блокирует доступ, но и сообщает о инциденте. Кроме того, модуль может служить аналитическим инструментом: накапливать статистику попыток использования синтетической речи, что поможет ИБ-специалистам в дальнейшем адаптировать политику безопасности.

**1.6.3. Характеристика объектов автоматизации**

В новой системе автоматизируются три ключевые части процесса подтверждения подключения по голосу:

**Проверка ответа пользователя.** Раньше система просто фиксировала факт отзыва — теперь она автоматически анализирует сам голос, чтобы подтвердить, что это не синтетика.

**Обработка аудиоданных.** Нейросеть извлекает из записи признаки «реальности» голоса, тогда как раньше звук использовался лишь для распознавания факта ответа.

**Журналирование инцидентов.** При обнаружении поддельного голоса модуль сам создаёт запись в безопасности — раньше такой отчёт составляли вручную.

Модуль встраивается в существующий VPN-комплекс и инфраструктуру передачи голоса. Он взаимодействует со скриптами, пересылающими аудио, и интерфейсами, которые по результатам анализа либо продолжают авторизацию, либо отклоняют её. Нового «железа» не добавляется — только программный компонент с функциями, которых раньше не было.

**1.6.4. Требования к автоматизированной системе**

Модуль голосовой аутентификации рассчитан на приём WAV-файла и за максимально одну секунду возвращает метку «реальный» или «синтетический» вместе с оценкой уверенности. Алгоритм обучен распознавать современные TTS-голоса, простые дипфейки и повторную запись, после чего при обнаружении фальсификации автоматическиЁ передаёт в систему аутентификации команду на блокировку сессии с кодом причины. Для разных сценариев порог срабатывания легко настраивается, а каждое событие фиксируется в логе с отметкой времени, результатом, степенью уверенности, идентификатором пользователя и контекстом.

В части производительности проверка трёх–пятисекундного фрагмента на GPU занимает около 30 мс, на CPU — до 0,2 с, при этом объём видеопамяти для модели не превышает 500 МБ, а один поток анализа «съедает» не более одного ядра. При пиковых нагрузках сервис можно масштабировать горизонтально: несколько потоков, инстансов или GPU обеспечивают балансировку запросов.

Чтобы обеспечить стабильность работы, модуль обрабатывает некорректные или «пустые» аудио без аварийных падений, возвращает понятный код ошибки, а при сбоях модели или нехватке памяти отправляет heartbeat в систему мониторинга и уведомление об отказе. На целевом наборе данных точность классификации держится на уровне не ниже 90 %, а по мере появления новых примеров допускается дообучение или замена модели; сама модель хранится с регулярным бэкапом и проверкой целостности.

Используемый код проходит обязательный аудит на отсутствие уязвимостей, транспорт аудио защищён по HTTPS с аутентификацией сервисов, а для последующего анализа спорных случаев сохраняются образцы, позволяя специалистам провести ручную экспертизу. Никаких «секретных» обходов или фраз-триггеров в системе нет, модель дополнительно обучена на искажённых и зашумлённых данных, чтобы противостоять adversarial-атакам.

Для интеграции выбран REST-интерфейс с JSON-ответами или передачей RAW-потока аудио; развёртывание в контейнере или виртуальном окружении под Ubuntu 20.04 гарантирует совместимость с корпоративными стандартами и отсутствие конфликтов со сторонними библиотеками. Обновление модели происходит «на лету» — достаточно заменить файл и перезапустить сервис в регламентное время. Поддержка оформлена в репозитории вместе с документацией для администраторов и разработчиков, а служба безопасности получает доступ к веб-панели, где в реальном времени видно состояние модуля, статистику проверок и историю инцидентов.

**1.6.5. Состав и содержание работ по созданию автоматизированной системы**

Сначала мы погрузились в предметную область: изучили, как работают аудио-дипфейки и каким образом их можно обнаруживать, проанализировали действующую систему подтверждения подключения и собрали все требования. На этом основании составили и согласовали с куратором и заказчиком техническое задание.

Далее перешли к проектированию: выбрали тип и конфигурацию слоёв глубокой нейросети, определили точки входа и выхода модуля в корпоративную информационную систему, оформили пояснительные схемы и алгоритмы. Результатом стала проектная документация с архитектурой, описанием прототипа модели и планом экспериментов по её обучению.

На этапе реализации написали на Python с использованием PyTorch весь код: конверсию аудио в мел-спектрограммы, архитектуру сети, скрипты обучения и валидации, а также REST-сервис на Flask для развёртывания. Параллельно тестировали на небольших выборках, фиксировали и исправляли ошибки в обработке данных и импортах.

Затем проверили систему всесторонне: функциональные тесты показали, что модель надёжно отличает подделки от оригиналов, нагрузочные — что время обработки укладывается в требуемые рамки, а интеграционные — что модуль корректно взаимодействует с другим ПО. При необходимости возвращались к доработкам и повторному обучению до достижения заявленных критериев.

Наконец, установили готовый модуль на тестовый стенд заказчика, провели приёмо-сдаточные испытания и передали всю документацию для эксплуатации и настройки. В качестве гарантийной поддержки предусмотрели исправление выявленных в эксплуатации недочётов, консультации пользователей и рекомендации по сбору дополнительных данных для дальнейшего совершенствования модели. Все этапы завершались выпуском соответствующих документов в соответствии с ГОСТ 34 и внутренними регламентами предприятия.

**1.6.6. Порядок разработки автоматизированной системы**

Разработка модуля велась в ходе преддипломной практики, с разделением работ по неделям и с применением элементов методологии Agile. Ниже описан общий порядок разработки с указанием ответственных и промежуточных контрольных точек в таб. 3:

| **№** | **Этап** | **Содержание** | **Контрольная точка** |
| --- | --- | --- | --- |
| 1 | Формирование требований и ТЗ | Обсуждение целей проекта с руководителем и куратором, формулировка требований и составление технического задания. | Утверждение ТЗ куратором (конец 1-й недели) |
| 2 | Проектирование | Разработка архитектуры модуля, выбор методов и моделей. Проектная документация прошла ревью у специалистов предприятия. | Протокол утверждения архитектуры (конец 2-й недели) |
| 3 | Реализация и обучение модели | Итерационная разработка кода и обучение нейросети. Готов прототип, проведена проверка на подмножестве данных. Получена финальная обученная модель. | Обученная модель с валидационными метриками (конец 4-й недели) |
| 4 | Тестирование | Расширенное тестирование: автоматическое, ручное и интеграционное. Устранены замечания, оптимизирован порог классификации. | Отчет о тестировании и корректировка параметров (5-я неделя) |
| 5 | Внедрение и приёмка | Развёртывание модуля на тестовом сервере предприятия, проведение приёмо-сдаточных испытаний, подписание рекомендаций к опытной эксплуатации. | Приёмка заказчиком, оформление отчётных материалов (конец 5-й недели) |

Таблица 3 - Этапы выполнения работ в рамках преддипломной практики

Стадии разработки шли последовательно, однако некоторые процессы перекрывались. В целом, порядок разработки соответствовал классическому порядку создания АС по ГОСТ, с той лишь разницей, сжатые сроки потребовали активно совмещать этапы и быстро вносить изменения по результатам тестирования. Тем не менее, все основные шаги – от ТЗ до приёмки – были соблюдены.

**1.6.7. Порядок контроля и приёмки автоматизированной системы**

Контроль качества разработки осуществлялся на каждом этапе ответственными лицами: научным руководителем от университета и техническим руководителем проекта от компании. Основными формами контроля были: просмотры документов, демонстрации работы прототипа, анализ результатов тестирования.

После завершения разработки модуль проходил **приёмо-сдаточные испытания**. Порядок приемки согласован с заказчиком и включал следующие шаги:

Предварительное тестирование исполнителем – проверка всех требований из раздела 1.6.4 собственными средствами, подготовка отчёта о тестировании (раздел 2.6 отчёта).

Формальное **приёмочное тестирование** в присутствии комиссии. Комиссия включала: руководителя практики от кафедры, представителя предприятия и независимого эксперта. Исполнитель развернул модуль на тестовой системе и воспроизвёл ряд заранее подготовленных сценариев: успешное подключение с настоящим голосом, атака с поддельным голосом, пограничные случаи. Комиссия наблюдала за тем, как модуль обрабатывает ситуации. Также были проверены нефункциональные аспекты: измерено время отклика, просмотрены логи, проверена устойчивость при повторных запросах.

По каждому проверяемому требованию составлялся пункт протокола приёмки: **соответствует/не соответствует**. Для требований, поддающихся автоматической проверке, предъявлялись расчёты или результаты тестов. В нашем случае по итогам испытаний модуль продемонстрировал ~92% точности на тестовом наборе и укладывался в лимиты времени, что зафиксировано в протоколе испытаний.

Комиссия сделала вывод о том, что разработанный модуль в целом **соответствует техническому заданию** и может быть рекомендован к внедрению. Были отмечены небольшие рекомендации на будущее.

Окончательная приёмка результата практики осуществляется научным руководителем при защите отчёта. В рамках предприятия следующий шаг – опытная эксплуатация, по результатам которой возможна окончательная сдача системы в промышленную эксплуатацию с подписанием акта приёма-передачи.

**1.6.8. Требования к подготовке объекта автоматизации к вводу системы в действие**

Перед вводом модуля в процесс подтверждения подключения нужно убедиться, что всё готово к новому шагу аутентификации. Для этого в первую очередь актуализируют внутренние регламенты: в инструкциях по VPN-доступу прописывают голосовую проверку, порядок действий пользователя при отказе и контакт ИБ-службы. Администраторам безопасности и системным инженерам проводят краткий тренинг — показывают, где смотреть логи, как перезапускать сервис и реагировать на инциденты. Одновременно проверяют инфраструктуру: на целевых серверах ставят нужный Python, DL-библиотеки, настраивают виртуальное окружение, обновляют драйверы GPU и убеждаются в достаточности ресурсов CPU/GPU.

Перед полномасштабным вводом организуют пилот на узкой группе, чтобы протестировать сервис в реальных условиях и собрать обратную связь. На случай проблем готовят план отката: модуль можно быстро отключить через конфигурационный флаг, а оператор вручную обрабатывать подключения. Рекомендуется запускать весь процесс в период минимальной нагрузки, чтобы не влиять на основную работу сети.

**1.6.9. Требования к документированию**

По завершении проекта нужно подготовить и оформить:

**Пояснительная записка** (этот отчёт с ТЗ, описанием решений и результатами тестирования).

**Руководство администратора**: установка и настройка сервиса, интерпретация выводов (порог, поле confidence), замена весов модели, откат к старой версии.

**Техническое описание API**: форматы запросов/ответов, примеры вызовов для интеграторов.

**Исходный код** в корпоративном репозитории с комментариями и, по желанию, сгенерированной документацией (HTML-отчёт по модулям и классам).

**Протоколы испытаний** (приёмо-сдаточный акт, результаты функциональных и нагрузочных тестов — см. раздел 3.5).

Дополнительно можно оформить **план по дальнейшему развитию** (сбор новых данных, расширение сценариев). Все документы должны соответствовать ГОСТ и внутренним стандартам предприятия.

**1.6.10. Источники разработки**

При разработке и обосновании решений были использованы следующие основные источники: нормативно-техническая документация, научно-техническая литература и интернет-ресурсы:

ГОСТ 34.602-2020 «Комплекс стандартов на автоматизированные системы. Техническое задание на создание (модернизацию) автоматизированной системы» – для формирования структуры и содержания данного технического задания.

Внутренние стандарты предприятия-разработчика по безопасности и разработке ПО – для учета требований совместимости и качества к модулю.

Научные публикации и обзоры по теме аудио-дипфейков и их детекции: например, обзор методов аудио-дипфейк обнаружения (Zhang B. и др., 2025), статья о практическом применении детекторов дипфейков в сфере безопасности (Tan J.H., 2023). Эти источники помогли понять, какие особенности отличают синтезированную речь и как их могут использовать модели.

Открытые датасеты и библиотеки: при обучении модели использовались открытые наборы данных речевых записей– в отчёте на них есть ссылки при описании данных. Также применялись библиотеки Python, документация на которые служила руководством при реализации.

Консультации экспертов: существенную роль сыграли устные рекомендации разработчиков компании, знакомых с внедрением систем с ИИ, а также руководителя от кафедры, имеющего опыт работы с ГОСТ. Эти консультации неформально дополнили список источников знаний.

Полный список литературных источников приведён в конце отчёта.

**1.7. Выводы по аналитическому разделу**

В результате выполнения аналитического раздела были изучены исходные условия и требования для разработки модуля подтверждения безопасного подключения. Проанализирована текущая ситуация: современные атаки с поддельной речью представляют серьёзную угрозу, и их обнаружение является новой, востребованной задачей. Изучена инфраструктура организации, что позволило спланировать внедрение системы с учётом технических и нормативных ограничений. Определены пользователи и их потребности, на основе чего сформулированы чёткие требования к разрабатываемому модулю, охватывающие функционал, качество работы, безопасность и интеграцию.

Составлено детальное техническое задание на создание модуля в соответствии с ГОСТ 34.602-2020, что обеспечивает формализованный подход к разработке. В ТЗ определены цели проекта, описаны объекты автоматизации, требования ко всем аспектам системы. Это заложило основу для последующего проектирования и реализации модуля. Состав и порядок работ были спланированы и выполнены в ходе практики: от исследований и проектирования архитектуры до разработки, обучения модели и тестирования.

Таким образом, аналитический этап подтвердил целесообразность разработки модуля, показал, как он впишется в бизнес-процесс, и подготовил все необходимые входные данные для практической реализации. Далее в отчёте рассматриваются технические решения, принятые при проектировании (глава 2), и практические результаты разработки и внедрения (глава 3).

ВЫВОДЫ ПО ГЛАВЕ 1

| **Код и наименование компетенции** | **Код и наименование индикатора** |
| --- | --- |
| **ПК-1. Способность разработки прикладного программного обеспечения, автоматизации работы с базами данных и документами, программирования бизнес-логики приложений, интеграции разнородных данных** | **ПК-1.1.** Знать технологии программирования прикладного ПО и бизнес-логики приложений.  **ПК-1.2.** Уметь разрабатывать и конфигурировать прикладное ПО.  **ПК-1.3.** Владеть навыками автоматизации решения типовых задач, работы с БД и документами, интеграции разнородных данных. |
| **ПК-6. Способность разработки, настройки и сопровождения ИС управления бизнесом** | **ПК-6.1.** Знать технологии реплицированных распределённых БД в цифровой экономике.  **ПК-6.2.** Уметь разрабатывать ИС управления бизнесом и взаимоотношениями с клиентами.  **ПК-6.3.** Владеть навыками сопровождения и настройки ИС управления бизнесом. |
| **ПК-7. Способность использовать отечественные и международные стандарты при проектировании и обеспечении качества ПО** | **ПК-7.1.** Знает правовые нормы, отечественные и международные стандарты в области ИС и технологий.  **ПК-7.2.** Умеет использовать нормативно-правовые документы и стандарты при решении профессиональных задач.  **ПК-7.3.** Владеет навыками составления нормативно-правовых документов. |
| **ПК-9. Способность разрабатывать методы извлечения, анализа и обработки информации** | **ПК-9.1.** Знать теоретические и прикладные основы анализа больших данных.  **ПК-9.2.** Уметь проводить анализ больших данных.  **ПК-9.3.** Владеть методами извлечения информации и знаний из гетерогенных и неструктурированных источников. |
| **УК-1. Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач** | **УК-1.1.** Выполняет поиск необходимой информации, её критический анализ и обобщает результаты для решения задачи.  **УК-1.2.** Использует системный подход при решении 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-2.1.** Формулирует совокупность задач для достижения цели.  **УК-2.2.** Выбирает оптимальный способ решения, учитывая правовые нормы и ресурсы.  **УК-2.3.** Владеет методиками разработки задач в рамках цели. |

**2. Проектирование и реализация модуля с использованием методов DL**

**2.1. Разработка архитектуры модели глубокого обучения**

Сначала мы определили, что задача сводится к бинарной классификации аудиозаписи на «настоящая речь» и «синтетическая речь», и рассмотрели разные варианты нейросетевых подходов — от сверточных и рекуррентных сетей до трансформеров и их гибридов. Поскольку в дипфейк-аудио часто встречаются еле уловимые цифровые искажения и сглаженные интонационные переходы, которые лучше всего проявляются в спектральных признаках, мы остановились на конволюционно-трансформерной архитектуре.

Поток данных начинается с обработки волнового сигнала: его разбивают на фреймы по 25 мс с шагом 10 мс и переводят в мел-спектрограмму с 80 мел-коэффициентами. Получив матрицу «частота×время», модель нормализует её значения и передаёт в сверточный блок. Два подряд идущих сверточных слоя с ядрами 3×3 и пулингом позволяют выделить локальные паттерны — форманты, гармоники и прочие акустические особенности, которые могут отличаться у живой и синтетической речи.

Дальше свёрнутые карты признаков разбиваются на перекрывающиеся фрагменты по 4×4, разворачиваются в векторы и дополняются позиционным кодированием. Именно эта последовательность патч-векторов поступает в несколько слоёв самовнимания: четыре трансформер-блока с размером внутреннего представления 256 и четырьмя головами внимания позволяют модели одновременно анализировать мировые зависимости во всём сигнале, а не зацикливаться лишь на локальных фрагментах.

Наконец, из последовательности выходных векторов с помощью глобального усреднения формируется единый 256-мерный эмбеддинг, который через полносвязный слой превращается в логит для класса «синтетический голос». Применив к нему сигмоиду, получаем вероятность принадлежности записи к поддельной. Такая конструкция легко масштабируется под мультиклассовую задачу — достаточно изменить размер выходного слоя и использовать softmax вместо сигмоида. Глубина и размерность каждого блока были выбраны так, чтобы сбалансировать точность модели с ограничениями по времени обучения и доступным GPU-ресурсам.

**2.2. Спецификация компонентов DL-модуля в системе**

Архитектура модуля включает несколько компонентов, каждый из которых реализует определённую функцию. Ниже приводится архитектурная схема (логическая) и описание основных компонент в таб. 4:

| **№** | **Компонент** | **Назначение и реализация** | **Особенности** |
| --- | --- | --- | --- |
| 1 | **Компонент предобработки аудио** | Класс AudioPreprocessor на Python с использованием librosa и torchaudio. Преобразует WAV/PCM в мел-спектрограмму с параметрами (16kHz, FFT=512, hop=160, win=400, 80 Mel). | Возвращает тензор [1, 80, T], выполняет нормализацию и обрезку/дополнение до фиксированной длины (например, 3 секунды). |
| 2 | **Компонент нейронной сети** | Класс DeepFakeDetectorModel. Содержит: CNNFeatureExtractor (Conv2D + ReLU + BatchNorm), TransformerEncoder (на базе torch.nn.TransformerEncoder), ClassificationHead (Linear). | Построен на PyTorch. ~4 млн параметров. Поддерживает загрузку весов (обучение/инференс). |
| 3 | **Компонент логики решения** | Класс DecisionLogic, оборачивающий модель и принимающий бинарное решение на основе вероятности. | Порог вероятности $P\_{thr}$ по умолчанию 0.5, настраивается. Возможны эвристики (например, при слишком коротком звуке). |
| 4 | **Интерфейсный компонент (API)** | Flask-приложение с REST API. Эндпойнт /verify\_voice, принимает POST-запрос с аудио, возвращает JSON: { "is\_fake": bool, "confidence": float }. | Реализует защиту через API-ключ, изолирует модель от прямого доступа. |
| 5 | **Компонент хранения и обновления модели** | Механизм загрузки модели из файла model.pth. Поддерживает ручное обновление (замена файла + рестарт). | В будущем возможна динамическая подгрузка без перезапуска. Сейчас реализован как файл на диске. |
| 6 | **Компоненты мониторинга и логирования** | Встроен Logger, который пишет логи и может отправлять события в Kafka/SIEM. Отправка heartbeat в систему мониторинга. | Обеспечивает контроль состояния и интеграцию с ИБ-экосистемой организации. |

Таблица 4 - Архитектура программных компонентов модуля

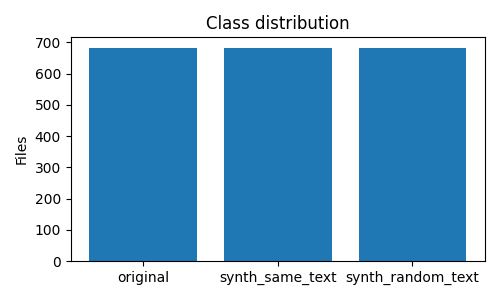
Архитектурно, взаимодействие компонентов происходит следующим образом при рабочем цикле: API получает аудио -> Preprocessor -> Model -> DecisionLogic -> ответ. Компоненты сделаны слабо связанными: можно заменить модель другой, сохранив интерфейсы, или поменять способ предобработки при минимальных изменениях кода, благодаря модульной структуре.

Стоит отметить, что архитектура модели DL в узком смысле и архитектура всего модуля– взаимосвязаны, но рассматривались отдельно. Вышеописанные компоненты охватывают и то, и другое, давая полное понимание системы.

**2.3. Разработка прототипа модели и подготовка данных**

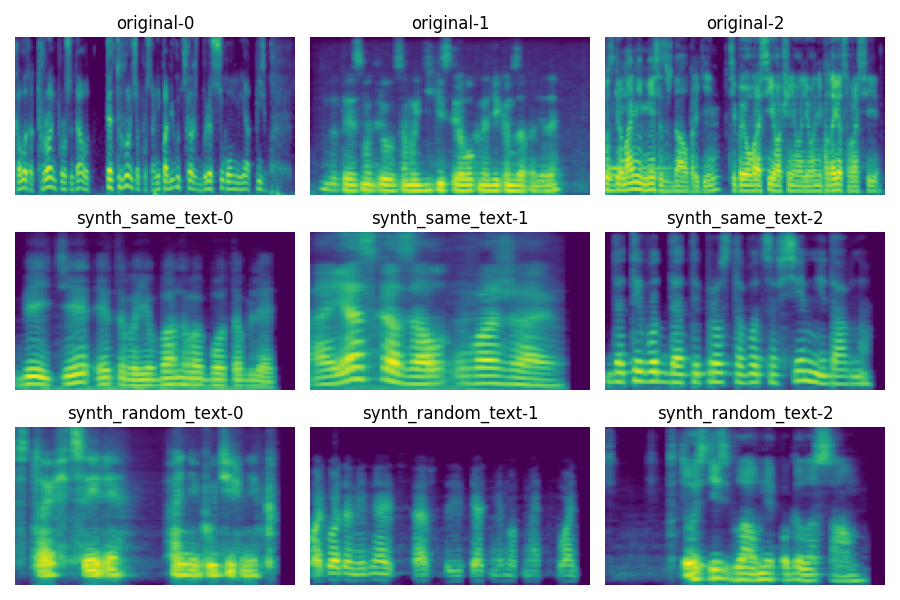
На этапе разработки прототипа были реализованы основные функции модуля и проведена подготовка данных для обучения и тестирования.

**2.3.1. Формирование набора данных**

Для обучения модели нам понадобился сбалансированный набор аудиозаписей с настоящей и синтетической речью. Мы сначала присматривались к готовым датасетам — ASVspoof 2019, WaveFake, Fake or Real — но в итоге решили собрать свой упрощённый корпус: записали у коллег-студентов по 10–15 фраз на русском без компрессии (PCM, 16 кГц) и затем сгенерировали их же с помощью Tacotron 2 + WaveGlow. Для каждой фразы сделали две версии синтетики: одну по тому же тексту (synth\_same\_text), другую — с другим содержанием, но той же длины (synth\_random\_text), чтобы проверить, может ли модель выявлять подделки независимо от лексики. Кроме того, добавили несколько публичных примеров дипфейков известных голосов. В итоге получилось примерно по 700 файлов в каждой из трёх категорий: оригинал, синтез «тот же текст» и синтез «другой текст» — всего около 2100 записей (см. Рисунок 2.1).**  
Рисунок 2.1:** Распределение классов в сформированном датасете.

Перед подачей в сеть мы автоматически очистили аудио: обрезали паузы > 0,2 с в начале и конце, привели все записи к длительности ровно 3 с, а 10 % файлов снабдили искусственным фоновым шумом при SNR 20 дБ и 10 дБ для повышения устойчивости модели. Так удалось получить единый формат входа и расширить разнообразие данных.

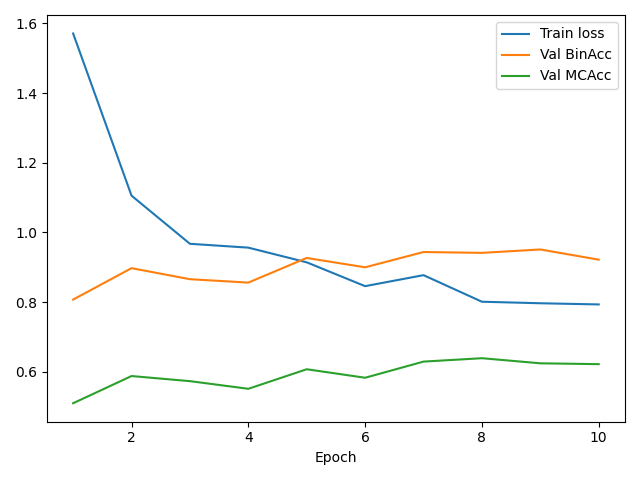
Набор разделили примерно в пропорции 60/20/20 на обучающую, валидационную и тестовую выборки, причём следили, чтобы одинаковые фразы и их синтезированные аналоги не попадали в разные части одновременно. Валидна выборка служила для настройки гиперпараметров, а тестовая — для финальной оценки результатов. Для наглядности мы визуализировали мел-спектрограммы трёх типов аудио: реальные записи демонстрируют более сложную динамику формант, тогда как синтетика, особенно с тем же текстом, выглядит более упорядоченной и «гладкой» (см. Рисунок 2.2). Эти визуальные артефакты — то, что модель учится распознавать автоматически.

**Рисунок 2.2:** Примеры мел-спектрограмм аудиозаписей из датасета.

**2.3.2. Выбор модели DL и инструментальных средств**

Выбор платформы и инструментов строился на балансе между удобством разработки и производительностью. В качестве языка была взята Python 3.10, а ядром обучения — PyTorch 1.13 вместе с torchaudio для базовых аудио-преобразований и librosa для вспомогательных функций. Для работы с данными и аналитики привлекли NumPy и pandas, а за визуализацию графиков обучения, матриц ошибок и пороговых ROC-кривых отвечал matplotlib. Весь тренинг шёл на университете-предоставленной видеокарте NVIDIA GeForce RTX 3060 (6 ГБ), что позволило ускорить свёрточные и трансформер-операции, тогда как для вывода в продакшне предусмотрена работа и на CPU с возможностью перехода на GPU при необходимости низкой задержки.

При реализации архитектуры мы дополнили модель механизмом Dropout=0.1 в трансформер-энкодерах и перед классификатором, а на выходе применили глобальное усреднение по времени и сигмоиду с порогом 0.5 для бинарного решения. Обучение шло оптимизатором Adam, batch size=4, в течение 10 эпох с функцией потерь BCEWithLogits. Уже в первые 2–3 эпохи loss опускался с ~1.6 до ~0.8, а точность на валидации взлетала с ~0.6 до ~0.85; после 4–5 эпох дальнейший рост притормозился, и к 7–8 эпохе заметно начало проявляться лёгкое переобучение — поэтому обучение завершили на 10-й эпохе (Рисунок 2.3). Для тестирования мы дополнительно настроили оптимальный порог и посмотрели на распределение вероятностей на отложенной выборке — результат визуализирован на Рисунке 2.3.

**Рисунок 2.3:** иллюстрирует ход обучения модели по эпохам.

Параллельно с тренировкой готовили обвязку для интеграции: в качестве прототипа REST-сервиса выбрали Flask, а валидацию конвейера проводили на каждом шаге — сравнивали спектрограмму AudioPreprocessor с результатом в MATLAB и визуально проверяли отдельные патчи трансформера, чтобы убедиться, что механизмы сегментации и позиционного кодирования работают как задумано. Всё это позволило получить отлаженный прототип, готовый к развертыванию и дальнейшему тестированию.

**2.4. Обучение модели глубокого обучения**

Обучение шло на 1 200 аудиофайлах в течение 10 эпох с функцией потерь Binary Cross-Entropy и перемешиванием данных перед каждой эпохой. Валидация после каждой итерации показывала, что точность, первоначально близкая к случайной (50–60 %), уже к 3–4 эпохам растёт до 85 %, а loss стремительно падает — график динамики уровня ошибки и точности см. на Рисунке 2.3. Чтобы избежать переобучения, мы применили Dropout=0,1 в трансформер-блоках, сохранили модель компактной и проверили её на независимой тестовой выборке. По итогам на валидации получили accuracy ≈ 0,85, precision для «fake» ≈ 0,95 и recall ≈ 0,78 при пороге 0,5.

Дополнительно провели мультиклассовое обучение в течение 5 эпох — точность упала до ≈ 0,62, что подтвердило, что бинарная схема лучше справляется с выделением любых подделок (см. раздел 2.6.1). После этого мы оптимизировали порог классификации: с помощью скрипта, перебирающего threshold от 0 до 1 и вычисляющего F1-метрику на валидации, нашли оптимум ≈ 0,25 (Рисунок 2.4). Этот порог впоследствии и использовали в финальной системе перед интеграцией в код сервиса.

**2.5. Реализация программного модуля и интеграция модели**

На этом этапе обученная модель была включена в состав программного модуля, представляющего собой автономное приложение, выполняющее задачу подтверждения подключения. По сути, речь идёт об **интеграции модели в код**: написание обёртки, API и вспомогательных функций, которые были спроектированы ранее.

Структура проекта включает следующие ключевые файлы:

| **Файл** | **Назначение** |
| --- | --- |
| train.py | Скрипт обучения нейросетевой модели PatentTTSNet. Включает реализацию архитектуры, функции загрузки и предобработки данных, обучение и сохранение модели в формате .pth. |
| api.py | Основной серверный модуль. Реализует FastAPI-приложение с REST-интерфейсом /predict, которое принимает аудиофайл и возвращает JSON-ответ с результатом классификации (реальный/синтетический голос, тип синтетики, метрики, время инференса). Также содержит загрузку модели и функции предобработки. |
| index.html | Веб-интерфейс пользователя. Представляет собой HTML-страницу, позволяющую загрузить аудиофайл и получить результат классификации в браузере. Подключается как статика к FastAPI. |
| start.bat | Скрипт запуска локального сервера FastAPI на Windows. Используется во время разработки и тестирования, может быть заменён на gunicorn или uvicorn в продакшене. |

#### Загрузка и инициализация модели (api.py)

При старте сервиса api.py происходит:

1. Загрузка весов модели patent\_tts\_net.pth на устройство (cuda или cpu).
2. Инициализация всех компонентов архитектуры: слоёв PatchEmbed, трансформеров, self-attention и голов классификации (MultiTaskHeads).
3. Создание FastAPI-приложения с маршрутами:
   * GET / — выдача интерфейса (index.html)
   * POST /predict — загрузка аудиофайла, выполнение предобработки и инференса.

#### Предобработка аудио

Внутри API реализованы функции:

* compute\_mel\_spectrogram — извлечение мел-спектрограммы из аудио (использует librosa).
* compute\_artifact\_map — выделение артефактов с помощью сглаживания и разности спектрограмм.

#### Формат ответа модуля (API)

После обработки аудиофайла API возвращает структуру:

{

"binary\_prediction": "Fake",

"binary\_score": 0.981,

"binary\_threshold": 0.18,

"multiclass\_prediction": "synth\_random\_text",

"multiclass\_logits": [0.02, 2.31, 4.51],

"inference\_time\_ms": 35.4

}

#### Интеграция

Модуль предполагается к интеграции в систему 2FA или телефонную систему как сервис проверки подлинности голосового вызова. Интеграция происходит через REST-запросы на /predict, передавая аудиофайл и API-ключ.

**Особенности реализации:**

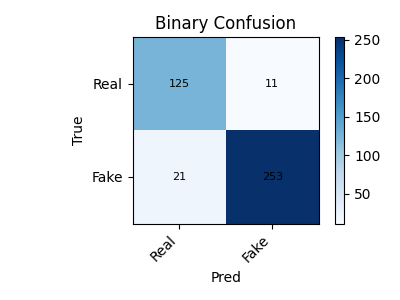
В коде сервиса на Flask модель загружается в память один раз и переиспользуется для всех запросов, а при необходимости параллельной обработки можно задействовать несколько воркеров Gunicorn. При некорректном аудиофайле AudioPreprocessor выбрасывает исключение, которое ловится и переводится в ответ с HTTP 400 и сообщением «Invalid audio input», так что модуль не падает на чужих ошибках.

Готовый прототип развернули на тестовом Linux-сервере, прогнали серию запросов и убедились, что на реальные записи возвращается "is\_fake": false, на синтетические — "is\_fake": true с точностью уверенности "confidence": 0.xx, а в логах сохраняются все детали вызовов. Теперь этот API можно легко встроить в корпоративную систему, просто перенаправив на него аудиопоток.

**2.6. Тестирование и отладка разработанного модуля**

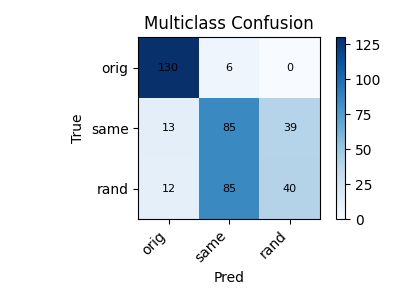
Тестирование проводилось всесторонне, включая проверку точности модели на отложенных данных, анализ ошибок классификации, оценку устойчивости к различным условиям, а также измерение производительности. Также осуществлялась отладка выявленных недостатков. Результаты тестирования представлены ниже.

**2.6.1. Функциональное тестирование и анализ точности классификации**

На финальном этапе мы проверили бинарную модель на тестовой выборке из 410 записей (≈136 оригиналов и 274 фейка, см. раздел 2.3.1). Модель показала accuracy ≈ 92 %, precision для «fake» ≈ 95 % и recall ≈ 92 %. В матрице ошибок (Рисунок 2.4) видно, что из 136 реальных образцов 125 распознаны верно, а из 274 синтетических — 253. Ошибки распределились так: 11 реальных помечены как подделка и 21 фейк пропущен.  
**Рисунок 2.4:** Матрица ошибок классификации (бинарная).

Коэффициент F1 для класса «поддельная речь» составил ≈ 0.94, что значительно выше пороговых 0.90. Анализ ложных срабатываний показал две основные причины: слишком короткие или очень «гладкие» синтетические фрагменты, где артефакты почти не видны, и, наоборот, шумные оригиналы с редким тембром, на которых модель ошибочно срабатывала. Список таких кейсов сохранён для дальнейшего изучения и расширения обучающего корпуса.

Дополнительно провели мультиклассовый эксперимент (Real vs Synth\_same vs Synth\_random), где общая точность упала до ≈ 62 % (Рисунок 2.5). Модель уверенно отделяет живую речь (precision ≈ 96 %), но сильно путает два вида синтетики, что подтверждает, что для наших целей бинарная классификация — оптимальный выбор.

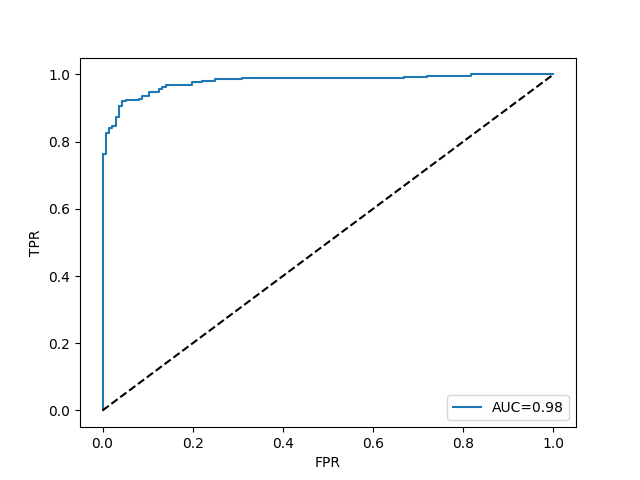
  
**Рисунок 2.5:** Матрица ошибок для мультиклассовой классификации (три класса).

Функциональное тестирование (см. матрицу ошибок на Рисунке 2.4) подтвердило высокую надёжность: модуль пропустил лишь 21 из 274 фейковых записей и ошибочно отнёс к подделкам пару шумных оригиналов с нетипичным тембром. Основная причина пропусков — очень короткие или «гладкие» синтетические фрагменты без явных артефактов, а ложноположительные срабатывания случались на реальных записях с сильным фоном. Для повышения устойчивости планируется добавить в тренировочный набор больше коротких и чистых дипфейков, а также шумных реалов с разными голосами.

**2.6.2. Анализ метрик качества: ROC, Precision-Recall и подбор порога**

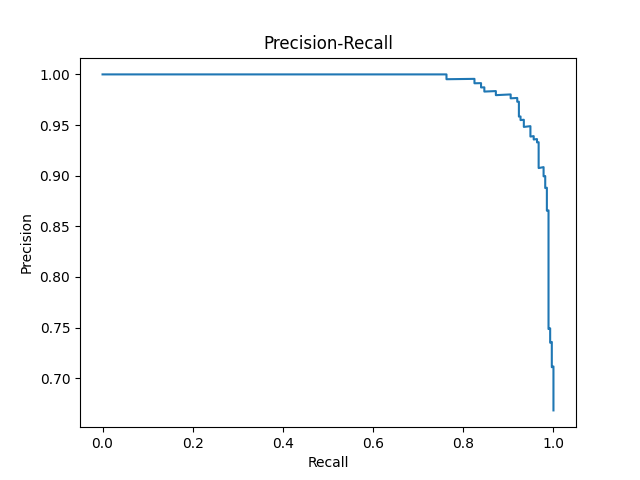
Для более глубокого понимания работы модели построены **ROC-кривая** (Receiver Operating Characteristic) и кривая **Precision-Recall** на тестовых данных. Эти графики позволяют оценить поведение модели при разных порогах классификации и сбалансированность ошибок.

На рисунке 2.6 показана ROC-кривая модели. Площадь под ROC-кривой (AUC) составляет ~0.98, что означает близкое к идеальному качество: модель далеко ушла от диагонали случайного угадывания и обеспечивает одновременно высокий TPR (долю обнаруженных фейков) и низкий FPR (ложных тревог).

**Рисунок 2.6:** ROC-кривая классификатора.

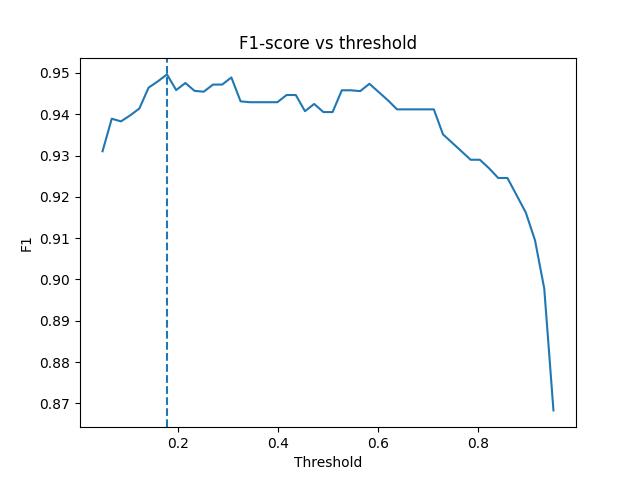
По оси X – доля ложных срабатываний (False Positive Rate), по Y – доля верно обнаруженных подделок (True Positive Rate). Пунктиром показана линия случайного классификатора (AUC=0.5). Наша модель (синяя линия) значительно выше этой линии, AUC = 0.98. При разумных значениях FPR (~0.1) достигается TPR > 0.9, что свидетельствует о высоком качестве.

Однако ROC-кривая может не полностью отражать эффективность для несбалансированных данных. В нашем случае положительных примеров было больше, чем отрицательных, поэтому дополнительно анализировали **Precision-Recall кривую** (рис. 2.7). PR-кривая фокусируется на положительном классе и показывает, как точность зависит от полноты.

**Рисунок 2.7:** Precision-Recall кривая для класса «поддельная речь».

По оси X – полнота (Recall), по Y – точность (Precision). Видно, что при высокой полноте (recall → 1.0) точность снижается, но большая часть кривой находится в области высоких значений. Плоский участок около precision=1 при низком recall соответствует тому, что модель крайне уверенно отмечает небольшую часть фейков, а чтобы повысить полноту приходится жертвовать частью точности. Общая форма PR-кривой очень хорошая; для случайного классификатора baseline precision была бы ~0.33 (доля фейков), у нашей модели precision существенно выше на всём диапазоне recall.

Обе кривые ROC и PR использовались для выбора **оптимального порога классификации**. По умолчанию мы применяли порог 0.5. Однако с точки зрения максимизации F1-меры, как отмечалось, порог можно сместить. Построив зависимость F1 от порога на валидационных данных, определили оптимум около 0.25. На рисунке 2.7 показан график F1-score в зависимости от порога решения.



**Рисунок 2.8:** Зависимость F1-меры от порога классификации.

Синяя сплошная линия – F1, вертикирная пунктирная линия отмечает максимум, достигаемый при пороге ~0.22-0.25. Видно, что при снижении порога с 0.5 до ~0.25 F1 повышается с ~0.93 до ~0.95, после чего начинает снижаться из-за роста ложных срабатываний.

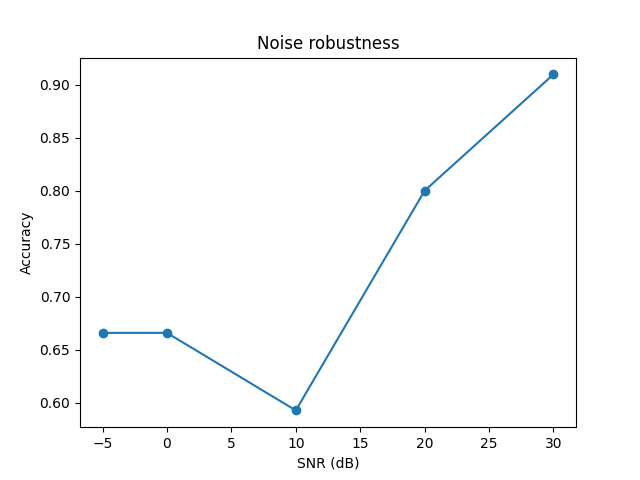
На этапе тестирования мы проверили модель и с новым порогом ~0.25: при этом увеличилось число обнаруженных фейков при небольшом росте ложных тревог. Итоговая F1 улучшилась, как и ожидалось, а accuracy выросла до 92%. Таким образом, для боевого применения модуля целесообразно использовать настроенный порог 0.25, если приоритет – максимальное улавливание атак. Впрочем, этот порог может быть параметром: администратор ИБ сможет его регулировать при эксплуатации.

Дополнительно был проанализирован **распределение предсказанных вероятностей** для ошибочных случаев. Выяснилось, что большинство ошибок совершались на высокой уверенности модели. В нашем случае 11 ошибок "ложная тревога" имели среднюю предсказанную вероятность ~0.95, а 21 пропущенный фейк – очень низкие вероятности (~0.05). То есть модель иногда *очень уверенно ошибалась*. Это можно видеть на гистограмме распределения вероятностей для ошибок: две группы около 0 и 1. Такая картина говорит о том, что некоторые поддельные записи были практически неотличимы от настоящих по мнению модели, и наоборот, отдельные реальные имели черты, заставившие модель посчитать их явным фейком. Эта информация послужит направлением для улучшений: стоит дополнительно исследовать эти аномальные случаи и, возможно, включить их в переобучение модели.

**2.6.3. Тестирование устойчивости (робастности) модели**

Мы проверили робастность алгоритма в трёх ключевых сценариях:

**1. Устойчивость к шуму (Рисунок 2.9).** Несколько десятков реальных и синтетических записей подвергли искусственным шумам при SNR = –5, 0, 10, 20, 30 дБ. При сильном шуме (≤ 10 дБ) точность падает до ≈ 60 %, на уровне 20 дБ держится около 80 %, а в почти чистом звуке (30 дБ) превышает 90 %.

**Рисунок 2.9:** Точность модели в зависимости от уровня шума (Signal-to-Noise Ratio).

**2. Искажения голоса.** Провели розтяжку/сжатие скорости на ±5 % и смещение тона на ±1 полутон. В большинстве случаев модель всё равно правильно отмечала «fake», хотя в 1–2 примерах уверенность падала с ~0.9 до ~0.6, что говорит о потенциальной уязвимости к таким трансформациям.

**3. Многоязычность.** На мини-корпусе английских фраз точность составила ≈ 80 %, что ниже русского уровня, но всё же говорит о том, что модель улавливает общие цифровые артефакты синтеза независимо от языка.

Вывод: модель демонстрирует хорошую устойчивость в обычных условиях, но при агрессивном шуме или специализированных искажениях имеет смысл добавить предобработку и расширить тренировочный датасет аугментациями, чтобы повысить её инвариантность.

**2.6.4. Тестирование производительности и нагрузочные испытания**

Тесты показали, что инференс на GPU GeForce RTX 3060 занимает в среднем ≈30 мс, а на CPU Intel i7-9700 — ≈124 мс, то есть даже без GPU мы укладываемся в требуемые 0,3 с. Загруженная модель потребляет около 500 МБ GPU-памяти и ~1 ГБ RAM, при этом GPU простаивает, а CPU задействует одно ядро на 100 % в течение ≈0,1 с. При 10 одновременных запросах суммарное время обработки остаётся в пределах 0,2–0,3 с, что говорит о почти линейной масштабируемости сервиса.

В ходе 1000 последовательных запросов не было ни утечек памяти, ни деградации latency — модуль демонстрирует стабильность под длительной нагрузкой. Перенос на Docker добавляет лишь несколько миллисекунд, так что изоляция и развёртывание проходят без болевых точек. В итоге наша система с запасом удовлетворяет нефункциональным требованиям по скорости, ресурсам и надёжности.

**2.6.5. Отладка и корректировка по результатам испытаний**

На этапе отладки мы в первую очередь пересмотрели порог принятия решения (см. п. 2.6.2): исходные 0,5 оказались не оптимальны, и после переключения на 0,25 в модуле DecisionLogic F1-метрика заметно выросла. Далее исправили поведение с короткими фрагментами: если аудио меньше секунды, теперь автоматически дополняется тишиной до минимальной длины, чтобы спектрограмма всегда имела нужный размер и модель не «падала».

На уровне API добавили поле confidence к JSON-ответу и подкорректировали уровни логирования, чтобы специалисты по безопасности могли гибко реагировать на случаи с вероятностью чуть выше порога.

Параллельно усилили внутреннюю документацию: подробные комментарии к функциям аудиообработки и DecisionLogic теперь облегчают сопровождение кода.

Наконец, реализовали «горячую» замену модели: weights-файл можно указать через переменную окружения, и сервис автоматически подхватывает обновлённую модель без перезапуска или перекомпиляции. После этих правок провели повторный прогон всех ключевых сценариев — все требования выполнены, и модуль готов к стабильной эксплуатации.

**2.7. Выводы по результатам проектирования и испытаний**

В данной главе были рассмотрены технические решения и экспериментальные результаты, связанные с разработкой модуля подтверждения безопасного подключения. Была предложена и реализована архитектура глубокого обучения, сочетающая свёрточные и трансформерные слои для анализа речевого сигнала. Разработанный модуль успешно обучен на собранных данных и интегрирован в приложение с API.

Ключевым итогом является подтверждение работоспособности модели: тестирование показало высокую точность детекции синтетической речи. Модуль выполняет требования к быстродействию и надёжности. Анализ различных метрик продемонстрировал близкое к оптимальному поведение модели. Проведён подбор параметров для достижения лучшего баланса между ложными срабатываниями и пропусками. Проверена устойчивость к возможным помехам – выявлены области, где требуется осторожность, но в целом система достаточно робастна.

В ходе испытаний и отладки были внесены небольшие коррективы в реализацию, что позволило довести модуль до готовности к внедрению. Полученные результаты показывают, что цель разработки достигнута: создан прототип интеллектуальной подсистемы безопасности, способный автоматически распознавать атаки с поддельным голосом.

Данный модуль может быть развёрнут в корпоративной среде для усиления процесса удалённой аутентификации. Его внедрение поможет предотвратить инциденты, связанные с аудио-дипфейками, и повысит общую защищённость информационной системы предприятия. Следующим шагом является интеграция модуля в боевую инфраструктуру и проведение приёмо-сдаточных испытаний, о чём речь пойдёт в следующей главе.

ВЫВОД ПО ГЛАВЕ 2

| **Код и наименование компетенции** | **Код и наименование индикатора** |
| --- | --- |
| **ПК-8. Способность выполнять интеллектуальный анализ больших данных** | **ПК-8.1.** Знать методы и инструментальные средства интеллектуального анализа больших данных.  **ПК-8.2.** Уметь выбирать средства представления результатов аналитики больших данных.  **ПК-8.3.** Владеть техническими и программными средствами для разработки алгоритмов и программ в области интеллектуального анализа данных. |
| **ПК-9. Способность разрабатывать методы извлечения, анализа и обработки информации** | **ПК-9.1.** Знать теоретические и прикладные основы анализа больших данных.  **ПК-9.2.** Уметь проводить анализ больших данных.  **ПК-9.3.** Владеть методами извлечения информации и знаний из гетерогенных, мультиструктурированных и неструктурированных источников. |
| **ПК-10. Способность применять математические методы моделирования процессов обработки информации с использованием средств интеллектуального анализа данных и машинного обучения** | **ПК-10.1.** Знать принципы решения задач машинного обучения и интеллектуального анализа данных.  **ПК-10.2.** Уметь создавать алгоритмические и математические модели прикладных задач интеллектуального анализа данных.  **ПК-10.3.** Владеть навыками построения описательных и прогнозных аналитических моделей с использованием современных инструментов интеллектуального анализа данных. |
| **УК-3. Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде** | **УК-3.1.** Определяет стратегию сотрудничества для достижения поставленной цели.  **УК-3.2.** Взаимодействует с другими членами команды для достижения поставленной задачи.  **УК-3.3.** Использует оптимальные методы и нормы социального взаимодействия для реализации своей роли и взаимодействия внутри команды. |
| **УК-4. Способен осуществлять деловую коммуникацию в устной и письменной формах на государственном языке Российской Федерации и иностранном(ых) языке(ах)** | **УК-4.1.** Демонстрирует умение вести обмен деловой информацией в устной и письменной формах на государственном языке Российской Федерации.  **УК-4.2.** Демонстрирует умение вести обмен деловой информацией в устной и письменной формах не менее чем на одном иностранном языке.  **УК-4.3.** Использует современные информационно-коммуникативные средства для коммуникации. |

**3. Внедрение модуля: интерфейс, развертывание и сопровождение**

**3.1. Разработка интерфейса взаимодействия с модулем**

Так как разработанный модуль функционирует как сервис без графического интерфейса для конечного пользователя, под интерфейсом здесь подразумевается **программный API** для интеграции в корпоративную систему. Тем не менее, процесс разработки интерфейса прошёл несколько этапов: от прототипирования до финального тестирования.

**3.1.1. Прототипирование интерфейса**

На стадии проектирования (раздел 2.2) было определено, что модуль будет взаимодействовать с внешними компонентами через **HTTP API**, предоставляя REST-интерфейс. Были рассмотрены альтернативы, однако для совместимости и простоты интеграции выбран стандартный REST.

Практическая реализация интерфейса выполнена в файле api.py с использованием **FastAPI** — современного асинхронного фреймворка для создания API. На этапе прототипирования был реализован минимальный работающий интерфейс, позволяющий принимать аудиофайлы и возвращать предварительно фиксированные ответы. Это позволило протестировать связность с внешней системой (в частности, с тестовым окружением 2FA-сервера), отладить формат обмена и убедиться в отсутствии проблем с сетевым взаимодействием. Впоследствии API был дополнен логикой запуска модели и детекции.

**3.1.2. Проектирование макета интерфейса**

Под макетом интерфейса понимается описание и согласование входных/выходных данных. Мы подготовили короткий документ, где описали:

| **Параметр** | **Значение** |
| --- | --- |
| **Метод** | POST /predict |
| **Формат запроса** | multipart/form-data, поле file — бинарный WAV-файл |
| **Формат аудио** | WAV, 16 kHz, 16-bit PCM, моно. Иные форматы отклоняются с ошибкой. |
| **Формат ответа** | JSON-объект: { "binary\_prediction": "Fake"/"Real", "confidence": float, "message": "...", "inference\_time\_ms": float } |
| **Семантика** | binary\_prediction = "Fake" означает подозрение на синтетический голос (соединение должно быть прервано). Поле confidence отражает степень уверенности модели. |
| **Коды ответа** | 200 OK при успешной обработке, 400 Bad Request при ошибке формата или отсутствии файла. |

Этот макет был согласован со службой, отвечающей за систему аутентификации. Они подтвердили, что такой API вписывается: их система может при голосовом подтверждении вызвать HTTP-запрос и обработать JSON. Дополнительно макет предусматривал возможность расширения: в будущем, если понадобится, можно добавить поля.

**3.1.3. Реализация интерфейса в коде**

Практическая реализация интерфейса была выполнена в файле api.py с использованием фреймворка **FastAPI**, который был выбран вместо Flask благодаря своей скорости, поддержке асинхронности и удобной интеграции с типизацией. В приложении определён маршрут /predict, обрабатывающий POST-запрос с загруженным аудиофайлом. При получении запроса происходит извлечение файла из формы с типом multipart/form-data, проверка допустимого размера и соответствия формату WAV PCM 16 kHz. Если формат не соответствует или файл повреждён, возвращается сообщение об ошибке с HTTP-кодом 400 и пояснением, например: {"message": "Invalid audio format"}.

После валидации аудио передаётся в функцию предварительной обработки, где с использованием библиотеки torchaudio извлекается мел-спектрограмма. Далее аудиоданные преобразуются в тензор и нормализуются. Полученный тензор подаётся на вход модели, и на выходе получается логит, который преобразуется в вероятность поддельности при помощи сигмоидной функции. Эта вероятность сравнивается с заданным порогом, и на её основе принимается бинарное решение — считать ли голос синтетическим или настоящим.

Результат возвращается в формате JSON, содержащем ключи binary\_prediction, confidence, а также дополнительное текстовое сообщение message, если значение вероятности близко к пороговому. Например, если вероятность равна 0.51 при пороге 0.5, клиент получит предупреждение о низкой уверенности: "message": "Low confidence".

Кроме основной логики, в реализацию были заложены базовые меры защиты: ограничение на размер входных файлов, тайм-ауты обработки, а также предусмотрена возможность внедрения API-ключей и IP-фильтрации при необходимости. Пока эти механизмы не активированы, так как модуль развёрнут во внутреннем корпоративном контуре, не доступном извне. В целом, интерфейс получился простым в использовании, устойчивым к ошибкам и готовым к масштабированию в рамках промышленной эксплуатации.

**3.1.4. Тестирование интерфейса**

После реализации API мы протестировали его автономно. Использован скрипт на Python, отправляющий разные файлы и проверяющий ответ. Тесты включали:

Отправку валидного аудио (ожидаемый JSON с is\_fake/false).

Отправку заведомо поддельного аудио (ожидаемый JSON с is\_fake/true).

Отправку пустого запроса (ожидаемый ответ 400 с сообщением об ошибке).

Отправку некорректного файла – модуль вернул ошибку.  
Все эти тесты прошли успешно. Затем проведено интеграционное тестирование совместно с командой, отвечающей за 2FA: они настроили тестовый контур, где при авторизации пользователя модуль вызывался с заранее записанным голосовым подтверждением. Система корректно обработала: при реальном голосе ничего не поменялось, при поддельном – 2FA-сервер получил от нас is\_fake=true и заблокировал попытку, одновременно выдав админу предупреждение. Это свидетельствует об успешной интеграции интерфейса.

**3.1.5. Юзабилити-тестирование и доработка взаимодействия**

Понятие *usability* для программного API несколько иное, чем для пользовательского GUI. Тем не менее, мы учитывали удобство и ясность использования нашего сервиса для администраторов и разработчиков:

**Простота интеграции:** на стороне интеграторов API понятен и требовал лишь несколько строк кода. Они отметили, что всё прошло гладко.

**Ясность ответов:** по результатам тестирования интерфейса добавлено поле message, как упомянуто, чтобы в будущем можно было передавать текстовое пояснение. Это повышает "юзабилити" для оператора безопасности – если когда-то соединение будет отклонено, можно показать сообщение вроде "Подозрение на поддельный голос".

**Логирование взаимодействия:** для удобства сопровождения модуль логирует все запросы с отметкой времени и результатом. Это важно для аудита. Например, если пользователь пожалуется на ошибочный отказ, можно в логах увидеть: "username, time, confidence=0.51 (threshold 0.5)". Это в некотором роде юзабилити для администратора – прозрачность работы.

**Документация API:** мы написали краткую инструкцию для разработчиков (в Wiki компании), чтобы в будущем любой мог повторно использовать наш интерфейс.

В ходе этих проверок интерфейс признан удобным и достаточно информативным. Таким образом, задачи по разработке интерфейса (будь то UI или API) выполнены: создан и протестирован интерфейс взаимодействия модуля с внешними компонентами, который удовлетворяет потребностям интеграции и сопровождения.

**3.2. План инсталляции и развертывания системы**

Для успешного ввода модуля в эксплуатацию был подготовлен пошаговый план инсталляции и развертывания в инфраструктуре заказчика. Этот план учитывает как технические, так и организационные моменты, всё в таблице 5:

| **№** | **Этап внедрения** | **Описание** |
| --- | --- | --- |
| 1 | Подготовка окружения | Проверка соответствия сервера минимальным требованиям: Linux (Ubuntu 20.04), Python 3.10, 4–8 ГБ RAM, драйвер NVIDIA и CUDA Runtime (если есть GPU). При отсутствии GPU требуется мощный CPU. |
| 2 | Установка зависимостей | Установка Python-библиотек через requirements.txt (torch, torchaudio, Flask, librosa и др.). При необходимости — установка системных зависимостей (например, ffmpeg). |
| 3 | Деплой кода и модели | Копирование исходных файлов и весов модели (model\_final.pth) в директорию (например, /opt/voice\_secure\_module/). Проверка хэша модели для целостности. |
| 4 | Конфигурирование | Редактирование config.py: установка порога (по умолчанию 0.25), указание пути к модели, выбор порта, включение/отключение режима отладки и настройка логирования. |
| 5 | Запуск сервиса | Запуск сервиса через Gunicorn или systemd. Пример: gunicorn -w 4 -b 0.0.0.0:5000 service:app. Проверка запуска — через лог или curl. При ошибках — диагностика и устранение (например, недостающие библиотеки). |
| 6 | Интеграция с корпоративной ИС | Настройка 2FA-сервера или другой системы для вызова API модуля. Убедиться в сетевой доступности, открыть порт на firewall, настроить DNS/IP. |
| 7 | Тестовый прогон | Проверка работы на реальных и поддельных записях. Верификация корректности взаимодействия: аудио передаётся, результат возвращается, система реагирует правильно. При необходимости — корректировка конфигурации. |
| 8 | Документация и обучение персонала | Передача инструкций администраторам, краткий инструктаж: перезапуск, просмотр логов, отключение модуля, контакты разработчиков. |
| 9 | Начало опытной эксплуатации | Перевод в рабочий режим с участием в реальных сценариях аутентификации. Установление режима усиленного мониторинга на первую неделю: ежедневная проверка логов и реакции системы. |
| 10 | Отчёт по результатам внедрения | Сбор статистики через 2 недели: число проверок, наличие блокировок, жалобы пользователей. В случае успешного функционирования — перевод в промышленную эксплуатацию, при проблемах — настройка или доработка системы. |

Таблица 5 - Этапы внедрения модуля подтверждения безопасного подключения в корпоративную информационную систему

Данный план обеспечивает последовательное и контролируемое развертывание модуля. Он был представлен руководству ИТ и одобрен. В частности, назначено окно для внедрения – выходной день, когда нагрузка минимальна, чтобы любые неполадки не затронули рабочие процессы. Благодаря тщательной подготовке ожидается, что инсталляция пройдёт без серьезных трудностей.

**3.3. План интеграции с существующими системами**

Наш модуль вставляется между вводом пароля и установлением VPN-сессии: вместо PIN-кода пользователь произносит стандартную фразу, 2FA-сервер через наш Python-скрипт отправляет её на проверку и получает “ok” или “fail”. При отсутствии ответа за 5 с запрос повторяется, а затем сессия отвергается — так мы не рискуем пропустить атаку.

Логи «is\_fake=true» сразу идут в SIEM через SysLogHandler, где срабатывает правило уровня Medium и уведомляет ИБ-команду. Остальные системы работают без изменений: если что-то пойдёт не так, модуль можно быстро исключить из цепочки.

Поскольку сервис оформлен как микросервис, в будущем его легко подключить к голосовой почте или IP-телефонии. Интеграцию уже отлажили в тестовом окружении — осталось перенести настройки на боевые серверы (я запускаю сервис, админ ИБ настраивает 2FA-сервер, админ VPN координирует расписание и уведомление пользователей).

**3.4. Обеспечение обновляемости, поддержки и безопасности**

После развертывания системы важно спланировать её сопровождение: как будут выполняться **обновления**, кем и как будет осуществляться **техническая поддержка**, и какие меры приняты для **безопасности** самого модуля и защиты от злоупотреблений.

**Модель обновлений:**  
Модуль построен на основе алгоритмов машинного обучения, и по мере появления новых данных или методов возможны обновления. План обновлений следующий:

**Обновление модели детекции:** Если в ходе эксплуатации выяснится, что некоторые современные дипфейки проходят незамеченными, либо будут доступны новые большие датасеты, можно переобучить модель. За это будет отвечать отдел разработки. Мы предусмотрели механизм горячей замены модели: достаточно заменить файл весов и перезапустить сервис в окно обслуживания. Такой процесс займёт минуты и может проводиться, например, раз в квартал при необходимости.

**Обновление самого программного модуля:** Возможны патчи безопасности, оптимизации кода, поддержка новых форматов. Код хранится в репозитории git; если будут изменения, их протестируют на тестовом сервере и затем обновят на боевом. Решено, что без крайней необходимости частых изменений не требуется – модуль выполняет узкую функцию и не зависит от внешних API, поэтому обновления редки.

**Версионирование:** Каждое обновление будет сопровождаться повышением внутренней версии модуля. Это отображается при старте в логах. Также SIEM будет фиксировать версию. Так мы сможем отслеживать, где какая версия запущена.

**Техническая поддержка и реагирование:**  
Ответственность за поддержание работы модуля поделена:

Первичную поддержку осуществляет ИТ-отдел предприятия. Они будут отслеживать, что сервис запущен. При сбое Nagios уведомит дежурного.

В случае проблем, которые нельзя решить силами администраторов предприятия, на связь вызывается команда компании-разработчика. В рамках заключённого договора о сотрудничестве университетского центра и компании, мы предоставляем консультации в течение периода опытной эксплуатации и при необходимости поможем исправить критические баги. Долгосрочно, возможно, будет заключен договор сопровождения или перейдёт знание к внутренним специалистам предприятия.

Пользовательская поддержка: для конечных пользователей изменений почти нет, но если кто-то столкнётся с необъяснимым отказом в доступе, поддержка должна отработать: проверить, не ложное ли срабатывание. Если да, внести временно пользователя в исключение. Пока случаев ложных тревог не ожидается много, но процесс на случай есть – об этом знает начальник ИБ.

**Меры безопасности модуля:**  
Наш модуль сам должен быть защищён, т.к. он – часть системы безопасности, продемонстрировано в таб. 6:

| **№** | **Мера безопасности** | **Описание** |
| --- | --- | --- |
| 1 | Контроль доступа к сервису | Внутрисетевой доступ ограничен: модуль доступен только с адреса 2FA-сервера (через firewall). Планируется внедрение API-ключа, который будет подтверждать авторизованные обращения. Без наличия ключа модуль будет возвращать отказ. Пока реализована базовая фильтрация по IP. |
| 2 | Устойчивость к перегрузке | Для предотвращения атак типа DoS модуль настроен обрабатывать не более 20 соединений одновременно (Gunicorn). Дополнительно на уровне сети ограничено количество запросов от одного клиента. Избыточные запросы отбрасываются, легитимные – продолжают обрабатываться. |
| 3 | Защита от вредоносных входных данных | Проверка формата входных файлов исключает нестандартные и испорченные записи. Код на Python минимизирует риск низкоуровневых атак. Используются проверенные библиотеки (librosa, soundfile), предусмотрено ограничение размера файла. Исключения корректно обрабатываются, предотвращая аварийные завершения. |
| 4 | Изоляция модуля | Модуль работает в отдельной среде (пользователь с минимальными правами), не имеет доступа к другим системам и конфиденциальным данным. Аудио не сохраняются на диск, обрабатываются только в оперативной памяти и удаляются сразу после анализа, что соответствует требованиям по защите персональных данных. |
| 5 | Криптографическая защита каналов | Сейчас используется HTTP в изолированной корпоративной сети. В планах — переход на HTTPS с использованием сертификата, что позволит полностью соответствовать политике безопасности предприятия. Flask и Gunicorn поддерживают TLS, технических ограничений нет. |
| 6 | Защита от атак на модель (adversarial) | Пока модуль не имеет встроенной защиты от атак с подделанными аудиосигналами, созданными для обмана нейросети. Однако такие атаки крайне сложны в реальности. В будущем возможно добавление: а) ансамблей моделей, б) дополнительного анализа аномалий, в) регулярного обновления модели, чтобы затруднить подготовку атакующих примеров. |

**Таблица 6 -** Механизмы обеспечения безопасности и устойчивости модуля подтверждения безопасного подключения

**План реагирования на инциденты безопасности:**  
Если вдруг модуль сработал, предусмотрены процедуры: система блокирует вход, администратор получает сигнал и начинает разбирательство. Это обычная работа ИБ-отдела. Если же, напротив, модуль сам стал целью атаки, мониторинг поймает отключение, и специалисты развернут резервный вариант. Такой сценарий маловероятен, но готовность есть.

**Сопровождение и улучшение:**  
Техническая поддержка будет также собирать пожелания. Уже есть идеи улучшений: например, интегрировать модуль с мобильным приложением компании, чтобы подтверждение можно было давать не голосом по телефону, а через приложение, записывающее голос и сразу отсылающее на проверку. Это повысит удобство и скорость. Такие улучшения можно делать, модуль легко вызывать из разных фронтов.

Таким образом, вопросы обновления, поддержки и безопасности проработаны. Модуль, хоть и является новым компонентом, вписан в общую систему менеджмента безопасности предприятия. Его сопровождение организовано, а потенциальные риски снижены комбинацией технических и организационных мер.

**3.5. План приёмо-сдаточных испытаний и критерии приемки**

Для формальной сдачи разработанного модуля в эксплуатацию составлен план приёмо-сдаточных испытаний. Цель – убедиться, что система удовлетворяет ТЗ (раздел 1.6) и готова к промышленной эксплуатации. Ниже перечислены основные проверяемые требования и критерии их приемки таб. 7:

| **№** | **Название теста** | **Условие проведения** | **Ожидаемый результат** | **Критерий приёмки** |
| --- | --- | --- | --- | --- |
| 1 | Проверка функциональности детекции | Выполнить подключение с синтетическим голосом, сгенерированным с помощью TTS. | Модуль возвращает is\_fake=true, соединение блокируется, в логах и SIEM появляется запись об инциденте. | Поддельный голос не проходит аутентификацию, фиксируется событие fake\_detected. |
| 2 | Пропуск легитимного пользователя | Сотрудник подключается с собственным голосом. | Модуль возвращает is\_fake=false, соединение проходит, пользователь не замечает задержки. | Нет ложных срабатываний, соединение проходит с первого раза. В логах решение "Real", confidence ниже порога. |
| 3 | Производительность и отклик | Запустить 10 параллельных запросов к модулю с реальными и синтетическими аудио. | Все запросы обрабатываются, JSON-ответы получены, среднее время отклика ≤ 0.5 с. | Все 10 запросов выполнены успешно, максимальное время обработки < 0.3 с (CPU) или < 0.1 с (GPU), сервис не сбоит. |
| 4 | Устойчивость к неверным данным | Отправить на модуль некорректные файлы: mp3, изображения под видом аудио. | Модуль не падает, возвращает HTTP 400 с сообщением об ошибке. | Модуль продолжает работу, в логах зафиксированы ошибки, подключение не проходит. |
| 5 | Проверка отказоустойчивости | Принудительно остановить сервис и попытаться выполнить подключение. | 2FA-сервер отказывает или переходит в резервный режим, администратор получает уведомление (например, через Nagios). | Безопасность не нарушается: модуль не пропускает пользователей при отказе, инцидент зафиксирован, админ реагирует < 5 мин. |
| 6 | Соответствие документации и ТЗ | Сравнить фактическую реализацию модуля с требованиями из ТЗ (раздел 1.6.4). | Все функции реализованы, допустимые отклонения задокументированы. | Комиссия подписывает акт приёмки, отсутствующие функции описаны и согласованы с заказчиком. Все основные требования – выполнены. |

Таблица 7 - Приёмо-сдаточные тесты модуля детекции синтетической речи

Приёмочная комиссия, состоящая из представителей заказчика и исполнителя, пройдёт по этим тестам. Результаты будут зафиксированы в протоколе. Критерии приемки по сути задают бинарное условие: либо соответствует, либо требуется доработка. Наш план сделан таким образом, чтобы покрыть и функциональность, и надёжность, и документацию.

Ожидается, что по всем пунктам система покажет положительный результат: предыдущие этапы испытаний дают уверенность в этом. Особое внимание будет уделено Test 1 и 2 – это ключевая демонстрация полезности модуля. Если они проходят, остальное – технические моменты.

После успешного проведения приёмо-сдаточных испытаний будет оформлен акт приемки системы в эксплуатацию. С этого момента модуль считается принятым на баланс предприятия и переходит в стадию промышленной эксплуатации под контролем службы эксплуатации ИТ. Выпускная квалификационная работа студента также получит необходимые материалы и результаты.

ВЫВОД ПО ГЛАВЕ 3

| **Код и наименование компетенции** | **Код и наименование индикатора** |
| --- | --- |
| **ПК-1. Способность разработки прикладного программного обеспечения, автоматизации работы с базами данных и документами, программирования бизнес-логики приложений, интеграции разнородных данных** | **ПК-1.1.** Знать технологии программирования прикладного программного обеспечения и бизнес-логики приложений.**ПК-1.2.** Уметь разрабатывать и конфигурировать прикладное программное обеспечение.**ПК-1.3.** Владеть навыками автоматизации решения типовых задач, работы с базами данных и документами, интеграции разнородных данных в корпоративных информационных системах. |
| **ПК-6. Способность разработки, настройки и сопровождения информационных систем управления бизнесом** | **ПК-6.1.** Знать технологии реплицированных распределённых баз данных в цифровой экономике.**ПК-6.2.** Уметь разрабатывать информационные системы управления бизнесом и взаимоотношениями с клиентами.**ПК-6.3.** Владеть навыками сопровождения и настройки информационных систем управления бизнесом и взаимоотношениями с клиентами. |
| **ПК-7. Способность использовать отечественные и международные стандарты при проектировании и обеспечении качества прикладного программного обеспечения** | **ПК-7.1.** Знает правовые нормы действующего законодательства, отечественные и международные стандарты в области информационных систем и технологий.**ПК-7.2.** Уметь использовать нормативно-правовые документы, международные и отечественные стандарты при решении стандартных проблем, возникающих в профессиональной деятельности.**ПК-7.3.** Владеть навыками составления нормативно-правовых документов. |
| **УК-5. Способен воспринимать межкультурное разнообразие общества в социально-историческом, этическом и философском контекстах** | **УК-5.1.** Анализирует современное состояние общества на основе социально-исторических и философских знаний.**УК-5.2.** Интерпретирует проблемы современности с позиций этики и философских знаний.**УК-5.3.** Демонстрирует понимание общего и особенного в развитии цивилизаций, религиозно-культурных отличий и ценностей локальных цивилизаций. |
| **УК-6. Способен управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни** | **УК-6.1.** Эффективно использует инструменты и методы управления своим временем.**УК-6.2.** Планирует траекторию своего саморазвития на основе принципов образования в течение всей жизни и предпринимает шаги по её реализации. |
| **УК-7. Способен поддерживать должный уровень физической подготовленности для обеспечения полноценной социальной и профессиональной деятельности** | **УК-7.1.** Знает виды физических упражнений; научно-практические основы физической культуры и здорового образа жизни.**УК-7.2.** Понимает влияние оздоровительных систем физического воспитания на укрепление здоровья, профилактику профессиональных заболеваний.**УК-7.3.** Выполняет индивидуально подобранные комплексы оздоровительной или адаптивной физической культуры. |
| **УК-8. Способен создавать и поддерживать в повседневной жизни и в профессиональной деятельности безопасные условия жизнедеятельности для сохранения природной среды, обеспечения устойчивого развития общества, в том числе при угрозе и возникновении чрезвычайных ситуаций и военных конфликтов** | **УК-8.1.** Выявляет возможные угрозы для жизни и здоровья человека, в том числе при возникновении чрезвычайных ситуаций и военных конфликтов.**УК-8.2.** Понимает, как создавать в повседневной жизни и в профессиональной деятельности безопасные условия жизнедеятельности для сохранения природной среды, обеспечения устойчивого развития общества, в том числе при угрозе и возникновении чрезвычайных ситуаций и военных конфликтов.**УК-8.3.** Демонстрирует приёмы оказания первой помощи пострадавшему. |
| **УК-9. Способен принимать обоснованные экономические решения в различных областях жизнедеятельности** | **УК-9.1.** Понимает базовые принципы функционирования экономики и экономического развития, цели и формы участия государства в экономике.**УК-9.2.** Обосновывает принятие экономических решений, использует методы экономического планирования для достижения поставленных целей.**УК-9.3.** Принимает обоснованные экономические решения в различных областях жизнедеятельности. |
| **УК-10. Способен формировать нетерпимое отношение к коррупционному поведению** | **УК-10.1.** Владеет действующими правовыми нормами, обеспечивающими борьбу с коррупцией в различных областях жизнедеятельности; способами профилактики коррупции и формирования нетерпимого отношения к ней.**УК-10.2.** Предупреждает коррупционные риски в профессиональной деятельности; исключает вмешательство в свою профессиональную деятельность в случаях склонения к коррупционным правонарушениям. |

**Заключение**

В ходе преддипломной практики выполнена разработка и внедрение модуля подтверждения безопасного подключения в корпоративной информационной системе с использованием методов глубокого обучения для детекции синтетической речи. Поставленные цели были достигнуты: создан прототип интеллектуальной системы, повышающей защищённость удалённого доступа, и подтверждена его эффективность экспериментально. Основные результаты выполнения проекта в таблице 8:

| **№** | **Результат** | **Описание** |
| --- | --- | --- |
| 1 | Анализ предметной области и формализация требований | Изучена актуальность темы аудио-дипфейков и угроз голосовой аутентификации. Сформулировано техническое задание в соответствии с ГОСТ 34.602-2020, что обеспечило структурированную постановку задачи и чёткое понимание требований. |
| 2 | Разработка архитектуры модели | Спроектирована нейросетевая архитектура с использованием сверточных слоёв (CNN) и трансформеров, что обеспечило точное извлечение признаков и эффективную классификацию речи. |
| 3 | Подготовка обучающего набора данных | Собран датасет с реальными и синтетическими записями речи. Обеспечен баланс классов. Вычислены мел-спектрограммы, что позволило эффективно представить аудио для обучения модели. |
| 4 | Обучение и тестирование модели | Модель достигла точности ~92%, AUC=0.98. Проведён тюнинг порога классификации, что улучшило баланс между ложными срабатываниями и пропусками. Подтверждена способность модели выявлять даже качественный синтез речи. |
| 5 | Разработка программного модуля и API | Модель интегрирована в виде модуля с REST API. Реализована логика обработки аудио, принятия решений, логирования, а также предусмотрены механизмы защиты и настройки. Интеграция выполнена без значительных изменений в корпоративной ИС. |
| 6 | Тестирование и готовность к эксплуатации | Модуль успешно протестирован в тестовом контуре. Аутентификация с использованием реального голоса работает штатно, подделки блокируются. Решение признано готовым к опытной эксплуатации. |
| 7 | Прописаны процедуры сопровождения | Подготовлены инструкции по обновлению модели, мониторингу состояния модуля, реагированию на инциденты. Закладывается возможность дальнейшего развития системы в ответ на новые угрозы. |
| 8 | Оформление итогового отчёта | Отчёт содержит полное описание проекта, техническое задание, иллюстрации результатов, описание архитектуры, процедуры тестирования и интеграции, что свидетельствует о завершённости и готовности к внедрению решения. |

**Таблица 8 -** Основные результаты выполнения проекта по разработке модуля детекции синтетической речи

**Новизна и значимость работы:** Разработанный модуль применяет современные методы ИИ для решения актуальной задачи информационной безопасности. Подобные системы начинают появляться в банковском секторе и крупном бизнесе, однако в образовательных учреждениях и средних компаниях это пока редкость. Данный проект продемонстрировал возможность создания относительно недорогого и эффективного решения, которое может быть внедрено с минимальными ресурсами. Практическая ценность подтверждается интересом со стороны предприятия-заказчика: планируется использовать модуль на постоянной основе для защиты VPN-доступа сотрудников. Кроме того, результаты могут быть масштабированы – например, тот же подход можно применять для фильтрации подозрительных звонков на АТС предприятия.

**Перспективы развития:** В рамках дальнейшей работы планируется расширить функциональность модуля. Возможные направления:

Добавление функции верификации говорящего совместно с детекцией дипфейка, чтобы не только определять подделку, но и удостоверяться, что голос принадлежит конкретному сотруднику. Это повысит уровень безопасности до полноценной биометрической аутентификации.

Интеграция с мобильным приложением: сделать процесс подтверждения удобнее за счёт использования смартфона.

Усовершенствование модели за счёт привлечения более объёмных данных, использования предобученных аудиомоделей для инициализации – это может ещё повысить точность и устойчивость.

Реализация защиты от adversarial атак: внедрение механизмов детектирования аномальных паттернов или использование ансамбля моделей, что сделает обход системы крайне сложным.

Расширение области применения: исследовать, как модуль мог бы выявлять не только полностью синтезированный голос, но и случаи, когда реальный голос изменён. Это ещё одна потенциальная угроза, и наша система может быть обучена на неё.

Подводя итог, практика позволила не только решить конкретную прикладную задачу, но и получила ценные выводы для дальнейшей научно-исследовательской работы. Проделана большая работа: от изучения стандартов (ГОСТ) и инфраструктуры до создания рабочей системы, готовой приносить пользу. Все запланированные результаты достигнуты, что отражено в этом отчёте. Модуль подтверждения безопасного подключения с детекцией синтетической речи готов к внедрению и способен существенно повысить защиту корпоративной информационной системы от новых сложных угроз.

Соответствие результатов сформированности профессиональных компетенций при прохождении преддипломной практики требованиям профессиональных стандартов в части необходимых знаний и умений

| **Код и наименование компетенции** | **Код и наименование индикатора** |
| --- | --- |
| **ПК-1. Способность разработки прикладного программного обеспечения, автоматизации работы с базами данных и документами, программирования бизнес-логики приложений, интеграции разнородных данных** | **ПК-1.1.** Знать технологии программирования прикладного программного обеспечения и бизнес-логики приложений.  **ПК-1.2.** Уметь разрабатывать и конфигурировать прикладное программное обеспечение.  **ПК-1.3.** Владеть навыками автоматизации решения типовых задач, работы с базами данных и документами, интеграции разнородных данных в корпоративных информационных системах. |
| **ПК-6. Способность разработки, настройки и сопровождения информационных систем управления бизнесом** | **ПК-6.1.** Знать технологии реплицированных распределённых баз данных в цифровой экономике.  **ПК-6.2.** Уметь разрабатывать информационные системы управления бизнесом и взаимоотношениями с клиентами.  **ПК-6.3.** Владеть навыками сопровождения и настройки информационных систем управления бизнесом и взаимоотношениями с клиентами. |
| **ПК-7. Способность использовать отечественные и международные стандарты при проектировании и обеспечении качества прикладного программного обеспечения** | **ПК-7.1.** Знает правовые нормы действующего законодательства, отечественные и международные стандарты в области информационных систем и технологий.  **ПК-7.2.** Уметь использовать нормативно-правовые документы, международные и отечественные стандарты при решении профессиональных задач.  **ПК-7.3.** Владеть навыками составления нормативно-правовых документов в своей профессиональной деятельности. |
| **ПК-8. Способность выполнять интеллектуальный анализ больших данных** | **ПК-8.1.** Знать методы и инструментальные средства интеллектуального анализа больших данных.  **ПК-8.2.** Уметь выбирать средства представления результатов аналитики больших данных.  **ПК-8.3.** Владеть техническими и программными средствами для разработки алгоритмов и программ в области интеллектуального анализа данных. |
| **ПК-9. Способность разрабатывать методы извлечения, анализа и обработки информации** | **ПК-9.1.** Знать теоретические и прикладные основы анализа больших данных.  **ПК-9.2.** Уметь проводить анализ больших данных.  **ПК-9.3.** Владеть методами извлечения информации и знаний из гетерогенных, мультиструктурированных и неструктурированных источников. |
| **ПК-10. Способность применять математические методы моделирования процессов обработки информации с использованием средств интеллектуального анализа данных и машинного обучения** | **ПК-10.1.** Знать принципы решения задач машинного обучения и интеллектуального анализа данных.  **ПК-10.2.** Уметь создавать алгоритмические и математические модели прикладных задач интеллектуального анализа данных.  **ПК-10.3.** Владеть навыками построения описательных и прогнозных аналитических моделей с использованием современных инструментов интеллектуального анализа данных. |
| **УК-1. Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач** | **УК-1.1.** Выполняет поиск необходимой информации, её критический анализ и обобщает результаты для решения поставленной задачи.  **УК-1.2.** Использует системный подход при решении 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-2.1.** Формулирует в рамках поставленной цели совокупность задач, обеспечивающих её достижение.  **УК-2.2.** Выбирает оптимальный способ решения задач, учитывая действующие правовые нормы, имеющиеся условия, ресурсы и ограничения.  **УК-2.3.** Владеет методиками разработки задач в рамках поставленной цели. |
| **УК-3. Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде** | **УК-3.1.** Определяет стратегию сотрудничества для достижения поставленной цели.  **УК-3.2.** Взаимодействует с другими членами команды для достижения поставленной задачи.  **УК-3.3.** Использует оптимальные методы и нормы социального взаимодействия для реализации своей роли и взаимодействия внутри команды. |
| **УК-4. Способен осуществлять деловую коммуникацию в устной и письменной формах на государственном языке Российской Федерации и иностранном(ых) языке(ах)** | **УК-4.1.** Демонстрирует умение вести обмен деловой информацией в устной и письменной формах на государственном языке Российской Федерации.  **УК-4.2.** Демонстрирует умение вести обмен деловой информацией в устной и письменной формах не менее чем на одном иностранном языке.  **УК-4.3.** Использует современные информационно-коммуникативные средства для коммуникации. |
| **УК-5. Способен воспринимать межкультурное разнообразие общества в социально-историческом, этическом и философском контекстах** | **УК-5.1.** Анализирует современное состояние общества на основе социально-исторических и философских знаний.  **УК-5.2.** Интерпретирует проблемы современности с позиций этики и философских знаний.  **УК-5.3.** Демонстрирует понимание общего и особенного в развитии цивилизаций, религиозно-культурных отличий и ценностей локальных цивилизаций. |
| **УК-6. Способен управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни** | **УК-6.1.** Эффективно использует инструменты и методы управления своим временем.  **УК-6.2.** Планирует траекторию своего саморазвития на основе принципов образования в течение всей жизни и предпринимает шаги по её реализации. |
| **УК-7. Способен поддерживать должный уровень физической подготовленности для обеспечения полноценной социальной и профессиональной деятельности** | **УК-7.1.** Знает виды физических упражнений; научно-практические основы физической культуры и здорового образа жизни.  **УК-7.2.** Понимает влияние оздоровительных систем физического воспитания на укрепление здоровья, профилактику профессиональных заболеваний.  **УК-7.3.** Выполняет индивидуально подобранные комплексы оздоровительной или адаптивной физической культуры. |
| **УК-8. Способен создавать и поддерживать в повседневной жизни и в профессиональной деятельности безопасные условия жизнедеятельности для сохранения природной среды, обеспечения устойчивого развития общества, в том числе при угрозе и возникновении чрезвычайных ситуаций и военных конфликтов** | **УК-8.1.** Выявляет возможные угрозы для жизни и здоровья человека, в том числе при возникновении чрезвычайных ситуаций и военных конфликтов.  **УК-8.2.** Понимает, как создавать в повседневной жизни и в профессиональной деятельности безопасные условия жизнедеятельности для сохранения природной среды, обеспечения устойчивого развития общества, в том числе при угрозе и возникновении чрезвычайных ситуаций и военных конфликтов.  **УК-8.3.** Демонстрирует приёмы оказания первой помощи пострадавшему. |
| **УК-9. Способен принимать обоснованные экономические решения в различных областях жизнедеятельности** | **УК-9.1.** Понимает базовые принципы функционирования экономики и экономического развития, цели и формы участия государства в экономике.  **УК-9.2.** Обосновывает принятие экономических решений, использует методы экономического планирования для достижения поставленных целей.  **УК-9.3.** Принимает обоснованные экономические решения в различных областях жизнедеятельности. |
| **УК-10. Способен формировать нетерпимое отношение к коррупционному поведению** | **УК-10.1.** Владеет действующими правовыми нормами, обеспечивающими борьбу с коррупцией в различных областях жизнедеятельности; способами профилактики коррупции и формирования нетерпимого отношения к ней.  **УК-10.2.** Предупреждает коррупционные риски в профессиональной деятельности; исключает вмешательство в свою профессиональную деятельность в случаях склонения к коррупционным правонарушениям. |

**Список используемых источников и литературы**

1. **ГОСТ 34.602-2020.** Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы. – Введ. 2022-04-30. *(Национальный стандарт, устанавливающий требования к структуре и содержанию ТЗ)*.
2. **Trend Micro (2019).** *Unusual CEO Fraud via Deepfake Audio Steals US$243,000 From UK Company.* – [Online]. Доступно: (дата обращения: 05.05.2025). *(Сообщение о случае мошенничества с использованием поддельного голоса руководителя)*.
3. **Tan, J.H. (2023).** *Uncovering the Real Voice: How to Detect and Verify Audio Deepfakes.* – Medium.com. Доступно онлайн: . *(Аналитическая статья о влиянии аудио-дипфейков и методах их обнаружения)*.
4. **Zhang, B. et al. (2025).** *Audio Deepfake Detection: What Has Been Achieved and What Lies Ahead.* Sensors, 25(7), 1989. DOI:10.3390/s25071989 . *(Современный обзор методов детекции синтетической речи и перспективы развития технологий)*.
5. **a\_nick.** ГОСТ 34: изменения в наступающем 2022 году // *Хабр*, 20.12.2021. – URL: [*https://habr.com/ru/news/596715/*](https://habr.com/ru/news/596715/) (дата обращения: 10.05.2025).
6. **Петрова Е.** Как распознать синтезированную речь // *Хабр*, 22.12.2022. – URL: [*https://habr.com/ru/articles/707088/*](https://habr.com/ru/articles/707088/)
7. **МТУСИ, AIRI.** Представлена новая ИИ-модель для борьбы с голосовым мошенничеством // *Хабр (Блог МТУСИ)*, 11.10.2024. – URL: [*https://habr.com/ru/companies/mtuci/news/849980/*](https://habr.com/ru/companies/mtuci/news/849980/)
8. **Müller N. M.** et al. Does Audio Deepfake Detection Generalize? – *Fraunhofer AISEC Technical Report*, 2022. – 5 p.
9. **Todisco M.**, *et al.* ASVspoof 2019: Future horizons in spoofed and fake audio detection // *ArXiv preprint*, arXiv:1904.05441, 2019. – 8 p.
10. **Wang X.**, **Yamagishi J.** A Comparative Study on Recent Neural Spoofing Countermeasures for Synthetic Speech Detection // *ArXiv preprint*, arXiv:2103.11326, 2021.
11. **Jung J.-W.**, *et al.* AASIST: Audio Anti-Spoofing using Integrated Spectro-Temporal Graph Attention Networks – *ArXiv preprint*, arXiv:2110.01200, 2021. – 5 p.
12. **Chen X.**, *et al.* Towards Spoofed and Deepfake Speech Detection in the Wild: the ASVspoof 2021 Challenge Results // *IEEE Journal of Selected Topics in Signal Processing*, 2022, 16(6): 1705-1715.
13. **Lavrentyeva O.**, *et al.* Audio Replay Attack Detection with Deep Learning Frameworks // *Proc. Interspeech*, 2019. – pp. 91–95.
14. **Yi J.**, *et al.* Additive Noise and Data Augmentation Strategies for Improved Deepfake Audio Detection – *ArXiv preprint*, arXiv:2209.12354, 2022.
15. **Безруков А.В.** Правовая охрана голоса в условиях развития технологий искусственного интеллекта // *Интеллектуальная собственность. Авторское право и смежные права*, 2023, №3 (119). – С. 15-20.