

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Б1.О.08
(индекс дисциплины)

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Информационные технологии и безопасность в системах искусственного интеллекта

(наименование дисциплины)

по направлению подготовки

01.04.02 Прикладная математика и информатика

направленность (профиль)

Искусственный интеллект и машинное обучение в беспилотных мобильных системах и комплексах

Форма обучения: очная

Год набора: 2026

Общая трудоемкость: 6 ЗЕ

Распределение часов дисциплины по семестрам

Семестр		Итого
Вид занятий	Форма контроля	
	Экзамен	
Лекции	12	12
Лабораторные		
Практические	24	24
Руководство: курсовые работы (проекты) / РГР	-	-
Промежуточная аттестация	0,35	0,35
Контактная работа	36,35	36,35
Самостоятельная работа	144	144
Контроль	35,65	35,65
Итого	216	216

Рабочую программу составил(и):

Доцент института цифровых технологий, канд. экон. наук. Т.А. Раченко

(должность, ученое звание, степень, Фамилия И.О.)

(должность, ученое звание, степень, Фамилия И.О.)

Рецензирование рабочей программы дисциплины:



Отсутствует



Рецензент

(должность, ученое звание, степень, Фамилия И.О.)

Рабочая программа дисциплины составлена на основании ФГОС ВО и учебного плана
направления подготовки

01.04.02 Прикладная математика и информатика

Срок действия рабочей программы дисциплины до «31» августа 2028 г.

УТВЕРЖДЕНО

На заседании института цифровых технологий

(протокол заседания № 1 от «05» сентября 2025 г.)

1. Цель освоения дисциплины

Цель дисциплины – формирование у обучающихся системных знаний и практических навыков в области применения информационно-коммуникационных технологий (ИКТ) в системах искусственного интеллекта (ИИ) с учётом требований информационной безопасности, а также способности адаптировать и комбинировать существующие ИКТ для решения профессиональных задач.

Задачи дисциплины:

1. изучить современные информационные технологии, используемые при разработке и эксплуатации систем ИИ;
2. освоить основные угрозы и уязвимости систем ИИ, методы их выявления и нейтрализации;
3. сформировать умения комбинировать и адаптировать ИКТ для решения задач профессиональной деятельности с соблюдением требований информационной безопасности;
4. развить навыки оценки и выбора средств защиты информации в контексте систем ИИ.

2. Место дисциплины в структуре ОПОП ВО

Дисциплина Б1.О.08 «Информационные технологии и безопасность в системах искусственного интеллекта» относится к обязательной части Блока 1 «Дисциплины (модули)» учебного плана.

Для успешного освоения дисциплины обучающийся должен владеть базовыми знаниями в области:

- алгоритмов и структур данных;
- основ машинного обучения и нейронных сетей (дисциплина Б1.О.06);
- математического и компьютерного моделирования (Б1.О.09);
- вероятностных и статистических методов анализа данных (Б1.О.05).

Знания, умения и навыки, полученные при изучении данной дисциплины, необходимы для выполнения выпускной квалификационной работы (Б3.01(Д)), прохождения преддипломной практики (Б2.В.02(Пд)), а также для профессиональной деятельности в области разработки и эксплуатации защищённых систем ИИ.

3. Планируемые результаты обучения

Формируемая и контролируемая компетенция	Индикаторы достижения компетенции	Планируемые результаты обучения
ОПК-4 Способен комбинировать и адаптировать существующие информационно-коммуникационные	ОПК-4.1. Знает современные информационно-коммуникационные технологии и основные требования	Знать: – современные ИКТ, применяемые в системах ИИ (облачные платформы, контейнеризация, микросервисная архитектура, API); – основные угрозы информационной

Формируемая и контролируемая компетенция	Индикаторы достижения компетенции	Планируемые результаты обучения
технологии для решения задач в области профессиональной деятельности с учетом требований информационной безопасности	информационной безопасности.	безопасности систем ИИ (атаки на этапе обучения, на этапе вывода, на конфиденциальность и целостность); – нормативные и правовые требования к обработке данных в РФ (ФЗ-152, требования ФСТЭК).
	ОПК-4.2. Умеет комбинировать и адаптировать информационно-коммуникационные технологии для решения профессиональных задач с учетом требований информационной безопасности.	Уметь: – выбирать и адаптировать существующие ИКТ для прикладных задач в области ИИ с учётом условий эксплуатации и требований безопасности; – оценивать уязвимости информационной системы на базе ИИ и определять необходимые меры защиты (дифференциальная приватность, шифрование, контроль доступа); – применять методы защиты моделей машинного обучения (adversarial training, обнаружение аномалий).
	ОПК-4.3. Владеет навыками применения и оценки информационно-коммуникационных технологий в профессиональной деятельности с соблюдением требований информационной безопасности.	Владеть: – навыками настройки и использования инструментов безопасной разработки и развёртывания систем ИИ (Docker, Kubernetes, API-шлюзы, системы логирования и мониторинга); – методами оценки эффективности применяемых средств защиты информации в системах ИИ; – технологиями интеграции средств защиты в жизненный цикл разработки систем ИИ (DevSecOps для ИИ).

4. Структура и содержание дисциплины

Модуль	Вид учебной работы	Наименование тем занятий	Семестр	Объём, ч	Баллы (max)	Интерактив, ч	Формы текущего контроля
1. Современные ИКТ в системах ИИ	лекция	Тема 1. Обзор современных информационно-коммуникационных технологий для систем искусственного интеллекта. Облачные платформы, контейнеризация, микросервисы.	4	2	—	—	—
	самост.	Изучение лекционного материала, подготовка к лабораторным работам	4	20	—	—	—
	лекция	Тема 2. Архитектуры безопасных систем ИИ: компоненты, точки уязвимости.	4	2	—	—	—
	Пр.	Лабораторная работа №1. Развёртывание защищённого REST API для модели ИИ (Docker, HTTPS, JWT).	4	6	—	—	Отчёт по ЛР (защита)
	самост.	Выполнение заданий по настройке контейнеризации, подготовка отчёта	4	30	—	—	—

Модуль	Вид учебной работы	Наименование тем занятий	Семестр	Объём, ч	Баллы (max)	Интерактив, ч	Формы текущего контроля
2. Угрозы и защита систем ИИ	лекция	Тема 3. Классификация угроз системам ИИ: атаки на этапе обучения (poisoning, backdoor), на этапе вывода (adversarial examples, evasion), атаки на конфиденциальность.	4	2	—	—	—
	Пр.	Лабораторная работа №2. Анализ атак на модель машинного обучения (генерация состязательных примеров, adversarial training).	4	6	—	—	Отчёт по ЛР (защита)
	самост.	Изучение методов генерации состязательных примеров, выполнение кейсов	4	25	—	—	—
	лекция	Тема 4. Методы защиты моделей: дифференциальная приватность, федеративное обучение, шифрование моделей.	4	2	—	—	—
	Пр.	Лабораторная работа №3. Применение дифференциальной приватности (DP-SGD) при обучении модели.	4	6	—	—	Отчёт по ЛР (защита)

Модуль	Вид учебной работы	Наименование тем занятий	Семестр	Объём, ч	Баллы (max)	Интерактив, ч	Формы текущего контроля
	самост.	Реализация DP-SGD, сравнение точности и приватности	4	30	—	—	—
3. Безопасная разработка и мониторинг	лекция	Тема 5. Безопасная разработка и развёртывание систем ИИ: DevSecOps, управление секретами, безопасная настройка контейнеров.	4	2	—	—	—
	Пр.	Лабораторная работа №4. Безопасное логирование и мониторинг ИИ-сервиса (сбор метрик, обнаружение аномалий).	4	6	—	—	Отчёт по ЛР (защита)
	самост.	Настройка систем мониторинга, подготовка отчёта	4	39	—	—	—
	пром. аттест.	Промежуточная аттестация	4	0,35	—	—	—
	контроль	Экзамен	4	35,65			Вопросы к экзамену
	Итого			216	—	—	

5. Образовательные технологии

В рамках изучения дисциплины предусмотрено использование следующих образовательных технологий:

- технология традиционного обучения (лекции, практические занятия);
- интерактивные технологии: учебные дискуссии, разбор кейсов, работа в малых группах;
- проектные технологии: выполнение практических работ, моделирующих реальные задачи бизнеса.

6. Методические указания по освоению дисциплины

6.1 Рекомендации по подготовке к практическим занятиям

Обучающимся следует при подготовке к занятиям использовать не только учебную литературу, но и актуальные источники (статьи, обзоры рынка CRM, документацию систем). В начале занятий задавать преподавателю вопросы по материалу, вызвавшему затруднения, на занятии доводить каждую задачу до окончательного решения, демонстрировать понимание применяемых методов.

При самостоятельном решении задач нужно обосновывать каждый этап, опираясь на теоретические положения курса. Результаты практических работ оформляются в виде отчётов, содержащих описание хода выполнения, использованные инструменты, полученные результаты и выводы.

6.2 Рекомендации по подготовке к зачету

Подготовка к зачету способствует закреплению, углублению и обобщению знаний, а также применению их к решению практических задач. Необходимо ориентировать обучающихся на систематическую подготовку в течение семестра, что позволит использовать время зачётной сессии для систематизации знаний. При подготовке рекомендуется:

- повторить основные понятия, методы и инструменты data-driven маркетинга и CRM;
- разобрать выполненные практические работы;
- ознакомиться с дополнительной литературой по современным CRM-системам и аналитике.

7. Оценочные средства

7.1. Паспорт оценочных средств

Семестр	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
4	ОПК-4	Вопросы к экзамену Отчёты по лабораторным работам №1–4

7.2. Типовые задания или иные материалы, необходимые для текущего контроля

7.2.1. Отчеты по лабораторным работам

(наименование оценочного средства)

Типовые задания для текущего контроля

Лабораторная работа №1. Развёртывание защищённого REST API для модели машинного обучения

Цель работы

Получить практические навыки контейнеризации модели машинного обучения с использованием Docker, а также настройки базовых механизмов безопасности API: HTTPS и JWT-аутентификации.

Задание

1. Разработать простое веб-приложение (на FastAPI или Flask), которое загружает предобученную модель (например, классификатор изображений или регрессор) и возвращает предсказание по POST-запросу.
2. Упаковать приложение в Docker-образ и запустить контейнер.
3. Настроить HTTPS-соединение (использовать самоподписанный сертификат или nginx как reverse proxy).
4. Реализовать JWT-аутентификацию: только авторизованные пользователи могут отправлять запросы к модели.
5. Продемонстрировать работоспособность API через Postman или curl.
6. Подготовить отчёт с описанием выполненных шагов, скриншотами и кодом.

Порядок выполнения

Шаг 1. Подготовка модели

- Используйте любую предобученную модель из scikit-learn, TensorFlow или PyTorch. Пример: модель логистической регрессии на наборе Iris.
- Сохраните модель в файл (.pkl, .h5 или .pt).

Шаг 2. Создание API на FastAPI

```
python
# app.py
from fastapi import FastAPI, Depends, HTTPException, status
from fastapi.security import HTTPBearer, HTTPAuthorizationCredentials
import jwt
import numpy as np
```

```

import pickle

app = FastAPI()
security = HTTPBearer()

SECRET_KEY = "your-secret-key"
ALGORITHM = "HS256"

# Загрузка модели
with open("model.pkl", "rb") as f:
    model = pickle.load(f)

def verify_token(credentials: HTTPAuthorizationCredentials = Depends(security)):
    token = credentials.credentials
    try:
        payload = jwt.decode(token, SECRET_KEY, algorithms=[ALGORITHM])
        return payload
    except jwt.PyJWTError:
        raise HTTPException(status_code=status.HTTP_401_UNAUTHORIZED, detail="Invalid token")

@app.post("/predict")
def predict(features: list, user=Depends(verify_token)):
    prediction = model.predict([features])[0]
    return {"prediction": int(prediction)}

```

Шаг 3. Docker-контейнеризация

- Создайте Dockerfile:


```

dockerfile
FROM python:3.9-slim
WORKDIR /app
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt
COPY . .
CMD ["uvicorn", "app:app", "--host", "0.0.0.0", "--port", "8000"]

```

 - Укажите зависимости в requirements.txt: fastapi, uvicorn, python-jose, pyjwt, numpy, scikit-learn, pickle.
 - Соберите образ: `docker build -t secure-ml-api .`
 - Запустите контейнер: `docker run -p 8000:8000 secure-ml-api`

Шаг 4. Настройка HTTPS

- Сгенерируйте самоподписанный сертификат: `openssl req -x509 -newkey rsa:4096 -keyout key.pem -out cert.pem -days 365 -nodes`
- Измените запуск Uvicorn: `uvicorn app:app --host 0.0.0.0 --port 8000 --ssl-keyfile=key.pem --ssl-certfile=cert.pem`
- Либо используйте Nginx как reverse проху с SSL-терминацией.

Шаг 5. Генерация JWT-токена

- Напишите отдельный скрипт для выдачи токена (эндпоинт /login или вне API).
- Пример получения токена:

```
payload = {"sub": "user123"}; token = jwt.encode(payload, SECRET_KEY,  
algorithm=ALGORITHM)
```

Шаг 6. Тестирование

- В Postman отправьте POST-запрос на `https://localhost:8000/predict` с заголовком `Authorization: Bearer <token>` и телом `{"features": [5.1, 3.5, 1.4, 0.2]}`.
- Убедитесь, что без токена возвращается ошибка 401.

Форма отчёта

Отчёт должен содержать:

1. Титульный лист.
2. Цель работы.
3. Краткие теоретические сведения (контейнеризация, JWT, HTTPS).
4. Листинги кода с пояснениями.
5. Скриншоты: сборка Docker, запуск контейнера, запросы через Postman.
6. Выводы о проделанной работе.

Критерии оценки (максимум 20 баллов)

Показатель	Макс. балл
Корректность создания Docker-образа и запуска контейнера	5
Настройка HTTPS и аутентификации (JWT)	5
Реализация API-методов (предсказание, проверка статуса)	5
Качество отчёта (описание, скриншоты, выводы)	5

Лабораторная работа №2. Анализ атак на модель машинного обучения: генерация состязательных примеров и защита

Цель работы

Изучить механизм состязательных атак (adversarial attacks) на модели глубокого обучения, освоить генерацию состязательных примеров с использованием метода Fast Gradient Sign Method (FGSM) и реализовать базовую защиту – adversarial training.

Задание

1. Загрузить предобученную модель (например, простую свёрточную сеть для MNIST или CIFAR-10) в TensorFlow/Keras или PyTorch.
2. Выбрать несколько тестовых изображений и получить исходные предсказания.
3. Реализовать атаку FGSM: для каждого изображения вычислить градиент функции потерь по входу и добавить возмущение со знаком градиента.
4. Продемонстрировать, как меняется предсказание модели после добавления состязательного возмущения.
5. Реализовать защиту **adversarial training**: дообучить модель на смеси чистых и состязательных примеров.
6. Сравнить точность модели на чистых и состязательных примерах до и после защиты.
7. Визуализировать оригинальные и состязательные изображения.

Порядок выполнения

Шаг 1. Загрузка модели и данных (MNIST)

```
python
import tensorflow as tf
from tensorflow.keras.datasets import mnist

(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train = x_train / 255.0
x_test = x_test / 255.0
model = tf.keras.models.load_model('mnist_model.h5') # предобученная модель
```

Шаг 2. Реализация FGSM

```
python
def fgsm_attack(model, image, label, epsilon=0.1):
    image = tf.convert_to_tensor(image, dtype=tf.float32)
    with tf.GradientTape() as tape:
        tape.watch(image)
        prediction = model(image)
        loss = tf.keras.losses.sparse_categorical_crossentropy(label, prediction)
    gradient = tape.gradient(loss, image)
    perturbation = epsilon * tf.sign(gradient)
    adversarial_image = image + perturbation
    return tf.clip_by_value(adversarial_image, 0, 1).numpy()
```

Шаг 3. Атака на тестовые примеры

- Выбрать несколько изображений, вычислить состязательные варианты.
- Сравнить предсказания исходной модели для чистых и атакованных изображений.

Шаг 4. Adversarial training

- Создать расширенный набор данных: для каждого тренировочного примера добавить его состязательную версию.
- Дообучить модель на этом наборе (несколько эпох).
- Оценить точность на чистых и состязательных тестовых примерах.

Шаг 5. Визуализация

- Отобразить оригинальные изображения и их состязательные варианты (можно разницу усилить для наглядности).

Форма отчёта

1. Цель работы.
2. Теоретическое описание FGSM и adversarial training.
3. Код с комментариями.
4. Графики точности и таблицы сравнения.
5. Визуализация примеров атак.
6. Выводы об эффективности защиты.

Критерии оценки (максимум 25 баллов)

Показатель	Макс. балл
Корректная реализация FGSM	6
Демонстрация изменения предсказаний	5
Реализация adversarial training	6
Сравнительный анализ точности	4
Качество отчёта и визуализации	4

Лабораторная работа №3. Применение дифференциальной приватности при обучении нейронной сети

Цель работы

Освоить метод дифференциальной приватности (DP) для защиты конфиденциальности обучающих данных. Реализовать алгоритм DP-SGD (Differentially Private Stochastic Gradient Descent) с использованием библиотеки TensorFlow Privacy и сравнить качество модели при различных уровнях приватности.

Задание

1. Установить библиотеку tensorflow-privacy.
2. Загрузить набор данных (например, MNIST).
3. Обучить baseline-модель (нейронную сеть) без приватности, зафиксировать точность.
4. Обучить модель с применением DP-SGD, задав параметры: epsilon (уровень приватности), delta, l2_norm_clip, noise_multiplier.
5. Провести серию экспериментов с разными значениями epsilon (например, 0.5, 1.0, 2.0, 5.0).
6. Построить график зависимости точности модели от epsilon.
7. Сделать выводы о компромиссе между приватностью и полезностью модели.

Порядок выполнения

Шаг 1. Установка и импорт

```
bash
pip install tensorflow-privacy
```

Шаг 2. Baseline-модель (без DP)

- Определить простую модель: Sequential с несколькими Dense слоями.
- Обучить на MNIST, достичь accuracy > 98%.

Шаг 3. DP-SGD модель

```
python
from tensorflow_privacy.privacy.optimizers.dp_optimizer_keras import DPKerasSGDOptimizer

optimizer = DPKerasSGDOptimizer(
    l2_norm_clip=1.0,
    noise_multiplier=0.7,
    num_microbatches=256,
    learning_rate=0.15
)

model.compile(optimizer=optimizer, loss='sparse_categorical_crossentropy', metrics=['accuracy'])
model.fit(x_train, y_train, batch_size=256, epochs=10, verbose=1)
```

Шаг 4. Подсчёт epsilon

- Использовать compute_dp_sgd_privacy для расчёта epsilon при заданных параметрах.

Шаг 5. Эксперименты

- Изменять `noise_multiplier` и `l2_norm_clip`, фиксируя итоговое `epsilon` и `accuracy`.

Построить таблицу:

<code>epsilon</code>	accuracy на тесте
0.5	...
1.0	...
2.0	...
5.0	...
без DP	...

Шаг 6. Анализ

- Объяснить, почему с уменьшением `epsilon` падает точность.
- Дать рекомендацию по выбору `epsilon` для практической задачи.

Форма отчёта

1. Цель работы.
2. Теоретические основы дифференциальной приватности (определение, ϵ , δ).
3. Описание DP-SGD.
4. Код экспериментов.
5. График зависимости точности от ϵ .
6. Выводы о применимости метода.

Критерии оценки (максимум 25 баллов)

Показатель	Макс. балл
Корректная установка и использование TensorFlow Privacy	5
Baseline-обучение и DP-обучение	6
Проведение экспериментов с разными ϵ	6
Построение графика и анализ	4
Качество отчёта	4

Лабораторная работа №4. Безопасное логирование и мониторинг ИИ-сервиса

Цель работы

Научиться организовывать безопасное логирование и мониторинг для сервиса машинного обучения, выявлять аномалии в запросах и настраивать алертинг.

Задание

1. Развернуть ранее созданное API (из ЛР №1) или создать новое простое API для предсказаний.
2. Реализовать логирование всех входящих запросов и ответов в структурированном формате (JSON) с обязательным исключением чувствительных данных.
3. Настроить сбор метрик: количество запросов, время ответа, количество ошибок, распределение предсказанных классов.
4. Реализовать механизм обнаружения аномалий в запросах (например, необычно большие значения признаков или слишком частые запросы с одного IP).
5. Настроить отправку уведомлений (email или Telegram) при превышении пороговых значений метрик (например, >10 ошибок в минуту).
6. Создать простой дашборд (можно через Prometheus + Grafana или просто скрипт визуализации).
7. Подготовить отчёт с описанием архитектуры логирования, скриншотами дашборда и примером аномалии.

Порядок выполнения

Шаг 1. Структурированное логирование (Python + logging)

```
python
import logging
import json
from datetime import datetime

logger = logging.getLogger("ml_api")
handler = logging.FileHandler("api.log")
formatter = logging.Formatter("{\"time\": \"%(asctime)s\", \"level\": \"%(levelname)s\", \"message\": %(message)s\"}")
handler.setFormatter(formatter)
logger.addHandler(handler)

# В эндпоинте:
def predict(features):
    start = datetime.now()
    # ... вычисление
    duration = (datetime.now() - start).total_seconds()
    log_entry = {
        "timestamp": start.isoformat(),
        "client_ip": request.client.host,
        "features_hash": hashlib.md5(str(features).encode()).hexdigest(),
        "prediction": prediction,
        "duration": duration
    }
    logger.info(json.dumps(log_entry))
```

Шаг 2. Сбор метрик (Prometheus client)

1. Установить prometheus-client.
2. Определить метрики: requests_total, request_duration_seconds, error_counter.
3. Выставить эндпоинт /metrics для Prometheus.

•

Шаг 3. Обнаружение аномалий

- Реализовать простой детектор: если значение любого признака выходит за пределы 3 сигм (по предварительно вычисленным статистикам), то запись помечается как аномальная и логируется отдельно.
- Детектор частоты запросов: хранить словарь IP → список времен запросов за последние 60 секунд; если частота > N, то отправлять уведомление.

Шаг 4. Уведомления (Telegram bot)

- Создать бота через BotFather, получить токен.
- Написать функцию send_alert(message).
- Вызывать её при обнаружении аномалии или превышении порога.

Шаг 5. Дашборд (Grafana + Prometheus)

- Запустить Prometheus и Grafana (можно в Docker).
- Настроить источник данных Prometheus.
- Создать панель с графиками: RPS, latency, error rate, количество аномалий.

Шаг 6. Тестирование

- Сгенерировать нагрузку (например, с помощью locust или ab).
- Имитировать аномальные запросы (высокие значения признаков, большое количество запросов).
- Проверить, что логируются события и приходят уведомления.

Форма отчёта

1. Цель работы.
2. Описание архитектуры мониторинга и логирования.
3. Код основных компонентов.
4. Скриншоты дашборда Grafana, примеры логов, скриншоты уведомлений.
5. Анализ эффективности обнаружения аномалий.
6. Выводы.

Критерии оценки (максимум 30 баллов)

Показатель	Макс. балл
Реализация структурированного логирования	6
Сбор метрик (Prometheus) и настройка дашборда	8
Реализация детекторов аномалий	6
Настройка уведомлений (Telegram)	5
Качество отчёта и демонстрация работы	5

Все лабораторные работы должны выполняться в среде Python (версия 3.8+). Код и отчёты предоставляются в электронном виде. Защита работ происходит в форме собеседования, где студент объясняет принятые решения и демонстрирует работу системы.

Форма отчета по практическим работам

В отчет по практической работе должны быть включены:

1. титульный лист (оформляется по установленному образцу);
2. цель работы (формулируется в соответствии с заданием);
3. краткие теоретические сведения (основные определения, формулы, понятия, необходимые для выполнения работы);
4. описание хода выполнения работы (последовательность действий, используемые инструменты, расчёты, построенные диаграммы);
5. результаты выполненной работы (полученные данные, скриншоты, таблицы, графики, диаграммы, числовые значения);
6. выводы (краткий анализ полученных результатов, достижение цели работы).

Требования к оформлению

1. Работа выполняется согласно методическим указаниям по выполнению лабораторных работ (прилагаются к РПД).
2. По каждой лабораторной работе создаётся отдельный отчёт.
3. Отчёт оформляется и сдаётся в цифровом виде (форматы PDF или DOCX).
4. Отчёт выполняется на листах формата A4. Допускается два способа оформления:
5. машинописный (шрифт Times New Roman, 14 pt, межстрочный интервал 1,5);
6. рукописный (разборчивым почерком, аккуратно).
7. Каждый новый структурный элемент отчёта (теоретическая часть, описание хода работы, результаты, выводы, приложения) начинается с новой страницы.
8. В заголовках не допускаются переносы слов.
9. Все таблицы, рисунки, диаграммы должны быть выполнены в соответствии с требованиями действующих стандартов:
10. ГОСТ 2.105-95 «Общие требования к текстовым документам»;
11. ГОСТ 7.32-2017 «Отчёт о научно-исследовательской работе». Каждый рисунок и таблица должны иметь подпись и номер (например, *Рисунок 3 – Архитектура API с JWT-аутентификацией*).
12. При использовании программных средств (Python, Docker, TensorFlow, Prometheus, Grafana и др.) к отчёту прилагаются файлы исходного кода, Dockerfile, конфигурационные файлы, а также скриншоты работающих систем. Файлы могут быть переданы отдельным архивом или размещены в репозитории (ссылка указывается в отчёте).

Критерии оценки отчётов по лабораторным работам

Максимальный балл за каждую лабораторную работу входит в общий рейтинг по дисциплине. Оценка производится по следующей шкале:

Лабораторная работа	Максимальный балл
№1 (Docker + JWT)	20
№2 (FGSM + adversarial training)	25
№3 (DP-SGD)	25
№4 (мониторинг)	30

Для лабораторных работ, где предусмотрена программная реализация, обязательным условием получения оценки не ниже 4 баллов является **работоспособность кода** и возможность его запуска преподавателем или демонстрация на защите.

7.3 Вопросы к промежуточной аттестации (экзамену)

Модуль 1. Информационно-коммуникационные технологии в системах ИИ (вопросы 1–15)

1. Опишите современные информационно-коммуникационные технологии, используемые при разработке и эксплуатации систем искусственного интеллекта. Приведите примеры облачных платформ, инструментов контейнеризации и оркестрации.
2. Что такое контейнеризация (Docker) и оркестрация контейнеров (Kubernetes)? Какие задачи они решают в контексте развёртывания ИИ-сервисов?
3. Объясните понятие API-шлюза (API Gateway). Какую роль он выполняет в архитектуре микросервисов ИИ-систем?
4. Какие протоколы обеспечивают безопасную передачу данных между клиентом и сервером? Опишите принцип работы HTTPS/TLS.
5. Что такое «бессерверные вычисления» (serverless)? Приведите примеры использования serverless-архитектур для ИИ-приложений.
6. Какие инструменты и практики используются для автоматизации сборки, тестирования и развёртывания ИИ-моделей (CI/CD)? Назовите основные этапы пайплайна MLOps.
7. Опишите форматы данных и протоколы обмена, применяемые при взаимодействии микросервисов в ИИ-системах. В чем преимущества JSON, gRPC, Kafka?
8. Что такое «вычислительный кластер» и распределённое обучение? Какие технологии (MPI, NCCL, Horovod) используются для параллельного обучения моделей?
9. Какие сервисы предоставляют облачные платформы (AWS SageMaker, Azure ML, Google AI Platform) для машинного обучения? Какие задачи они автоматизируют?
10. Что такое «инференс» (inference) и какие подходы существуют для его масштабирования (batch inference, real-time inference, edge inference)?
11. Опишите инструменты для версионирования данных и моделей (DVC, MLflow, Weights & Biases). Зачем они нужны в MLOps?
12. Что такое «ONNX» и каково его назначение? Как он способствует переносимости моделей между фреймворками?
13. Какие методы оптимизации моделей (квантование, pruning, дистилляция) используются для ускорения инференса и уменьшения размера модели?
14. Что такое «Feature Store» и как он помогает обеспечить консистентность признаков при обучении и инференсе?

15. Опишите архитектуру Lambda (пакетная + потоковая обработка) применительно к пайплайнам обработки данных для ИИ.

Модуль 2. Угрозы и уязвимости систем искусственного интеллекта (вопросы 16–30)

16. Классифицируйте угрозы безопасности систем ИИ по этапам жизненного цикла (сбор данных, обучение, инференс, эксплуатация). Приведите примеры для каждого этапа.

17. Что такое атака «отравление данных» (data poisoning)? Каковы возможные последствия и способы обнаружения?

18. Опишите атаки с закладкой (backdoor attacks). Как злоумышленник может внедрить триггер в модель и как это проявляется?

19. Что такое состязательные примеры (adversarial examples)? Объясните принцип атаки FGSM (Fast Gradient Sign Method).

20. Какие виды атак на этапе инференса существуют (evasion attacks)? Приведите примеры физических состязательных атак (adversarial patches).

21. Что такое атака инверсии модели (model inversion)? Какие данные можно восстановить и какие риски это создаёт для конфиденциальности?

22. Объясните суть атаки определения принадлежности (membership inference). Как злоумышленник может определить, использовалась ли конкретная запись в обучающей выборке?

23. Что такое атака извлечения модели (model extraction)? Какие методы используются для копирования модели через API и как это угрожает интеллектуальной собственности?

24. Опишите атаки на конфиденциальность градиентов (gradient leakage) в федеративном обучении. Как возможно восстановить данные по градиентам?

25. Какие уязвимости могут возникать при использовании сторонних библиотек и предобученных моделей? Что такое атака на цепочку поставок (supply chain attack)?

26. Что такое атака «отказ в обслуживании» (DoS) на ИИ-сервис? Приведите примеры и методы защиты.

27. Какие риски связаны с использованием небезопасных десериализаторов при загрузке моделей? Приведите пример уязвимости.

28. Опишите атаки на системы логирования и мониторинга. Как злоумышленник может скрыть свою активность или вызвать ложные срабатывания?

29. Что такое «атака на этапе предобработки данных»? Как модификация препроцессора может повлиять на работу модели?

30. Какие угрозы существуют для моделей, развёрнутых на периферийных устройствах (edge devices)? В чем особенности атак на on-device модели?

Модуль 3. Методы и средства обеспечения безопасности систем ИИ (вопросы 31–45)

31. Что такое дифференциальная приватность (DP)? Объясните смысл параметров ϵ (epsilon) и δ (delta).

32. Опишите алгоритм DP-SGD. Как добавление шума в градиенты обеспечивает защиту конфиденциальности?

33. В чем заключается компромисс между приватностью и полезностью модели при использовании дифференциальной приватности? Приведите примеры.

34. Что такое федеративное обучение (Federated Learning)? Как оно помогает защитить данные пользователей?

35. Какие методы защиты от состязательных атак существуют? Опишите adversarial training и defensive distillation.

36. Что такое «обнаружение состязательных примеров» (adversarial detection)? Какие подходы используются (статистические, на основе отдельной модели)?
37. Как работает гомоморфное шифрование (homomorphic encryption) и где оно может применяться в ИИ-системах?
38. Какие методы используются для защиты от атак извлечения модели (model extraction)? Опишите rate limiting, обфускацию, watermarking.
39. Что такое «безопасное многостороннее вычисление» (secure multi-party computation) и как оно может применяться для совместного обучения без раскрытия данных?
40. Какие механизмы аутентификации и авторизации (JWT, OAuth 2.0, API-ключи) применяются для защиты API моделей?
41. Опишите методы защиты от атак инверсии модели. Как ограничение детализации выходов и добавление шума помогают предотвратить восстановление данных?
42. Что такое «политика минимальных привилегий» (least privilege) в контексте ИИ-систем? Как она реализуется на уровне инфраструктуры и доступа к данным?
43. Какие методы используются для защиты целостности модели (контрольные суммы, цифровые подписи, проверка при загрузке)?
44. Что такое «оценка воздействия на защиту данных» (DPIA) и зачем она проводится для ИИ-систем, обрабатывающих персональные данные?
45. Опишите нормативно-правовые требования к обработке персональных данных в РФ (ФЗ-152) и их влияние на проектирование защищённых ИИ-систем.

Модуль 4. Безопасная разработка, мониторинг и эксплуатация (вопросы 46–57)

46. Что такое DevSecOps? Как интегрируются практики безопасности в CI/CD-конвейер для ИИ-приложений?
47. Какие инструменты статического и динамического анализа кода (SAST/DAST) используются для выявления уязвимостей в коде ИИ-сервисов? Приведите примеры.
48. Что такое «инфраструктура как код» (IaC) и как она помогает обеспечить безопасность конфигураций (сканирование шаблонов Terraform, CloudFormation)?
49. Как организовать безопасное хранение и управление секретами (пароли, ключи API) в контейнерных средах (Kubernetes Secrets, HashiCorp Vault)?
50. Что такое SBOM (Software Bill of Materials) и почему он важен для отслеживания уязвимостей в зависимостях ИИ-проектов?
51. Опишите подходы к мониторингу безопасности ИИ-систем в runtime (Falco, системные детекторы аномалий, обнаружение аномальных запросов).
52. Как настроить структурированное логирование для ИИ-сервиса? Какие данные должны логироваться, а какие – исключаться для защиты конфиденциальности?
53. Какие метрики безопасности следует отслеживать (количество ошибок аутентификации, частота запросов, распределение предсказаний) и как настроить алертинг?
54. Что такое «анализ дрейфа модели» (model drift) и «дрейфа данных» (data drift)? Как мониторинг этих явлений помогает выявлять аномалии и потенциальные атаки?
55. Опишите процесс аудита безопасности модели перед её внедрением. Какие проверки должны быть проведены?
56. Какие практики безопасной разработки (безопасное кодирование, peer review, threat modeling) применимы к ИИ-проектам?
57. Что такое «красная команда» (Red Team) для ИИ-систем? Как проводятся состязательные тесты на проникновение (adversarial testing) для оценки защищённости?

Практические кейсы для экзамена

На экзамене студенту предлагается решить один из приведённых кейсов (или аналогичный), демонстрируя умение применять полученные знания для анализа ситуации, выбора мер защиты и обоснования решений.

Кейс 1. Компания разрабатывает систему распознавания лиц для контроля доступа. При тестировании обнаружено, что система ошибается, если на лицо наклеить специальный паттерн (очки с определённым рисунком). Как называется такая атака? Предложите способы защиты, включая как архитектурные меры, так и методы обучения модели.

Кейс 2. Вам необходимо развернуть модель классификации медицинских изображений в больничной сети, где действуют строгие требования к конфиденциальности (ПДн). Опишите архитектуру развёртывания (с использованием контейнеризации, шифрования, контроля доступа, логирования) и обоснуйте выбор технологий. Какие нормативные требования необходимо учесть?

Кейс 3. Вы обнаружили, что кто-то отправляет в ваш API предсказаний тысячи запросов с незначительно изменёнными изображениями, копируя ответы. Как называется эта атака? Какие меры вы предпримете немедленно, а какие – в долгосрочной перспективе для предотвращения подобных инцидентов?

Кейс 4. При обучении модели для банковского скоринга вы использовали дифференциальную приватность. После развёртывания модель показывает точность 92% при $\epsilon=2.0$. Руководство требует повысить точность до 95%, снизив приватность до $\epsilon=5.0$. Какие риски это несёт? Обоснуйте свой ответ и предложите компромиссное решение.

Кейс 5. В федеративной системе обучения мобильных устройств один из участников начал присылать градиенты, которые ухудшают глобальную модель. Как называется эта атака? Какие методы агрегации (robust aggregation) и детекции вредоносных участников можно применить для защиты?

Кейс 6. Ваша компания использует предобученную модель NLP из открытого репозитория. В одном из обновлений библиотеки-зависимости обнаружена критическая уязвимость. Опишите шаги по реагированию на инцидент: как выявить затронутые системы, провести анализ воздействия, устранить уязвимость и предотвратить повторение. Какие инструменты помогут в этом процессе?

Кейс 7. При мониторинге ИИ-сервиса вы заметили резкое увеличение доли запросов, содержащих необычно большие значения числовых признаков (выходящие за пределы 5 сигм от обучающего распределения). Что может быть причиной? Как вы будете расследовать это событие и какие меры примете для защиты?

7.3.2. Критерии и нормы оценки

Процедура оценивания по экзаменационным билетам

Если экзамен проводится в устной или письменной форме с использованием экзаменационных билетов, каждый билет содержит:

- **два теоретических вопроса** (из перечня, приведённого в п. 7.3);
- **один практический кейс** (из перечня кейсов).

Время на подготовку – **35 минут**. После подготовки экзаменуемый последовательно излагает ответы. Преподаватель может задавать уточняющие и дополнительные вопросы.

Требования к ответу:

1. Ответ должен быть научным, логически стройным и опираться на соответствующие теоретические положения, концепции, а также на практический опыт выполнения лабораторных работ.
2. Необходимо строить ответ в единстве теории и практики, подкрепляя теоретические положения примерами.
3. При ответе на теоретические вопросы следует чётко формулировать определения, классификации, перечислять методы и инструменты, объяснять принципы их работы.
4. При решении практического кейса требуется:
 - определить суть проблемы и возможные причины;
 - предложить пошаговый план решения;
 - обосновать выбор конкретных методов, моделей или инструментов;
 - оценить эффективность предлагаемых мер.

Критерии оценки ответа по билетам:

Оценка	Критерии
«отлично»	Обучающийся полностью раскрыл содержание всех вопросов билета: даны исчерпывающие, аргументированные ответы, демонстрирующие глубокое понимание материала. Практический кейс решён верно, предложены обоснованные меры, использованы профессиональные термины. Ответ логичен, грамотен, структурирован. На дополнительные вопросы даны правильные ответы.
«хорошо»	Обучающийся полностью раскрыл содержание всех вопросов билета, но допустил незначительные неточности или недостаточно полно аргументировал отдельные положения. Практический кейс решён верно, но предложенные меры недостаточно детализированы или обоснованы. На дополнительные вопросы ответил правильно, но с некоторыми затруднениями.
«удовлетворительно»	Обучающийся раскрыл содержание вопросов билета в минимально необходимом объёме, допустил отдельные ошибки, которые исправил после наводящих вопросов. Практический кейс решён частично или с ошибками, но основные подходы определены верно. На дополнительные вопросы ответил неуверенно.
«неудовлетворительно»	Обучающийся не раскрыл содержание вопросов билета, допустил принципиальные ошибки, не решил практический кейс или предложил неверные решения. Не может ответить на дополнительные вопросы.

Обучающийся не сдавший экзамен направляется на пересдачу в установленном порядке.

8. Учебно-методическое и информационное обеспечение дисциплины

8.1. Обязательная литература

№ п/п	Авторы, составители	Заглавие (заголовок)	Тип (учебник, учебное пособие, учебно-методическое пособие, практикум, др.)	Год издания	Количество в научной библиотеке / Наименование ЭБС
1	Сазонов, С. Н.	Системы искусственного интеллекта : учебное пособие / С. Н. Сазонов. — Ульяновск : Ульяновский государственный технический университет, 2023. — 84 с. — ISBN 978-5-9795-2352-1.	Учебное пособие	2023	ЭБС “IPRbooks”
2	Затонский, А. В.	Информационные технологии: разработка информационных моделей и систем : учебное пособие / А.В. Затонский. — Москва : РИОР : ИНФРА-М, 2023. — 344 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-369-01183-6.	Учебное пособие	2023	ЭБС «Znanium»
3	Ложников П.С., Самотуга А.Е., Жумажанова С.С., Сулавко А.Е.	Безопасность систем искусственного интеллекта. Ч.2. Доверенный искусственный интеллект : учебное пособие / П. С. Ложников, А. Е. Самотуга, С. С. Жумажанова, А. Е. Сулавко. — Омск : Омский государственный технический университет, 2023. — 74 с. — ISBN 978-5-8149-3614-1, 978-5-8149-3731-5 (ч.2)	Учебное пособие	2023	ЭБС “IPRbooks”

8.2. Дополнительная литература

№ п/п	Авторы, составители	Заглавие (заголовок)	Тип (учебник, учебное пособие, учебно- методическое пособие, практикум, др.)	Год издания	Количество в научной библиотеке / Наименование ЭБС
4	Менисов, А. Б.	Технологии искусственного интеллекта и кибербезопасность : монография / А. Б. Менисов. — Москва : Ай Пи Ар Медиа, 2022. — 133 с. — ISBN 978-5-4497-1788-7.	Монография	2022	ЭБС “IPRbooks”
5	Блюмин, А. М.	Информационный менеджмент: автоматизация информационных технологий и систем управления : учебник / А. М. Блюмин. - Москва : Издательско-торговая корпорация «Дашков и К°», 2024. - 378 с. - ISBN 978-5-394-05487-7.	Учебник	2024	ЭБС “IPRbooks”

8.3. Перечень профессиональных баз данных и информационных справочных систем

№ пп	Наименование	Ссылка
1	ЭБС «Лань»	ЭБС Лань
2	ЭБС "ZNANIUM.COM"	Электронно-библиотечная система Znanium
3	ЭБС “IPRbooks”	IPR SMART / Главная
4	OWASP Machine Learning Security Top 10 (рекомендации по безопасности ИИ)	https://owasp.org/www-project-machine-learning-security-top-10/
5	Adversarial Robustness Toolbox (ART) – библиотека и документация IBM	https://github.com/IBM/adversarial-robustness-toolbox
6	TensorFlow Privacy (документация и примеры)	https://www.tensorflow.org/responsible_ai/privacy
7	OpenML (открытая платформа для датасетов и экспериментов по машинному обучению)	https://openml.org/
8	Kaggle (датасеты, ноутбуки, соревнования)	https://www.kaggle.com/

8.4 Перечень программного обеспечения

№ п/п	Наименование ПО	Реквизиты договора (дата, номер, срок действия)
1	Adversarial Robustness Toolbox (ART)	Свободное ПО (лицензия MIT)
2	Docker Desktop	Бесплатная версия (Personal edition)
3	Minikube (локальный Kubernetes)	Свободное ПО (лицензия Apache 2.0)
4	Postman (тестирование API)	Бесплатная версия (Basic)

№ п/п	Наименование ПО	Реквизиты договора (дата, номер, срок действия)
5	Prometheus (система мониторинга)	Свободное ПО (лицензия Apache 2.0)
6	Grafana (визуализация метрик)	Свободное ПО (лицензия Apache 2.0)
7	Git (система контроля версий)	Свободное ПО (лицензия GPL)
8	Jupyter Notebook / JupyterLab	Свободное ПО (лицензия BSD)

8.5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

№ п/п	Наименование оборудованных учебных кабинетов, лабораторий, мастерских и др. объектов для проведения практических и лабораторных занятий, помещений для самостоятельной работы обучающихся (номер аудитории)	Перечень основного оборудования
1	Компьютерный класс. Учебная аудитория для проведения занятий лекционного типа. Учебная аудитория для проведения занятий семинарского типа. Учебная аудитория для проведения лабораторных работ. Учебная аудитория для курсового проектирования (выполнения курсовых работ). Учебная аудитория для проведения групповых и индивидуальных консультаций Учебная аудитория для проведения занятий текущего контроля и промежуточной аттестации. (УЛК-408)	Компьютер, проектор Acer P1303W., стол преподавательский, стол ученический, стол компьютерный, стул, доска аудиторная (маркерная).
2	Компьютерный класс. Помещение для самостоятельной работы. Учебная аудитория для проведения занятий семинарского типа. Учебная аудитория для курсового проектирования (выполнения курсовых работ). Учебная аудитория для проведения групповых и индивидуальных консультаций. Учебная аудитория для проведения занятий текущего контроля и промежуточной аттестации (Г-401)	Столы ученические, стулья ученические, ПК с выходом в сеть Интернет