

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ



УТВЕРЖДАЮ
Декан РТФ

УТВЕРЖДАЮ /А.Н. Дедов/
(Ф.И.О. декана (директора института))

13.02.2025 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Б.1.2.10 Специализированные программно-аппаратные отраслевые комплексы

(код и наименование дисциплины по учебному плану)

Направление подготовки
(специальность)

09.03.02 Информационные системы и технологии

Квалификация выпускника

Бакалавр

(бакалавр/магистр/специалист)

Направленность

Интеллектуальные информационные системы и
технологии

Курс 3, 4
Семестр 6, 7

Распределение учебного времени

Трудоемкость по учебному плану	252 / 7	часов/зачетных единиц
Лекции	32	часов
Лабораторные работы	-	часов
Практические занятия	64	часов
Иная контактная работа	-	часов
Всего контактной работы (без учета экз.)	96	часов
Контактная работа по экзамену	6	часов
Курсовой проект (работа)	-	семестр
Самостоятельная работа обучающихся (без учета экз.)	120	часов
Самостоятельная работа по подготовке к экзамену	30	часов
Экзамен	6	семестр
Зачет	-	семестр
БРК, ДЗ	7	семестр

(год)

Программа составлена в соответствии с требованиями ФГОС ВО направления подготовки (специальности) 09.03.02 Информационные системы и технологии

Программу составили:

заведующий кафедрой	ИТРС	СОГЛАСОВАНО	Н.В. Рябова
(должность)	(кафедра)		(И.О. Фамилия)
доцент	ИТРС	СОГЛАСОВАНО	А.А. Кислицын
(должность)	(кафедра)		(И.О. Фамилия)

РАССМОТРЕНА и ОДОБРЕНА на заседании кафедры, за которой закреплена дисциплина
Кафедра информационных технологий, радиотехники и связи

(наименование кафедры)			
13.01.2025	протокол №	2	
(дата)			
Заведующий кафедрой	СОГЛАСОВАНО	Н.В. Рябова	
		(И.О. Фамилия)	

Рабочая программа СОГЛАСОВАНА с факультетом (институтом), выпускающей(ими)
кафедрой(ами).

СООТВЕТСТВУЕТ действующей ОП.

Заведующий кафедрой	СОГЛАСОВАНО	Н.В. Рябова
		(И.О. Фамилия)

Председатель методической комиссии факультета (института), в который входит
выпускающая кафедра

СОГЛАСОВАНО	А.Н. Дедов
	(И.О. Фамилия)

Эксперт(ы): Дудин Александр Николаевич , Исполнительный директор ООО «Трэвел Лайн
Системс»

Рабочая программа проверена и зарегистрирована в УМЦ 17.02.2025 г.

Специалист учебно-методического центра СОГЛАСОВАНО /И.Р. Валиева/

Раздел 1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью освоения дисциплины является достижение планируемых результатов обучения, соответствующих установленным в ОПОП индикаторам достижения компетенций:

Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Результаты обучения
1. ПК-2 Способность выполнять работы по обслуживанию программными и программно-аппаратными средствами сетей и инфокоммуникаций	ПК-2.1. Знать общие принципы функционирования аппаратных, программных и программно-аппаратных средств администрируемой сети, архитектуры аппаратных, программных и программно-аппаратных средств администрируемой сети, а также общие основы решения практических задач по работе с установленной БД	знания: Знает: общие принципы функционирования аппаратных, программных и программно-аппаратных средств инфокоммуникационных сетей с компонентами машинного обучения. Понимает архитектуры распределённых ML-систем в телекоммуникационной инфраструктуре, включая edge computing и fog computing для обработки данных сетевого трафика. умения: Умеет применять методы машинного обучения для мониторинга и оптимизации работы сетевого оборудования. навыки:
	ПК-2.2. Уметь конфигурировать периферийные и абонентские программно-аппаратные устройства, а также предотвращать основные угрозы безопасности БД	знания: умения: Умеет: конфигурировать периферийные и абонентские программно-аппаратные устройства для сбора данных и развёртывания ML-моделей в инфокоммуникационных системах. Применять методы машинного обучения для предотвращения основных угроз безопасности в сетях и БД, включая обнаружение вторжений, аномального трафика и DDoS-атак. навыки:
	ПК-2.3. Иметь навыки установки и настройки программного обеспечения, регламенты проведения профилактических работ на администрируемой инфокоммуникационной системе а также выявления угроз безопасности на уровне БД	знания: умения: навыки: Владеет: навыками установки и настройки программного обеспечения для ML-систем в инфокоммуникационной инфраструктуре, включая фреймворки для обработки потоковых данных и обучения моделей в реальном времени. Знает регламенты проведения профилактических работ на администрируемой инфокоммуникационной системе с интеграцией ML-компонентов. Владеет методами выявления угроз безопасности на уровне БД и сетевой инфраструктуры с использованием алгоритмов машинного обучения для детекции аномалий.

2. ПК-3 Способность проводить анализ требований к программному обеспечению, выполнять работы по проектированию программного обеспечения	ПК-3.1. Знать возможности современных и перспективных средств разработки программных продуктов, технических средств; методологии разработки программного обеспечения и технологии программирования; теоретические основы разработки средств реализации информационных технологий.	знания: Знает: возможности современных и перспективных средств разработки ML-систем и нейронных сетей на языке Python, включая фреймворки (PyTorch, TensorFlow, JAX). Понимает методологии разработки программного обеспечения для систем искусственного интеллекта и технологии программирования нейронных сетей. Знает теоретические основы разработки средств реализации информационных технологий на базе глубокого обучения. Владеет знаниями об архитектурах специализированных программно-аппаратных комплексов для обучения и инференса нейронных сетей (GPU, TPU, специализированные ускорители). умения: навыки:
	ПК-3.2. Уметь проводить анализ исполнения требований; вырабатывать варианты реализации требований; проводить оценку и обоснование рекомендуемых решений; осуществлять коммуникации с заинтересованными сторонами; разрабатывать и реализовывать методические, информационные, математические, алгоритмические, технические и программные средства информационных технологий.	знания: умения: Умеет: проводить анализ исполнения требований к ML-системам и нейросетевым решениям, вырабатывать варианты их реализации с учётом ограничений производительности и ресурсов. Проводить оценку и обоснование рекомендуемых архитектурных решений для систем ИИ. Осуществлять коммуникации с заинтересованными сторонами по техническим аспектам разработки нейронных сетей. Разрабатывать и реализовывать методические, информационные, математические и алгоритмические средства на Python для создания систем искусственного интеллекта. Программировать и оптимизировать нейронные сети под специализированные программно-аппаратные комплексы. навыки:

	ПК-3.3. Иметь навыки анализа возможностей реализации требований к программному обеспечению; оценки времени и трудоемкости реализации требований к программному обеспечению; Согласования требований к программному обеспечению с заинтересованными сторонами; оценки и согласование сроков выполнения поставленных задач; разработки средств реализации информационных	знания: умения: навыки: Владеет: навыками анализа возможностей реализации требований к программному обеспечению систем ИИ и нейронных сетей на Python. Способен оценивать время и трудоёмкость реализации ML-проектов с учётом сложности архитектур нейронных сетей и объёмов данных. Владеет методами согласования требований к программному обеспечению ИИ-систем с заинтересованными сторонами. Умеет оценивать и согласовывать сроки выполнения задач разработки нейронных сетей.
3. LC-5 Способен применять и (или) проектировать различные инструменты и инженерные практики промышленной разработки систем искусственного интеллекта, развертывания и сопровождения моделей	LC-5.1. Осуществляет выбор инструментов и инженерных практик промышленной разработки систем ИИ, развертывания и сопровождения моделей машинного обучения в продуктивной среде	знания: Экспертный: разрабатывает специализированные инструменты и инженерные практики промышленной разработки систем ИИ, развертывания и сопровождения моделей машинного обучения в рабочей версии ПО Знает: современные инструменты, архитектуры и инженерные практики промышленной разработки, развертывания и сопровождения систем ИИ и моделей машинного обучения умения: Умеет: проектировать и разрабатывать специализированные инструменты и практики MLOps для промышленной эксплуатации ИИ-систем в продуктивной среде навыки: Владеет: навыками внедрения, автоматизации и сопровождения рабочих версий ML-моделей с обеспечением надежности, масштабируемости и жизненного цикла систем искусственного интеллекта.

машинного обучения в продуктивной среде	LC-5.2. Осуществляет выбор инструментов и инженерных практик по управлению данными с необходимым уровнем доступа, контроля качества, резервирования и скоростью выполнения запросов	знания: Экспертный: кастомизирует и разрабатывает специализированные инструменты управления данными Знает: архитектуры и современные инструменты управления данными, включая механизмы доступа, контроля качества, резервирования и оптимизации производительности умения: Умеет: кастомизировать и разрабатывать специализированные инструменты и инженерные практики управления данными с учётом требований ИИ-систем навыки: Владеет: навыками проектирования и внедрения решений по управлению данными, обеспечивающих надежность, безопасность и высокую скорость выполнения запросов в промышленной среде
4. ML-2 Способен применять фундаментальные принципы и методы машинного обучения включая подготовку данных оценку качества моделей и работу с признаками	ML-2.3. Решает проблемы несбалансированных данных и оценивает качество моделей	знания: Экспертный: применяет продвинутые методы работы с несбалансированными данными (SMOTE weighted learning). Настраивает кастомные метрики и функции потерь. Проводит статистический анализ значимости результатов Знает: продвинутые типы и постановки задач машинного обучения, современные методы их решения и критерии оценки качества; умения: Умеет: проектировать и реализовывать комплексные решения машинного обучения для нестандартных задач, включая разработку сквозных пайплайнов, оптимизацию моделей и интерпретацию результатов навыки: Владеет: навыками системной разработки, настройки и внедрения ML-решений, обеспечивающих воспроизводимость, масштабируемость и обоснованность получаемых выводов

Раздел 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина относится к части, формируемой участниками образовательных отношений ОПОП.

Дисциплина является обязательной

Для продолжения формирования заявленных компетенций необходимы знания предшествующих дисциплин: Инфраструктура для машинного обучения (LC-5), Контейнеризация и оркестрация в машинном обучении (LC-5), Основы технологий искусственного интеллекта (ML-2), Введение в машинное обучение и интеллектуальные системы (ML-2); практик: Учебная практика. Ознакомительная практика (ПК-2), Производственная практика. Технологическая (проектно-технологическая) практика (распределенная) (ПК-2), Производственная практика. Технологическая (проектно-технологическая) практика (ПК-2), Производственная практика. Научно-исследовательская работа (ПК-2), Учебная практика. Ознакомительная практика (ПК-3), Производственная практика. Технологическая (проектно-технологическая) практика (распределенная) (ПК-3), Производственная практика. Технологическая (проектно-технологическая) практика (ПК-

3), Производственная практика. Научно-исследовательская работа (ПК-3), Учебная практика. Ознакомительная практика (LC-5), Производственная практика. Технологическая (проектно-технологическая) практика (рассредоточенная) (LC-5), Производственная практика. Технологическая (проектно-технологическая) практика (LC-5), Производственная практика. Научно-исследовательская работа (LC-5), Учебная практика. Ознакомительная практика (ML-2), Производственная практика. Технологическая (проектно-технологическая) практика (рассредоточенная) (ML-2), Производственная практика. Технологическая (проектно-технологическая) практика (ML-2), Производственная практика. Научно-исследовательская работа (ML-2)

Изучаемая дисциплина является основой для продолжения формирования указанных компетенций в следующих дисциплинах: ML-инженерия в облачных платформах (ML-2), Промышленные системы машинного обучения (ML-2); практиках: Преддипломная практика (ПК-2), Преддипломная практика (ПК-3), Преддипломная практика (LC-5), Преддипломная практика (ML-2); государственной итоговой аттестации в форме: Выполнение и защита выпускной квалификационной работы (ПК-2), Выполнение и защита выпускной квалификационной работы (ПК-3), Выполнение и защита выпускной квалификационной работы (LC-5), Выполнение и защита выпускной квалификационной работы (ML-2)

Раздел 3. ОПИСАНИЕ ОБРАЗОВАТЕЛЬНЫХ ТЕХНОЛОГИЙ

Для формирования заявленных компетенций используются методологические технологии, реализующие деятельностный, личностно-ориентированный, практико-ориентированный подходы.

Основными стратегическими технологиями являются: лекционные занятия, практические занятия

На достижение конкретных целей обучения направлены применяемые тактические технологии: задания, классическая лекция, лекция-провокация, мини-проекты, информационные технологии обучения (применение специализированного ПО, облачных сервисов, инструментов ml ops), работа с кейсами, лекция с обсуждением новых научных достижений в сфере ИИ по результатам конференции и публикаций и др.), семинары с ит-практиками, конкурсы, хакатоны, кейс-чемпионаты, мастер-классы, командные проектные задания, контрольно-измерительные тесты на электронном курсе образовательного портала пгту на платформе lms moodle (elearning.volgatech.net), интегрированная федеральная платформа гис спос online.edu.ru, открытый образовательный портал (mooped.net), ресурсы открытых образовательных платформ (яндекс-практикум, сберуниверситет)

Раздел 4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

6 семестр

Виды и темы занятий	Количество часов	Формируемые компетенции
Введение и аппаратные основы	48	LC-5, ML-2, ПК-2, ПК-3
Лекция. Лекция 1. Введение в специализированные программно-аппаратные комплексы Роль программно-аппаратных систем в ИИ, понятия «комплекс» и «отрасль», обзор рынка (GPU, TPU, FPGA, ASIC), отрасли-потребители.	2	
Лекция. Лекция 2. Архитектуры ускорителей Устройство и отличия GPU, TPU, FPGA и ASIC; вычислительные ядра, память, пропускная способность, энергопотребление, отечественные платформы.	2	

Практическое занятие. Практическая работа 1. Настройка среды и проверка оборудования Установка драйверов (CUDA, cuDNN, ROCm) и библиотек, настройка виртуальных окружений, тестовые вычисления на GPU/TPU/FPGA.	4	
Практическое занятие. Практическая работа 2. Программирование на CUDA/OpenCL Реализация простых параллельных алгоритмов (например, суммирование векторов), управление памятью, измерение ускорения.	4	
Практическое занятие. Практическая работа 3. Построение потоковой pipeline Симуляция датчиков, отправка данных в Kafka, обработка в Spark/Flink, базовый feature engineering.	4	
Задания для самостоятельной работы, в том числе выполнение Выполнение курсового проекта: «Разработка специализированного программно-аппаратного комплекса мониторинга и интеллектуальной диагностики радиоканала на основе методов машинного обучения» Чтение материалов из «белого списка» и конференций A*/A, изучение документации TensorFlow/PyTorch, решение задач на Kaggle/Stepik, выполнение индивидуального мини-проекта по выбранной промышленной задаче, работа с электронными курсами (moored.net, Яндекс-Практикум, СберУниверситет).	32	
Фреймворки, контейнеризация и CI/CD	38	LC-5, ML-2
Лекция. Лекция 3. Основы параллельного и распределённого программирования Модели параллелизма (data-parallel, model-parallel), основы CUDA/OpenCL, распределённые вычисления (MPI), паттерны оптимизации.	2	
Лекция. Лекция 4. Data pipelines и источники данных Сенсоры и IoT-устройства; потоковые шины (Kafka, Pulsar); распределённая обработка (Spark, Flink); подготовка данных для ML.	2	
Лекция. Лекция 5. ML-фреймворки и базовые MLOps Обзор фреймворков (TensorFlow, PyTorch, ONNX, XLA), инструменты ускорения инференса (TensorRT, OpenVINO); основы контейнеризации (Docker) и CI/CD.	2	
Задания для самостоятельной работы, в том числе выполнение Выполнение курсового проекта: «Разработка специализированного программно-аппаратного комплекса мониторинга и интеллектуальной диагностики радиоканала на основе методов машинного обучения» Чтение материалов из «белого списка» и конференций A*/A, изучение документации TensorFlow/PyTorch, решение задач на Kaggle/Stepik, выполнение индивидуального мини-проекта по выбранной промышленной задаче, работа с электронными курсами (moored.net, Яндекс-Практикум, СберУниверситет). Мини-хакатон: Командная работа над небольшим промышленным кейсом: разработка прототипа решения с использованием специализированного комплекса.	32	
Работа с данными, Edge AI и безопасность	58	LC-5, ML-2

Лекция. Лекция 6. Работа с признаками и несбалансированными данными Методы feature engineering, techniques для несбалансированных данных (oversampling, undersampling), основы SMOTE; метрики качества (precision/recall, ROC AUC).	2
Лекция. Лекция 7. Edge AI и IoT Специализированные платы (Jetson, Movidius), микроконтроллеры с нейроускорителями; особенности развёртывания на «крае» сети, ограниченные ресурсы	2
Лекция. Лекция 8. Безопасность и надёжность Угрозы (перехват данных, DDoS), доверенная среда, резервирование и отказоустойчивость; основы безопасной разработки.	2
Практическое занятие. Практическая работа 4. Работа с несбалансированными данными Применение методов oversampling/undersampling, реализация SMOTE, настройка взвешенных функций потерь, оценка влияния на метрики.	4
Практическое занятие. Практическая работа 5. Feature engineering Создание новых признаков, кодирование категориальных переменных, подбор масштабирования; построение Pipelined Transformer в scikit?learn.	4
Практическое занятие. Практическая работа 6. Базовое обучение с подкреплением Реализация Q?learning или DQN в простом симуляторе; настройка коэффициента дисконтирования, проверка политики	4
Практическое занятие. Практическая работа 7. Развёртывание модели на Edge Портирование модели на Jetson Nano/Raspberry Pi, квантование до 8?bit, измерение производительности и энергопотребления.	4
Практическое занятие. Практическая работа 8. Профилирование производительности Использование Nsight Systems, Nsight Compute, TensorBoard для анализа времени и использования GPU; поиск узких мест.	4
Задания для самостоятельной работы, в том числе выполнение РГР Выполнение РГР: «Разработка специализированного программно-аппаратного комплекса мониторинга и интеллектуальной диагностики радиоканала на основе методов машинного обучения» Чтение материалов из «белого списка» и конференций А*/А, изучение документации TensorFlow/PyTorch, решение задач на Kaggle/Stepik, выполнение индивидуального мини?проекта по выбранной промышленной задаче, работа с электронными курсами (moored.net, ЯндексПрактикум, СберУниверситет). Лаборатория по безопасности: Настройка шифрования	32
Иная контактная работа:	0
Подготовка к экзамену	30
Проведение экзамена	6

7 семестр

Виды и темы занятий	Количество часов	Формируемые компетенции
Оптимизация и управление данными	34	LC-5, ML-2, ПК-2, ПК-3
Лекция. Лекция 9. Распределённое и гиперпараметрическое оптимизирование Многонодовое обучение, PyTorch DDP, Horovod; методы поиска гиперпараметров (Grid Search, Random Search, Bayesian Optimization).	2	
Лекция. Лекция 10. Продвинутое методы для несбалансированных данных Варианты SMOTE (Borderline, ADASYN), Focal loss, кастомные функции потерь; статистический анализ значимости результатов.	2	
Лекция. Лекция 11. DataOps и управление данными Практики непрерывной интеграции данных; управление версиями данных; инструменты для DataOps и Feature Store.	2	
Практическое занятие. Практическая работа 9. Distributed Training Развёртывание PyTorch DDP/Horovod на нескольких узлах, измерение масштабируемости, устранение узких мест.	4	
Практическое занятие. Практическая работа 10. Гиперпараметрическая оптимизация Реализация Random Search и Bayesian Optimization; сравнение методов; автоматизация с помощью Optuna или Ray Tune.	4	
Практическое занятие. Практическая работа 11. Advanced SMOTE и кастомные метрики Применение ADASYN, Borderline SMOTE; реализация Focal loss; настройка кастомных метрик и функций потерь; статистическое тестирование результатов.	4	
Практическое занятие. Практическая работа 12. DataOps pipeline Настройка управления версиями данных (DVC или LakeFS), создание feature store, автоматическое обновление данных и мониторинг качества.	4	
Задания для самостоятельной работы, в том числе выполнение Изучение стандартов и ГОСТов, отработка навыков гиперпараметрического поиска и DataOps, чтение передовых статей по специализированным архитектурам и федеративному обучению. Изучение кейса от индустриального партнёра: Работа над реальной задачей (напр., предиктивное обслуживание оборудования): разработка полного pipeline, общение с представителями партнёра.	12	
Специализированное проектирование и генеративные модели	22	LC-5, ML-2
Лекция. Лекция 12. Специализированное аппаратное проектирование Основы высокоуровневого синтеза (HLS), проектирование ускорителей для CNN/RNN, критерии выбора FPGA/ASIC.	2	
Лекция. Лекция 13. Генеративные модели и специализированные архитектуры Вариационные автокодировщики, диффузионные модели,	2	

RAG/transformer?архитектуры; оптимизация для специализированных платформ.		
Практическое занятие. Практическая работа 13. Разработка специализированного инструмента Создание утилиты для сборки и развертывания модели на выбранной платформе (например, скрипт для упаковки модели, генерации конфигураций).	4	
Практическое занятие. Практическая работа 14. FPGA/HLS лаборатория Проектирование простого ускорителя CNN с использованием HLS4ML или Vitis HLS; синтез, оценка задержки и энергопотребления.	4	
Практическое занятие. Практическая работа 15. Генеративные модели на специализированных устройствах Реализация вариационного автоэнкодера или диффузионной модели; оптимизация памяти; портирование модели на ускоритель.	4	
Задания для самостоятельной работы, в том числе выполнение Подготовка обзора статьи по темам HLS, федеративного обучения или генеративных моделей, обсуждение новых трендов. Развёртывание Kubeflow, ведение экспериментов, регистрация моделей; настройка мониторинга и алертинга.	6	
Федеративное обучение, безопасность и стандарты	16	LC-5, ML-2
Лекция. Лекция 14. Federated AI и безопасность Принципы федеративного и распределённого обучения, конфиденциальность данных, противодействие атакующим; private AI.	2	
Лекция. Лекция 15. Изучение кейса и стандарты отрасли Обзор промышленных проектов, стандарты (ГОСТ, ПНСТ), сертификация программно-аппаратных комплексов.	2	
Лекция. Лекция 16. Научный семинар. Обзор статьи из конференции уровня A*/A или журнала первого уровня белого списка; обсуждение методик и результатов.	2	
Практическое занятие. Практическая работа 16. Federated Learning Hands-On Настройка простой федеративной системы (Flower/FedAvg) с несколькими клиентами; защита данных; оценка качества.	4	
Задания для самостоятельной работы, в том числе выполнение Анализ безопасности и приватности Проведение экспериментов по атаке и защите (Backdoor attack, FGSM), применение методов дифференциальной приватности. Изучение кейса с промышленным партнёром Работа над реальной задачей (напр., предиктивное обслуживание оборудования): разработка полного pipeline, общение с представителями партнёра.	6	
Иная контактная работа:	0	

Раздел 5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Изучение дисциплины рекомендуется начать с ознакомления с рабочей программой, ее структурой и содержанием разделов. Учебный материал структурирован, изучение дисциплины осуществляется в тематической последовательности.

Подготовка к практическим занятиям включает ознакомление с планом занятия, работу с учебной и учебно-методической литературой, научными изданиями и электронными образовательными ресурсами, рекомендованными рабочей программой дисциплины. Содержание самостоятельной работы определяется рабочей программой дисциплины, оценочными и методическими материалами, заданиями и указаниями преподавателя. Самостоятельная работа может осуществляться в аудиторной и внеаудиторной формах. Эффективным средством осуществления самостоятельной работы является электронная информационно-образовательная среда университета, которая обеспечивает доступ к образовательной программе, рабочей программе дисциплины (модуля), к электронным библиотечным системам, профессиональным базам данных и информационным справочным системам.

Контроль и аттестация

Периодичность проведения, формы текущего контроля успеваемости, система оценивания хода освоения дисциплины представлены в рабочей программе.

Формой промежуточной аттестации по дисциплине в 6 семестре – экзамен, в 7 семестре БРК.

В 6 семестре предусмотрен курсовой проект в рамках кейса научной школы ПГТУ:

Тема РГР: «Разработка специализированного программно-аппаратного комплекса мониторинга и интеллектуальной диагностики радиоканала на основе методов машинного обучения»

Постановка задачи

В радиотехнических системах связи (КВ/УКВ радиоканалы, сенсорные радиосети, радиолинии телеметрии и навигации) качество передачи информации существенно зависит от состояния радиоканала, уровня помех, ионосферных условий и параметров аппаратуры.

Требуется разработать прототип программно-аппаратного комплекса, обеспечивающего сбор радиотехнических параметров, их обработку и интеллектуальную диагностику состояния радиоканала с использованием методов машинного обучения, а также прогнозирование ухудшения качества связи и выявление аномальных режимов работы.

Задание на РГР:

1. Анализ предметной области

- Описать выбранный тип радиотехнической системы:
 - КВ-радиоканал;
 - радиолиния телеметрии;
 - сенсорная радиосеть;
 - приём GNSS/радионавигационных сигналов;
 - радиоканал «передатчик–ретранслятор–приёмник».
- Определить основные параметры радиоканала:

- уровень сигнала (RSSI);
- SNR;
- BER / FER;
- задержка и фазовые искажения;
- доплеровский сдвиг;
- параметры помех и замираний.
- Сформулировать задачу диагностики:
 - классификация состояния канала;
 - обнаружение аномалий;
 - прогноз ухудшения качества связи.

2. Сбор и подготовка данных

- Организовать сбор данных:
 - с SDR-приёмника (USRP, HackRF, RTL-SDR — реальный или эмулированный);
 - либо из симулятора радиоканала (GNU Radio, MATLAB/Simulink, Python-модель).
- Реализовать поток передачи данных (файлы / сокеты / очередь сообщений).
- Выполнить предобработку:
 - фильтрация;
 - агрегация временных окон;
 - формирование признаков (spectral features, статистические параметры).
- Проанализировать дисбаланс классов (редкие аномальные состояния канала) и применить:
 - SMOTE / weighted learning;
 - кастомные функции потерь.

3. Разработка модели машинного обучения

- Выбрать и реализовать модель:
 - классификатор состояний радиоканала;
 - модель обнаружения аномалий;
 - модель прогнозирования качества канала во времени.
- Использовать Python-фреймворки:
 - scikit-learn / PyTorch / TensorFlow.
- Реализовать:
 - кастомные метрики (взвешенный F1, PR-AUC);
 - статистическую проверку значимости результатов.

- Провести сравнительный анализ нескольких моделей.

4. Программно-аппаратная реализация

- Разработать прототип комплекса:
 - модуль приёма/эмуляции радиосигнала;
 - модуль обработки данных;
 - ML-модуль диагностики.
- Выполнить оптимизацию под аппаратную платформу:
 - CPU / GPU;
 - edge-устройство (Jetson Nano / аналог).
- Оценить:
 - задержку обработки;
 - пропускную способность;
 - энергопотребление (при возможности).

5. Инженерные практики и сопровождение

- Оформить проект в виде сервиса (CLI или REST API).
- Использовать контейнеризацию (Docker).
- Реализовать базовый мониторинг:
 - метрики качества модели;
 - метрики работы системы.
- Подготовить документацию по архитектуре комплекса.

6. Безопасность и надёжность

- Рассмотреть угрозы:
 - искажение данных;
 - помехи;
 - аномальный трафик.
- Реализовать элементарное обнаружение аномалий в потоке данных.
- Оценить устойчивость модели к шумам и искажениям.

7. Документация и защита

- Подготовить отчёт, включающий:
 - описание радиотехнической модели;
 - архитектуру программно-аппаратного комплекса;

- результаты экспериментов;
- графики и диаграммы.
- Подготовить презентацию и защитить проект, продемонстрировав работу прототипа.

Программное обеспечение и инструментальные средства

1. Среды разработки и инструменты программирования

- Visual Studio Code — универсальная среда разработки с поддержкой Python, C/C++, CUDA, Docker и удалённых серверов. <https://code.visualstudio.com/>
- PyCharm Community Edition — IDE для разработки ML- и AI-проектов на Python. <https://www.jetbrains.com/pycharm/>
- Jupyter Notebook / JupyterLab — интерактивная среда для исследований, экспериментов и визуализации результатов. <https://jupyter.org/>
- NVIDIA Nsight Systems / Nsight Compute — профилировщики для анализа производительности GPU-приложений (CUDA, TensorRT). <https://developer.nvidia.com/tools-overview>
- Intel VTune Profiler — инструмент анализа производительности для CPU/FPGA приложений. <https://www.intel.com/content/www/us/en/developer/tools/vtune/overview.html>
- VSCode Remote Development / SSH — для работы с серверными кластерами и edge-устройствами.

2. ML-фреймворки и специализированные библиотеки

- TensorFlow — обучение и инференс моделей, поддержка XLA-компиляции, TPU. <https://www.tensorflow.org/>
- PyTorch — основной фреймворк для исследований, распределённого обучения и разработки DL-моделей. <https://pytorch.org/>
- JAX — высокоскоростные вычисления и автоматическая дифференциация с XLA. <https://github.com/google/jax>
- ONNX / ONNX Runtime — универсальный формат обмена моделями между фреймворками. <https://onnx.ai/>
- TensorRT, OpenVINO, TVM — оптимизация и ускорение инференса для GPU/CPU/FPGA/ASIC.
- Horovod / PyTorch Distributed — инструменты распределённого обучения для мульти-GPU и мульти-узловых кластеров.
- Scikit-learn — обработка данных, классические ML-алгоритмы.
- HLS4ML / Vitis HLS — генерация ускорителей для FPGA.

3. Средства работы с большими данными и потоковыми системами

- Apache Kafka — высокопроизводительная потоковая шина данных. <https://kafka.apache.org/>
- Apache Spark / PySpark — распределённая обработка данных. <https://spark.apache.org/>
- Apache Flink — потоковая обработка в реальном времени. <https://flink.apache.org/>
- Airflow / Prefect — построение ETL-/DataOps-пайплайнов.
- DVC или LakeFS — контроль версий данных.

- **4. Платформы и инструменты для MLOps**
- Docker — контейнеризация ML- и AI-приложений. <https://www.docker.com/>
- Kubernetes — оркестрация, масштабирование и управление нагрузкой. <https://kubernetes.io/>
- Kubeflow — MLOps-платформа для полного ML-цикла (pipeline → обучение → деплой). <https://www.kubeflow.org/>
- MLflow — трекинг экспериментов, управление моделями и регистрами моделей. <https://mlflow.org/>
- Prometheus + Grafana — мониторинг сервиса и моделей.
- Git / GitHub / GitLab — система контроля версий и CI/CD.

5. Средства разработки и оптимизации для специализированного оборудования GPU / TPU

- CUDA Toolkit (<https://developer.nvidia.com/cuda-toolkit>)
- cuDNN (<https://developer.nvidia.com/cudnn>)
- ROCm (AMD GPU stack): <https://rocm.docs.amd.com/>

FPGA / ASIC

- Xilinx Vitis / Vitis HLS
- Intel Quartus Prime
- HLS4ML — преобразование нейросетей в FPGA-совместимые реализации

Edge-устройства

- NVIDIA Jetson SDK (JetPack)
- Intel OpenVINO Toolkit
- Raspberry Pi tools + Coral USB Accelerator

6. Платформы для удалённых и коллективных экспериментов

- Google Colaboratory — облачные ноутбуки с поддержкой GPU/TPU. — <https://colab.research.google.com/>
- Kaggle Notebooks — бесплатные GPU, датасеты и соревнования. — <https://www.kaggle.com/>
- Yandex DataSphere (если требуется отечественная альтернатива).
- Серверные кластеры университета / корпоративные GPU-кластеры — основная вычислительная платформа курса.

7. Инструменты обеспечения безопасности и приватности

- OpenSSL / WireGuard — шифрование соединений
- Tools for adversarial robustness (IBM ART, Foolbox)
- Differential Privacy Libraries (Opacus, Google DP)

8. Дополнительные утилиты и сервисы

- Anaconda / Miniconda — управление окружениями
- Weights & Biases / Neptune.ai — отслеживание экспериментов
- FastAPI / Flask — развёртывание ML-сервисов
- Postman / Swagger — тестирование API
- S3-совместимые хранилища (MinIO, AWS S3, Yandex Object Storage) — хранение данных и моделей

Дополнительные ссылки на источники информации (КРМ)

1. Э: Применяет продвинутые методы работы с несбалансированными данными (SMOTE weighted learning). Настраивает кастомные метрики и функции потерь. Проводит статистический анализ значимости результатов
 - Оценка качества систем искусственного интеллекта. Общие положения ГОСТ Р 59898- 2021
 - Системная и программная инженерия. Тестирование программного обеспечения. Часть 11. Тестирование систем искусственного интеллекта ПНСТ965-2024
 - <https://education.yandex.ru/handbook/ml/article/metriki-klassifikacii-i-regressii>
 - <https://education.yandex.ru/handbook/ml/article/kross-validaciya>
 - <https://education.yandex.ru/handbook/ml/article/bias-variance-decomposition>
 - https://imbalanced-learn.org/stable/user_guide.html#user-guide
 - <https://education.yandex.ru/handbook/data-analysis/article/bazovye-statisticheskie-testy>
1. Э: Разрабатывает специализированные инструменты и инженерные практики промышленной разработки систем ИИ, развертывания и сопровождения моделей машинного обучения в рабочей версии ПО
 - Oviedo J. et al. ISO/IEC quality standards for AI engineering //Computer Science Review. – 2024. – Т. 54. – С. 100681.
1. Э: Кастомизирует и разрабатывает специализированные инструменты управления данными
 - Oviedo J. et al. ISO/IEC quality standards for AI engineering //Computer Science Review. – 2024. – Т. 54. – С. 100681.

Дополнительные ссылки на источники информации (из КРМ)

2. Э: Применяет продвинутые методы работы с несбалансированными данными (SMOTE weighted learning). Настраивает кастомные метрики и функции потерь. Проводит статистический анализ значимости результатов
 - Оценка качества систем искусственного интеллекта. Общие положения ГОСТ Р 59898- 2021
 - Системная и программная инженерия. Тестирование программного обеспечения. Часть 11. Тестирование систем искусственного интеллекта ПНСТ965-2024
 - <https://education.yandex.ru/handbook/ml/article/metriki-klassifikacii-i-regressii>

- <https://education.yandex.ru/handbook/ml/article/kross-validaciya>
 - <https://education.yandex.ru/handbook/ml/article/bias-variance-decomposition>
 - https://imbalanced-learn.org/stable/user_guide.html#user-guide
 - <https://education.yandex.ru/handbook/data-analysis/article/bazovye-statisticheskie-testy>
1. Э: Разрабатывает специализированные инструменты и инженерные практики промышленной разработки систем ИИ, развертывания и сопровождения моделей машинного обучения в рабочей версии ПО
- Oviedo J. et al. ISO/IEC quality standards for AI engineering //Computer Science Review. – 2024. – Т. 54. – С. 100681.
1. Э: Кастомизирует и разрабатывает специализированные инструменты управления данными
- Oviedo J. et al. ISO/IEC quality standards for AI engineering //Computer Science Review. – 2024. – Т. 54. – С. 100681.

Языки программирования и библиотеки

1. Языки программирования (Python, C/C++, CUDA, SQL)

- 1.1.Python 3.x — основной язык разработки в области ИИ, поддержка большого числа библиотек. – Курсы и справочники:
- Palladain. Python_1_HSE_2024 — курс «Язык программирования Python» (ввод-вывод, типы данных, основы синтаксиса) – https://github.com/Palladain/Python_1_HSE_2024
- Основы Python. Хендбук (Яндекс Образование) – <https://education.yandex.ru/handbook/python>
- Smartika.ru — задачник по Python (списки, структуры данных) – <https://smartika.ru/python-workbook/list#2>
- 1.2.C/C++ — разработка высокопроизводительных модулей, взаимодействие с CUDA/ FPGA, оптимизация вычислений.
- 1.3.CUDA C/C++ — язык для разработки программ под GPU NVIDIA. – Документация: <https://docs.nvidia.com/cuda/>
- 1.4.SQL — работа с хранилищами данных, Data Lake, MLOps-конвейерами.

1. Библиотеки Python для ввода-вывода, структур данных и регулярных выражений

- 2.1.re – регулярные выражения в Python – <https://docs.python.org/3/library/re.html>
- 2.2.Regex101 — тестирование и разбор регулярных выражений – <https://regex101.com/>
- 2.3.itertools, collections — эффективные структуры и итераторы для обработки данных (deque, Counter, chain, product).
- 2.4.f-strings:
- f-string cheat sheet — <https://fstring.help/cheat/>

- Статья: «f-строки в Python: применение и производительность» — <https://shultais.education/blog/python-f-strings>

1. Объектно-ориентированное программирование и структура модулей

- 3.1.Auriga. Алгоритмы автоматической проверки требований... (подходы к проектированию систем) — <https://habr.com/ru/companies/auriga/articles/724030/>
- 3.2.Python Descriptor Guide — дескрипторы, фундамент ООП-механизмов Python — <https://docs.python.org/3/howto/descriptor.html>

1. Параллелизм и конкурентность (многопроцессность, многопоточность, асинхронность)

- 4.1.Wunder Fund. Разбираемся с параллельными и конкурентными вычислениями в Python — <https://habr.com/ru/companies/wunderfund/articles/>
- 4.2.Модули Python:
 - multiprocessing — процессы и распараллеливание
 - threading — потоки
 - asyncio — асинхронные вычисления
 - concurrent.futures — высокоуровневые пулы потоков/процессов

1. Библиотеки для машинного обучения и работы с данными (feature engineering, несбалансированные данные, статистика, ML-пайплайны)

- 5.1.NumPy — векторные вычисления, матричные операции
- 5.2.Pandas — табличные данные, подготовка датасетов
- 5.3.Scikit-learn — классические ML-алгоритмы, метрики, preprocessing
- 5.4.Imbalanced-Learn (imblearn) — SMOTE, ADASYN, весовые функции потерь
- 5.5.SciPy — статистика, оптимизация, тесты значимости
- 5.6.Matplotlib / Seaborn / Plotly — визуализация данных
- 5.7.PySpark — распределённая обработка больших данных

1. Библиотеки глубокого обучения и специализированных вычислений (GPU/TPU, оптимизация инференса, компиляторы моделей)

- 6.1.PyTorch — основная библиотека DL, поддержка распределённого обучения
- 6.2.TensorFlow / Keras — разработка нейронных сетей, XLA-компиляция
- 6.3.JAX — автодифференцирование + ускорение через XLA
- 6.4.ONNX / ONNX Runtime — переносимость и оптимизация моделей
- 6.5.TensorRT — ускорение инференса на GPU NVIDIA
- 6.6.OpenVINO Toolkit — оптимизация под CPU, iGPU, VPU
- 6.7.TVM — компилятор для оптимизации моделей под различные ускорители

1. Средства распределённого и высокопроизводительного обучения (multi-GPU, multi-node, кластерные вычисления)

- 7.1.PyTorch Distributed (DDP)
- 7.2.Horovod — универсальный распределённый тренинг (TensorFlow/PyTorch)
- 7.3.NCCL — коммуникации GPU-GPU (all-reduce, broadcast)
- 7.4.Ray / Ray Tune — распределённое обучение и гиперпараметрический поиск

1. Библиотеки для потоковой обработки и DataOps

- 8.1. Apache Kafka (kafka-python) — потоковые данные
- 8.2. Apache Flink / PyFlink — обработка событий в реальном времени
- 8.3. Apache Spark / PySpark — батч-и стрим-обработка
- 8.4. Airflow / Prefect — оркестровка данных и ETL-конвейеры
- 8.5. DVC / LakeFS — контроль версий данных

1. Инструменты для FPGA, Edge AI и аппаратного ускорения (HLS, синтез, edge-устройства)

- 9.1. HLS4ML — перенос нейросетей на FPGA
- 9.2. Vitis HLS / Vivado — проектирование ускорителей
- 9.3. Intel OpenCL FPGA SDK
- 9.4. NVIDIA JetPack SDK — набор инструментов для Jetson
- 9.5. TensorFlow Lite / TFLite Micro — оптимизация для микроконтроллеров

1. Инфраструктура, деплой и сервисы (MLOps, API, CI/CD)

- 10.1. Docker — контейнеризация
- 10.2. Kubernetes / KubeFlow — оркестрация, ML-пайплайны
- 10.3. MLflow — трекинг экспериментов
- 10.4. FastAPI / Flask — построение inference-сервисов
- 10.5. gRPC / REST — взаимодействие сервисов
- 10.6. Prometheus + Grafana — мониторинг моделей

Содержание дисциплины включает изучение исследований и разработок, представленных в научных публикациях на конференциях уровня А*, статьях, опубликованных в журналах 1 уровня Белого списка научных журналов Минобрнауки России (УММ на основе публикаций в научных журналах 1 или 2 уровня Белого списка научных журналов Минобрнауки России, публикаций на конференциях уровня А*/А.

Ниже приведен перечень конференций, рекомендуемых для подачи работ, соответствующих уровню подготовки студентов, освоивших принципы программирования и аналитики данных на языке Python.

1. Машинное обучение, инженерия ИИ и оптимизация вычислений (Machine Learning Engineering, Systems for ML)

- 1.1. International Conference on Machine Learning (ICML)

<https://icml.cc/> — Одна из ключевых конференций А*. Имеет Systems for ML Track, где рассматриваются оптимизация вычислений, распределённое обучение, компиляторы моделей, ускорение инференса — темы, напрямую связанные с GPU/TPU/FPGA и программно-аппаратной оптимизацией.

- 1.2. Conference on Neural Information Processing Systems (NeurIPS)

<https://neurips.cc/> — Включает ML Systems, Optimization, Efficient ML, а также Student

- Paper Award tracks, где принимаются студенческие работы по ускорению моделей, работе с несбалансированными данными, оптимизации памяти, генерации на специализированных устройствах.
- 1.3. International Conference on Learning Representations (ICLR)

<https://iclr.cc/> – Открытое рецензирование, большое количество workshops по практическим ML-инструментам, edge-AI, оптимизации нейросетей, low-bit quantization и методам ускорения обучения.
- 1.4. AAAI Conference on Artificial Intelligence

<https://aaai.org/> – Имеет Student Abstract and Poster Program, где принимаются инженерные и экспериментальные работы, связанные с MLOps-практиками, кастомными метриками, SMOTE-подобными методами, оптимизацией inference, разработкой специализированных пайплайнов.
- 1.5. International Joint Conference on Artificial Intelligence (IJCAI)

<https://ijcai.org/> – Содержит AI Engineering, Applied AI, AI Infrastructure треки — подходит для студенческих проектов по разработке инструментов для ML-платформ, edge-устройств, IoT-аналитики и распределённых вычислений.
- 1. **Системы обработки данных, стриминг, высокопроизводительные вычисления (Data Engineering, HPC, Big Data)**
 - 2.1. ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)

<https://www.kdd.org> – Applied Data Science Track принимает работы по реальным промышленным ML-кейсам, включая анализ сенсорных данных, потоковые пайплайны, оптимизацию DataOps — то, что студенты изучают в рамках курса.
 - 2.2. IEEE International Conference on Big Data

<https://bigdataieee.org> – Тематика включает: потоковую обработку данных (Kafka/Flink), распределённые вычисления (Spark), оптимизацию ETL-процессов, edge-аналитику
 - 2.3. IEEE International Conference on High Performance Computing (HiPC)

<https://hipc.org> – Конференция по высокопроизводительным вычислениям: GPU - оптимизация, параллелизм, MPI, CUDA, hardware-aware ML.
 - 2.4. USENIX OSDI / USENIX ATC (операционные системы и системная инженерия)

<https://www.usenix.org> – Подходят для проектов, связанных с: разработкой собственных инструментов для ускорителей, контейнеризацией, оптимизацией runtime для ML-моделей, MLOps-инфраструктурой

1. Встраиваемые системы, FPGA, аппаратные ускорители (Embedded AI, Hardware for ML)

- 3.1.ACM/IEEE International Symposium on Computer Architecture (ISCA)

<https://iscaconf.org> – Главная конференция по архитектурам процессоров. Секции по: FPGA/ASIC для AI, системным оптимизациям, энергоэффективным архитектурам, аппаратной поддержке quantization/pruning

- 3.2.Design Automation Conference (DAC)

<https://www.dac.com> – Фокус на проектировании ASIC/FPGA для ML. Подходит для проектов студентов 7 семестра, связанных с HLS, HLS4ML, оптимизацией latency/power.

- 3.3.International Conference on Field-Programmable Logic and Applications (FPL)

<https://fpl.org> – Работы по FPGA-реализациям нейросетей, edge-AI и pipeline - архитектурам.

- 3.4.Embedded Systems Week (ESWEEK)

<https://esweek.org> – Темы: IoT-аналитика, real-time ML, энергоэффективные устройства — полностью соответствуют edge-части курса.

1. Поточковые данные, IoT, промышленная аналитика (Industry 4.0, Streaming Analytics)

- 4.1.ACM DEBS — Distributed and Event-Based Systems

<https://debs.org> – для проектов по Kafka/Flink/Spark Streaming, которые входят в практики 6 семестра.

- 4.2.IEEE Internet of Things Conference

Подходит для работ по edge-AI, сенсорным сетям, распределённой обработке телеметрии.

- 4.3.IEEE International Conference on Industrial Informatics (INDIN)

Темы: анализ данных оборудования, обнаружение аномалий, промышленный IoT, edge-вычисления

1. Примеры тем учебных задач

(Задачи ориентированы на GPU/TPU/FPGA, MLOps, потоковые данные, оптимизацию моделей и работу с аппаратными комплексами.)

1.

- 5.1. Сравнение производительности ускорителей (CPU vs GPU vs TPU vs FPGA) на реальной ML-задаче.
- 5.2. Проектирование pipeline для потоковых данных оборудования (Kafka → Spark → ML-модель).
- 5.3. Обнаружение аномалий в телеметрии промышленного датчика с применением SMOTE и кастомных метрик.
- 5.4. Квантование модели до 8-bit и 4-bit и анализ trade-off между скоростью и качеством.
- 5.5. Оптимизация инференса модели на специализированном оборудовании (TensorRT/OpenVINO/TVM).
- 5.6. Построение edge-ML-аналитики на Jetson Nano или Raspberry Pi + Movidius.
- 5.7. Создание собственного инструмента для автоматической упаковки ML-модели (DataOps + MLflow + Docker).
- 5.8. FPGA-реализация сверточного слоя с использованием HLS4ML / Vitis HLS.
- 5.9. Проектирование системы мониторинга температурных и вибрационных данных оборудования.
- 5.10. Реализация распределённого обучения на нескольких GPU с PyTorch Distributed.
- 5.11. Создание MLOps-конвейера для автоматического обучения и деплоя модели.
- 5.12. Построение energy-profiling отчёта (GPU/edge device) и оптимизация потребления.
- 5.13. Исследование методов Borderline-SMOTE и Focal Loss на задаче промышленной диагностики.
- 5.14. Разработка защищённого ML-сервиса с мониторингом атак и отказоустойчивостью.
- 5.15. Анализ latency и throughput ML-системы при нагрузочном тестировании.

Изучение современных технологий по материалам конференций:

1. **A Survey on Distributed Machine Learning** / Joost Verbraeken et al. // arXiv preprint, 2020.

Этот обзор представляет современные методы распределённого обучения, описывает архитектуры, схемы параллелизации и проблемы масштабируемости, а также анализирует инструменты и подходы для распределённого тренинга ML-моделей. Результаты исследования полезны для понимания многопроцессного обучения, коммуникации между узлами и системной оптимизации на GPU-кластерах, что связано с практиками по распределённому обучению и оптимизации вычислений. URL: <https://arxiv.org/abs/1912.09789>

2. **Hardware Acceleration of LLMs: A Comprehensive Survey and Comparison** / Nikoletta Koilia & Christoforos Kachris // arXiv preprint, 2024.

В этой обзорной статье сравниваются и классифицируются различные аппаратные ускорители (GPU, FPGA, ASIC, in-memory computing) для больших языковых моделей (LLMs), оценивая производительность, энергоэффективность и

1. архитектурные особенности. Материал особенно полезен для студентов при изучении аппаратных ускорителей, оптимизации моделей и системного проектирования под специализированное оборудование. URL: <https://arxiv.org/abs/2409.03384>
2. **A Survey on FPGA-based Accelerators for ML Models** / Feng Yan, Andreas Koch, Oliver Sinnen // arXiv preprint, 2024.

Данное исследование систематизирует разработки аппаратных ускорителей на базе FPGA для ускорения инференса и обучения ML-моделей, анализирует тенденции исследований, основные подходы и перспективы. Этот материал полезен для понимания программно-аппаратной интеграции и оптимизации ML-вычислений на FPGA и связан с практическими заданиями по разработке аппаратных модулей и FPGA-оптимизации. URL: <https://arxiv.org/abs/2412.15666>

3. **A Survey on Hardware Accelerators for Large Language Models (LLMs)** / Christoforos Kachris // Applied Sciences, 2025.

Статья представляет обзор аппаратных ускорителей (GPU, FPGA, ASIC) для LLM, включая анализ архитектурных решений, метрик производительности и энергоэффективности. Материал помогает студентам понять критерии выбора аппаратной платформы, оптимизацию вычислений и соотношение нагрузок CPU/GPU /FPGA при работе с крупными моделями. URL: <https://www.mdpi.com/2076-3417/15/2/586>

4. **Intelligent Edge Computing and Machine Learning** / SA Cajas Ordóñez // Future Internet (MDPI), 2025.

Статья анализирует технологии Intelligent MLOps на edge-устройствах, включая развертывание и мониторинг моделей в распределённых сетях IoT, а также особенности управления жизненным циклом ML-систем в ресурсно-ограниченных условиях. Работа коррелирует с темами курса о edge-AI, MLOps, IoT-интеграции и оркестрации ML-процессов. URL: <https://www.mdpi.com/1999-5903/17/9/417>

5. **Hardware Acceleration of Explainable Machine Learning using TPUs** / Zhixin Pan & Prabhat Mishra // arXiv preprint, 2021.

В статье предложена методика ускорения алгоритмов объяснимого машинного обучения (Explainable ML) с помощью TPU/GPUs, что обеспечивает значительное ускорение интерпретации результатов. Материал полезен для понимания аппаратной поддержки интерпретируемых моделей, что актуально при разработке real-time ML-систем. URL: <https://arxiv.org/abs/2103.11927>

6. **A review on TinyML: State-of-the-art and prospects** / L. Capogrosso // Sensors International, 2024.

Обзор TinyML-подходов — ML на микроконтроллерах и ultra-low-power устройствах. Обсуждаются производительность, потребление энергии и аппаратно-программные оптимизации для TinyML-систем. Связан с практиками по edge-AI и ресурсно-ограниченным ML-выполнениям. URL: https://iris.univr.it/retrieve/36d129f6-e5ae-4345-baa7-e38f06b5e593/A_Machine_Learning-Oriented_Survey_on_Tiny_Machine_Learning.pdf

Раздел 6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

6.1. Учебно-методическое обеспечение

№№ п/п	Список используемой литературы	Количество экземпляров печатных изданий, имеющихся в библиотеке, или электронный адрес издания (ресурса) в сети Интернет
УЧЕБНЫЕ, УЧЕБНО-МЕТОДИЧЕСКИЕ И НАУЧНЫЕ ИЗДАНИЯ		
1.	Хахаев, И. А. Практикум по алгоритмизации и программированию на Python [Электронный ресурс] / Хахаев И. А. 2-е изд. Москва: ИНТУИТ, 2016. - 178 с.	https://e.lanbook.com/book/100377
2.	Сузи, Р. А. Язык программирования Python [Электронный ресурс] / Сузи Р. А. 2-е изд. Москва: ИНТУИТ, 2016. - 350 с. ISBN 5-9556-0058-2.	https://e.lanbook.com/book/100546
3.	Северенс, Ч. Введение в программирование на Python [Электронный ресурс] / Северенс Ч. 2-е изд. Москва: ИНТУИТ, 2016. - 231 с.	https://e.lanbook.com/book/100703
ЭЛЕКТРОННЫЕ ОБРАЗОВАТЕЛЬНЫЕ РЕСУРСЫ		
1.	Научная электронная библиотека eLIBRARY.RU	http://elibrary.ru
2.	Научная электронная библиотека «Киберленинка»	http://cyberleninka.ru
3.	Chollet F. Deep Learning with Python / 2nd ed. – Manning, 2021. – 770 p. (англ.)	https://www.manning.com/books/deep-learning-with-python-second-edition
4.	Geron A. Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow / 2nd ed. – O'Reilly, 2019. – 850 p. (англ.)	https://learning.oreilly.com/library/view/hands-on-machine-learning/9781492032632/
5.	Aurelien Geron. Machine Learning Engineering – O'Reilly, 2023. – 400 p. (англ.)	https://learning.oreilly.com/library/view/machine-learning-engineering/9781098103010/
6.	Muller A., Guido S. Introduction to Machine Learning with Python – O'Reilly, 2016. – 400 p. (англ.)	https://learning.oreilly.com/library/view/introduction-to-machine/9781449369880/
7.	CUDA Programming Guide– NVIDIA Developer Documentation	https://docs.nvidia.com/cuda/pdf/CUDA_C_Programming_Guide.pdf
8.	TensorRT Developer Guide – NVIDIA Developer Documentation	https://docs.nvidia.com/deeplearning/tensorrt/developer-guide/index.html
9.	OpenVINO™ Toolkit Documentation – Intel	https://docs.openvino.ai/

ПРОФЕССИОНАЛЬНЫЕ БАЗЫ ДАННЫХ И ИНФОРМАЦИОННЫЕ СПРАВОЧНЫЕ СИСТЕМЫ		
1.	Справочно-правовая система Консультант+	http://www.consultant.ru
2.	Информационно-правовой портал Гарант	http://www.garant.ru
3.	Профессиональные справочные системы Техэксперт	http://www.cntd.ru

6.2. Материально-техническая база и программное обеспечение

№№ п/п	Аудитории для проведения учебных занятий, самостоятельной работы и проведения государственной итоговой аттестации	Перечень основного оборудования	Программное обеспечение
1.	333б (III)	GPS приемник в составе (1), VICTORIA 3065 C (1), Аппаратно-программный комплекс (адаптер+ПО) (1), Вольтметр В7-37 (3), ГЕНЕРАТОР Г2-57 (4), Генератор шумовых сигналов (1), ИЗМЕРИТ.ПОМЕХ LMZ-4 (2), ИЗМЕРИТЕЛЬ ПОМЕХ (1), ИЗМЕРИТЕЛЬ X6-5 (1), Измерительный прибор SNT LITE PSTN (1), Комплект дополнит.оборудования к VICTORIA (1), Лабораторный стенд "Цифровая электроника" 1060x256x654 (2), Монитор 19" Samsung 940N (KSB) TFT Silver. Round Simple (3), Монитор 19"Samsung 940N (LKSB) TFT (2), Монитор LG LCD 19" L1919S-SF (1), Осциллограф GDS-820C (1), ПРИБОР Д/ИС КОР X6-8 (3), ПРИБОР ИКХ-X6-5 (1), Проектор мультимедийный Hitachi CP-X2514WN (1), Систем.блок Athlon 64 3500/512Mb*2/160Gb/FDD/DVD-RW клав.мышь.ковр. (2), Систем.блок Core 2Duo E6320/2Гб/320Гб/512Мб клав.мышь (2), Систем.блок АМД3000+(512*2)/160Gb/DVD+R Wrkfd/+мышь+коврик+клав. (1), Системный блок RAY P360.3 ,клав,мышь оптич, коврик+монитор 19" ViewSonic VA916 (1), Системный блок AMD*2 4000/2*512 MB/160Gb/512 MB/ (1), Учебная Лабораторная установка "Изучение КМ-кодека" (1), Учебная	Microsoft Windows Enterprise, Справочная правовая система "Консультант Плюс", Microsoft Office Standard, Агент Dr.Web, Комплект ГАРАНТ-Мастер, Microsoft Access, Microsoft Visio Professional, Microsoft Project Professional, Microsoft Visual Studio Enterprise, Комплект ПО для решения основных пользовательских задач

	Лабораторная установка "Изучение принципов временного разделения каналов" (1), Учебная телевиз.установка "УТС-2004" (1), Экран настенный 200*200см Braun Roll Vision (1), Комплект учебной мебели (1)	
--	---	--

Раздел 7. ФОРМЫ КОНТРОЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ/ ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Критерии оценивания индикаторов достижения компетенций направлены на:

- усвоение теоретического материала (объем знаний, глубина усвоения), предусмотренного рабочей программой;
- умение излагать материал (четкость, грамотность изложения материала, точность и полнота воспроизведения учебного материала);
- умение применять теоретические знания при решении практических заданий.

Шкала оценивания представлена ниже.

Уровень сформированности элементов компетенции	Критерии оценивания	Шкала оценивания
Пороговый уровень	Обучающийся имеет знания основного материала, проявляет умение логично его излагать, но может допускать неточности в изложении материала, недостаточно правильные формулировки, испытывает затруднения в выполнении практических заданий.	удовлетворительно
Продвинутый уровень	Обучающийся твердо знает программный материал, излагает его грамотно и по существу, не допускает существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения	хорошо
Высокий уровень	Обучающийся глубоко и прочно усвоил программный материал, грамотно и логически стройно его излагает, дает исчерпывающие ответы на поставленные вопросы. В ответе тесно увязывается теория с практикой, при этом обучающийся не затрудняется с ответом при видоизменении задания, свободно справляется с задачами, вопросами и другими видами применения знаний, показывает знакомство с монографической литературой, периодическими изданиями, правильно обосновывает принятые решения, свободно владеет разносторонними навыками, приемами выполнения практических работ	отлично

7.1. Текущий контроль успеваемости

Текущий контроль успеваемости обеспечивает оценивание хода освоения дисциплины (модуля) и производится с применением технологии рейтингового контроля в соответствии с технологической картой дисциплины. Порядок составления технологической карты и алгоритм проведения процедуры оценивания видов деятельности обучающихся, направленных на освоение знаний, умений, навыков и/или опыта деятельности, по

накопительной системе в баллах устанавливается положением о системе РИТМ в ФГБОУ ВО «ПГТУ»

7.2. Промежуточная аттестация обучающихся

Промежуточная аттестация обучающихся направлена на оценивание результатов обучения по дисциплине (модулю) и проводится с использованием фондов оценочных средств.

Примеры типовых контрольных заданий из базы фонда оценочных средств по образовательной программе.

Примеры типовых контрольных заданий из базы фонда оценочных средств по образовательной программе.

1. Какой метод NumPy обеспечивает наиболее эффективное выполнение операций над массивами за счет применения операций к целым массивам без явных циклов?

- a) Индексирование
- b) Векторизация
- c) Broadcasting
- d) Slicing

2. Что произойдет при выполнении операции `arr1 + arr2`, где `arr1.shape = (3, 1)` и `arr2.shape = (1, 4)` в NumPy?

- a) Ошибка
- b) Форма (3, 4) благодаря broadcasting
- c) Форма (1, 1)
- d) Дублирование элементов

3. В многопоточном программировании на Python, что такое GIL (Global Interpreter Lock)?

- a) Механизм синхронизации
- b) Библиотека
- c) Блокировка, позволяющая выполнять байт-код Python только одному потоку
- d) Инструмент отладки

4. Какой модуль Python следует использовать для CPU-bound задач вместо threading?

- a) asyncio
- b) multiprocessing
- c) concurrent.futures с ThreadPoolExecutor
- d) queue

5. В scikit-learn, какой параметр в `train_test_split` обеспечивает воспроизводимость?

- a) shuffle
- b) stratify
- c) random_state
- d) test_size

6. Что такое переобучение (overfitting) в ML?

- a) Низкое качество на всех выборках
- b) Высокое на обучающей, низкое на тестовой
- c) Долго обучается
- d) Много признаков

7. Какой метод кросс-валидации наиболее подходит для несбалансированных классов?

- a) KFold
- b) StratifiedKFold
- c) ShuffleSplit
- d) LeaveOneOut

8. Что делает `fit_transform()` в scikit-learn?

- a) Только трансформация
- b) Обучает и применяет
- c) Быстрее, другой результат
- d) Синтаксический сахар

9. В PyTorch, какой метод обнуляет градиенты перед backward?

- a) `optimizer.step()`
- b) `optimizer.zero_grad()`
- c) `loss.backward()`
- d) `model.train()`

10. Что представляет собой autograd в PyTorch?

- a) Выбор оптимизатора
- b) Автоматическое дифференцирование
- c) Аугментация данных
- d) Настройка гиперпараметров

11. Какая архитектура НС создана для последовательностей и долгосрочных зависимостей?

- a) CNN
- b) Perceptron
- c) LSTM
- d) Autoencoder

12. Что такое Transfer Learning?

- a) Перенос данных
- b) Использование предобученной модели для новой задачи
- c) Передача параметров

d) Метод распределённого обучения

13. Какой метод регуляризации случайно отключает нейроны?

a) L1

b) L2

c) BatchNorm

d) Dropout

14. В TensorFlow/Keras, чем Functional API отличается от Sequential API?

a) Быстрее

b) Позволяет сложные топологии

c) Только полносвязные

d) Нет различий

15. Какой Python-фреймворк асинхронный и высокопроизводительный для ML-сервисов?

a) Django

b) Flask

c) FastAPI

d) Tornado

16. Что такое DAG в Apache Airflow?

a) Формат данных

b) Граф задач без циклов

c) Тип сети

d) Протокол

17. В PySpark структура, представляющая распределённую неизменяемую коллекцию?

a) DataFrame

b) RDD

c) Dataset

d) Array

18. Какая функция PySpark применяется к каждому элементу RDD?

a) apply()

b) transform()

c) map()

d) foreach()

19. Что такое cold-start problem в Spark?

a) Медленный запуск кластера

b) Время инициализации задач

c) Ошибка архивных данных

d) Проблема памяти

20. Какой компонент MLOps отвечает за отслеживание метрик и экспериментов?

a) CI/CD

b) Model registry

c) Experiment tracking

d) Feature store

Перечень вопросов для проведения промежуточной аттестации

Перечень вопросов по модулям для проведения промежуточной аттестации

Модуль 1. Инструментальные средства разработки на Python

1. Экосистема Python, virtual environments, pip/conda.

2. Векторизация в NumPy, broadcasting, линейная алгебра.

3. Основные операции с DataFrame и Series в pandas.

4. Основы визуализации (Matplotlib, Seaborn, Plotly).

5. Параллелизм и многопоточность в Python, GIL, multiprocessing.

Модуль 2. Машинное обучение и классические алгоритмы

1. Обзор типов задач ML: классификация, регрессия, кластеризация.

2. Метрики качества моделей, переобучение, кросс-валидация.

3. Препроцессинг, масштабирование, кодирование категорий.

4. Алгоритмы: линейная/логистическая регрессия, деревья решений, случайный лес.

5. Градиентный бустинг (XGBoost, LightGBM, CatBoost), гиперпараметры.

Модуль 3. Глубокое обучение и нейронные сети

1. Архитектура нейронных сетей, forward/backward propagation.

2. Фреймворки PyTorch и TensorFlow, autograd.

3. CNN: свёртки, pooling, архитектуры LeNet/VGG/ResNet.

4. RNN, LSTM, GRU, Attention.

5. Регуляризация (Dropout, BatchNorm), оптимизаторы, Transfer Learning.

Модуль 4. Серверные приложения и обработка больших данных

1. REST API для ML (FastAPI, Flask, DRF), валидация, документирование.

2. MLOps: Git, DVC, MLflow, Weights & Biases, мониторинг моделей.

3. Архитектура Spark и PySpark: RDD, DataFrame, Dataset.

4. MLlib в Spark: распределённое обучение моделей.

5. Apache Airflow: концепция DAG, операторов и автоматизация ML-пайплайнов.

