МИНИСТЕРСТВО ПРОМЫШЛЕННОСТИ И ТОРГОВЛИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ УНИТАРНОЕ ПРЕДПРИЯТИЕ
«НАУЧНО-ТЕХНИЧЕСКИЙ ЦЕНТР ОБОРОННОГО КОМПЛЕКСА «КОМПАС»

АНАЛИТИЧЕСКИЕ МАТЕРИАЛЫ

ТЕХНИЧЕСКИЕ АСПЕКТЫ РАЗРАБОТКИ И РАЗВЕРТЫВАНИЯ СИСТЕМ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА НА КОСМИЧЕСКИХ АППАРАТАХ

МОСКВА

СОДЕРЖАНИЕ

Обозначения и сокращения……..……..…….. 3
Введение…….…… 5
1. Подходы министерства обороны сша к развитию и применению технологий искусственного интеллекта в космической технике …… 6
1.1 Взгляды руководства США на использование технологий искусственного интеллекта для повышения автономности спутниковых операций…….…….. 8
1.2 Особенности систем искусственного интеллекта в зависимости от среды развертывания …… 12
2. Особенности процессов разработки и развертывания систем искусственного интеллекта для космической техники ….……..…… 18
2.1 Эталонная архитектура рабочего процесса разработки и развертывания систем искусственного интеллекта …… 18
2.2 Оптимизация модели рабочего процесса AI/ML ….……..…. 30
3. Особенности операций машинного обучения в системах искусственного интеллекта на космической технике…….……... 38
3.1 Операции машинного обучения MLOps …… 38
3.2 Особенности операций машинного обучения MLOps для наземной среды …… 41
3.3 Особенности операций машинного обучения MLOps для периферийной/мобильной среды ……. 42
3.4 Особенности операций машинного обучения MLOps в космической среде ……..… 44
4. Особенности выбора электронных компонентов для систем искусственного интеллекта на космической технике ……... 47
4.1 Обоснование выбора вычислительных устройств для космических систем AI/ML ……..…… 47
4.2 Особенности аппаратной среды для гетерогенных вычислений в системах AI/ML …… 58
Заключение ……..……..… 60
Список использованных источников …… 62

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

ИИ	–	искусственный интеллект
КА	–	космический аппарат
МО	–	Министерство обороны
НОО	–	низкая околоземная орбита
ПН	–	полезная нагрузка
СПРН	–	система предупреждения о ракетном нападении
УСП	–	унифицированная спутниковая платформа
3D	–	Three-Dimensional (трехмерный)
AI	–	Artificial Intelligence (искусственный интеллект)
AutoCA	–	Automatic Conjunction Assessment (автоматическая оценка конъюктуры)
BNN	–	Bayesian Neural Network (байесовская нейронная сеть)
CPS	–	Cyber-Physical Systems (киберфизическая система)
$CHMI^2$	–	Cognitive Human-Machine Interfaces and Interactions (когнитивные человеко-машинный интерфейс и взаимодействие)
DARPA	–	Defense Advanced Research Projects Agency (Управление перспективных исследований МО США)
DSS	–	Distributed Satellite System (распределенная спутниковая система)
iCPS	–	intelligent CPS (интеллектуальная киберфизическая система)
HDFS	–	Hadoop Distributed File System (распределенная файловая система Hadoop)
HMS		Human-Machine System (человеко-машинная система)
EGP	–	Extended General Perturbation ([методы] расширенного общего возмущения)
MAPE	–	Mean Absolute Percentage Error (средняя абсолютная процентная ошибка)
ML	–	Machine Learning (машинное обучение)
MLOps	–	Machine Learning Operations (операции машинного обучения)
MW/MT	–	Missile Warning/Missile Tracking (система предупреждения о ракетном нападении и сопровождения ракет)
NASA	–	National Aeronautics and Space Administration (Национальное управление по аэронавтике и исследованию космического пространства США)

OMM	–	Orbit Mean-Elements Message (сообщение о средних элементах орбиты)
OSP	–	Orbital State Propagator (пропагатор орбитального состояния [модель прогнозирования])
SDA	–	Space Domain Awareness (ситуационная осведомленность о космическом пространстве)
SNN	–	Standard Neural Network (стандартная нейронная сеть)
TASO	–	Trusted Autonomous Satellite Operation (доверительная автономная спутниковая операция)

ВВЕДЕНИЕ

Развитие и внедрение технологически готовых и надежных приложений искусственного интеллекта и машинного обучения рассматривается руководством МО США в качестве ключевого условия для реализации современных межвидовых операций (Joint All Domain Operations) и достижения решающего превосходства в космосе.

В обзоре рассматриваются современные подходы руководства ведущих аэрокосмических и консалтинговых компаний, космических сил и Министерства обороны США к формированию требований по обеспечению развертывания и использования приложений искусственного интеллекта и машинного обучения (AI/ML) на борту космических аппаратов с оценкой элементов сходства и различия в структуре моделей AI/ML в зависимости от наземной, периферийной или космической среды развертывания, особенностей эталонной архитектуры и услуг, необходимых для обеспечения разработки и развертывания приложений AI/ML для использования в космосе. Раскрываются общие и частные тенденции в развитии аппаратной среды с уточнением преимуществ и недостатков современных процессоров основных типов для приложений AI/ML на борту КА.

Рассматриваются общие тенденции и особенности, структурно-логические схемы разработки и программирования моделей искусственного интеллекта и машинного обучения для наземной, периферийной или космической сред развертывания. Показано, что для космических систем обработки данных на борту КА на основе приложений AI/ML, и особенно с использованием алгоритмов глубокого обучения, требуется многократное увеличение вычислительной производительности и мощности по сравнению с тем уровнем, который доступен в системах на основе только радиационно-стойких процессоров. Следующее поколение процессоров для космических приложений искусственного интеллекта и машинного обучения на борту КА в зависимости от целевого назначения будет включать различные комбинации коммерческих и новых радиационно-стойких центральных (CPU) и графических (GPU) процессоров, программируемых вентильных матриц (FPGA) и специализированных интегральных схем ASIC.

В документе представлены данные исследовательских отчетов ведущих американских аэрокосмических компаний Lockheed Martin Space и Boeing, консалтинговой корпорации RAND и Национального управления по аэронавтике и исследованию космического пространства США с оценками основных направлений и проблем разработки и внедрения приложений искусственного интеллекта и машинного обучения в составе ботовых систем космических аппаратов

.

1 ПОДХОДЫ МИНИСТЕРСТВА ОБОРОНЫ США К РАЗВИТИЮ И ПРИМЕНЕНИЮ ТЕХНОЛОГИЙ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА В КОСМИЧЕСКОЙ ТЕХНИКЕ

Согласно новой стратегии, опубликованной Министерством обороны США 19 марта 2025 года, технологии искусственного интеллекта в космической технике являются критически важным элементом стратегии национальной безопасности Соединенных Штатов Америки. Развитие и внедрение апробированных и надежных технологий искусственного интеллекта и машинного обучения (AI/ML) является ключевым подходом военного ведомства к реализации усилий по объединенному командованию и управлению в совместных межвидовых операциях (Joint All Domain Operations Command and Control - JADC2) и требует активного сотрудничества со стороны Министерства обороны, академических научных кругов и космической промышленности для создания и предоставления наиболее передовых решений с поддержкой ИИ.

По заявлению заместителя министра обороны США, обновленное руководство «не только основано на прошлогодних стратегиях ведомства в области искусственного интеллекта и управления данными, но и включает дополнительные уточнения, учитывающие последние достижения космической отрасли в области федеративных сред, децентрализованного управления данными, генеративного искусственного интеллекта, киберфизической защиты и интеграции технологий.

В частности, новый документ заменяет стратегию МО США в области искусственного интеллекта 2018 года издания и стратегию в области управления данными 2020 года, которые соответственно заложили доктринальную основу для качественного расширения стратегических и оперативных возможностей ВС США с поддержкой искусственного интеллекта. Стратегия 2025 года создает концептуальную основу и определяет современный подход МО США в области использования наиболее продвинутых технологий ИИ, устанавливая «иерархию потребностей» их развития и внедрения. В рамках условной пирамиды этих потребностей безусловный приоритет отдается высококачественным данным как фундаментальной основе для глубокой аналитики и разработки ответственного искусственного интеллекта (рисунок 1).

Рисунок 1. Иерархия потребностей для развития искусственного интеллекта в интересах МО США

По данным ведущих космических компаний Lockheed Martin Space и Boeing, структура и содержание предлагаемой МО США «иерархии потребностей» ИИ в значительной мере соответствуют концепции и архитектуре продвинутых киберфизических систем (cyber-physical systems — CPS), главной характеристикой которых является интеграция и сетевое взаимодействие вычислительных, технологических и производственных процессов, физических элементов и их цифровых двойников в цифровом слое управления для оптимизации процессов достижения назначенных задач1 (рисунок 2).

Рисунок 2. Типовая архитектура киберфизической системы

[1] Национальный институт стандартов и технологий США (NIST) определяет киберфизические системы (CPS) как совокупность цифровых, аналоговых, физических и человеческих компонентов, функционирующих посредством интегрированной физической технологии и логики.

Согласно концепции CPS, в этих системах вычислительная компонента распределена по всей физической среде, которая является ее носителем, и синергетически увязана со всеми составляющими элементами. Такие системы взаимодействуют между собой с использованием стандартных интернет-протоколов для прогнозирования, самонастройки, реконфигурирования и адаптации к новым задачам и условиям работы.

По мере развития технологий искусственного интеллекта и машинного обучения, облачных и периферийных вычислений предполагается существенно сокращать задержки, возникающие при передаче данных, и обеспечивать новые возможности развития интеллектуальных систем.

В то же время развитие киберфизических систем рассматривается МО США в качестве эффективного инструмента для преодоления таких проблем высокоавтоматизированных систем автономного управления, как распределенность, надежность, отказоустойчивость, масштабируемость и автономность работы в удаленных и труднодоступных местах, как например космическое пространство.

Согласно обновленной стратегии 2025 года, одна из наиболее крупных проблем использования ИИ в военных, и особенно космических, системах заключается в том, что эти операционные среды являются очень динамичными со многими, зачастую неизвестными, факторами и последствиями. В свою очередь, это создает проблемы в выборе, приобретении и предоставлении большого объема данных, требуемых для обучения наиболее продвинутых систем искусственного интеллекта. Даже при существовании возможности обучить систему ИИ для выполнения конкретной военной задачи, «адаптация ее к новой задаче или среде, как правило, невозможна без существенного переобучения, что может сохранить или не сохранить компетентность этой системы для выполнения ранее освоенных задач и операций.

1.1 Взгляды руководства США на использование технологий искусственного интеллекта для повышения автономности спутниковых операций

По данным МО США, современные системы искусственного интеллекта хорошо работают в тех военных и космических приложениях, где последствия «неправильного понимания» ими поставленных задач терпимы и не катастрофичны. Растущая сложность и интенсивность военных и космических операций устанавливают высокую планку для систем ИИ, которые поддерживают принятие и реализацию решений «быстрее, чем предполагалось» в ситуациях, когда человеческие жизни находятся под угрозой. А реалистичное тестирование и оценка систем ИИ с точки зрения того, как они будут работать в таких приложениях, чрезвычайно сложна и до настоящего времени не отработана.

В частности, значительные достижения в США в области искусственного интеллекта и киберфизических систем для космических приложений открыли новые возможности для быстрорастущей космической отрасли и американских космических сил. Поэтапное внедрение распределенных и гибридных спутниковых группировок и связанных с ними концепций космических операций стимулирует разработку архитектур интеллектуальных киберфизических систем (iCPS), которые могут поддерживать высокий уровень гибкости и устойчивости работы в условиях все более перегруженного околоземного космического пространства. В свою очередь растущая потребность в более высоких уровнях автоматизации и автономности в работе космических аппаратов и выполнении спутниковых операций вызывает расширение исследований и разработок, направленных на улучшение системной производительности, включая решение проблем достоверности, целостности и кибернетической защиты данных, а также связанных с ними методов мониторинга и реконфигурации программно-аппаратных средств, которые способны поддерживать доверенные автономные спутниковые операции (Trusted Autonomous Satellite Operations - TACO).

Несмотря на эти достижения, уровень автономности значительной части современных спутниковых платформ ограничен определенным набором (шаблоном) правил и типовых случаев, в то время как переход к концепции TASO требует смены парадигмы в проектировании как космических аппаратов, так и наземных систем космической инфраструктуры. В частности, широкое внедрение технологий ИИ рассматривается как важный фактор для реализации доверенных автономных спутниковых операций в связи с тем, что их использование позволяет существенно повысить пропускную способность и адаптируемость космической системы, а также целостность и достоверность передаваемых данных, особенно в распределенных спутниковых системах (Distributed Satellite Systems - DSS).

При этом отмечается, что в определенных направлениях космической деятельности полностью автономные спутниковые операции либо нецелесообразны либо нежелательны, в основном потому, что даже незначительная ошибка в управлении КА может привести к значительному ущербу и расходам, а в некоторых случаях - и к человеческим жертвам (например, в случаях с суборбитальным воздушно-космическим транспортом или обитаемой космической станции на околоземной орбите). Поэтому для околоземных операций требуется гарантированный приемлемый уровень доверия, особенно с учетом постоянного увеличения количества развернутых космических объектов на низких околоземных и геостационарных орбитах.

Вместе с тем анализ современных тенденций развития киберфизических систем для космических приложений показывает, что наиболее передовая архитектура киберфизико-человеческих (Cyber-Physical-Human - CPH) систем развивается с широким использованием адаптированных технологий машинного обучения и гибридного искусственного интел-

лекта (например, нейронечеткие механизмы вывода), что позволяет модулировать уровни ее автономности и функции управления/контроля оператором для решения конкретных задач.

В последние 10 лет распределенные спутниковые системы (DSS) с использованием малых космических аппаратов на околоземных орбитах стали приоритетным направлением наращивания возможностей и трансформации космических сил США. В связи с этим с высокой вероятностью можно утверждать, что использование технологий машинного обучения и искусственного интеллекта в создаваемых в США распределенных системах DSS имеет критическое значение для упрощения перехода на качественно новый технологический уровень доверенных автономных спутниковых операций (TASO). Чтобы соответствовать требованиям доверительности и надежности создаваемых автономных космических аппаратов и их интеллектуального функционирования в условиях насыщенного околоземного космического пространства, требуется радикальный отход от традиционных систем проектирования и разработки. В дальнейшем, особенно в операциях на низких околоземных орбитах, где требуется одновременно решать вопросы безопасности движения и предупреждения столкновений в космосе, устойчивости космических группировок и защиты космических аппаратов, а также правовые требования, решающее значение будут иметь доступность и сертификация систем на основе искусственного интеллекта.

На современном этапе развития такие требования все еще сложно удовлетворить в полной мере в системах искусственного интеллекта и машинного обучения (AI/ML), разворачиваемых на создаваемых гибридных и масштабируемых космических системах с распределенной многоуровневой архитектурой и доступом к облачным вычислительным ресурсам и графическим процессорам. Подобная задача является более сложной для реализации на борту КА в условиях ограниченных вычислительных возможностей программно-аппаратных средств и интерфейса взаимодействия с другими аппаратами. Требования по разработке и внедрению приложений AI/ML для обработки данных на борту КА можно разделить на три категории: эксплуатационные, по безопасности и на основе пользовательского опыта (таблица 1).

Эксплуатационные требования охватывают способность системы AI/ML выполнять требуемую задачу в требуемой среде. Конечным пользователям нужны решения искусственного интеллекта и машинного обучения, способные обрабатывать большие объемы данных в режиме реального времени с минимальным количеством оборудования, занимаемой площадью и высокой точностью. Эти решения должны также развертываться на различных платформах и оборудовании архитектуры, масштабируемые до переменных входов и пропускной способности между несколькими процессорами и КА при сохранении требуемого уровня производительности.

Таблица 1.

Определяющие требования и общие решения по использованию приложений искусственного интеллекта и машинного обучения в космических аппаратах

Требования	Определяющие параметры	Решения
	Эксплуатационные	
Эффективность	• Время реакции/ожидания • Пропускная способность • Рабочий цикл • Время запуска • Потребляемые ресурсы	• Оптимизация модели и потоков •Планирование потока данных и/или загрузки конвейера моделей • Среда тестирования • Минимальное время выполнения
Развертываемость	• Архитектура модели (поддерживаемые операции) • Язык программирования • Фреймворк AI/ML • Перенацеливаемость	• Многоспутниковая поддержка • Совместная поддержка множества ускорителей • Контейнеризация
Масштабируемость	• Количество одновременных операций ввода-вывода • Горизонтальная масштабируемость • Вертикальная масштабируемость	• Переменные вход/пропускная способность • Распределенные обработка и рассуждения • Многоагентная оркестровка
Устойчивость	• Частота обновления модели • Доля нисходящих данных • Размер артефактов операций машинного обучения • Требуемая производительность обновлений	• Автоматический конвейер моделей машинного обучения (MLOps) • Бортовой мониторинг • Цифровые двойники • Активное обучение • Дифференциальные (выборочные) обновления
	Безопасность	
Защищенность	• Угроза уязвимости • Угроза возможностей обнаружения • Угроза изменения чувствительности и реакции сенсорной системы • Устойчивость и надежность модели • Избыточность прогноза	• Разработка и испытание робастных моделей • Обновление моделей безопасности • Состязательная подготовка • Верификация и валидация • Кибербезопасность на основе технологий AI/ML
Надежность	• Отказоустойчивость • Робастность • Время простоя • Рабочий цикл	• Мониторинг моделей • Встроенная диагностика/устранение неисправностей • Несколько вариантов резервирования • Оптимизация ресурсов
Объяснимость	• Модель неопределенности • Модель сложности архитектуры • Объяснимость	• Объяснимые методы AI/ML • Непрерывный мониторинг • Анализ первопричин
	Пользовательские требования	
Быстродействие	• Период разработки • Простота обновления	• Хранилище данных и моделей • Передовые коммерческие инструменты и услуги разработки
Стоимость	• Единовременные расходы • Периодические затраты • Улучшение по сравнению с другими методами	• Использование автоматических конвейеров и потоков • Повторное использование общих решений • Сокращение дублирования усилий
Простота в использовании	• Интуитивность пользовательского интерфейса приложений • Простота интеграции • Соответствие стандартам	• Ориентированные на пользователя проекты и модели • Внедрение стандартных интерфейсов/протоколов/методов

Требования безопасности предусматривают ожидаемую работу систем искусственного интеллекта и машинного обучения и их устойчивость к внешним воздействиям и угрозам, как физическим, так и цифровым. В частности, модели глубокого обучения зачастую описываются как «черные ящики», где данные поступают и выходят, и мало что известно о том, каким образом эти модели реализуют свои задачи и функции.

Аналогичным образом во многом недетерминированный характер систем AI/ML может вызвать у некоторых пользователей сомнения и недоверие к таким системам. Кроме того, системы AI/ML должны соответствовать более стандартным программным показателям надежности и безопасности в связи с тем, что они характеризуются аналогичной с типовыми программными системами уязвимостью. Наряду с этим требуются процедуры для снижения уровня отказов, вызванных радиационным излучением, а также возможности гибкого переключения на резервный ресурс и уменьшение влияния сбоев или повреждения системы AI/ML на космический аппарат и выполняемые им задачи.

Требования с позиций пользовательского опыта сосредоточены на удовлетворении потребностей пользователей в относительно недорогих и быстроразрабатываемых системах AI/ML, которые просты в использовании и обеспечивают существенные преимущества по сравнению с традиционными методами. Системы на основе AI/ML могут быть более дорогостоящими для разработки, внедрения и последующей поддержки. Однако заказчик в лице Министерства обороны и конечные пользователи должны видеть преимущества этих систем, которые стоят таких инвестиций.

1.2 Особенности систем искусственного интеллекта в зависимости от среды развертывания

По данным опытной эксплуатации систем AI/ML ведущими американскими компаниями, уникальные проблемы, оборудование и сценарии использования в значительной мере определяются характеристиками физической среды развертывания этих систем. В наиболее общем случае различают четыре таких среды: наземные, периферийные, миниатюрные портативные и космические объекты и компоненты. Такая классификация не является исчерпывающей, а служит основой для рассмотрения особенностей в разработке и развертывании систем искусственного интеллекта в различных средах.

Наземная среда характеризуется наличием систем AI/ML, развернутых на базе настольных компьютеров, ноутбуков, стационарных серверов или на масштабируемых системах с достаточными вычислительными ресурсами и услугами, включая продвину-

тые облачные вычисления и Интернет вещей. Вычислительная мощность в наземных средах является, как правило, наибольшей, а эффективность используемых ресурсов имеет тенденцию быть вторичной по отношению к точности, особенно в облачных вычислениях, где при необходимости достаточно просто привлекать дополнительные ресурсы. Наземные системы AI/ML работают чаще всего на компонентной базе процессоров x86 CPU и NVIDIA GPU, несмотря на то что использование других процессоров (например, ARM CPU, AMD GPU, TPU и FPGA) может предложить больше решений и приложений искусственного интеллекта в этой среде. Космические приложения AI/ML в наземной среде, как правило, развертываются для контроля и управления космическими аппаратами в космосе, сбора, обработки, анализа и распределения полученных от космических средств данных.

Периферийные, миниатюрные портативные и мобильные среды AI/ML содержат широкую номенклатуру устройств, характеризующихся малыми форм-факторами, низким уровнем энергопотребления и мобильностью. При этом периферийная среда предполагает использование современных вычислительных вредств, в том числе центральных (CPU) и графических (GPU) процессоров, тензорных нейронных процессоров (TPU), программируемых вентильных матриц (FPGA) и интегральных схем специального назначения (ASIC), а также многих других, как правило в виде одноплатного компьютера (SBC), систем на кристалле (SoC) или микроконтроллеров (MCU).

Периферийные устройства используются в беспилотных и пилотируемых наземных, воздушных и морских транспортных средствах. Кроме того, такие средства можно найти в системах управления и наведения высокоточного оружия, медицинских приборах, умных помощниках, средствах дистанционного зондирования, на объектах производства и в устройствах Интернета вещей. Ограниченные вычислительная мощность, объем памяти, обучающие данные и ряд других технических ограничений представляют проблемы, которые требуется преодолеть при развертывания систем искусственного интеллекта в этой среде. При этом сама система AI/ML должна быть не только достаточно компактной, чтобы соответствовать доступному объему энергонезависимой памяти, она должна быть способна функционировать, принимать решения и предлагать выводы в пределах требуемого уровня задержки и точности. Это обусловливает необходимость сбалансировать различные показатели производительности системы AI/ML в процессе ее разработки. Кроме того, несмотря на большое

разнообразие вычислительных средств, используемых в периферийных устройствах, значительная часть из них не обеспечивает полную поддержку операций искусственного интеллекта и машинного обучения.

Наряду с этим развертывание системы AI/ML на периферии по сравнению с наземной средой обычно требует более значительного объема инвестиций для ее интеграции с целевым устройством.

В миниатюрной портативной среде используются устройства, ориентированные на еще более низкое энергопотребление и меньший форм-фактор, такой как «голый металл» (bare metal). В частности, серверные платформы bare metal представляют собой устройства без операционной системы, которая позволяет программному обеспечению получать прямой доступ к аппаратной части. При этом алгоритмы выполняются непосредственно на логическом оборудовании без рабочей системы. К таким устройствам относятся различные миниатюрные датчики, носимое портативное оборудование и ряд других средств, требующих минимального форм-фактора и обеспечивающих работу тонких компьютерных узлов.

Устройства в мобильной среде основаны на технологиях систем на кристалле (System on Chip - SoC) — интегральных микросхем, которые построены на базе архитектуры ARM (Advanced RISC Mashine) микропроцессорных и микроконтроллерных ядер, разработанных британской компанией ARM Limited, и включают в себя графические процессоры, процессоры сигналов изображения (ISP), цифровые процессоры сигналов (DSP), а наиболее современные микросхемы - также блоки нейронной (NPU) или тензорной обработки (TPU). В состав устройств мобильной среды входят сотовые телефоны и планшеты. При этом программные приложения, как правило, выполняются в операционной системе iOS или Android (варианты для мобильной среды). Существуют и другие виды устройств, но ни один из них не используется так часто, как оба упомянутых выше.

Необходимо указать, что из-за существенного подобия периферийных, миниатюрных и мобильных сред при рассмотрении рабочего процесса AI/ML в дальнейшем в рамках данной работы, если не потребуется выделить явные различия, ссылки будут делаться только на одну из этих сред.

С вычислительной точки зрения для бортовых процессоров космических аппаратов существует множество дополнительных проблем из-за ограничений, связанных с осо-

бенностями космической среды. В частности, космическая среда характеризуется высоким уровнем радиационного, в том числе солнечного, излучения и высокоэнергетических частиц, которые непрерывно воздействует на бортовые электронные средства. Такие воздействия, включая накопительные эффекты полной ионизирующей дозы (Total Ionizing Dose - TID) и эффекты одиночных событий (SEE — сбои, вызванные попаданием одной высокоэнергетической частицы в чувствительный узел), в свою очередь вызывают целый ряд проблем, от неразрушающих временных ошибок (мягкие ошибки) и до катастрофических отказов и повреждений электронных устройств. Неисправность или отказ оборудования в результате TID или SEE могут не только поставить под угрозу выполнение задач космической операции, но и подвергнуть риску жизни людей в пилотируемом полете.

Наряду с воздействием радиации, разработчики космических аппаратов ограничены другими проблемами, включая требованиями по ограничению и оптимизации габаритов, массы, потребляемой мощности и стоимости (SWaP-C). Несмотря на то что современные радиационно-стойкие (rad-hard) процессоры обеспечивают необходимую устойчивость к влиянию космической радиации, зачастую они имеют чрезвычайно ограниченные показатели производительности, отставая на несколько поколений от коммерческих встроенных процессоров.

Эти ограничения на современном этапе развития делают развертывание передовых ИИ-фреймворков на радиационно-стойкой компонентной базе практически сложноосуществимым. В связи с этим наиболее привлекательным вариантом является использование коммерчески доступных (COTS) встроенных устройств, включая центральные и графические процессоры, программируемые вентильные массивы (FPGA) и специализированные интегральные схемы (ASIC). Однако необходимо, чтобы эти коммерческие компоненты были протестированы на предмет использования в космосе, включая радиационные испытания, чтобы проверить их реакцию на TID и SEE. Только после таких испытаний и с учетом возможных ошибок из-за влияния радиации могут использоваться различные технологии вычислений. Учитывая огромное увеличение производительности коммерческих электронных компонентов по сравнению с их радиационно-стойкими аналогами, использование COTS-устройств позволит реализовать существенно расширенный диапазон решений и приложений для развертывания ИИ-моделей на борту космических аппаратов.

В 2003 году компания Autonomous Sciencecraft в рамках программы «Новое тысячелетие» по совершенствованию технологий для инструментов съемки земли Earth Observing 1 (EO-1) провела свой эксперимент, в рамках которого было развернуто несколько бортовых автономных приложений дистанционного зондирования, включая обнаружение облаков, классификацию мест наводнения, обнаружение природных катаклизмов и других аномалий с использованием гиперспектральных данных с аппаратуры Hyperion. Сложность испытаний и используемых алгоритмов была ограничена вычислительными возможностями встроенного микропроцессора Mongoose V и использованием простых методов «дерева решений» и пороговых значений. Десять лет спустя в 2013 году на демонстрационном КА CubeSat-1U (10×10×10 $см^3$) в ходе космического эксперимента IPEX (Intelligent Payload Experiment) использовались более сложные алгоритмы автономной бортовой обработки данных для классификации изображений, шаблонов сигнатур и карт заметности для идентификации «интересных» областей на полученных изображениях и их передачи по нисходящей линии связи «КА-Земля».

На современном этапе для выполнения космических операций более высокого уровня, в том числе в интересах NASA и Национального разведывательного управления (NRO), в ряде случаев из-за невозможности управления оператором КА при определенных маневрах использовалась система автономной оптической навигации. Так, для выполнения маневра touch-and-go на поверхности астероида Бенну космический аппарат OSIRIS-Rex (Origins, Spectral Interpretation, Resource Identification, Security, Regolith Explorer) использовал алгоритм естественного отслеживания аномалий. Используя базу данных объектов, собранных во время предварительного полета около астероида Бенну, бортовой процессор выполнил алгоритмы взаимной корреляции и автономной идентификации этих особенностей на изображениях, полученных в процессе маневра touch-and-go, а также оценки положения космического аппарата и точки контакта. Аналогичным образом, чтобы повысить точность посадки и избежать опасностей (например, скал, кратеров, крутых склонов, песчаных полей) в месте посадки в кратере Езеро, научная миссия «Марс-2020» использовала алгоритмы автономной навигации относительно местности в период сближения, спуска и посадки (EDL). Эти алгоритмы сопоставляли характеристики снимков с камер, сделанных во время спуска, с бортовой картой опасностей, которая была сгенерирована с использо-

ванием продуктов данных Mars Reconnaissance Orbiter для определения местоположения и высоты космического аппарата и одновременно прокладки маршрута к безопасным зонам.

В таблице 2 представлены ключевые области и приложения, которые позволяют получить существенные преимущества от использования системы ИИ на борту КА для прогнозирования с низкой задержкой. Хотя исчерпывающий обзор примеров бортового ИИ выходит за рамки этой работы, мы выделим основные области, где исследования бортового ИИ активно развиваются, включая дистанционное зондирование и автономное управление, навигацию и контроль (GPS).

Таблица 2.

Ключевые области и приложения приоритетного использования искусственного интеллекта на борту космических аппаратов

Ключевая область	Бортовые приложения искусственного интеллекты
Дистанционное зондирование	• Быстрое реагирование на стихийные бедствия и чрезвычайные ситуации • Сортировка разноформатных данных, обработка и сжатие гиперспектральных изображений и видеоданных • Бортовые интеллектуальные обработка данных и подготовка решений
Контроль пространственного положения, навигация и управление	• Управление автономными марсо- и луноходом • Автономные обнаружение опасностей и посадка • Отслеживание линии горизонта и положения звезд • Классификация рельефа и ландшафта местности
Планирование космических операций и задач	• Интеллектуальное планирование • Операции и задачи распределенных космических систем
Ситуационная осведомленность и контроль космического пространства	• Предупреждение столкновений в космосе • Автономные обнаружение и уклонение от опасных сближений • Автономные операции разделения, сближения и стыковки
Связь и коммуникации	• Программно-определяемые системы радиосвязи • Криптография и киберзашита

В настоящее время множество космических исследований по развитию планетарной науки, науки о Земле и других в областях ограничены из-за серьезных проблем по обработке и передаче больших массивов данных. Например, четыре аппарата Magnetosphere Multiscale (MMS) генерируют в день до 100 ГБ научных данных в режиме быстродействующей передачи данных, но в среднем только 4% этих данных могут быть переданы на Землю. Системы AI/ML могут использоваться для классификации на изображении конкретных объектов или определения «чувствительных» об-

ластей, что позволяет космическому аппарату автономно определять приоритетность передачи таких данных, а также отслеживать интересующие особенности для их фиксирования и сопровождения. Например, с целью дистанционного наблюдения за поверхностью Земли алгоритмы бортового маскирования облаков могут сегментировать изображение на облачные и не облачные метки.

В связи с тем что облака составляют значительную часть захваченных изображений, но являются неактуальными для значительной части операций, бортовое маскирование облаков может значительно сократить объем данных, которые необходимо передать. В частности, эксперимент Ф-sat-1 проводился с использованием одного из первых демонстрационных космических аппаратов, который включал и тестировал такую возможность бортового маскирования облаков, используя глубокую сверточную нейронную сеть (CNN), известную как CloudScout, для обнаружения облаков на изображениях, захваченных гиперспектральной камерой HyperScout-2. Имея крайне ограниченные данные, полученные до полета, разработчики HyperScout-2, первоначально обучили ее цифровую модель, используя прокси-датасет, созданный путем аугментации существующих наборов данных КА Sentinel 2.

Следует отметить, что в настоящее время множество других моделей ИИ может быть предложено для маскирования облаков на различных датчиках, которые могут быть развернуты на борту КА при условии наличия достаточных вычислительных ресурсов. Аналогичным образом существует возможность предоставления классификации изображений Земли с низкой задержкой, что также критически важно для оперативного реагирования на стихийные бедствия. В последнее время развертывание крупных созвездий малых спутников, таких как группировка КА Dove компании Planet Lab, предлагает беспрецедентное пространственное и временное покрытие Земли по сравнению с флагманскими миссиями с одним космическим аппаратом.

По данным NASA, в области автономной навигации и управления КА сделаны значительные шаги к созданию крупных наборов данных, необходимых для обучения моделей ИИ для автономных планетарных роверов и посадочных аппаратов. Несмотря на то что несколько марсианских роверов, включая Spirit, Opportunity, Curiosity и Perseverance, использовали систему авторулевого и автономного вождения, известную как AutoNav, даже самые передовые версии их технологий машинного зрения были построены исключительно на классических алгоритмах компьютерного зрения, в основе которых использовалась геометрическая информация для передвижения по марсианскому рельефу. Однако, учитывая, что роверы Spirit и Curiosity застряли в песчаной местности Марса, а колеса Curiosity были проколоты на острых камнях, NASA указывает на необходимость автономно идентифицировать семантическую

информацию о типе местности для оценки проходимости ландшафта, аналогично тому, как автомобили с автопилотом используют модели семантической сегментации для определения проходимой поверхности. Таким образом, для будущих миссий марсианских роверов Лаборатория реактивного движения NASA (JPL) разработала набор данных AI4Mars, который включает примерно 326 тыс. экземпляров семантической сегментации четырех классов: «почва», «основная порода», «песок» и «большой камень» на 35 тыс. изображениях от роверов Spirit, Opportunity и Curiosity для глубокого обучения общих моделей обучения.

Для расширения возможностей автономных посадки КА в основе алгоритмов Mars 2020 TRN использовались классические техники компьютерного зрения, основанные на шаблонном сравнении и регистрации по заранее определенным картам опасностей. Для сравнительно неотмеченных и динамичных территорий, например таких как Европа, эти техники TRN могут быть непрактичны, поскольку они сильно зависят от заранее определенных карт опасностей. В настоящее время специалисты NASA пытаются адаптировать модели ИИ для автопилотов и автономного вождения для повышения уровня обобщенного их восприятия, используя такие модели, как YOCO (You Only Crash Once), которые сначала обучались на симулированных данных, а затем тестировались на реальных данных Mars 2020 TRN, чтобы предсказывать как местоположение, так и семантическую информацию во время посадки.

В отличие от Mars 2020 TRN, в значительной части научных космических полетов используются узкоспециализированные датчики, которые фиксируют новые, впервые полученные измерения, что ставит перед приложениями AI/ML дополнительные проблемы по их обновлению, моделированию и адаптации к новым условиям. Процесс захвата и получения требуемого количества новых данных для обучения модели может быть чрезмерно ограниченным, а аннотирование данных часто требует много времени и может потребовать участия высококвалифицированных экспертов в данной области. Сочетание этих факторов часто приводит к недостатку крупномасштабных размеченных обучающих наборов данных для научного инструмента, который может не охватывать полное распределение входных данных, наблюдаемых во время развертывания, как это на своем опыте испытали разработчики КА Ф-sat-1.

2 ОСОБЕННОСТИ ПРОЦЕССОВ РАЗРАБОТКИ И РАЗВЕРТЫВАНИЯ СИСТЕМ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ДЛЯ КОСМИЧЕСКОЙ ТЕХНИКИ

2.1 Эталонная архитектура рабочего процесса разработки и развертывания систем искусственного интеллекта

По результатам масштабного исследования в 2023-2034 годах специалистами компании Lockheed Martin Space (г. Литлтон, шт. Колорадо) были уточнены эталонная архитектура и модель рабочего процесса создания космических систем AI/ML (подготовка, моделирование, оптимизация, интеграция и сертификация), включая определение основных компонентов и функций на каждом шаге этого процесса, а также их различий и подобия для различных сред. Данные вопросы были рассмотрены применительно к функциональным возможностям существующего свободного и открытого программного обеспечения FOSS (Free and Open Source Software) и коммерческих средств COTS.

Шаг А. Подготовка данных

На шаге «Подготовка рабочего процесса AI/ML» данные собираются, сохраняются, анализируются и предварительно обрабатываются. В некоторых случаях этот шаг выполняется с помощью специальных сценариев и записных книжек для импорта, анализа и подготовки данных к обучению; однако существует постоянно растущий список программных средств, помогающих в отслеживании, управлении и исследовании данных. На рисунке 3 представлена структурно-логическая схема этапа подготовки данных.

Обучающие данные часто хранятся в различных форматах, включая текстовые файлы (например, CSV или другой разделенный текст, raw) текст, текст фиксированной ширины, JSON, XML), двоичные файлы (например, HDF5, Excel, Python Pickle, TFRecord), файлы изображений (например, PNG, JPEG, TIFF, NTF), базы данных (например, MySQL, PostgreSQL, NoSQL) и хранилища объектов.

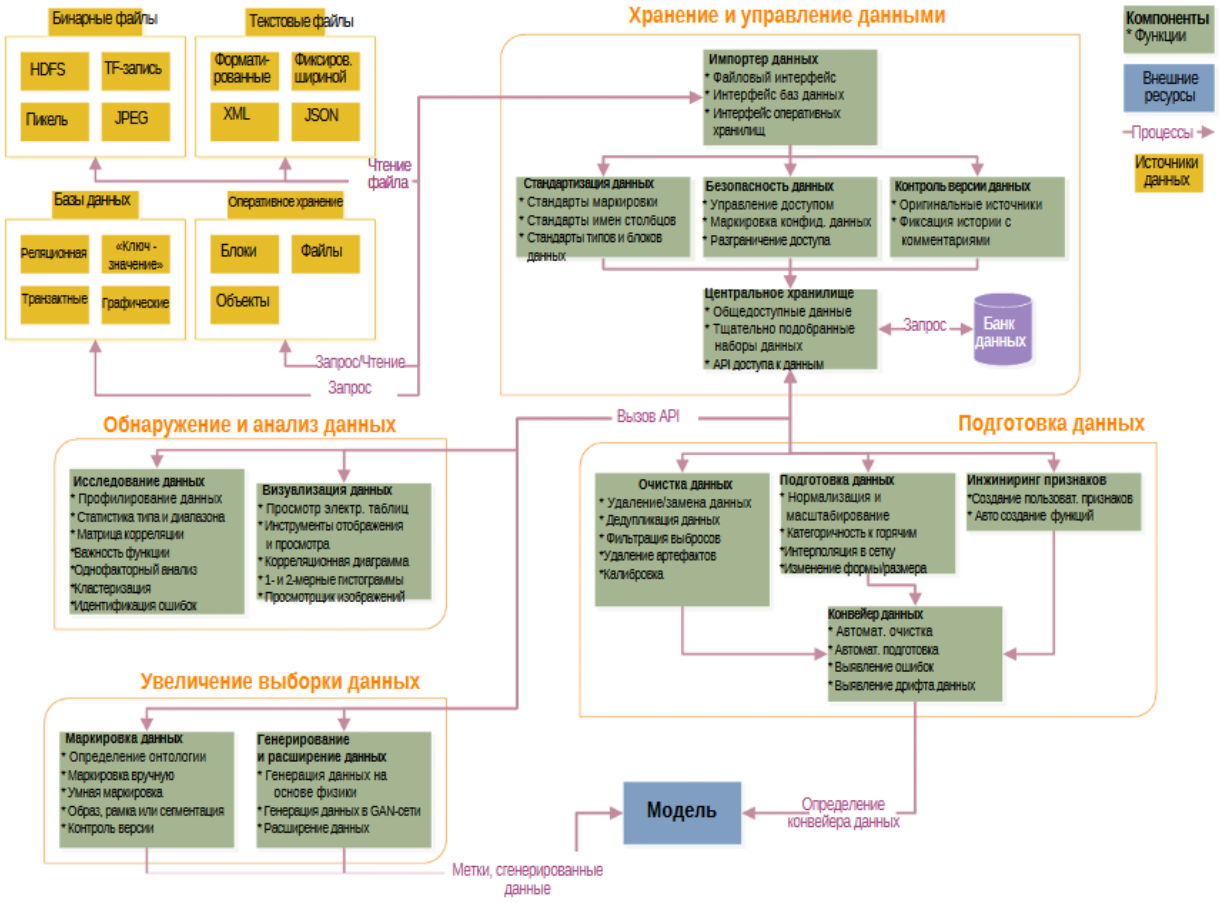

Рисунок 3. Структурно-логическая схема этапа подготовки данных для рабочего процесса AI/ML

Широкая поддержка различных форматов и методов хранения данных необходима для поддержки различных вариантов использования с одним фреймворком (платформой). Существует ряд решений свободного (бесплатного) программного обеспечения с открытым исходным кодом (FOSS) и коммерчески-доступных готовых продуктов (COTS), которые реализуются в системах AI/ML в качестве инструментов на уровне программных интерфейсов API (например, DVC или TF.data.Dataset module) или в качестве центрального хранилища наборов данных корпоративного уровня (например, Collibra).

Наряду с хранением данных централизованные репозитории наборов данных предоставляют дополнительные функции, которые обеспечивают управление этими данными. Центральные хранилища набора данных могут быть привязаны к корпоративным службам аутентификации, таким как каталог Active Directory, для обеспечения простого и безопасного способа управления разрешениями доступа для групп наборов данных. Для неограниченных наборов данных централизованный репозито-

рий упрощает поиск конкретных наборов данных и предлагает единый источник истинности для этих наборов данных. Стандарты могут быть внедрены на уровне компании-разработчика для сохранения с согласованной маркировкой, именованием столбцов, типом и единицей данных и т. д.

Средства хранения данных, реализованные на уровне проекта, как например DVC и интерфейс API набора данных TensorFlow, не могут предложить такие же функции корпоративного уровня, что и центральный репозиторий данных, но могут предоставить другие преимущества, включая более высокий уровень портативности, отсутствие блокировки поставщика и отсутствие требований корпоративных лицензии и управления. Решения для управления данными на уровне предприятия и проекта могут предоставить версию управления данными, которая отслеживает оригинальный источник и любые изменения в наборе данных, обеспечивая прослеживаемость (трассировку) линии данных.

Первым шагом в большинстве проектов машинного обучения после получения данных является исследовательский анализ данных (Exploratory Data Analysis - EDA). Целью EDA являются уяснение содержания набора данных, определение любых тенденций, которые могут быть использованы через механихзмы AI/ML, а также переход к формированию гипотез и подходов к исследованию. Шаг EDA, как правило, зависит от набора данных и требует высокопрофессиональных специалистов для создания визуализаций и других вариантов анализа данных, включая гистограммы признаков или статистическое резюме, графики рассеяния, сравнительные характеристики, линейные графики данных, визуализация изображений, анализ метаданных, статистика распределения данных, кардинальность, отсутствующие или противоречивые данные, корреляции между элементами, одномерный анализ и объединение данных в кластеры. Отдельные из этих вариантов анализа могут быть автоматизированы для определенных типов наборов данных. Например, инструмент профилирования Pandas Profiling может автоматически профилировать кадры данных, содержащие логические, числовые и категорированные данные, даты, URL, путь, файл и изображения.

В рамках проекта AI/ML шаг EDA может быть продолжен и повторен несколько раз. Однако в любом случае разработчик может использовать информацию, собранную в рамках EDA, чтобы начать очистку и подготовку данных для приема в модель. Целью очистки данных является исправление ошибок или других проблем в наборе данных, которые могут помешать производительности модели AI/ML.

Очистка данных может включать удаление или замену отсутствующих данных, дедупликацию (сжатие массива данных на основе исключения дублирующих копий повторяющихся данных), фильтрацию отклонений, фиксацию несоответствий в маркировке категориальных меток, удалении артефактов или других неисправных данных и в любом количестве специфичных для датчика калибровки, такие как данные изображения с плоским полем. Сегмент «Подготовка данных» принимает очищенные данные и переформатирует их в наиболее подходящий для моделей машинного обучения формат.

Подготовка данных может включать нормализацию или масштабирование данных, преобразование категориальных данных в одноступенчатое кодирование, интерполяцию в обычную сетку, изменение размеров, изменение формы и окна данных. Шаги подготовки некоторых данных могут применяться в обратном направлении к выходам модели, чтобы преобразовать выходы обратно в тот же, что и исходный вид данных.

В свою очередь сегмент «Инжиниринг признаков» представляет собой процесс «прочесывания» характеристик и функций набора данных для построения агрегированных или производных элементов, что наиболее упрощает извлечение результатов с использованием выбранной модели машинного обучения. В то время как проектирование признаков зачастую является ручным, итеративным и экспериментальным процессом, разработаны некоторые инструменты (например, Featuretools), которые позволяют автоматизировать поиск приемлемых комбинаций признаков. Эти три этапа - очистка, подготовка данных и разработка признаков - могут выполняться в разное время, а их результаты сохраняются в промежуточных наборах данных или могут быть объединены в определение конвейера данных и выполнены на необработанных данных в процессе обучения и вывода.

В отдельных случаях требуется дополнительный этап между предварительной обработкой данных и обучением модели - шаг расширения или генерации данных. Для моделей глубокого обучения, которые склонны к переформатированию, для повышения надежности обучаемой модели иногда требуется расширение данных обучения или генерирование дополнительных данных, основанных на этих данных обучения. Расширение данных, как правило, применяется к данным датчиков (например, изображения или речевые данные), где преобразования могут имитировать изменения отдельных характеристик, например, каким образом были собраны данные.

В частности, изменение яркости изображения может имитировать различные условия освещенности, а изменение скорости и шага звуковой выборки может имитировать изменения скорости речи. Процессы увеличения объема данных обычно определяются преобразованием и значением интенсивности, которое выбирается случайным образом в процессе обучения каждого образца в каждой новой партии. Например, процессом расширения может быть поворот изображения и его интенсивность - степень вращения, случайным образом выбранная из равномерного распределения в диапазоне ± 30 градусов. Такие расширения могут выполняться до или после очистки и подготовки данных.

Однако, в отличие от очистки и подготовки данных, расширение данных не является детерминированным и не применяется к данным в процессе их вывода. Существует много дополнений данных процессов, которые могут применяться независимо или одновременно. Процессы дополнения, применяемые к набору данных, как и значения интенсивности, выбранные случайным образом, могут быть объединены в конвейер, такой как Keras ImageDataGenerator.

Кроме того, для добавления дополнительных обучающих данных можно использовать увеличение, генерацию или синтез данных по сценарию. Данные могут быть сгенерированы с использованием инструментов физического моделирования или DL-подходов, таких как GAN, подготовленный на данных обучения. Результатом этапа подготовки является конвейер данных, ручной или автоматизированный, который включает в себя все операции по очистке, предварительной обработке, расширению и генерации данных, используемых на следующем шаге «Модель» для обучения моделей и проверки гипотез.

Шаг Б. Модель рабочего процесса AI/ML

Шаг «Модель рабочего процесса AI/ML» включает в себя выбор подходящей модели для приложения, определения архитектуры модели и итераций в процессе обучения модели, настройки гиперпараметров и оценки модели. В общем случае компоненты выбора, обучения и настройки модели, автоматизированные с использованием различных механизмов, образуют так называемый контур AutoML. Достижение заданных результатов определяет необходимость выбора новой модели или если разработчику необходимо вернуться к предыдущему рабочему процессу и повторно проверить доступные данные. Этот рабочий процесс остается в основном аналогичным

для наземной и периферийной сред или космоса, но условия предполагаемой среды могут ограничивать доступные модели на выбор.

Рабочий процесс выбора модели, показанный на рисунке 4, начинается с определения разработчиком класса модели или классов, наиболее подходящих для поставленной задачи. Применимость модели зависит от нескольких факторов, включая тип и доступность данных, доступность маркеров, цель проблемы и ограничения, обусловленные предполагаемой операционной средой использования. Например, если маркированные данные недоступны, разработчик в общем случае не выбирает модели контролируемого обучения, для которых требуются такие данные. В этом случае разработчик, как правило, построит архитектуру модели с фреймворком машинного обучения, например TensorFlow, и выберет метод инициализации модели параметров и гиперпараметров (параметры, которые не извлекаются непосредственно из данных в процессе обучения).

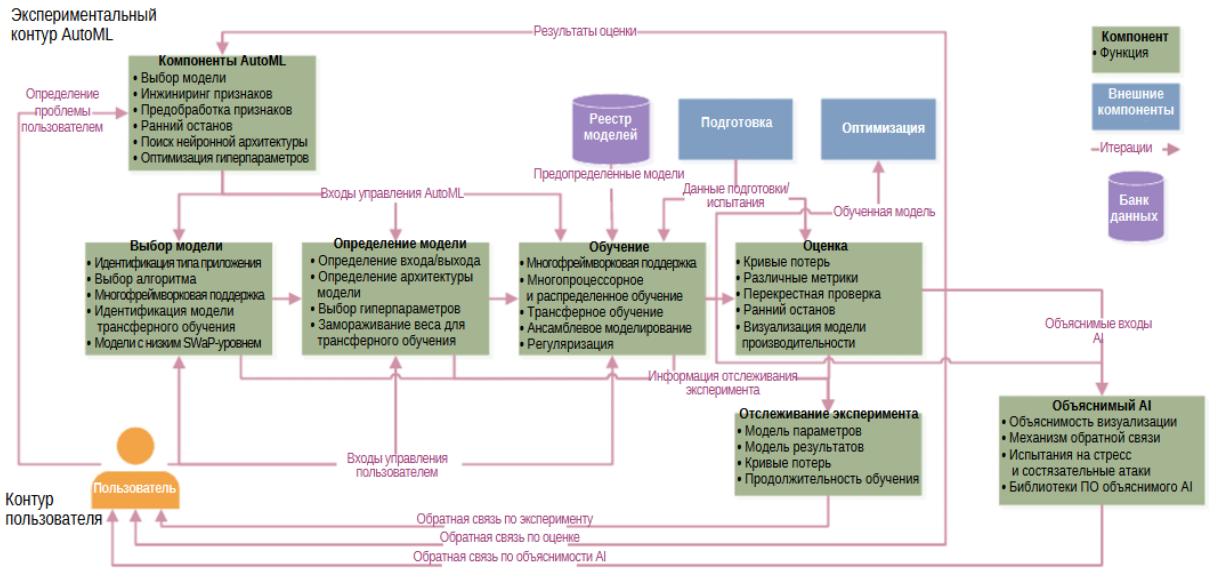

Рисунок 4. Структурно-логическая схема этапа выбора модели рабочего процесса AI/ML

Следующие шаги рабочего процесса AI/ML включают обучение модели по имеющимися данным, оценку результатов и настройку архитектуры модели или гиперпараметры при необходимости. Обучение относится к процессу предоставления модели с такими данными, что он пытается узнать оптимальные значения (веса) параметров модели для вывода. Это часто делается в сочетании с оценкой, когда необходимо оценить результаты (такие как точность, прецизионность и отзывчивость) с тем, чтобы уточнить, требуется ли дополнительная настройка и переобучение модели.

Настройка, как правило, относится к процессу изменения архитектуры модели, например количество слоев в нейронной сети или корректировка других гиперпараметров, включая размер партии, скорость обучения для тренировочных нейронных сетей или параметр регуляризации L1 в линейной модели Лассо. Этот процесс является итеративным, и для его реализации требуются квалифицированные специалисты с опытом машинного обучения в связи с тем,что производительность выбранного алгоритма существенно зависит от грамотного выбора гиперпараметров.

Различные компоненты выбора модели, цикла обучения и настройки могут быть автоматизированы с помощью разных подходов и, как правило, образуют контур AutoML. Как минимум средства AutoML обычно обеспечивают автоматическую настройку стандартной сетки или случайный поиск в пространстве предопределенных гиперпараметров (например, модуль Grid Search scikit-learn).

Исчерпывающий поиск в сетке может быть слишком ресурсоемким для решения ряда проблем с большой размерностью пространства поиска гиперпараметров, особенно при использовании перекрестной проверки в K-кратном порядке. Более продвинутые инструменты, такие как автопросмотр и AutoKeras, также способны автоматизировать предварительную обработку данных, выбор модели и конфигурации архитектуры модели (например, поиск нейронной архитектуры) и используют более эффективные методы поиска, например, такие как байесовская оптимизация.

Некоторые коммерчески доступные COTS программные платформы машинного обучения (ML) как Dataiku и автономного искусственного интеллекта H2O Driverless AI включают еще больше механизмов автоматизации и, зачастую, почти завершают наборы AutoML, позволяя с помощью интуитивно понятного графического интерфейса (GUI) импортировать данные, определять проблемы и сравнивать автоматически генерируемые модели. Однако такие более продвинутые и интуитивно понятные контуры AutoML, как правило, ограничены в применении и наилучшим образом работают по классификации и регрессии табличных данных с отдельными расширениями до нескольких зависящих от времени рядов, приложениями компьютерного зрения и обработки речевых сообщений.

После обучения и сравнения нескольких моделей для повышения их производительности и робастности используется подход, называемый ансамблевым моделированием. Ансамблевое моделирование относится к методам, предусматривающим

комбинирование большого количества слабых моделей для создания сильной модели (по типу «фасовка в мешки» или «наддув»). Эти методы часто непосредственно реализуются как их собственные модели, встроенные в рамках ансамбля в схему обучения. Однако подобное ансамблевое моделирование может также ссылаться на метод укладки нескольких сильных моделей для усреднения смещения каждой отдельной модели и создания более обобщенной и более высокопроизводительной модели, например сложение результатов из нескольких сверточных слоев, слоев пула и полностью связанных слоев архитектуры сверточной нейронной сети CNN (Convolutional Neural Networc) для достижения требуемого уровня производительности при выполнении задачи классификации изображений.

Выбор модели рабочего процесса AI/ML практически аналогичен определению зависимостей и особенностей приложения искусственного интеллекта и машинного обучения для развертывания в наземной, периферийной или космической среде, но, как правило, требует уточнения и учета дополнительных соображений в зависимости от среды развертывания.

В частности, если предполагаемая среда применения является периферийной, мобильной или космической, набор доступных вычислительных ресурсов существенно более ограничен и может работать разными способами, отличными от стандартного центрального или графического процессора. Производительность модели для этих сред, ограничения на обработку и выбор модели, подлежащей развертыванию на периферийном, мобильном устройстве или в космосе, необходимо учитывать более глубоко. С учетом ограниченных вычислений в такой среде разработчик должен учитывать размер модели, вычислительную сложность, требуемые скорость передачи данных и пропускную способность, ряд других факторов. Все они будут зависеть от доступных компонентов, аппаратных средств, решаемых задач и в зависимости от конфигурации будут иметь различную вычислительную мощность.

Еще одним логическим компонентом является обслуживание модели. В частности, регулярное обновление модели может быть затруднено, если целевое аппаратное обеспечение имеет ограниченную полосу пропускания двухсторонней (восходящей/нисходящей) линии связи. Ограниченная передача информации из космоса сокращает объем новых данных обучения, которые могут быть собраны для обновления модели. Активные методы обучения для выявления полезности данных с точки зре-

ния объема информации, полученной путем включения этих данных в набор учебных данных, позволяют определять, какие данные являются наиболее важными для нисходящей линии связи. Если выбранная модель является большой моделью глубокого обучения, она может не соответствовать размеру или временным ограничениям окон восходящего канала целевого аппаратного обеспечения. В этом случае вместо нее может быть использована более простая модель. При развертывании в различных средах разработчики должны учитывать доступные вычислительные ресурсы, доступность системы для обновления, а также возможности по эксплуатации и обслуживанию выбранной модели.

Оценка модели, как правило, проводится одновременно с итеративным обучением модели и включает анализ модели, непосредственное сравнение моделей, прозрачности и понятийной доступности процессов принятия решений — объяснимости искусственного интеллекта. Применимые методы оценки различаются в зависимости от категории и типа проблемы выбранной модели. Методы оценки в значительной мере обусловливают метрику или набор метрик, которые используются для оценки эффективности выбранной модели. Применимые метрики и рабочий процесс оценки, как правило, также могут отличаться в зависимости от предполагаемой среды развертывания.

Визуализация обучения и оценка функции потерь по выполнению итерации обучения позволяют определить, насколько хорошо сходилась модель глубокого обучения. Для контролируемых моделей классификации популярные метрики включают, но не ограничены точностью, прецизионностью, отзывами обратной связи, оценкой эффективности F1 и площадью под кривой (AUC) рабочей характеристики приемника (ROC). Визуализация по типу матрицы путаницы предоставляет более подробный обзор производительности модели, показывая количество истинно положительных, истинно отрицательных, ложно положительных и ложно отрицательных классов.

Для контролируемых регрессионных моделей такие метрики, как среднеквадратическая ошибка (RMSE) и средняя абсолютная ошибка (MAE), обеспечивают оценку производительности модели во всем наборе данных, в то время как остаточные диаграммы рассеяния могут визуализировать производительность как функцию входных параметров. Одни и те же метрики обычно не применяются к неконтролируемым моделям из-за разницы в выходе между алгоритмами. Неконтролируемые выходные

данные модели не сравниваются с достоверностью данных, поэтому метрики для контролируемых моделей в этом случае не применяются. Популярные метрики для кластеризации (в случае неконтролируемых моделей) включают индексы внутренней достоверности, такие как индекс силуэта, который предназначен для измерения когезии и разделения кластеров.

Объяснимый ИИ (Explainable AI- XAI) относится к методам и приемам машинного обучения, используемым для повышения прозрачности и понятности процессов принятия решений в системах AI/ML и в целом доверия к их рекомендациям и решениям. Признание эффективности систем AI/ML может быть существенно ограничено неспособностью пользователей в полной мере оценить и понять решения и действия этих систем. Благодаря лучшему пониманию логики прогнозирования и принятия решений системами искусственного интеллекта разработчики смогут лучше интегрировать эти знания в собственные процессы принятия инженерных решений. В свою очередь методы XAI позволяют выявлять и смягчать предвзятости в разрабатываемых моделях, повышая уровень объективности и целесообразности их использования, исправлять ошибки и в конечном итоге создавать более надежную и устойчивую систему AI/ML.

Одним из способов достижения более объяснимого ИИ является выбор моделей, которые по своей сути более объяснимы по сравнению с другими из-за их внутренней архитектуры. В частности, простые модели, такие как дерево принятия решений, линейная и логистическая регрессии, проще интерпретировать. Например, модели дерева решений принимают определенную последовательность решений для перехода к окончательному варианту, и при этом промежуточные решения могут быть проанализированы непосредственно для обоснования конечного результата. Для таких более простых моделей существуют методы расчета важности элементов, которые оказывают наибольшее влияние на принятие решений, интуитивно и непосредственно из модели. С другой стороны, нейронные сети, как правило, включают большое количество весовых коэффициентов, которые отражают разную степень влияния описываемых ими элементов на итоговое решение. Благодаря большому количеству этих весов и их нелинейным соотношениям, зачастую чрезвычайно сложно интерпретировать, как входные данные непосредственно влияют на итоговое решение системы. Для сложных моделей существуют некоторые методы объяснения, которые помогают

разработчикам лучше понять модели процесса принятия решения и включают метрики объяснимости, визуализации и пояснения прогнозирования. В частности, пакеты LIME (Local Interpretable Model-agnostic Explainations) и SHAP (SHapley Additive exPlanations) являются двумя так называемыми алгоритмами «черного ящика», которые даже для специалистов чрезвычайно сложно объяснить, и они являются агностическими по отношению к типу используемой модели.

Кроме того, для моделей глубокого обучения используются специфические, основанные на градиенте методы объяснения, такие как карты активации классов (Class Activation Maps - CAM) и Grad-CAM, способные выделять области в изображении, которые оказали наибольшее влияние на их классификацию. В другой модели на платформе Darwin AI's GenSynth используется контрафактный подход для определения входных данных (например, пиксели), удаление которых приводит к неправильной классификации.

Следует отметить, что в случае, когда предполагаемая среда использования системы AI/ML является периферийной или космической, могут потребоваться дополнительные методы оценки и оптимизации выбранной модели. Также в некоторых случаях проведение тщательной оценки моделей на месте до ее развертывания может оказаться невозможным. Однако использование обобщенных метрик, таких как количество параметров, размер модели в МБ, количество операций в GFLOPS и длина пути критических данных (CDL), способны дать представление о возможности развертывания модели на конкретной платформе и в определенной среде.

2.2 Оптимизация модели рабочего процесса AI/ML

Этап оптимизации моделирования рабочего процесса AI/ML является дополнительным и выполняется после обучения, оценки и выбора модели с использованием различных методов сжатия и ускорения. Целью этого этапа является упрощение модели для уменьшения ее размера на диске и сокращения задержек выполнения вывода и вычислительной мощности, используемой для вывода итогового решения.

Для моделей, развертываемых в наземной облачной среде, для которых существуют ограничения по уровням вычислительной мощности и энергопотребления, и которые вызывают мало опасений по поводу производительности, данный шаг С «Оптимизация», как правило, пропускается. Для моделей, развернутых на периферий-

ных/мобильных устройствах и в космической среде, оптимизация может дать существенные преимущества, а в ряде случаев может потребоваться для обеспечения эксплуатационных требований.

В общем случае существуют четыре основные категории оптимизации модели: оптимизация расчетных графиков, отсечение, квантование и аппаратная оптимизация. Оптимизация графиков вычислений и аппаратно-зависимая оптимизация, как правило, не изменяют принципиально результаты модели, и функциональные отображения входных данных в выходные данные должны быть эквивалентны до и после этих оптимизаций. С другой стороны, обрезка и квантование способны принципиально изменить функциональное отображение модели и результаты рабочего процесса.

Оптимизация вычислительных графов рассматривает сеть как направленный ациклический график (DAG) и пытается оптимизировать его путем исключения тензоров сквозного (no-ops) и нулевого (zero-dimension) выполнения операций, алгебраического упрощения, операторного слияния, констант-сворачивания, преобразования структуры данных и статического планирования памяти. Как правило, такие оптимизации не являются аппаратно-зависимыми и могут быть применены к высокоуровневому промежуточному представлению модели, такому как Relay и ONNX. Однако некоторые слияния оператора и преобразования структуры данных могут быть оптимизированы на основе целевых аппаратных средств.

В комплекте с набором обучающих данных для уменьшения общего размера модели без значительного снижения ее точности можно использовать методы отсечения узлов. Такие подходы к обучению со сжатием предполагают отсечение в модели таких операций, которые не оказывают существенного влияния на выходные результаты, но увеличивают объем и сложность модели. При этом следует учитывать, что только отсечение узла способно на порядок сократить количество параметров в таких моделях, как например AlexNet и VGG-16, с незначительным (<1%) влиянием на производительность.

Более совершенные методы обрезки могут изменять всю архитектуру модели для замены больших и, как правило, дорогостоящих частей вычислительных графов меньшими и более эффективными подграфами. Данные методы, например, аппроксимируют замененный подграф Generative Synthesis, созданный на платформе GenSynth компании Darwin AI и использующий пару «генератор-инквизитор» для генерации новых моделей, на которые распространяются пользовательские ограничения

и целевые показатели производительности. Для таких моделей, как ResNet-50 и InceptionV3, платформа GenSynth может сгенерировать модели с сокращенным на от 1/3 до 1/7 количеством параметров в исходной модели при сохранении в пределах нескольких процентов ее точности.

Другой метод «квантование» представляет собой мощный подход к сокращению объема и задержек модели за счет замены дорогостоящих 32-битных операций с плавающей запятой, как правило, необходимых для обучения, на 16-, 8- или даже 1-разрядными операции с фиксированной точкой. Например, только преобразование из FP-32 в INT-8 позволяет на 1/4 уменьшить исходный размер модели. Вместе с тем из-за уменьшения динамического диапазона при представлении модели с 8-битными целыми числами, может быть некоторое ухудшение производительности модели. Квантование с учетом обучения часто может обеспечить лучшие результаты точности, если обучающий набор данных доступен на этапе оптимизации модели. Использование программной платформы Latent Efficient Inference Platform позволяет выполнять квантование для сжатия модели как после ее обучения, так и с учетом процесса обучения. В частности, без обучающего набора квантование на платформе Latent AI’s после обучения модели позволяет достигать битовой глубины до 3 бит без значительного снижения точности (< 2%).

Аппаратные оптимизации могут включать элементы оптимизации графов вычислений, отсечения и квантования, адаптирования модели для вывода результатов на определенном устройстве. Используемые при этом методы оптимизации будут различаться в зависимости от типа аппаратных средств (CPU, GPU, FPGA, TPU) и их производителей (NVIDIA, Xilinx, AMD, Intel). Основной подход заключается в том, чтобы улучшить использование оптимизированных оператором «вручную» ядер, которые были разработаны для конкретных целевых аппаратных средств. Это достигается за счет использования библиотек глубокого обучения, таких как cuDNN для графических процессоров на основе CUDA, oneDNN для центральных и графических процессоров и FPGA компании Intel, Arm NN для центральных и графических процессоров Arm, а также MIOpen для графических процессоров AMD. Такие библиотеки данных часто могут быть реализованы в виде бэкэнда для фреймворков глубокого обучения и обеспечивают оптимизированные ядра для простых операций глубокого обучения (таких, как умножение и сложение), а также ядра более высокого уровня для глубоко-

го обучения специфичным операциям слияния, таким как 2D-свертка + ReLU + Batch Normalization. С этой целью для некоторых периферийных устройств модель обучения выполняется на устройстве с API-интерфейсом среды выполнения, таким как TensorFlow Lite, или со специальными библиотеками, такими как Android NN API для устройств Android и Core ML для устройств iOS.

Для других целевых аппаратных средств для преобразования модели обучения в механизм вывода в процессе выполнения требуются специализированные наборы (SDK) инструментов, библиотек, компиляторов, отладчиков. Необходимо отметить, что в случаях для периферийной/мобильной среды или космоса механизм вывода является одним из основных программных компонентов систем AI/ML, отвечающим за применение логических правил к базе знаний для вывода новых данных или принятия итогового решения. Например, комплект Vitis AI обеспечивает сжатие, квантование и синтез моделей глубокого обучения для выполнения вывода на основе программируемых пользователем матриц (FPGA) Xilinx. Аналогичным образом пакет TensorRT обеспечивает универсальную платформу для оптимизации на уровне графов, квантования и составление нейронных сетей на базе графических процессоров компании NVIDIA. Периодически аппаратная оптимизация проводится параллельно с компиляцией механизма вывода в процессе выполнения итерации как часть шага «Интеграция» рабочего процесса AI/ML.

Шаг D. Интеграция модели рабочего процесса AI/ML

На шаге «Интеграция» выполняются заключительные действия по переходу модели от проверки концепции к развернутой полноценной модели системы AI/ML. В общем случае эти усилия включают в себя компиляцию модели в механизм вывода в процессе рабочего цикла (runtime inference engine), который является программным продуктом, обеспечивает взаимодействие различных частей системы во время выполнения операций и может выполняться на различной компонентной базе. При этом формирование модели осуществляется путем разработки и интеграции вспомогательных услуг, необходимых для работы в целевой среде, а также методов обеспечения устойчивости для мониторинга данных и моделей в процессе производства.

На рисунке 5 представлена структурно-логическая схема оптимизации и интеграции модели рабочего процесса AI/ML.

Существует множество вариантов выполнения модели в качестве механизма вывода в

процессе выполнения рабочего цикла. Самый простой подход - развернуть прикладной интерфейс API - набор правил и спецификаций, которые позволяют различным программным компонентам взаимодействовать друг с другом и обмениваться данными в рамках модели. Интерфейс API упрощает разработки, позволяя программистам использовать готовые решения, может быть доступен через службу RESTful и развернут в той же среде, которая используется для обучения. Некоторые структуры, например TensorFlow Serving, имеют встроенную программную поддержку такого интерфейса. Другой подход заключается в использовании платформы развертывания модели, которая организует развертывание и поддержку моделей глубокого обучения в масштабируемой вычислительной среде. К инструментам подобного типа, обеспечивающим функции модели, относятся Dataiku17, MLflow18 и Modzy19.

Для развертывания подобной модели в периферийной/мобильной или космической среде требуются дополнительные шаги для создания механизма вывода. При этом, как правило, аппаратные и программные библиотеки ускорения, используемые для обучения модели, не задействуются на периферийных устройствах или космических платформах.

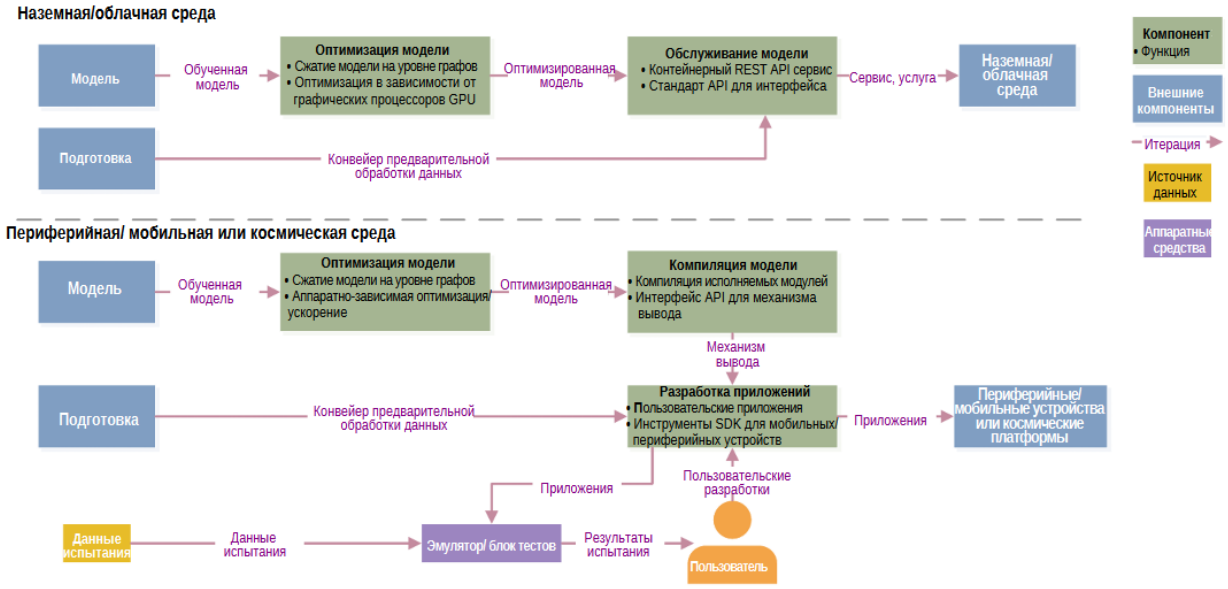

Рисунок 5. Структурно-логическая схема оптимизации и интеграции модели рабочего процесса AI/ML

В этом случае модель должна быть сначала скомпилирована для запуска на целевом оборудовании путем преобразования операций модели на низкоуровневом языке

программирования C, C++ или с использованием библиотеки нейронной сети, которая уже скомпилирована для запуска на целевом аппаратном обеспечении и реализована через интерфейс API. Несмотря на то что конкретные детали этого процесса могут различаться в зависимости от конкретной библиотеки, общий поток рабочего процесса сохранится в основном в том же виде.

Первоначально модель экспортируется или преобразуется в формат, который может быть считан библиотекой среды выполнения. После этого некоторые библиотеки предоставляют такие возможности оптимизации, как операции с предохранителями и использование высокооптимизированных ядер глубокого обучения, поддерживаемых на целевом оборудовании. Наконец, механизм вывода среды выполнения экспортирован как исполняемый файл на целевом оборудовании. TensorRT, VitisAI, и TFLite являются примерами таких библиотек, которые предлагают оптимизацию и компиляцию в одном пакете.

В то время как многие из этих библиотек рабочего цикла выполнения являются аппаратно специфичными или предлагают ограниченную поддержку для различного целевого оборудования, существуют более недавние усилия по разработке компиляторов глубокого обучения, которые обобщают оптимизацию и компиляцию модели глубокого обучения для широкого набора целевого оборудования. Так, в основе компиляторов глубокого обучения (DL) лежит промежуточное представление (IR). Компиляторы DL могут реализовывать несколько уровней IR. Для IR высокого уровня модель обычно представлена в виде вычислительного графа, где оптимизация может быть выполнена неаппаратными методами.

Низкоуровневое промежуточное представление обеспечивает более мелкозернистое представление вычислений, которое позволяют оптимизировать аппаратные средства, включая распределение памяти, оптимизацию посредством последовательности циклических преобразований, и распараллеливание. После этого промежуточное представление может быть обработано одним из многих бэкэндов, которые заменяют низкоуровневые IR-инструкции с аппаратными характеристиками и ядрами из доступных библиотек ускорения цикла. При этом может существовать огромное количество параметров для настройки аппаратной оптимизации перед компиляцией.

Согласно принятым методикам, автоматическое конфигурирование специфичных аппаратно-зависимых оптимизаций называется автоматической настройкой, и эти

подходы варьируются в зависимости от параметризации конфигурации, модели затрат и технологии поиска. В частности, TVM является компилятором глубокого обучения Apache, реализующим промежуточное представление RelayIR графа вычислений - низкоуровневого IR на основе галогенида и модели затрат на основе машинного обучения - для автоматической настройки. Другие примеры современных компиляторов глубокого обучения включают nGraph компании Intel, Tensor Integrations и Glow от Facebook и TensorFlow XLA компании Google.

После компиляции модели ее оптимизированный механизм вывода можно развернуть на целевом оборудовании, но этот шаг, как правило, не является завершением развития модели рабочего цикла. Наряду с механизмом вывода, ряд других вспомогательных сервисных приложений и программного обеспечения может потребоваться для обеспечения реализации и функционирования развернутой модели. В число таких услуг могут входить потоковая и предварительная обработка данных, постобработка результатов моделей, мониторинг данных и моделей. Эти сервисы могут быть особо важными для периферийных и космических приложений, где среда развертывания существенно отличается от среды разработки, когда, например, стандартного набора инструментов и программного обеспечения не существует. Кроме этих услуг, дополнительные функции могут потребоваться для поддержки операций машинного обучения MLOps, например, реестра модели или хранилища функций. Все такие услуги должны быть разработаны, интегрированы, протестированы и скомпилированы для целевого оборудования до готовности всей системы к развертыванию в конкретной среде.

Для развертывания в космической среде, которая негативно влияет на бортовое оборудование, датчики и системы КА и соответствующим образом определяет входные данные для модели глубокого обучения, особый интерес представляет надежный подход к оценке поддержки для оценки устойчивости и непрерывности обеспечения требуемого уровня производительности модели. В ряде случаев может потребоваться дополнительный мониторинг, например, дрейфа данных, где распределение входных данных меняется с течением времени; дрейф модели прогнозирования, где распределение модели прогнозирования меняется с течением времени; ухудшение производительности модели, когда точность модели с течением времени дает отклонения результатов или сообщения о снижении достоверности.

Наряду со случайными изменениями в эффективности данных и моделей, развернутые модели машинного обучения находятся под угрозой нападений с целью целенаправленного обмана и дискредитации систем AI/ML. В этих случаях дополнительные меры гарантии, такие как ансамблевое моделирование и методы верификации и валидации, могут обеспечить определенную защиту от подобных атак. Кроме того, применительно к космической среде развертывания, применение ограничительных операций, используемых в моделях глубокого обучения, и аппаратных средств их ускорения позволит расширить возможности по устранению отказов, специфичных для механизма вывода в моделях глубокого обучения.

Шаг E. Сертификация модели

Предыдущие разделы охватывали стандартный рабочий процесс для разработки и интеграции систем AI/ML с аппаратным обеспечением для конкретной среды и некоторые необходимые вспомогательные услуги. Каждый из компонентов в системе AI/ML необходимо испытать в подходящей среде для подтверждения выполнения ключевых эксплуатационных требований и оценки безопасности/риска системы. Проведение подобных испытаний имеет решающее значение для космической среды, где может возникнуть сбой или непреднамеренное решение, которые обойдутся в миллиардные потери и/или приведут к человеческим жертвам.

В отношении программного обеспечения и аппаратных средств, предназначенных для космических аппаратов такие испытания являются обычной практикой, полностью учитывающей требования принятой NASA и Министерством обороны США системы уровня технологической готовности (TRL) и используемой для конкретизации объема требуемых разработок и испытаний. Однако развертывание систем AI/ML сопряжено с дополнительным рисками, учитывающими сложный и в значительной мере недетерминированный характер моделей AI/ML, которые часто рассматриваются как условные «черные ящики», что делает логику принятия каждого решения непрозрачной, а зачастую и непонятной.

Необходимость обширных испытаний дополнительно подчеркивается тем, что в составе вычислительных ресурсов на борту космических аппаратов могут быть процессоры совершенно другого типа вместо тех, которые использовались для обучения модели. В частности, достаточно сложно предсказать, как модель рабочего цикла AI/ML будет работать на программируемых вентильных матрицах FPGA по сравне-

нию с тем, как она выполняется на графических (GPU) или центральных (CPU) процессорах. Для любой другой среды также требуется подобное строгое и тщательное тестирование уровней потребления ресурсов, взаимодействия между системой и средой, а также устойчивости и уязвимостей системы безопасности.

Каждый из компонентов или артефактов в системе AI/ML должен быть протестирован в условиях, максимально соответствущих предполагаемой среде развертывания. Для наземной среды это относительно просто, так как системы AI/ML уже традиционно работают в этой среде и существуют хорошо зарекомендовавшие себя методы их оценки. Широкодоступные облачные вычисления в наземной среде не только уменьшают потребность в оптимизации моделей по различным физическим ограничениям, но и позволяют запускать теневые модели (модели, которые проходят тестирование в режиме реального времени, но результаты которых сравниваются только с фактическим результатом и текущей версией реально развернутой модели) с непрерывным мониторингом.

Для периферийной, мобильной и особенно космической сред добиться тестирования в режиме реального времени сложнее. Ограниченные возможности по развертыванию в этих средах теневых моделей во время работы текущей версии модели является очень сложным, а зачастую и невозможным. В этом случае ситуация или период простоя сервиса для запуска теневой модели должны быть разрешены до тех пор, пока не будет продемонстрирована ее удовлетворительная работа или теневая модель должна быть запущена на Земле на аналогичном оборудовании и данных. При последнем способе развертывание и использование теневой модели осложняются проблемами временной задержки, обусловленными особенностями линий связи с космическими аппаратами.

Часть сертификации системы AI/ML должна включать разработку профилей данных обучения и тестирования для выявления аномальных данных и уровня предвзятости, которые, как правило, имеют место на этапах подготовки и моделирования рабочего процесса. Кроме того, высококачественные наборы данных для космической среды достаточно сложно получить и приходится полагаться на расширение данных или моделирование, что потенциально вносит неизвестные уязвимости. Кроме того, производительность системы AI/ML будет естественно ухудшаться с течением времени, независимо от учета дрейфа данных, изменений в конвейере данных или повреждения модели весов в памяти.

Например, фаза, амплитуда и относительное положение отдельных элементов в массиве фазированной решетки может изменяться из-за вибрации или других воздействий окружающей среды; сигнал, поступающий в матрицу, может быть результатом воздействия внешних механических и/или естественных источников шума. Результатом влияния этих двух факторов станет принятый сигнал со статистически значимыми различиями во времени. Смещение данных может привести к тому, что модель будет работать в соответствии с требованиями выполнять не лучше, чем случайное угадывание.

Частью процесса сертификации системы AI/ML должно быть понимание того, каким образом может произойти деградация модели, каковы признаки того, что при этом происходит, и что является наилучшим алгоритмом действий, когда это происходит. Такой процесс тестирования может включать в себя моделирование дрейфа данных для наблюдения за тем, как модель выполняет распределение данных до ее обучения, и с этой позиции определить границу того, когда результаты модели не должны быть доверенными.

В целом сертификация модели работы систем AI/ML является жизненно важным процессом снижения рисков и производства программных продуктов, в полной мере соответствующих высоким требованиям конечных пользователей. Стандартизация методов тестирования и валидации позволит автоматизировать сертификации в конвейере операционного машинного обучения (MLOps).

3 ОСОБЕННОСТИ ОПЕРАЦИЙ МАШИННОГО ОБУЧЕНИЯ В СИСТЕМАХ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА НА КОСМИЧЕСКОЙ ТЕХНИКЕ

3.1 Операции машинного обучения MLOps

В общем случае под операциями машинного обучения MLOps понимают совокупность практик непрерывной интеграции (CI), непрерывной поставки (CD), непрерывного обучения (CT) и непрерывного мониторинга (CM), которые позволяют автоматизировать и упростить процессы разработки, развертывания и поддержки моделей машинного обучения. По сути эти практики являются вариантом DevOps-технологий непрерывной разработки ПО для машинного обучения, направленным на стандартизацию и оптимизацию жизненного цикла модели AI/ML от ее разработки до внедрения и развертывания.

Использование операций MLOps позволяет, в частности, автоматизировать рутинные задачи, такие как подготовка данных, обучение, тестирование и развертывание, что сокращает продолжительность цикла выпуска модели. Операции MLOps включают мониторинг моделей, в том числе контроль версий моделей и данных, отслеживание их производительности и автоматическое переобучение при необходимости, что повышает стабильность работы и устойчивость модели.

Непрерывная интеграция (CI) включает создание, тестирование и упаковку по мере продвижения нового кода в репозитарий исходного кода. Для систем AI/ML этот шаг может также включать создание артефактов, необходимых для выполнения в среде целевого объекта, построение и тестирование совместимости с целевой средой и тестирование для случая, если обучение сходится, вход в ступень CI является модельным автоматизированным конвейером, а механизм выхода будет тем же конвейером, упакованным для целевой окружающей среды.

Непрерывная поставка (CD) включает в себя перемещение новых пакетов вручную или автоматически к целевой среде по мере их появления. Существуют два этапа непрерывной поставки с системой AI/ML. Первый пакет, содержащий автоматизированный конвейер модели, будет перемещен в целевую среду. Затем автоматизированная модель конвейера будет работать и служить моделью в качестве службы или приложения. В операции машинного обучения (MLOps), наряду со стандартным пакетом DevOps-практик, входят непрерывное обучение (CT) и непрерывный мониторинг (CM). Непрерывное обучение предусматривает обучение модели в процессе ее обслуживания для поддержания приемлемого уровня производительности, что также

реализуется автоматизированной моделью конвейера. Методика непрерывного мониторинга предполагает мониторинг входных данных и результатов вывода модели для аномалий или их дрейф. Наземные/ облачные операции обучения MLOps являются самыми простыми из-за наличия мощных вычислительных ресурсов, доступности и относительной простоты осуществления мониторинга.

3.2 Особенности операций машинного обучения MLOps для наземной среды

На рисунке 6 в обобщенном виде представлены поэтапные шаги и артефакты операций машинного обучения MLOps для наземной среды. В общем случае MLOps можно разделить на два основных компонента: этап разработки и этап развертывания. При этом непрерывная интеграция и непрерывное обучение выполняются на этапах разработки и развертывания соответственно. Непрерывная поставка осуществляется в процессе этих двух этапов, а именно на этапе разработки поставляется автоматизированный конвейер модели, а на этапе развертывания — сама модель. Непрерывный мониторинг выполняется на этапе развертывания.

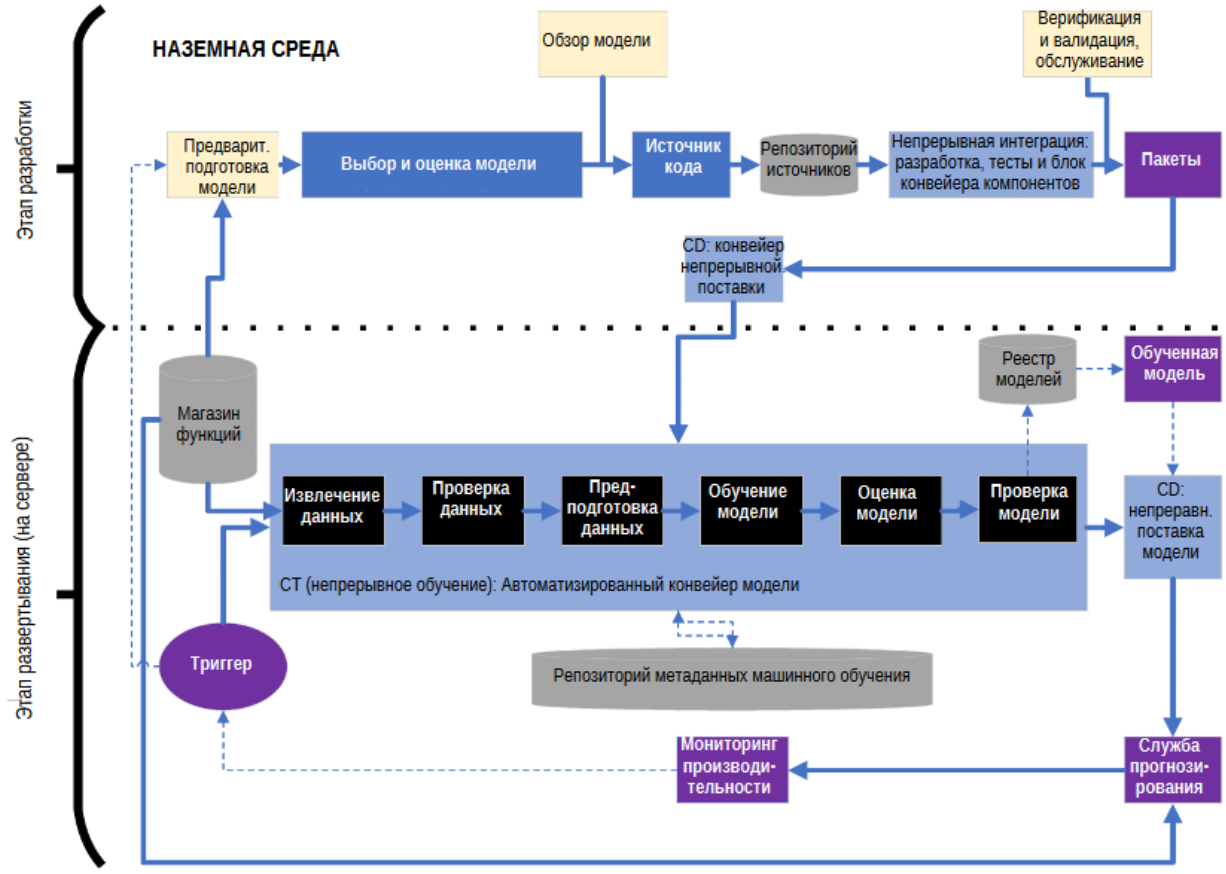

Рисунок 6. Структурно-логическая схема развертывания компонентов машинного обучения MLOps в наземной среде

Шаги подготовки, моделирования, оптимизации и интеграции, описанные в предыдущих разделах, выполняются на этапе разработки. Эти рабочие процессы обычно включают в себя исследовательский анализ, и в любой момент разработчик может перенаправить на предыдущий шаг или рабочий процесс.

Как правило, разработка начинается с подготовки данных и продолжается выбор модели, обучение и оценка. На этапе подготовки данных входные данные модели должны храниться в хранилище элементов, которое также будет доступно на этапе развертывания. Хранилище функций - это централизованный репозиторий для организации, стандартизации, хранения и обеспечения доступа к функциям для обучения и обслуживания. После оценки модель должна быть проверена с позиций соблюдения минимальных стандартов. После подтверждения соответствия этим стандартам исходный код модели сохраняется в репозитории, из которого он может быть извлечен для развертывания. На этом этапе начинается непрерывная интеграция (CI), которая охватывает рабочий процесс развертывания модели. Процесс развертывания модели включает преобразование исходного кода модели в формат, совместимый с целевым процессором, и разработку артефактов для поддержания и мониторинга модели.

Артефакты системы AI/ML тестируются для обеспечения надлежащего выполнения функций и работы и, как правило, включают:

• автоматизированный модельный конвейер, который выполняет следующие процессы:

- извлечение данных из хранилища элементов;

- проверку данных;

- подготовку данных для обучения;

- обучение и оценку новой модели;

- проверку производительности новой модели по сравнению с предыдущими версиями;

- включение оптимальной модели в сервис прогнозирования;

• реестр моделей, в котором хранятся предыдущие версии модели, созданные автоматизированным конвейером моделей;

• хранилище метаданных для отслеживания предыдущих запусков автоматизированного конвейера модели для обеспечения воспроизводимости, сравнения и анализ ошибок;

• система мониторинга для отслеживания различных статистических данных о производительности моделей;

• триггерная система для определения времени запуска автоматизированного конвейера модели или перезапуска цикла разработки. Механизм перезапуска определяется производительностью модели и/или графа.

После разработки артефакты тестируются на предмет обеспечения правильной работы и требуемой производительности. После этого проводится окончательный обзор перед выпуском упакованного автоматизированного конвейера модели для доставки в целевую среду.

Основные шаги на этапе разработки для наземной, периферийной и космической сред преимущественно одинаковы, а отдельные отличия в уровне рабочего процесса обсуждались в предыдущих разделах. Как правило, основные отличия заключаются в типах артефактов, построенных в процессе непрерывной интеграции, и рассматриваются в следующем разделе.

На этапе развертывания упакованный автоматизированный конвейер модели встраивается в целевую целевую среду. С этой целью автоматизированный конвейер модели после его построения тренируется и предоставляет обученную модель для службы прогнозирования. Этапы MLOps, отражающие процесс автоматизации конвейера машинного обучения, представлены в таблице 3.

Таблица 3.

Основные этапы операций MLOps автоматизации машинного обучения

Фаза операций MLOps	**Выход исполнения этапа**
Разработка и экспериментирование (алгоритмы MLOps, новые модели машинного обучения)	Исходный код для конвейеров; извлечение данных, проверка, подготовка, обучение и оценка модели, испытание модели
Непрерывная интеграция конвейера (сборка исходного кода и запуск тестов)	Компоненты конвейера, которые необходимо развернуть: пакеты и исполняемые файлы
Непрерывная поставка конвейера (развертывание конвейеров в целевой среде)	Развернутый конвейер с новой версией реализации модели
Автоматизированный запуск конвейеров в процессе рабочего цикла (используются расписание или триггер)	Обученная модель, которая хранится в реестре моделей
Модель непрерывной доставки (модель прогнозирования)	Развернутая служба прогнозирования (например, такая модель, как REST API)
Непрерывный мониторинг (сбор данных о производительности модели на основе реальных данных	Триггер для выполнения конвейера или начала нового цикла эксперимента

После обучения и проверки новой модели оптимальная производительность службы прогнозирования обновляется с помощью новой модели, и цикл рабочего процесса продолжается. Эта фаза наиболее варьируется для всех основных целевых сред развертывания систем AI/ML из-за особенностей и ограничений доступных вычислительных ресурсов и систем связи.

3.3 Особенности операций машинного обучения MLOps для периферийной/мобильной среды

Развертывание системы AI/ML в пограничной или мобильной среде усложняет выполнение операций машинного обучения MLOps. В этом случае вместо этапа развертывания, полностью реализуемого в наземной среде на подключенных серверах, MLOps разделяются как на сервер, так и на периферийные или мобильные устройства. Для упрощения изложения в оставшейся части исследования ссылка будет делаться только на периферийные устройства в связи с тем, что практики MLOps для периферийной, мобильной и миниатюрной портативной сред практически одинаковы.

Разделение MLOps на этапе развертывания между сервером и периферийными устройствами зависит от вычислительных требований системы AI/ML и вычислительной мощности периферийного устройства.

На рисунке 7 представлена структурно-логическая схема операций машинного обучения MLOps для периферийной среды с тремя различными уровнями сложности развертывания. В частности, начальный уровень Lvl.0. предполагает простейшее развертывание, при котором периферийное устройство обеспечивает только сбор данных и выполнение незначительной обработки данных перед передачей на сервер, на котором размещены остальные артефакты и сервисы.

Характерной особенностью уровня развертывания LV 1 является использование либо достаточно простой службы прогнозирования, либо достаточно мощного периферийного устройства для размещения службы прогнозирования. Этот уровень развертывания требует создания в процессе фазы развития двух новых дополнительных артефактов: системы локального протоколирования показателей производительности модели и менее развернутой версии хранилища функций. Целью использования этих двух артефактов является сокращение связи между сервером и периферийным устройством, что в свою очередь позволит увеличить скорость, с которой может быть выдано предупреждение о снижении производительности, и сократить время ожидания прогнозирова-

ния. Автоматизированный конвейер модели и другие артефакты, необходимые для непрерывного обучения, остаются на сервере хоста.

На уровне развертывания Deploy Lvl. 2 не добавляется никаких других артефактов, но модель достаточно упрощена или ресурсы периферийного устройства являются достаточно мощными и обеспечивают размещение на нем в полном объеме автоматизированного конвейера модели. Этот уровень развертывания позволяет минимизировать время реакции между замеченным ухудшением модели и автоматическим предоставлением новой модели.

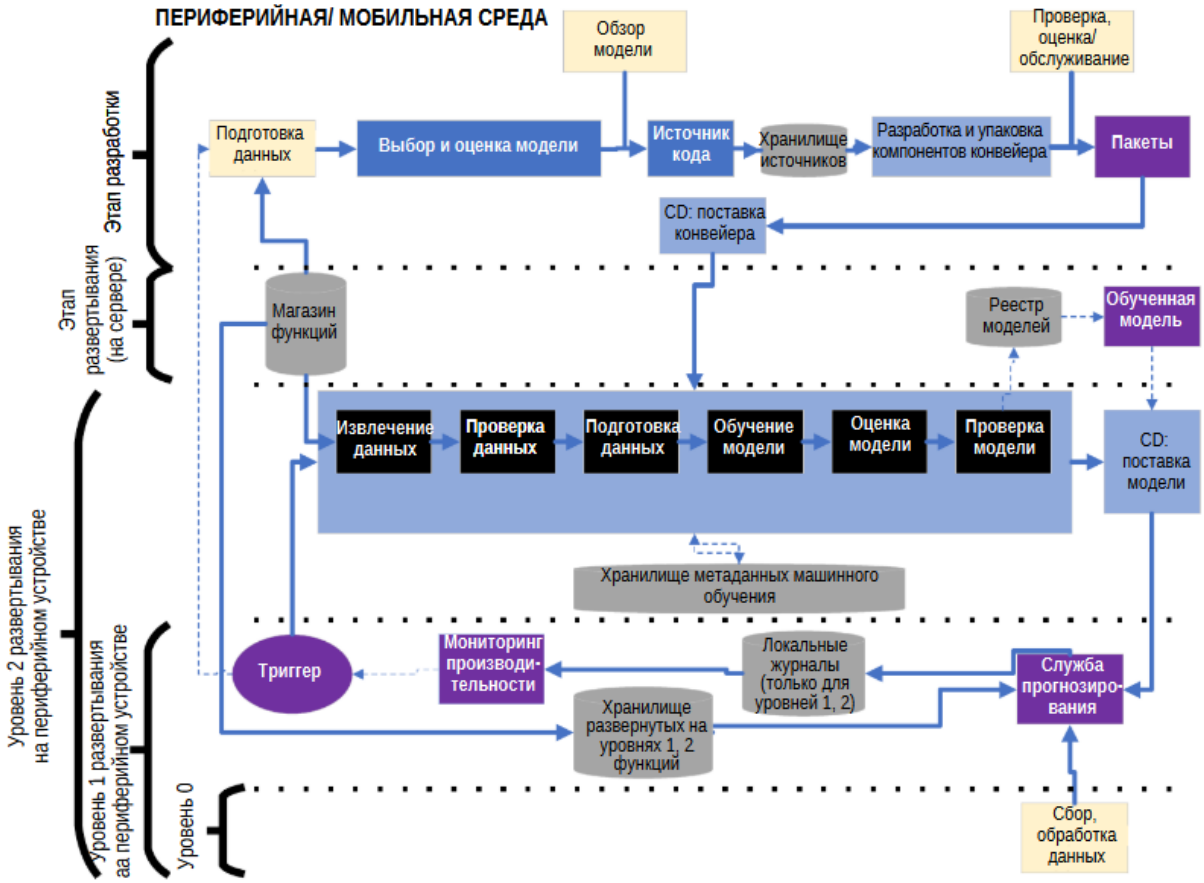

Рисунок 7. Структурно-логическая схема развертывания компонентов машинного обучения MLOps в периферийной или мобильной среде

3.4 Особенности операций машинного обучения MLOps в космической среде

Проведение MLOPS в пространстве требует устранения нескольких дополнительных усложняющих факторов. В космической среде мало того, что доступная вычислительная мощность чрезвычайно ограничена и совместно используется во многих приложениях, существуют тепловые и энергетические рабочие циклы, ограниченная телеметрия и вызванные внешней радиацией неисправности, с которыми необходимо бороться. При этом бортовые системы AI/ML на космических аппаратах должны быть изготовлены и работать в рамках оперативных заданий и концепции целевого назначения КА.

Из-за ограниченного количества вычислений, доступных на космическом аппарате, как правило, невозможно развернуть полностью автоматизированный конвейер моделей, способный обучать и доставлять новую модель на борт. Даже если бортовой процессор КА был бы способен работать по обучению модели AI/ML, частые и длительные периоды обучения будет мешать выполнению на борту других операций обработки данных, создавая предпосылки к срыву целевой миссии. Кроме того, для обучения моделей требуются большие объемы данных, что в свою очередь может вызвать проблемы с использованием доступного на борту КА ограниченного хранилища данных. Реально возможны только самые простые встроенные с бортовыми компьютерами схемы обучения, такие как онлайн-обучение простых моделей.

В качестве обходного пути преодоления подобных ограничений на борту космических аппаратов можно использовать два автоматизированных конвейера модели для систем AI/ML, как это показано на рисунке 8. Первый из этих конвейеров будет ставший на современном этапе традиционным автоматизированный конвейер модели, развернутый на сервере, способный обрабатывать поставку новой модели, начиная с этапа извлечения данных вплоть до проверки модели.

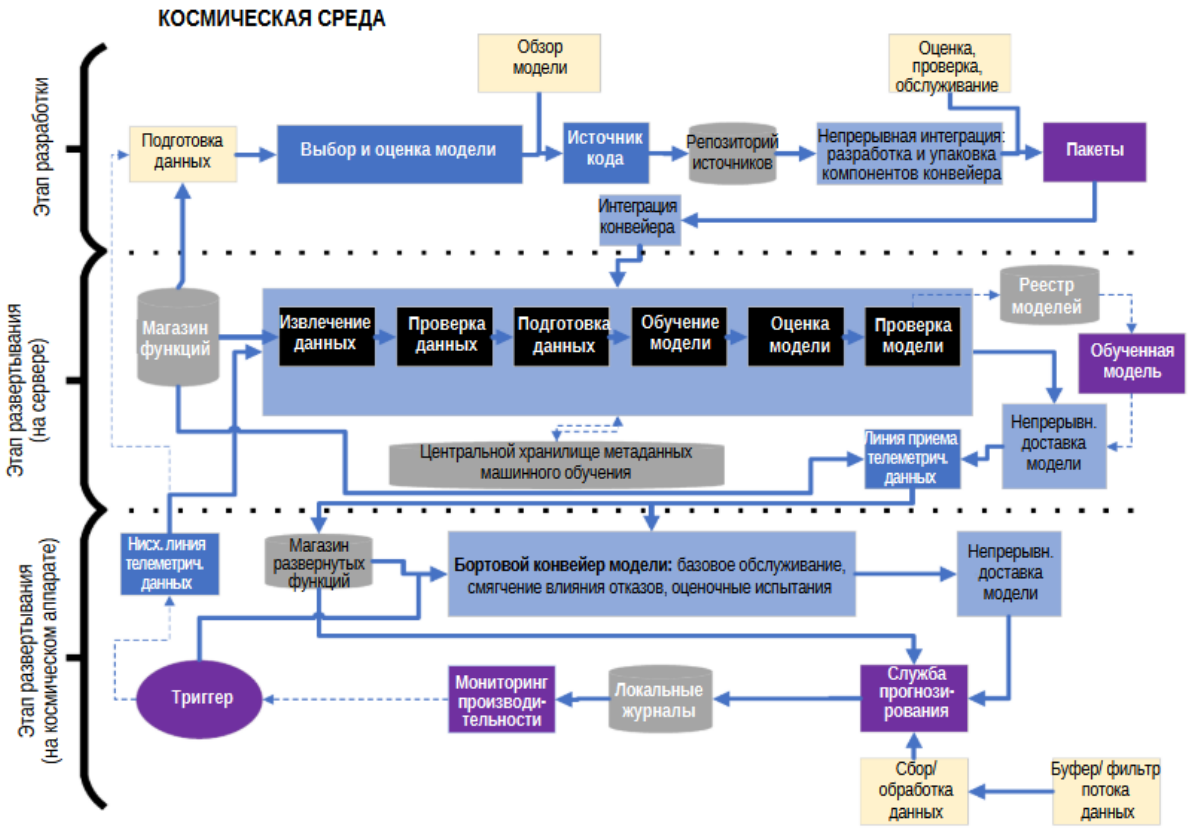

Рисунок 8. Структурно-логическая схема развертывания компонентов MLOps в космической среде

Второй, работающий на борту КА конвейер представляет собой упрощенный вариант базового, развернутого на сервере. Назначением бортового конвейера модели является базовое обслуживание модели, устранение неисправностей и проверка эффективности модели до того времени, пока по восходящей линии связи в бортовую аппаратуру КА не будет загружена и подключена новая модель.

В дополнение к бортовому конвейеру на КА требуется внедрить ПО обработки шаблонных обновлений восходящего (приема от других КА и средств) и нисходящего (передачи на Землю) каналов телеметрических данных. Для значительной части космических средств существует ограниченное время связи для передачи данных и план поддержания модели этих линий должен строиться вокруг этих возможных окон связи. Разработчикам потребуется детально рассмотреть предлагаемые стратегии с тем, чтобы сделать передаваемые обновления модели достаточно малыми, соответствующими одной попытке передачи за один сеанс связи, или создать возможность отправлять большие обновления за несколько сеансов связи.

Наряду с этим необходимо предусмотреть возможности передачи обратно на Землю различных служебных и дополнительных данных. В частности, переподготовка модели MLOps может потребовать сбора и передачи дополнительных данных, предназначенных для обработки на борту.

В космической среде факторы ограниченной вычислительной мощности и меняющихся рабочих циклов в значительной мере определяют подход, в рамках которого любая система AI/ML, вероятнее всего, не будет постоянно испытывать дрейф модели и смещаться. Все компоненты системы AI/ML должны быть спроектированы для обработки прерываний, вызванных планированием нормальной обработки заданий, потерей мощности из-за перегрева и воздействия внешней радиации. Каждый шаг процесса машинного обучения MLOps должен быть предназначен для выполнения только в том случае, если имеется достаточно ресурсов для завершения этого шага, но при этом требуется встроить дополнительные методы обработки незавершенного выполнения шагов MLOps.

Выполнение операций MLOps в космической среде значительно усложняет модель, но это является абсолютно необходимым элементом. Отдельные из этих дополнительных проблем обусловлены необходимостью создания дополнительных артефактов для поддержания, обслуживания и планирования усовершенствований системы AI/ML. Наиболее вероятно, что основным источником этой проблемы станет план эксплуатации, мониторинга и обновления системы в условиях ограниченности существующих концепций целевых задач и операций КА, описывающих, каким конкретно образом система и/или операции будут использоваться и поддерживаться для достижения требуемых целей. Основной вопрос не в том, ухудшится ли производительность системы AI/ML со временем, а в том, когда это произойдет. Различные системы AI/ML будут деградировать с различной скоростью, но это в любом случае это гарантированно произойдет, и процедуры их верификации, валидации и обслуживания должны в полной мере соответствовать требованиям обеспечения целостности и достоверности критически важных решений для автономных средств.

4 ОСОБЕННОСТИ ВЫБОРА ЭЛЕКТРОННЫХ КОМПОНЕНТОВ ДЛЯ СИСТЕМ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА НА КОСМИЧЕСКОЙ ТЕХНИКЕ

4.1 Обоснование выбора процессоров для космических систем AI/ML

Для обоснования выбора электронной компонентной базы для систем искусственного интеллекта и машинного обучения на космических аппаратах сравнение основных элементов вычислений — центральных (CPU), графических (GPU) процессоров и программируемых вентильных матриц (FPGA) — проводится по следующим эксплуатационным параметрам:

- пропускная способность;
- временная задержка:
- интеграция датчиков — процесс извлечения данных с датчика, выполнение предварительной обработки данных и доставка результатов в модели AI/ML;
- простота обновлений — сложность процесса, требуемого для изменения функциональности развернутой модели AI/ML;
- радиационная стойкость — уровень допуска каждого электронного элемента бортового оборудования по радиационной стойкости и коммерческой доступности;
- простота разработки — сложность развертывания модели на конкретно каждом типе электронного оборудования;
- вычислительная мощность — производительность и эффективность вычислений по уровню энергопотребления для конкретно каждого типа электронного оборудования.

Развертывание моделей ИИ на борту КА является сложной задачей, поскольку общие радиационно-стойкие космические процессоры имеют крайне ограниченные характеристики производительности, отставая на несколько поколений от коммерческих COTS-технологий. Процесс обеспечения радиационной стойкости сложных вычислительных устройств требует значительных финансовых и инженерных инвестиций и обязательно с серией испытаний на установках с жестким радиационным излучением.

В таблице 4 представлены широко распространенные радиационно-стойкие процессоры и примеры высококлассных миссий, в рамках которых они были использованы.

Таблица 4.

Современные радиационно-стойкие процессоры

Радиационно-стойкие процессоры	Описание, характеристики	Космические полеты, миссии
RAD6000	32-разрядный радиационно-стойкий одноплатный компьютер, построенный на базе RISC-процессора с архитектуры Power1 компании IBM, тактовая частота — 33 МГц, быстродействие — 35 MIPS, технология производства — 0,5 мкм	По состоянию на 2008 г. насчитывалось более 200 КА с RAD6000, в том числе: марсоходы Spirit и Opportunity, КА Deep Space 1, орбитальный аппарат Mars Odyssey, телескоп Spitzer
RAD750	Радиационно-стойкий одноплатный компьютер на базе одноименного процессора. Является приемником RAD6000, базируется на архитектуре семейства процессоров PowerPC 750 компании IBM, тактовая частота 110-200 МГц, производительность 266-400 MIPS, технология производства — 250-150 нм	По состоянию на 2022 г. с Земли запущено около 300 КА, использующих RAD750. Наиболее известные миссии: Deep Impact (2005), Mars Reconnaissance Orbiter (2020), WorldView-1 (2007), Curiosity Perseverance Rovers (2011), James Webb Space Telescope (2021), Europa Clipper (2024)
CAES Gaisler GR712RC	Радиационно-стойкий одноплатный компьютер на базе двух 32-разрядных процессоров с двумя ядрами LEON3FT SPARC V8, тактовая частота 100 МГц, производительность — 5,6 GOPS - 3,7 GFLOPS, технология производства — 180нм	Кубсаты DART и LICIA, Artemis-I и ArgoMoon
CAES Gaisler GR740	Радиационно-стойкая вариант системы на кристалле LEON4FT SPARC V8. Является приемником GR712RC, построен на базе 4-х ядерного 64-разрядного отказоустойчивого процессора LEON4FT компании CAES, тактовая частота 250 МГц, производительность 1700 DMIPS, технология производства — 20 нм	КА Copernicus, телескоп Nancy Grace Roman
RAD5545 SpaceVPX	Радиационно-стойкий одноплатный компьютер на базе 64-разрядного процессора с 4-мя ядрами RAD5500, построенный по архитектуре QoriQ PowerPC e5500v, производительность — 5,6 GOPS - 3,7 GFLOPS, технология производства — 45 нм	Микроэлектроника категории 1А для КА космических сил США. Планируется на окололунную автоматическую станцию Lunar Gateway

Стремление увязать этапы проектирования радиационно-стойких вычислительных устройств с циклами их разработки зачастую приводит к тому, что в создаваемых электронных компонентах используются архаичные архитектуры и устаревшие тех-

нологии изготовления с большими габаритами и низкой тактовой частотой, что делает их менее производительными и более энергозатратными по сравнению с передовыми коммерческими технологиями.

На рисунке 9 продемонстрирован существующий разрыв между радиационно-стойкими и коммерческими технологиями в зависимости от показателей вычислительной плотности, которые отражают статическую производительность для потока независимых целочисленных (Int8, Int16, Int32) и плавающих (SPFP - однородная точка с одинарной точностью, DPFP - однородная точка с двойной точностью) операций для современных радиационно-стойких (RAD750, Gaiser GR740 и RAD5545) и коммерческих встроенных процессоров (Xilinx Zynq 7020 и Intel Core i7-4610Y) в миллионах операций в секунду (MOPS) на логарифмической шкале.

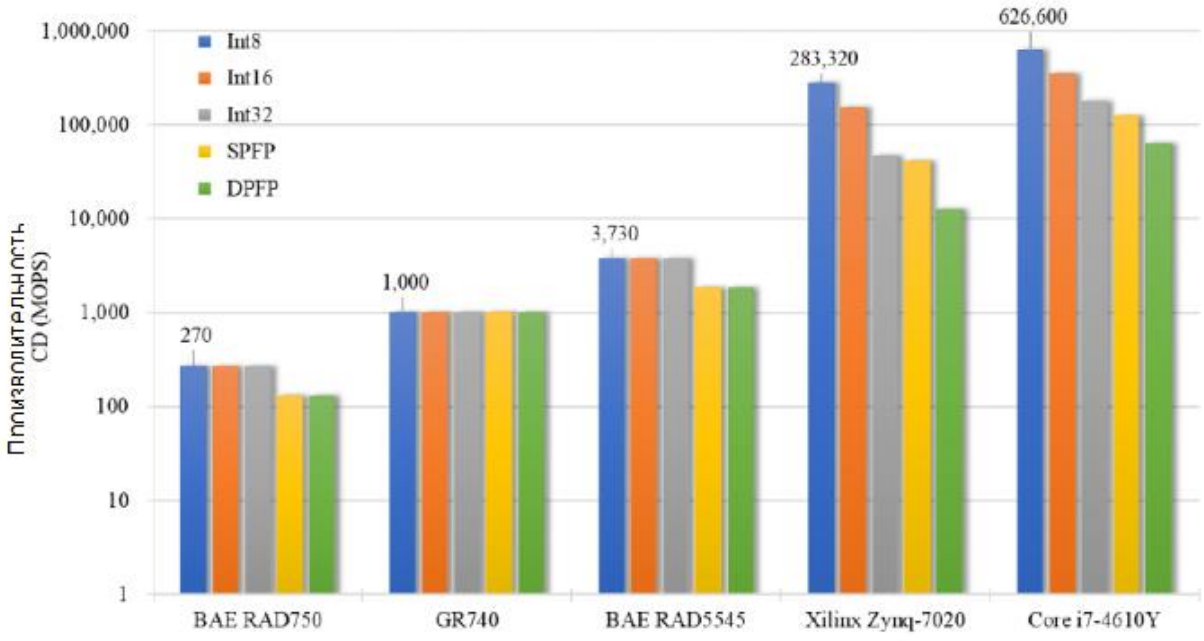

Рисунок 9. Сравнительные показатели производительности современных радиационно-стойких и коммерческих процессоров

Даже наиболее современный радиационно-стойкий процессор RAD5545 компании BAE демонстрирует на два порядка меньшую вычислительную плотность по сравнению с коммерческим процессором Xilinx Zynq 7020, построенном на основе архитектуры системы на кристалле (SoC), которая сочетает в себе два ядра ARM Cortex-A9 и FPGA, а также с коммерческим четырехъядерным процессором 4-го поколения Intel Core i7-4610Y, выпускаемым с 2013 года для мобильных и планшетных устройств.

Аналогичным образом эффективность потребления энергии радиационно-устойчивых процессоров, которую можно оценить по метрике вычислительной плотности на Ватт (CD/W), показанной на рисунке 10, почти на два порядка меньше, чем у коммерческих встроенных процессоров.

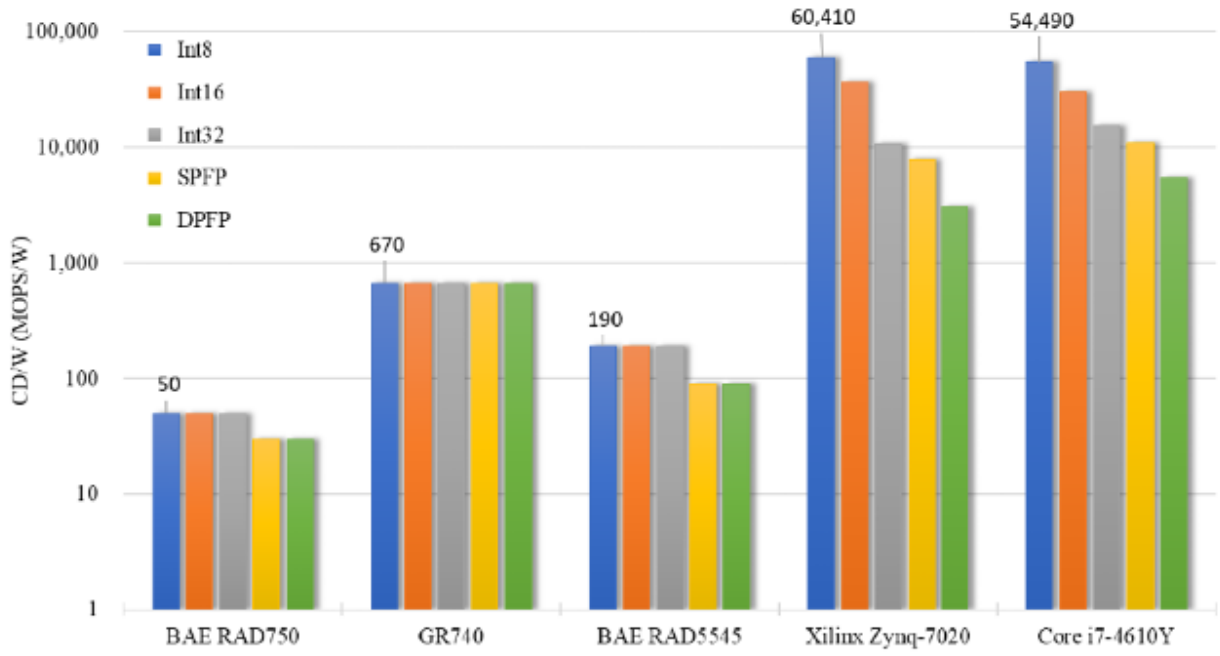

Рисунок 10. Сравнительные показатели эффективности энергопотребления современных радиационно-стойких и коммерческих процессоров

Наряду с показателями вычислительной плотности и эффективности потребления электроэнергии, размер памяти и пропускная способность вычислительных устройств также являются ключевыми факторами для систем искусственного интеллекта, демонстрируя возможности процессора эффективно хранить и передавать параметры модели и выполнять промежуточные вычисления. Поскольку радиационно-стойкие процессоры работают, как правило, на более низких тактовых частотах, пропускная способность устройств их встроенной памяти также значительно отстает от коммерческих решений DDR. Сниженные вычислительная плотность и пропускная способность устройств памяти существенно ограничивают размер и сложность моделей ИИ, которые можно на практике развернуть с использованием радиационно-стойких процессоров.

В целях снижения разрыва между радиационно-стойкими и коммерческими технологиями и значительного продвижения передовых достижений в области вычислений

в космической среде, NASA и военно-воздушные силы США совместно профинансировали проект создания высокопроизводительных космических вычислительных устройств (High-Performance Space Computing - HPSC). Согласно условиям заключенного в 2013 году контракта с компанией Boeing, для увеличения вычислительной плотности предполагалось на одной подложке чиплетного процессора[2] объединить с использованием быстродействующего интерфейса два четырехъядерных (ARM Cortex-A53) процессора. В процессе реализации проект HPSC претерпел значительные задержки и переносы сроков.Однако из-за того, что компания Boeing не уложилась в согласованные ключевые сроки работ, в конце 2021 года контракт был расторгнут.

В августе 2022 года NASA передало контракт по проекту HPSC на общую сумму 50 млн долл. компании Microchip Technology, которая до конца 2025 года планирует спроектировать, протестировать и сертифицировать 12-ядерный (включая восемь ядер приложения SiFive X280) чиплетный процессор PIC64-HPSC, построенный по 64-разрядной архитектуре с сокращенным набором команд (RISC-V) со встроенным коммутатором Ethernet для чувствительных ко времени сетей (TSN).

Важно отметить, что микропроцессоры PIC64-HPSC содержат большую часть интерфейсов, встроенного программного обеспечения, сетевых решений и поддержки искусственного интеллекта и машинного обучения, которые ранее не были широко доступны для бортовой космической авионики. Компания Microchip также создала раннюю экосистему партнеров по встраиваемым вычислениям и программному обеспечению, которые обладают всеми необходимыми компонентами и функциями, необходимыми для того, чтобы в конечном итоге предоставить вычислительное оборудование, готовое к космическим полетам, в течение следующих 1-2 лет.

Для быстрой инференции[3] используемых моделей ИИ и ускорения вычислений нейронной сети с заявленной производительностью 4,6 триллионов операций, ядра X280 включают расширения SiFive Intelligence. В Использование наряду с этим архитектуры RISC-V и встроенного коммутатора TSN Ethernet является ключевым факто-

[2] Чиплетный процессор (от англ. Chiplet) — сложная интегральная микросхема (например, центральный или графический процессор), в отличие от монолитной системы на кристалле (SoC) объединяющая на едином интерфейсе и в одном корпусе несколько кремниевых кристаллов (чиплетов), каждый из которых обеспечивает только свою часть общей функциональности процессора.

[3] Инференция — умозаключение, выводимое решение как результат и сам когнитивный процесс, связанный с извлечением, обработкой и интерпретацией получаемых сообщений.

ром качественного скачка в повышении вычислительной производительности. В частности, согласно техническому документу «Высокопроизводительные космические полетные вычисления», опубликованному NASA в июле 2024 года, RISC-V представляет собой 64-разрядную архитектуру набора основных инструкций (ISA) с открытым стандартом, на основе которых работает программное обеспечение процессора. Являясь набором инструкций ISA с открытым исходным кодом, архитектура RISC-V позволяет проектировщикам создавать процессорные модули, обеспечивающие расширенные возможности по настройке применительно к конечным целевым приложениям и оптимизации потребляемой мощности, производительности и занимаемого ими объема.

В свою очередь коммутатор TSN (Time Sensitive Networks) представляет собой набор стандартов, которые обеспечивают обмен стандартизированными сообщениями и управление сетевым трафиком, точно синхронизируя фиксированное (эталонное) время задержек при их передаче.

На современном этапе развития технологий существует широкая поддержка операций глубокого обучения на графических процессорах в связи с тем, что в настоящее время они являются основными ускорителями моделей глубокого обучения. Первоначальная установка драйверов и библиотек может оказаться сложной, но такие методы, как контейнеризация, позволяют упростить этот процесс, который необходимо выполнить только один раз. Необходимость установки дополнительных драйверов и библиотек делает развертывание модели AI/ML на базе графических процессоров (GPU) немного сложнее по сравнению с центральными (CPU), но учитывая существенное увеличение общей производительности модели из-за распараллеливания процессов, делает такие усилия обоснованными.

В свою очередь внедрение программируемых вентильных матриц (FPGA) может значительно усложнить модели AI/ML. Только сравнительно недавно производители ЭКБ стали предоставлять инструменты для упрощения развития моделей глубокого обучения на основе матриц FPGA, например такие, как Vitis AI компании Xilinx. Это вызывает еще одну проблему из-за того, каждый производитель предоставляет на рынок свой уникальный комплект инструментов. При этом, как правило, доступные инструменты компиляции моделей глубокого обучения на базе FPGA обеспечивают лишь ограниченную поддержку операциям глубокого обучения. Развертывание модели глубокого обучения на

FPGA может потребовать значительных изменений в этой модели для учетной записи неподдерживаемых операций таким образом, чтобы это согласовывалось со сложной кривой обучения, в свою очередь связанной с программированием для FPGA, что может привести к дорогостоящим разработкам.

Процесс обновления действующей развернутой модели также может варьироваться в зависимости от конкретного типа различных элементов и вычислительных платформ. Так, процесс обновления для центральных и графических процессоров, как правило, включает в себя традиционные и надежные способы замены управления в процессе перехода на новую модель. Такой подход позволяет сократить время вынужденного простоя и обеспечивает автоматический возврат в случае сбоя обновления. С другой стороны, при использовании FPGA требуется перезапуск логической матрицы, что вызывает перевод оборудования в автономный режим работы в процессе обновления. Также в случае сбоя матрица не может вернуться к предыдущей модели, а устройство, контролирующее перезапуск логической матрицы, должно будет сохранить эту логику.

На рисунке 11 представлена графическая диаграмма сравнения преимуществ использования каждого варианта электронных средств. Чем дальше представительная область продвинута по оси от начала диаграммы, тем лучше производительность модели в соответствующей категории.

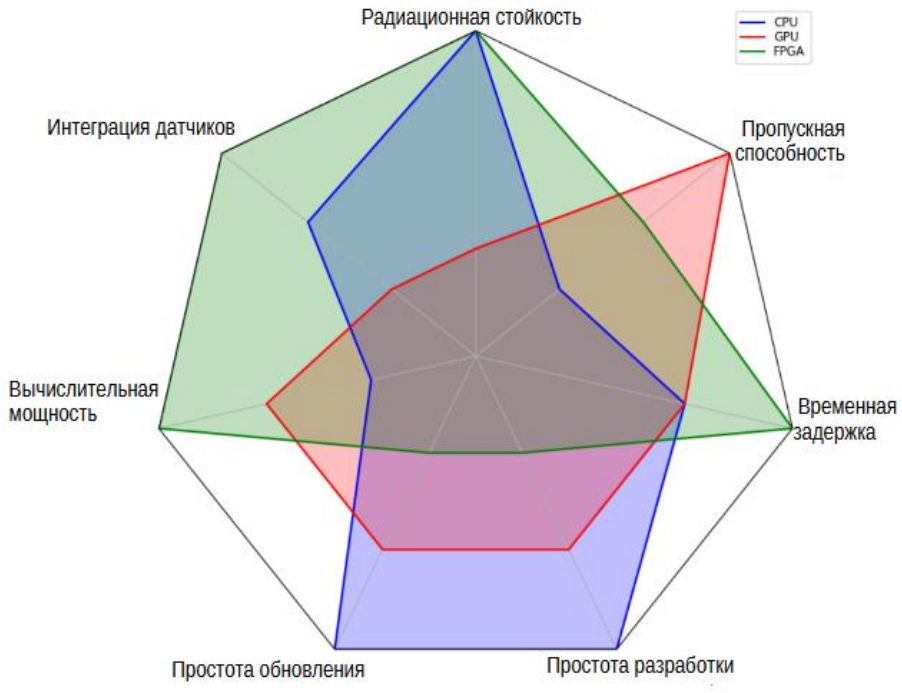

Рисунок 11. Диаграмма сравнения преимуществ основных типов электронных вычислительных элементов

В общем случае, чем больше площадь фигуры, охватываемая каждым типом аппаратных средств, тем эффективнее указанные аппаратные средства при выполнении операций AI/ML на борту космических средств. Например, для приложений, в которых ключевыми критериями являются задержка и радиационный допуск (оцененные в равной мере), использование матриц FPGA будет лучшим выбором в значительной мере из-за того, что графические процессоры GPU в настоящее время не имеют требуемого уровня радиационной стойкости и допуска.

Текущее поколение процессоров, отвечающих повышенным требованиям для использования в космосе (например, RAD 750 компании BAE), в пересчете на GFLOPS (Floating-Point Operations per Secund) - количество операций с плавающей запятой в секунду - на несколько порядков отстает от возможных радиационно не защищенных вариантов для наземного применения (зеленый график). Процессоры с плавающей запятой идеально подходят для вычислительно интенсивных приложений, наиболее используемых в области машинного обучения, особенно при обработке больших наборов данных, где некоторые данные могут иметь чрезвычайно большой диапазон числовых значений или где этот диапазон может быть непредсказуемым. При этом возведение в степень, присущее вычислениям с плавающей запятой, обеспечивает гораздо больший динамический диапазон представленных значений.

Кроме того, процессор RAD750 от BAE крайне дефицитен относительно других устройств, и возможность запуска ценных приложений AI/ML на этом типе процессоров маловероятна. В свою очередь процессоры RAD5545 компании BAE Systems Electronic Solutions и RTG4 компании Microsemi примерно в 10 раз превышают по количеству GFLOPS процессор RAD750. Однако, если их сравнивать для условий развертывания в периферийной среде (оранжевый график), они будут существенно ему уступать. Процессоры RAD5545 и RTG4 могут использоваться только для выполнения нескольких выводов в секунду в малых моделях MLOps, таких как StartNet (нейронная сеть для классификации изображений на мобильных устройствах).

На рисунке 12 представлен график рассеяния показателей GFLOPS в зависимости от пикового уровня энергопотребления для вычислительных устройств разных типов.

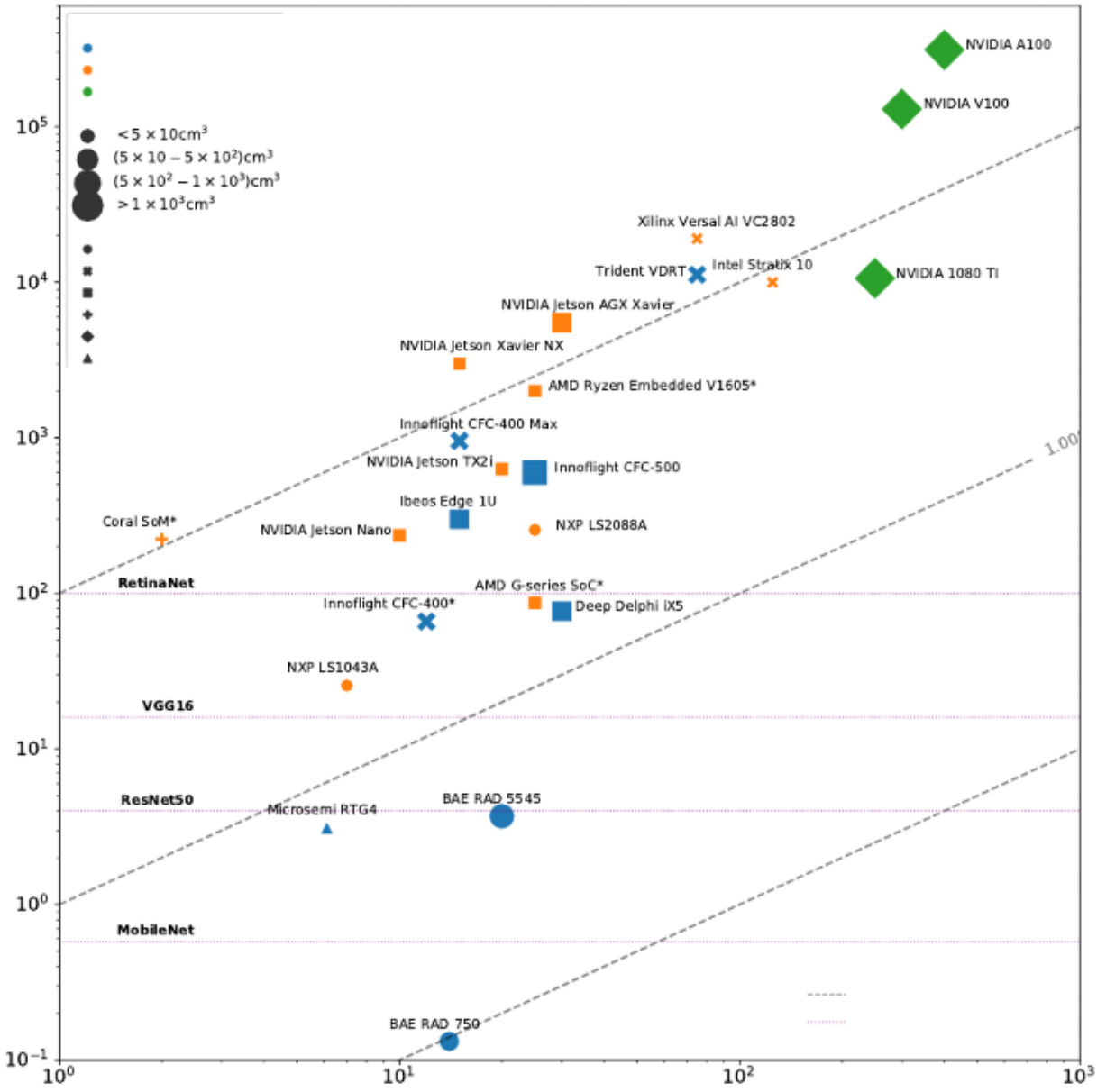

Рисунок 12. Распределение производительности вычислительных устройств в Гфлопс по величине пика энергопотребления в ваттах

Примечание: синий, оранжевый и зеленый цвета представляют устройства, предназначенные для космических, периферийных и наземной сред развертывания систем AI/ML соответственно. Размер точек данных представляет оценочный объем устройств. Тип фигуры указывает тип процессоров на устройстве. Пунктирные серые диагональные линии представляют контуры с постоянными величинами (GFLOPS/Вт), а пурпурные горизонтальные пунктирные линии — контуры производительности в GFLOP, необходимой для единого вывода стандартного выбора моделей глубокого обучения.

Следует отметить, что на рисунке 12 отражена приблизительная оценка производительности рассматриваемых вычислительных устройств, которая не претендует на окончательное представление качества каждого из них. В частности, несмотря на то что процессоры RAD750, RAD5545 и RTG4 являются наименее производительными, они обеспечивают наибольшую устойчивость к влиянию радиации, делая устройства этих трех типов наиболее подходящими для длительных космических полетов.

4.2 Особенности аппаратной среды для гетерогенных вычислений в системах AI/ML

Необходимо отметить, что в большинстве категорий и сред развертывания любой тип процессорных блоков может быть объединен в рамках одной системы для достижения максимальной производительности и энергоэффективности. Такой подход позволяет объединять различные типы ядер и может представлять собой комбинацию графического процессора с программируемой вентильной матрицей FPGA, которая оснащена встроенным микропроцессором.

В частности, для выполнения задач высокоуровневой логики, наряду с графическими (GPU), требуется центральный процессор CPU, который также полезен для подготовки пакетов данных, когда графический процессор выполняет операции вывода. Сочетание программируемой вентильной матрицы FPGA и графического процессора также может быть полезным, если запускать небольшие модели с низкой задержкой на FPGA и затем использовать GPU, когда для обработки данных должна применяться более сложная модель.

Примером подобного подхода может служить двухэтапная система автоматического распознавания целей (Automatic Target Recognition - ATR). Потоковая передача данных с камеры может обеспечивать съемку с более высокой скоростью в секунду по сравнению с производительностью оцениваемой модели ATR. В этом случае более эффективно было бы развернуть модель с низкой задержкой на базе программируемой матрицы FPGA для обнаружения важных кадров и только после этого отправлять выбранные кадры в более сложную модель на графическом процессоре GPU, где будут идентифицированы цели.

Системы гетерогенных вычислений характеризуются своими уникальными проблемами, которые не встречаются в однородных системах. В частности, наличие в них нескольких процессорных ядер обусловливает такие же проблемы, которые характерны для однородных систем параллельной обработки, но добавляет потенциальные неоднородности на этапе разработки системы в практике программирования и

общих возможностях системы. Для таких систем, как правило, требуется интерфейс, определенный для разных типов аппаратных средств и вычислительных блоков, и, например, в случае системы с центральным CPU и графическими GPU процессорами такой интерфейс может представлять собой четко определенный высокопроизводительный протокол на компьютерной шине PCI Express компании Intel.

Однако в варианте взаимодействия общего центрального процессора с матрицей FPGA интерфейс, как правило, не является четко определенным и для разработки решения могут потребоваться дополнительные накладные расходы. Однако этот вывод не всегда справедлив при использовании систем на чипах (SOC) компании Xilinx, разработчики встраиваемых процессоров которой достаточно часто используют свои уникальные решения, такие как шина AXI или DMA.

Кроме того, существует несколько вариантов, которые способны обеспечить улучшение общей производительности в будущем. Значительную часть этих примеров составляют варианты 64-битных интегральных схем специального назначения (ASIC), которые нацелены на приложения AI/ML для ускорения обучения и вывода моделей глубокого обучения или в некоторых случаях только на приложения вывода. Отдельные примеры включают такие схемы ASIC, как например тензорные (TPU) и периферийные тензорные процессоры Edge TPU, поставляемые компанией Google, а также нейроморфные процессоры, разрабатываемые компанией Intel и другими производителями.

Наиболее существенным недостатком интегральных схем ASIC для применения в условиях космоса является отсутствие возможности их перепрограммирования. Все центральные и графические процессоры и вентильные матрицы FPGA могут быть перепрограммированы для выполнения совершенно разных задач, но интегральные схемы ASIC могут выполнять только то, что было изначально разработано для периферийных тензорных процессоров Edge TPU и никогда не могут использоваться для каких-либо вычислений, кроме операций глубокого обучения.

На Edge TPU могут быть скомпилированы различные модели, но операции, необходимые для выполнения модели на этом устройстве, должны соответствовать краткому перечню его матричных операций. Поэтому интегральные схемы специального назначения ASIC никогда не станут единственным вычислительным устройством в полезной нагрузке КА. Вместо этого интегральные схемы ASIC могут использоваться для ускорения моделей глубокого обучения, в то время как процессоры другого типа будут обеспечивать другие вычислительные потребности.

ЗАКЛЮЧЕНИЕ

Одним из ключевых направлений современной ракетно-космической техники является разработка и развертывание крупных и чрезвычайно крупных космических систем различного назначения, состоящих из большого количества малых космических аппаратов. Это стало возможным благодаря существенному уменьшению массы и энергопотребления КА и его полезной нагрузки, в том числе за счет развития электронной компонентной базы, систем межспутниковой связи, повышения автономности обработки данных и принятия решений на периферийных устройствах и на борту космических аппаратов.

В связи с новыми тенденциями в развитии и использовании космических средств, в том числе для обеспечения военных действий в составе объединенных межвидовых формирований, требуются более быстрое по сравнению с физическими возможностями человека реагирование на возникающие угрозы, а также еще более высокий уровень автономности в функционировании КА и его бортового оборудования. Для решения этих и ряда других взаимосвязанных проблем предполагается широкое использование сетевых решений и технологий искусственного интеллекта и машинного обучения.

В свою очередь, для развертывания на борту космических аппаратов и использования передовых моделей и систем искусственного интеллекта и машинного обучения, особенно алгоритмов глубокого обучения, требуется многократное увеличение вычислительной производительности по сравнению с тем, что доступно в радиационностойких процессорах в бортовых системах современных КА. Следующее поколение космических процессоров для приложений AI/ML на борту КА, вероятно, будет включать разнородный ассортимент вычислительных устройств, включающий различные комбинации высокопроизводительных центральных (CPU) и графических (GPU) процессоров, программируемых вентильных матриц (FPGA) и специализированных интегральных схем ASIC.

Развертывание и использование многоспутниковых группировок, включающих сотни и тысячи аппаратов зачастую разных производителей, требует принципиально новых подходов к созданию и развитию приложений AI/ML для распределенной в космической среде системы на всех этапах ее эксплуатации и обслуживания. Методы обработки данных в системах AI/ML на борту КА определяют новые ограничения и условия обеспечения соот-

ветствия эксплуатационным требованиям, безопасности, и потребностям заказчика. При этом космическая среда создает новые требования по развитию и использованию моделей AI/ML, которые не просматриваются для условий наземной и периферийной сред, включая использование набора разнородных вычислительных устройств с низким уровнем массогаритных характеристик и энергопотреблением (SwaP), повышенной степенью радиационной стойкости.

Развитие приложений AI/ML для космоса требует также новых подходов к разработке эталонной архитектры для сквозной разработки, развертывания моделей и поддержки операций машинного обучения MLOps в среде с низким уровнем SwaP вычислительных устройств.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Lockheed Martin Space. AI/MI for Mission Processing Onboard Satellites. White Paper of R&D. July, 2024.

2 Air & Space Operations Review (ASOR): Uncertainty Quantification, Artificial Intelligence and Machine Learning in Military Systems. Vol. 2, No. 1, Spring 2023.

3 RAND Corporation. Artifical Intelligence and Machine Learning for Space Domain Awareness, RR-A2318-2. August, 2024.

4 NASA Goddard Space Flight Center. Current AI Technology in Space. July 24, 2023.