

ІТМО

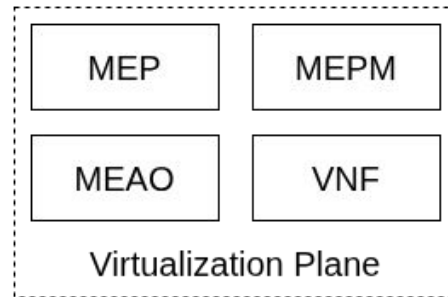
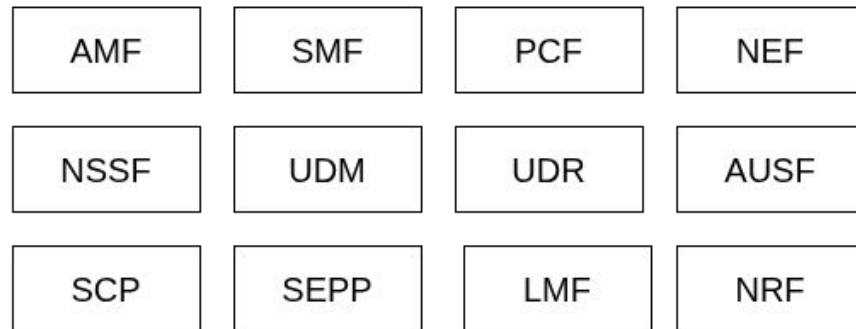
Актуальное состояния систем принятия решений в МЕС

**Авторы:
И.В. Филянин,
Н.А. Сорокин,
В.А. Богатырев**

MEC над SDN



Application Plane



Control Plane



Data Plane

Проблемы и цели.

- Рост требований к низкой задержке в сетях
- Появление MEC как ответ на эти вызовы
- Фрагментированность существующих исследований

Multi-Access Edge Computing — это архитектура, при которой вычислительные и сервисные функции выносятся из централизованных дата-центров на «край» сети, ближе к пользователю.

Цель:

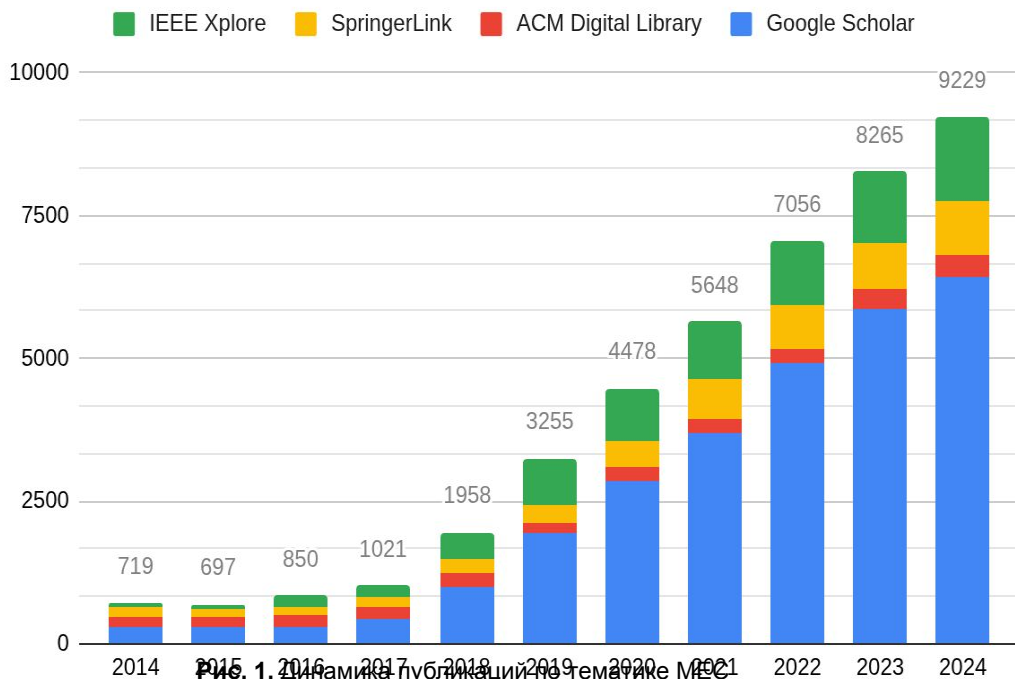
Провести системный анализ подходов к принятию решений при размещении вычислительных сервисов в гео распределенных узлах краевых вычислений, выявить ключевые проблемы и оценить перспективы применения разных алгоритмов.

Задачи:

- Систематизация существующих подходов
- Сравнительный анализ алгоритмов
- Оценка перспектив алгоритмов RL
- Выявление метрик для оценки систем

Популярность МЕС как технологии

ИТМО



2000-е — первые идеи граничных вычислений (CDN)

2013 — формализация ETSI как Mobile Edge Computing

2014 — выход первых стандартов и архитектурных рекомендаций

2017 — преобразование в **Multi-Access Edge Computing** (поддержка Wi-Fi, фиксированных сетей, IoT)

Концептуальная модель MEC

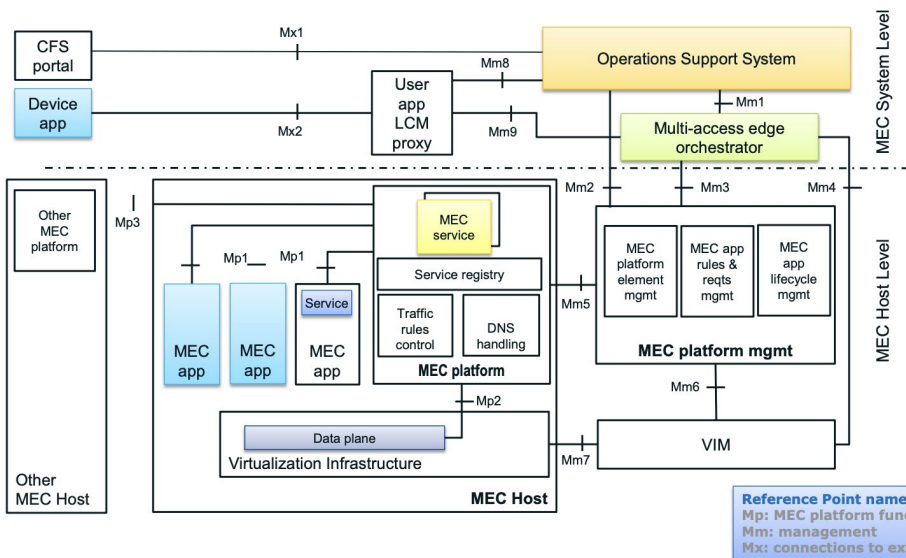


Рис. 2. Референсная архитектура MEC платформы

Mm1 – обмен данными и управление размещением.

Mm2 – передача команд и мониторинг состояния.

Mm3 – доступ к локальным службам хоста.

Mm4 – отчёты о состоянии и характеристиках хоста.

Mm5 – координация между менеджерами платформ и верхним MEC-оркестратором.

Mm6 – взаимодействие оркестратора с VIM.

Mm7 – связь с OSS/BSS оператора для учета, SLA и биллинга.

Mm9 – интеграция с NFV-оркестратором.

Mx1 – REST-интерфейс MEC-приложения к платформенным сервисам.

Mx2 – обмен данными и событиями между MEC-приложениями.

Математическая модель МЕС



$H = \{h_1, h_2, \dots, h_n\}$ – множество МЕС платформ;

$A = \{a_1, a_2, \dots, a_m\}$ – множество МЕС приложений;

$U = \{u_1, u_2, \dots, u_k\}$ – множество пользователей;

$R_h = (CPU_h, RAM_h, TX_h, RX_h, Read_h, Write_h)$ – ресурсы хоста

$r_a = (cpu_a, ram_a, tx_a, rx_a, read_a, write_a)$ – требуемые ресурсы приложения

$\min_x \sum_{u \in U} \sum_{a \in A} \sum_{h \in H} \lambda_{u,a} * L_{u,h} * x_{a,h}$ – целевая функция минимизации средней задержки

Poisson Point Process

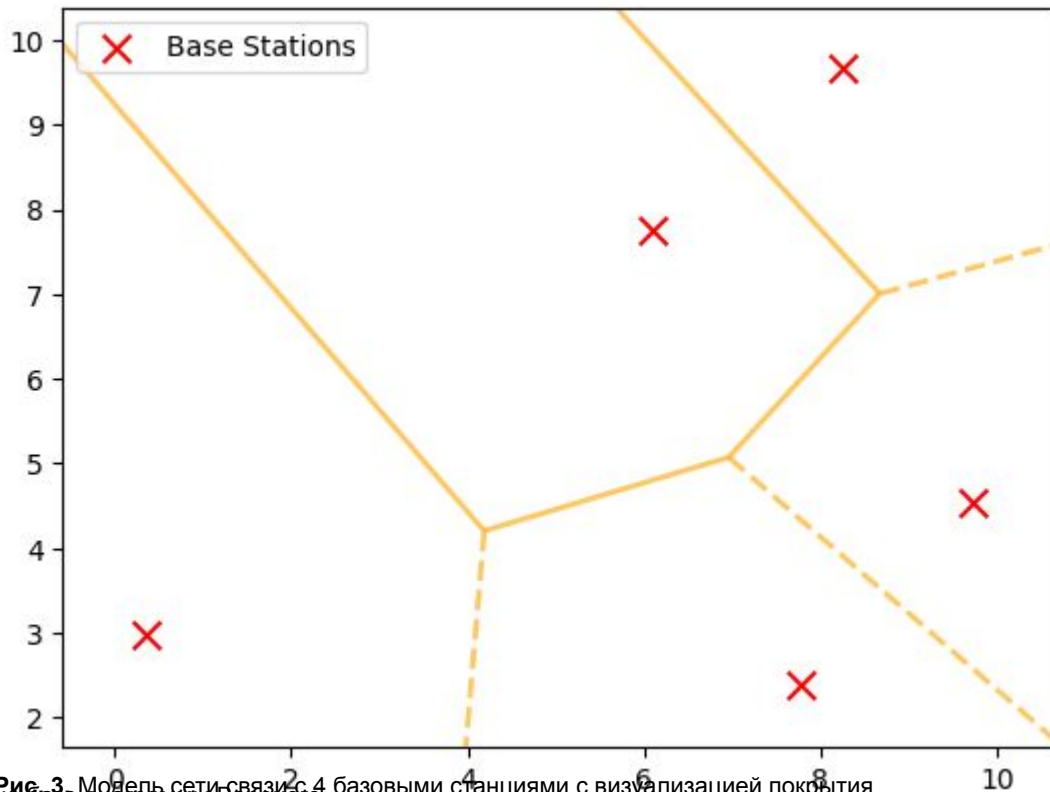


Рис. 3. Модель сети связи с 4 базовыми станциями с визуализацией покрытия на базе диаграммы Вороного

Poisson Point Process – стохастическая модель, используемая для описания случайного распределения объектов в пространстве, таких как базовые станции или пользователи в мобильных сетях.

	Критерий оптимизации	Сложность реализации	Сложность эксплуатации	Масштабируемость
Resource allocation	Максимизация утилизации ресурсов/сокращение средней задержки	Низкая	Низкая	Средняя
Task offloading	Сокращение задержки/минимизация потребления сетевого трафика	Высокая	Высокая	Средняя
Task Scheduling	Минимизировать среднее время отклика и долю просроченных задач/минимизировать джиттер	Средняя	Средняя	Высокая

Таблица 1. Сравнение подходов организации размещения вычислительных сервисов

Оптимизация размещения ресурсов ІІТМО

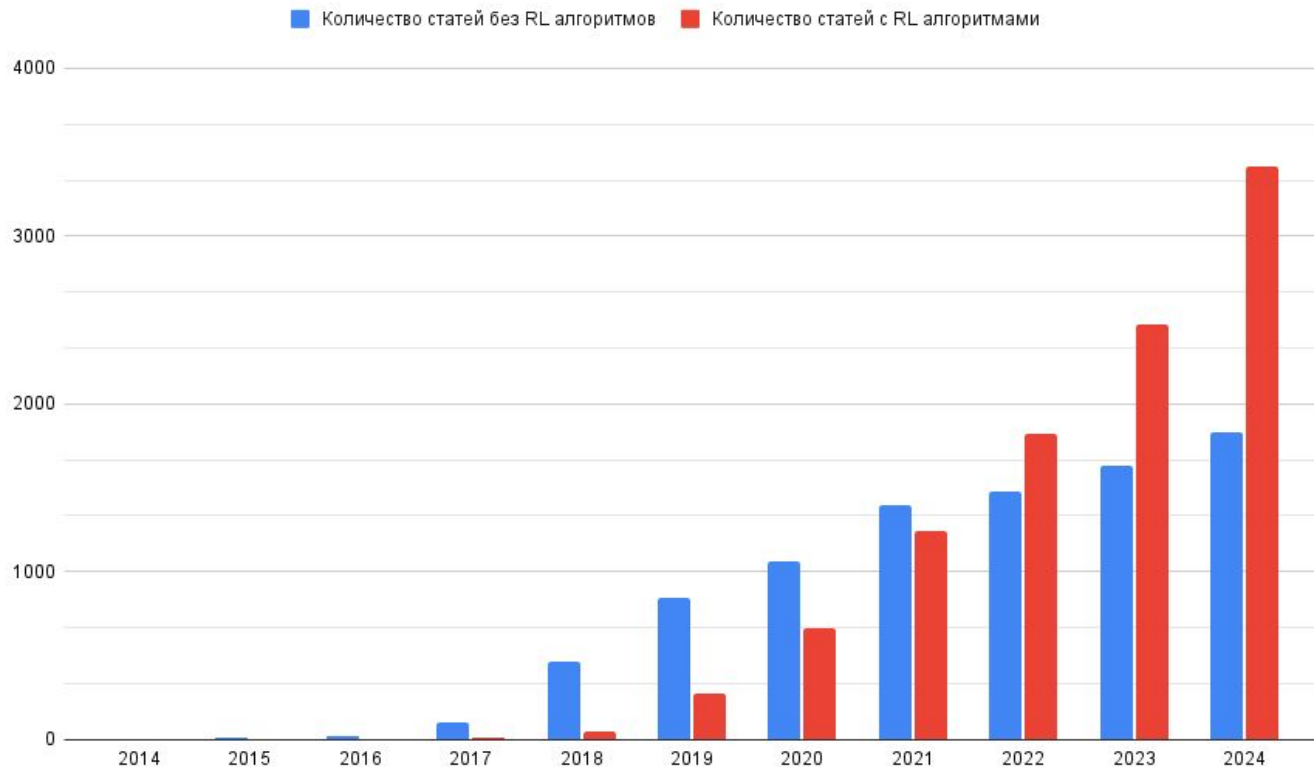


Рис. 4. Диаграмма популярности алгоритмов принятия решений на основе публикаций научных статей

Метод	Преимущества	Методики	Эффективность
Deep Q-networks (DQN)	Добивается лучшей производительности по сравнению с традиционными методами	Использует нейронную сеть для аппроксимации функции ценности и применяет алгоритм Q-обучения для выработки оптимальной политики	Превосходит традиционные методы в задаче восходящей (uplink) передачи
Asynchronous Advantage Actor-Critic (A3C)	Позволяет лучше исследовать пространство действий и ускоряет сходимость	Использует несколько агентов, одновременно исследующих разные части среды	Эффективное распределение ресурсов в крупномасштабных сотовых сетях с разными требованиями QoS
Proximal Policy Optimization (PPO)	Эффективно распределяет ресурсы, обеспечивая требуемый QoS для срезов	Применяет оптимизацию в доверительной области (trust region) для плавного и безопасного улучшения политики	Приводит к высокому уровню выполнения QoS
Soft Actor-Critic (SAC)	Объединяет идеи actor-critic и максимальной энтропии в RL	Добивается лучшей эффективности выборки и устойчивости при изменениях среды	Эффективное распределение ресурсов для сквозного (end-to-end) сетевого среза в сценариях с несколькими срезами и сервисами

Таблица 2. Сравнение методов RL-алгоритмов принятия решений распределения вычислительных сервисов

Популярность RL алгоритмов

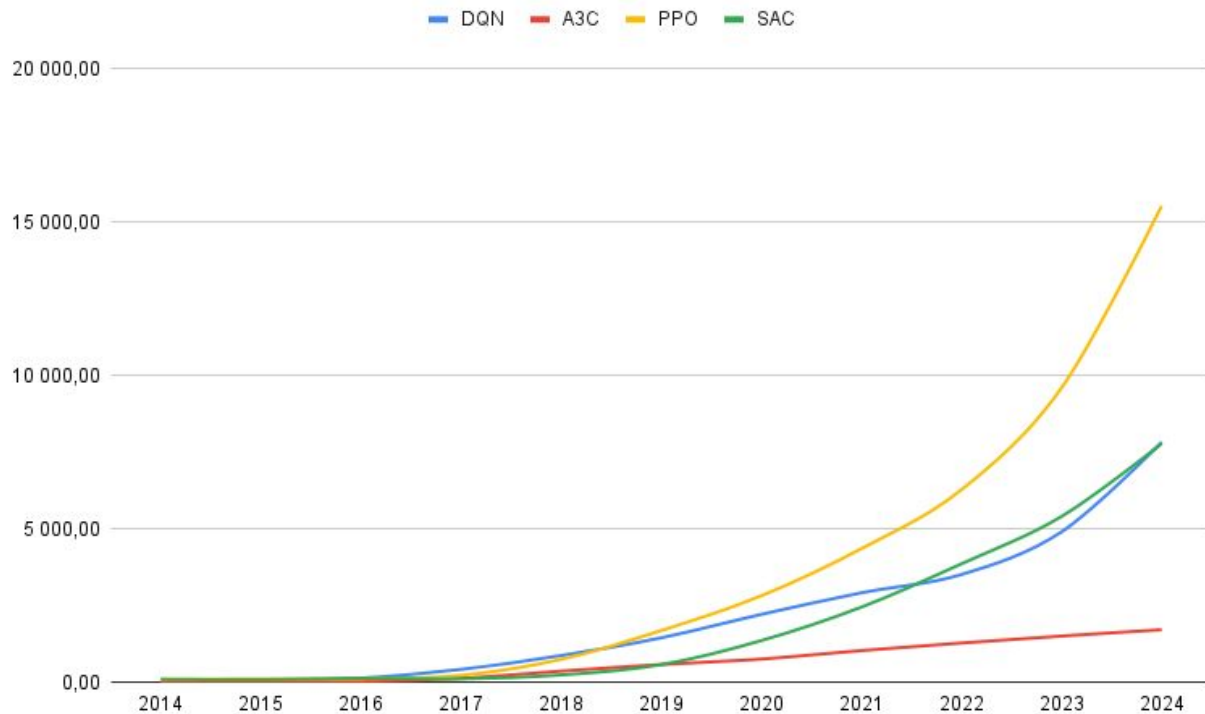


Рис. 5. График количества публикаций научных статей по алгоритмам машинного обучения с

Спасибо за внимание!

itMO *re than a*
UNIVERSITY

Сорокин Никита
308405@niuitmo.ru