Team04 - EdgeZilla

RESOURCE ALLOCATION IN IOT EDGE COMPUTING VIA CONCURRENT FEDERATED REINFORCEMENT LEARNING

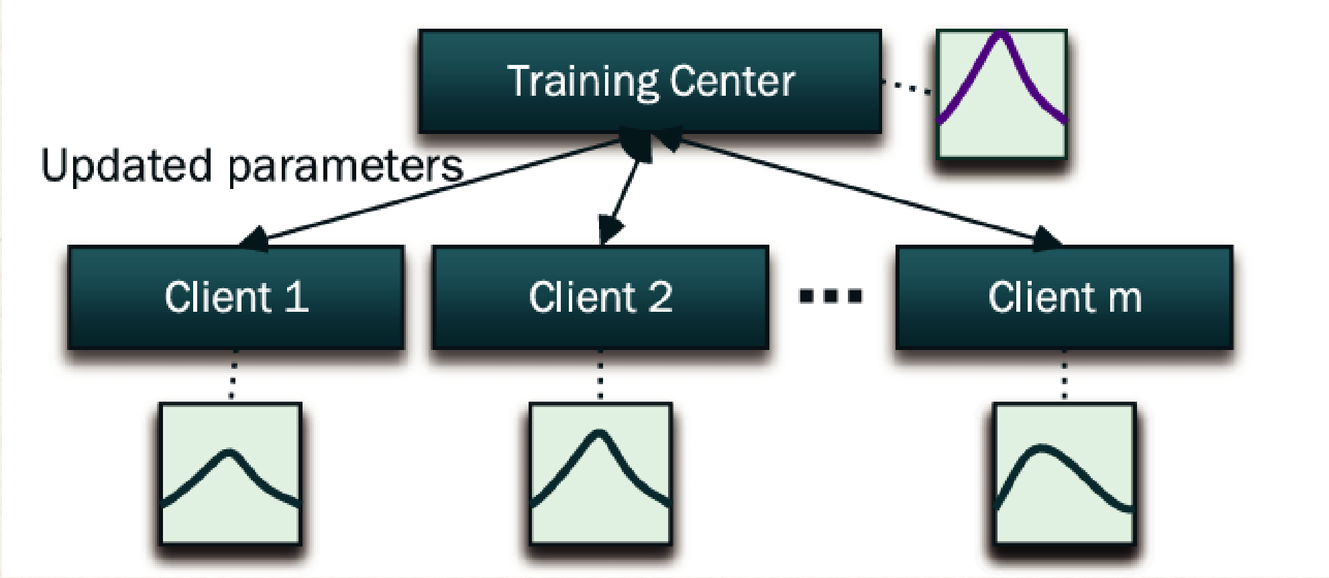
AUTHORS

Zhu Tianqing , Wei Zhou , Dayong Ye , Zishuo Cheng, and Jin Li

Agenda

01	**Overview of the Research Paper**
02	**Use Case taken to apply the startegy**
03	**Why its a Edge Problem?**
04	**State of the Art Literatures**
05	**Implementation Strategy and Plan of Action**
06	**Future Goals**
07	**Q&A**

What is Federated Learning ??



Federated Learning is a machine learning approach where multiple devices (clients) collaboratively train a model without sharing their raw data.

Contentions in the paper

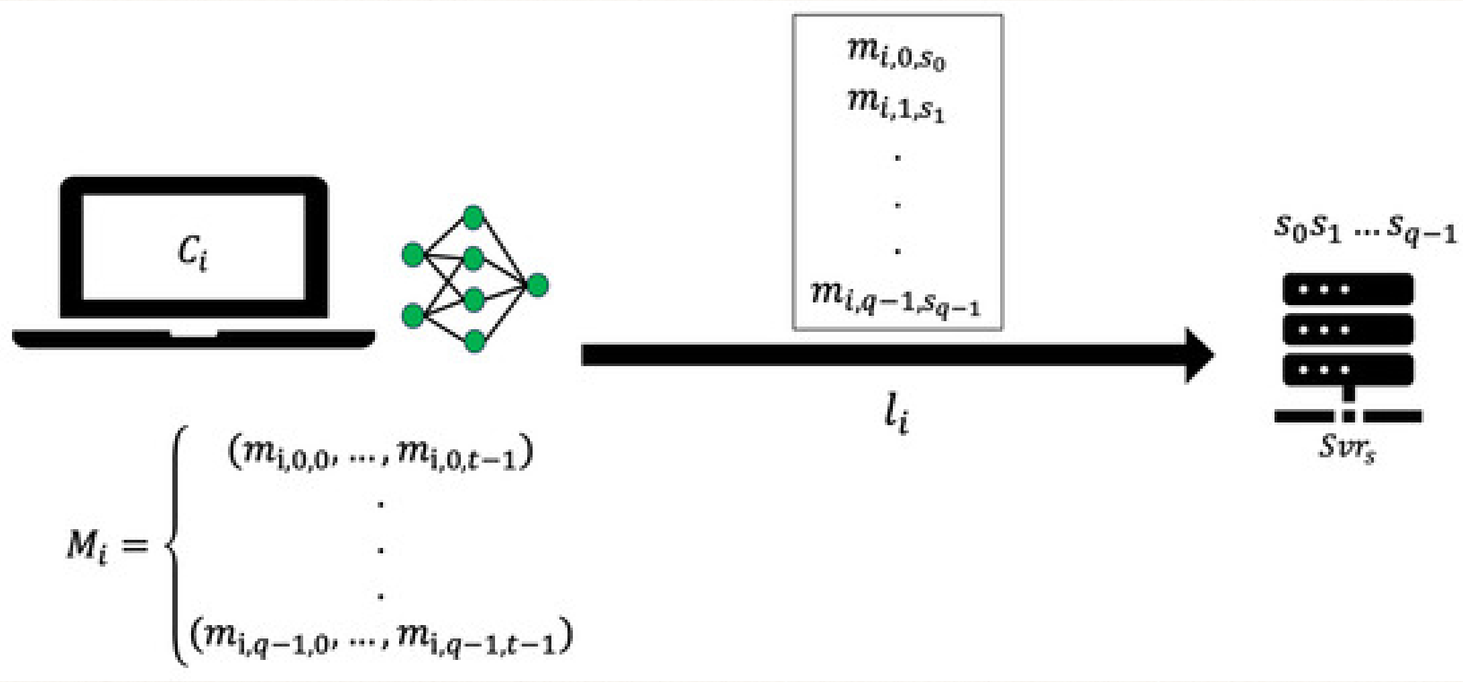
- Global info vs Local info that led to resource allocation problem

- Federated vs Reinforcement learning that led to FRL

Geiping method and discovery of CFRL

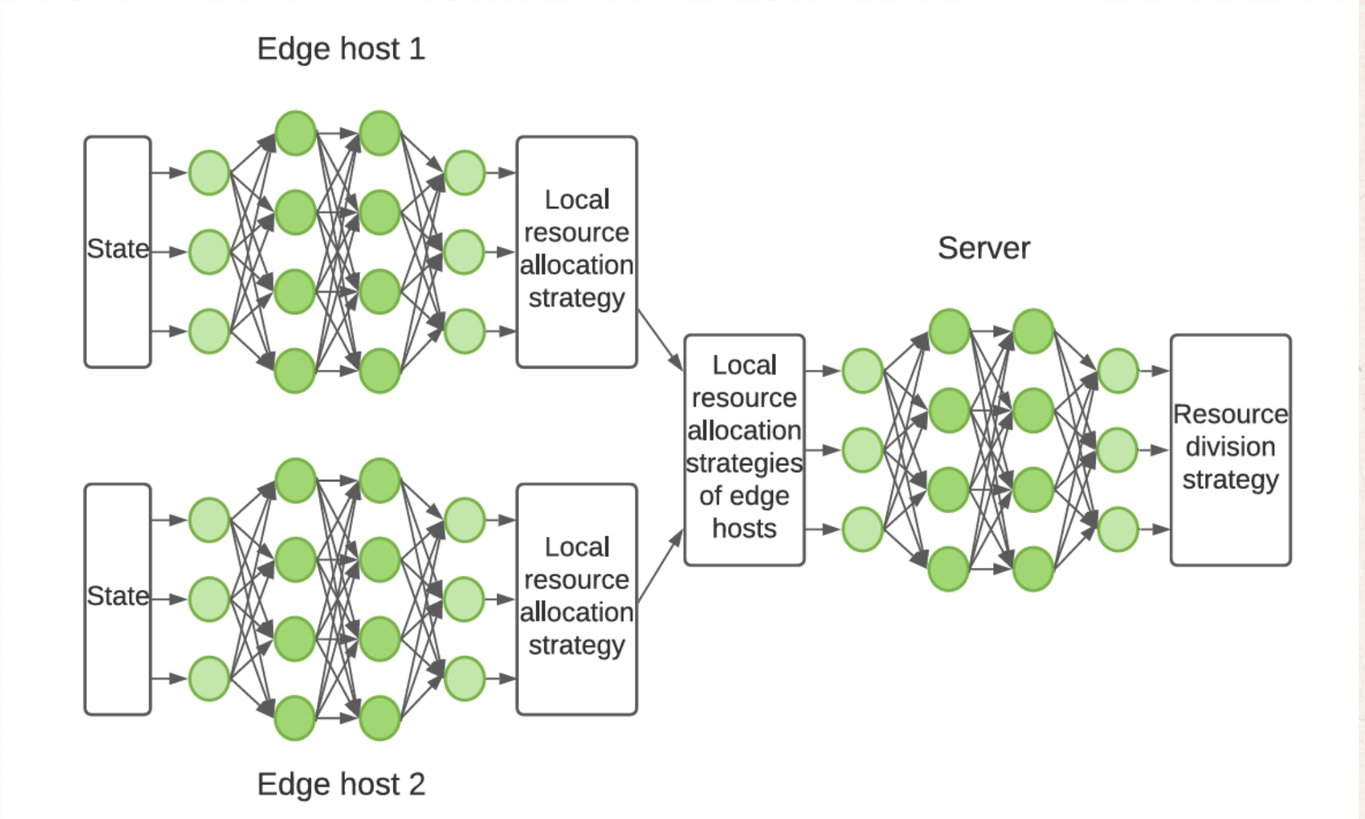
- Existing Federated Reinforcement Learning (FRL) requires local models to be uploaded to the server.

- Privacy Risk: Model gradients can leak input data (Geiping et al.).



Proposed CFRL:

- Server and edge hosts make joint decisions (concurrency).
- Only model outputs & rewards are shared; local data and models remain private.
- Stronger privacy and not vulnerable to model inference attacks.



Overview of Concurrent Federated Reinforcement Learning (CFRL)

States:

Edge hosts → Resource status + task requirements
Server → Resource allocation strategies of edge hosts

Actions:

Edge hosts → Allocate resources or offload tasks
Server → Divide and reserve resources for hosts

Rewards:

Based on task completion speed and resource usage efficiency

Strategy Creation

- Edge hosts create local resource strategies and share outputs with the server.
- Server generates a resource division strategy using Deep RL.

Strategy Execution

- Edge hosts process tasks locally or offload to the server.
- Rewards are given based on successful task completion and resource usage.

Model Update

Both server and edge hosts update their DQNs using received rewards.

Use Case - Early Sepsis Detection

This project addresses the challenge of dynamically allocating constrained edge and centralized computing resources for real-time, privacy-preserving early sepsis detection in heterogeneous hospital environments with variable and bursty task loads.

- **Sepsis is a life-threatening condition caused by the body's extreme response to an infection. It can lead to organ failure and death if not treated early.**
- **Every hour of delayed detection/treatment → 7–10% increase in mortality.**
- **Early warning systems based on vitals, lab tests, and symptoms can save lives.**

Why it's Hard to detect Sepsis?

- **Sepsis signs are non-specific (fever, rapid heart rate, low BP).**
- **Requires real-time monitoring and analysis of multi-modal signals:**
  - **Vitals: heart rate, temperature, respiratory rate, blood pressure.**
  - **Labs: WBC count, lactate, creatinine.**
  - **Trends: deterioration over hours.**

Why its an Edge problem?

1. **High Task Volume + Bursty Workload**
   - Each edge node handles tens of patients, each generating inference tasks every 5–15 minutes.
   - During surges (shift changes, emergencies), task arrival rate exceeds local processing capacity.
2. **Real-Time Requirements**
   - A delayed sepsis alert can be fatal.
   - Inference latency > 5–10 seconds may make predictions less actionable.
   - Offloading introduces network latency → risky unless well-optimized.

Where Our Approach Stands Out?

1. Privacy-Preserving Intelligence with No Raw Data Sharing

2. Adaptive Resource Sharing in Heterogeneous Hospital Environments

3. Joint Decision Making: Smart Local Inference or Server Offloading

Objectives and Goals

Enable Real-Time, Privacy-Preserving Sepsis Risk Prediction at the Edge

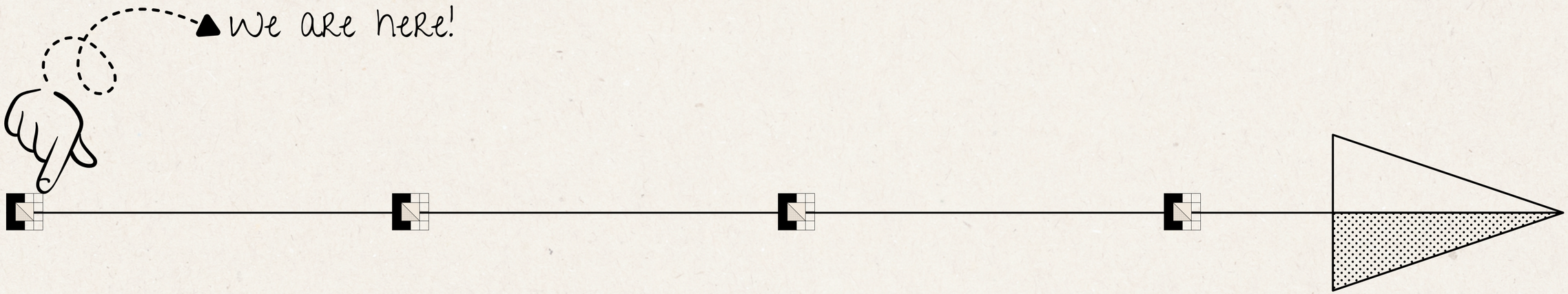
Optimize Resource Allocation Under Variable Workload Across Hospital Units

Evaluate CFRL Against Baseline Scheduling Strategies

State of the Art Literature

Feature	DRL for Local Optimization (Xiong et al., 2020)	Centralized Optimization (Wei et al., 2019)	Federated Learning with Model Sharing (Yu et al., 2021
Core Idea	Use Deep Reinforcement Learning (DRL) for independent, local resource allocation on each edge host	A central server makes all resource allocation decisions for the entire system to maximize global utility.	Combine DRL with Federated Learning to collaboratively train models while preserving data privacy.
Decision-Making	Decentralized: Each edge host decides for itself	Centralized: One server decides for everyone	Hybrid: Local training, but a central server aggregates models
Key Limitation	Poor system-wide efficiency due to lack of coordination	Sacrifices user privacy by design, as it requires collecting status data from all hosts.	Vulnerable to gradient inversion attacks, allowing an adversary to reconstruct private data from shared models

Plan Of Action



Architecture Design & System Planning

iFogSim Modeling & Edge Resource Simulation

Python API Integration with ML Module

Scenario Testing, Metrics & Final Evaluation

Tools Used to Validate the Architecture

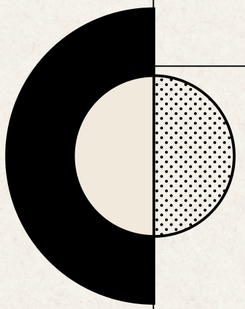
iFogSim2

Python + FastAPI

Postman - For Testing

Team split up

Shantharam P	Architecture Design, iFogSim2 Base Modeling, and API Integration with Python ML Module
Kavinesh P	DQN Algorithm 1 (Edge Node Scheduler) and Task Simulation + State/Reward Modeling in iFogSim2
Adhikesh S K	DQN Algorithm 2 (Server-Side Resource Allocator) and Server-Side Q-Vector Aggregation in iFogSim2
Nellore Sisir Reddy	Scenario Design, Baseline Comparisons, Final Simulation Runs, and Evaluation Metrics & Visualization



Q&A TIME!