Multi-user Edge-assisted Video Analytics Task Offloading Game based on Deep Reinforcement Learning

Yu Chen, Sheng Zhang, Mingjun Xiao, Zhuzhong Qian, Jie Wu, and Sanglu Lu

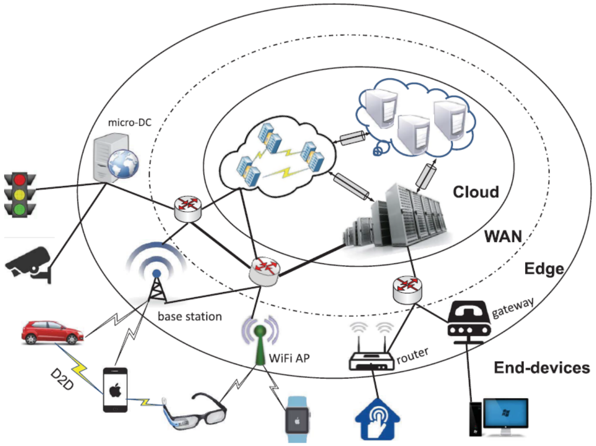
ICPADS 2020

Outline

- **Background**
- **Motivation**
- **Problem formulation**
- **Algorithm design**
  - Game theory-based solution
  - Reinforcement learning-based solution
- **Performance evaluation**
- **Conclusion**

Background

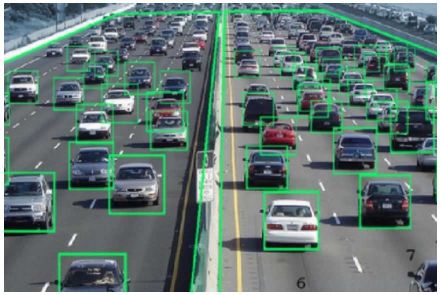
- **Conventional centralized network**
  - High transmission delay
  - Heavy loads on the backhaul links



- **Mobile edge computing**
  - Bring the computation and storage resources to the edge of networks
  - Connect users directly to the nearest service-enabled edge networks and provide computing and caching capabilities

Background

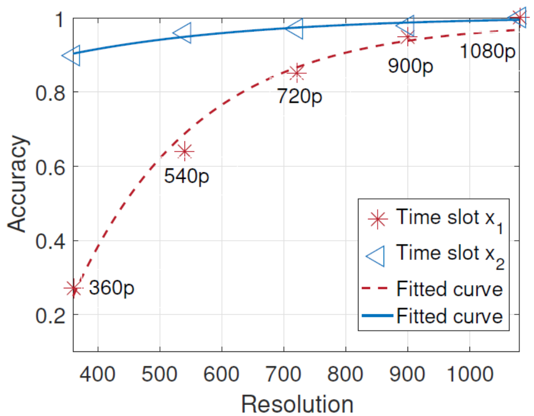
- **Emerging artificial intelligence application**
  - Recommendation systems, personal assistant, video surveillance, etc.
  - Real-time **video analytics** is envisioned as a killer application in the edge computing environment
- **Video analytics task**
  - Detect specific events, such as causing-trouble vehicle, abandoned luggage, lost child
  - Collect high-definition videos and require high bandwidth, high computation and low latency

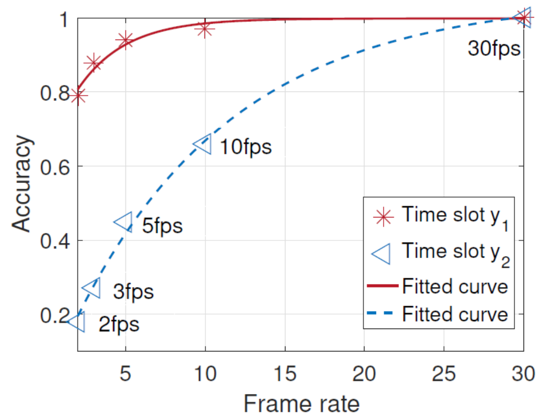


Edge computing is regarded as a promising solution to meet the strict requirements

Motivation

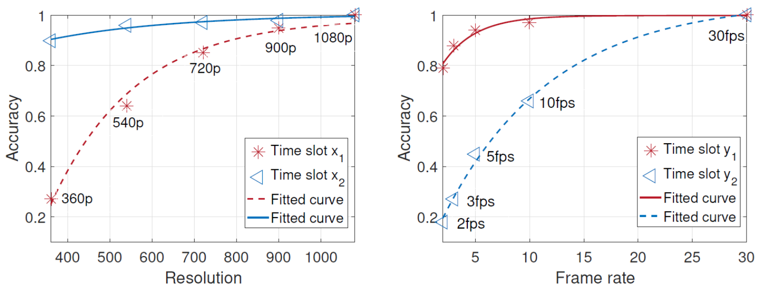
- **Video analytics task offloading**
  - Video frames are extracted at various sampling rates, compressed into different resolutions, and processed by CNN models
  - Refer to the combination of frame rate and resolution as **configuration**
  - Analytics accuracy & configuration:



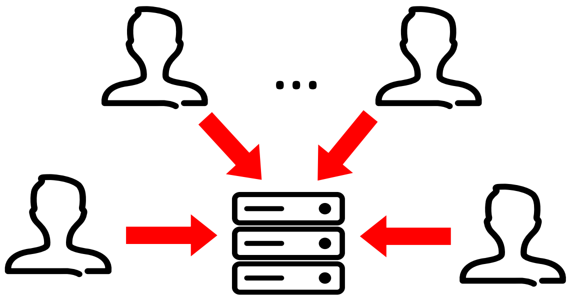


Motivation

- **Video analytics task offloading**
  - Analytics accuracy & configuration
  - Quality of experience (QoE) usually involves transmission delay, allocated computing resources, analysis accuracy, etc.



Higher accuracy decision ⇩ ***Higher Resolutions*** / ***Higher Sampling rates*** ⇨ ***More video data amount*** ⇧ ***<u>Higher Transmission delay</u>***

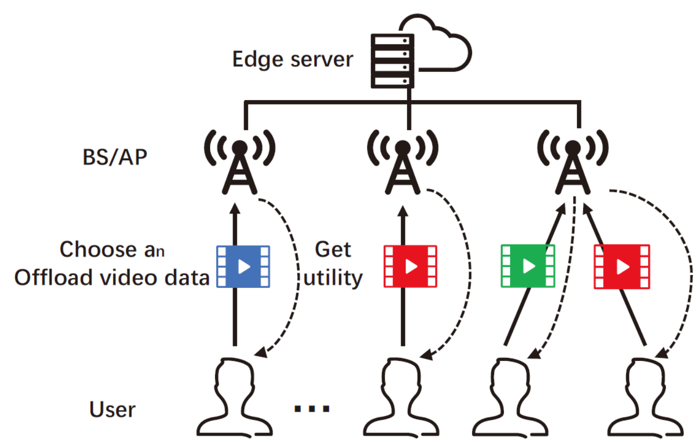


Limited Edge computing resources

Problem: *When all users share the limited edge computing resources, how to determine the accuracy decisions for them to maximize their respective QoE as much as possible and ensure their accuracy decisions are stable?*

Problem Formulation

- **Some notations**
  - Consider a set of N users, denoted by $\{1,2,\dots,N\}$
  - User n chooses the accuracy decision a_n
  - $M_n \leq a_n \leq 1$, where M_n denotes the minimum requirement on analysis accuracy for user n
  - Design the utility function based on the features of video analytics task offloading for each user n
    - **Transmission cost $T(a_n)$**
    - **Computation allocation $C(a_n)$**
    - **Accuracy satisfaction $Sat(a_n)$**



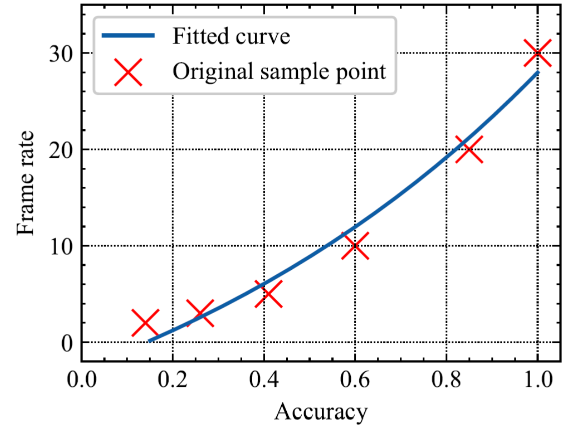
Problem Formulation

- **Transmission cost**
  - Accuracy decision & frame rate

$$F(a_n) = \frac{1}{r_n}(e^{a_n - s_n} - t_n)$$

  - Accuracy decision & transmission cost

$$T(a_n) = \frac{K \cdot F(a_n)}{b_n} = \frac{K \cdot \frac{1}{r_n}(e^{a_n - s_n} - t_n)}{b_n}$$



- **Computation allocation**
  - let E denote the amount of computation resource at the edge server
  - Computation resources allocated to the users depend on the proportion of their uploaded video data amount on the edge server

$$C(a_n) = E \cdot \frac{\frac{1}{r_n}(e^{a_n - s_n} - t_n)}{\sum_{i=1}^{N} \frac{1}{r_i}(e^{a_i - s_i} - t_i)}$$

Problem Formulation

- **Accuracy satisfaction**
  - If we use the deep learning approach like CNN for video analysis, the accuracy will be more difficult to improve when it is close to 100%

Accuracy Increasing	Accuracy satisfaction
80% -> 85%	😊
85% -> 90%	😊 😊
90% -> 95%	😊 😊 😊

  - The property of accuracy satisfaction is consistent with the convex functions, thus we describe the accuracy satisfaction in this work as

$$Sat(a_n) = e^{a_n}$$

Problem Formulation

- **Utility function design**
  - In terms of transmission cost, computation allocation and accuracy satisfaction, user n's utility function is defined as

$$u_n(a_n, \boldsymbol{a}_{-n}) = -\alpha_n T(a_n) + \ \beta_n C(a_n) + \ \gamma_n Sat(a_n)$$

  - Let $\boldsymbol{a}_{-n} = (a_1, \dots, a_{n-1}, a_{n+1}, \dots, a_N)$ denote the accuracy decisions from all users except user n
  - Given other users' decisions, each user n will choose the optimal accuracy decision a_n to maximize its utility $u_n(a_n, \boldsymbol{a}_{-n})$, i.e.,

For each user n:

$$\begin{aligned} &\max \ \ \boldsymbol{u_n(a_n, a_{-n})} \\ &\boldsymbol{s.t.} \ \ \boldsymbol{a_n} \in [\boldsymbol{M_n}, \mathbf{1}] \end{aligned}$$

GT-based Solution

- **Nash equilibrium (NE)**

For each user n, the strategy set $\{a_1^*, a_2^*, \dots, a_N^*\}$ constitutes a Nash equilibrium in the game of the problem if the individual utility cannot be improved by changing the accuracy strategy, i.e.,

$$u_n(a_n^*, \boldsymbol{a}_{-n}^*) \geq u_n(a_n, \boldsymbol{a}_{-n}^*)$$

- **Determining $\boldsymbol{a_n^*}$ corresponding to $\boldsymbol{a_{-n}^*}$**

$x_n^* = \sqrt{\frac{\beta_n E b_n O_n^*}{\alpha_n K - \gamma_n e^{s_n} r_n b_n}} - O_n^*$, where $x_n^* = \frac{e^{a_n^* - s_n - t_n}}{r_n}$ and $O_n^* = \sum_{i \neq n} \frac{e^{a_i^* - s_i - t_i}}{r_i}$

- **Figuring out $\boldsymbol{a_n^*}$**

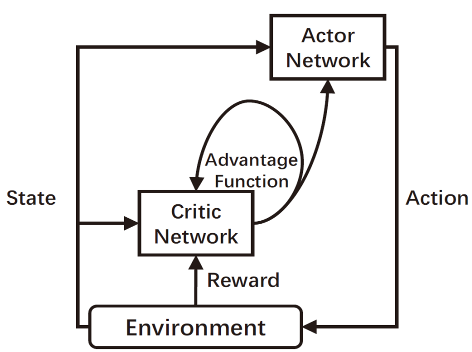
$$a_n^* = \ln\left(\frac{r_n(N-1)}{\sum_{n=1}^N s_n}\left(1 - \frac{s_n(N-1)}{\sum_{n=1}^N s_n}\right) + t_n\right) + s_n$$

Algorithm 1 GT-based Algorithm

```
for each user n = 1, 2, 3, ..., N do
    Prepare user n's information including r_n, s_n, t_n, b_n,
    α_n, β_n, γ_n, M_n.
    Publish information to a specified shared storage area.
    repeat
        Gather other users' information except user n.
    until All of other users' information is collected.
    Calculate the optimal accuracy decision a_n* according
    to Eqn. (19).
end for
```

RL-based Solution

- **Reinforcement learning-based solution**
  - It is unrealistic to share private information (e.g. allocated bandwidth, accuracy requirement) because of security and privacy concerns
  - Markov decision process model
    - Action space A: $A_n = \{a_n^k | k = 1,2, \dots\}$
    - State space ST: $st_n^k = [a_n^k, \boldsymbol{a}_{-n}^k, \dots, a_n^{k-B}, \boldsymbol{a}_{-n}^{k-B}]$
    - Reward space R: $r_n^k = u_n(a_n^k, \boldsymbol{a}_{-n}^k)$
    - Task offloading policy $\pi_{\theta_n}: ST_n \times A_n \to [0,1]$



Advantage Actor Critic (A2C)

Algorithm 2 RL-based Algorithm

Initialize $\boldsymbol{\theta}_n$, $\boldsymbol{w}_n$, $\alpha^{\boldsymbol{\theta}_n}$, $\alpha^{\boldsymbol{w}_n}$ and st_n^0.
for time slot $k = 0, 1, 2, \dots$ **do**
  for each user $n = 1, 2, 3, \dots, N$ **do**
    Acquire the past strategy set.
    Update its state st_n^k into st_n^{k+1}.
    Input st_n^k into the Actor network $\pi_{\boldsymbol{\theta}_n}$.
    Obtain the accuracy decision a_n^k from $\pi_{\boldsymbol{\theta}_n}$.
    Calculate reward $r_n^k = u_n(a_n^k, \boldsymbol{a}_{-n}^k)$ according to (5).
    Update $\boldsymbol{w}_n$ and $\boldsymbol{\theta}_n$ according to (24), (25).
  end for
end for

Performance Evaluation

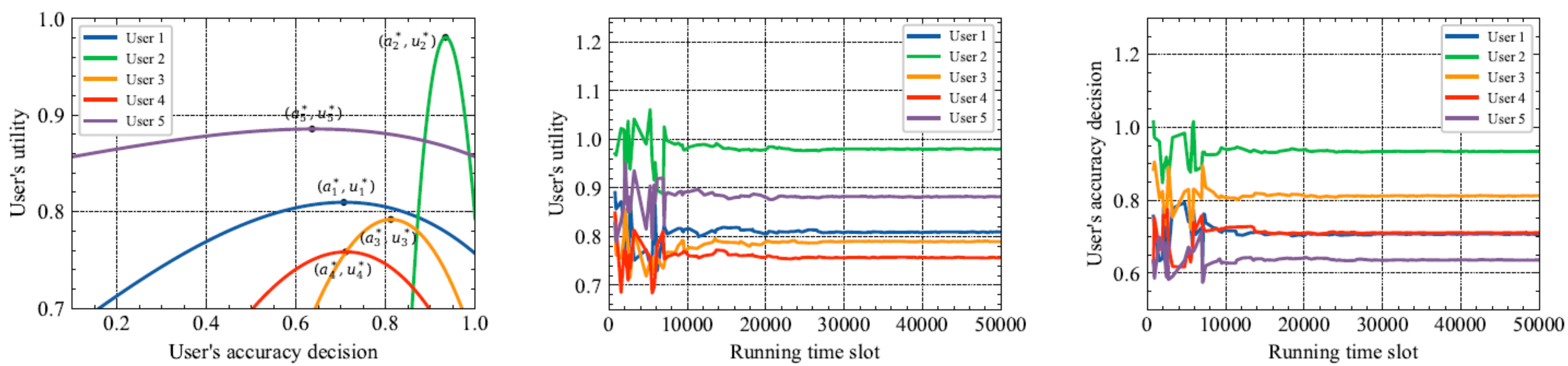
- **Simulation settings**

Para.	Value/ Distribution	Para.	Value/ Distribution
b_n	$N(1,0.1)Mb/s$	M_n	$U(0,1)$
E	$32Mb/s$	K	$0.1Mb$
N	5	M_l	0.7
M_2	0.9	M_3	0.8
M_4	0.7	M_5	0.6

Other baseline approaches:

- **MPPO:** Modified PPO, implemented for competitive multi-agent training.
- **AccuracyPrior:** Giving priority to the accuracy when making the decision.
- **LatencyPrior:** Giving priority to the latency when making the decision.
- **Greedy:** Making the decision with the maximum reward for each time slot.

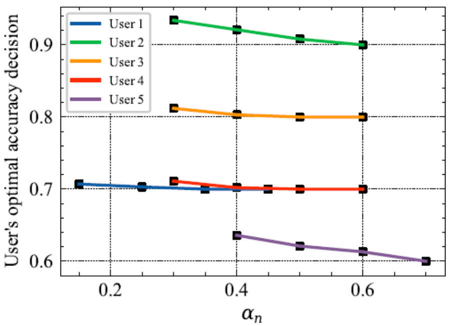
- **Performance of our algorithms**

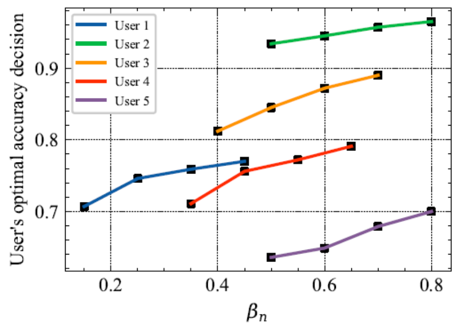


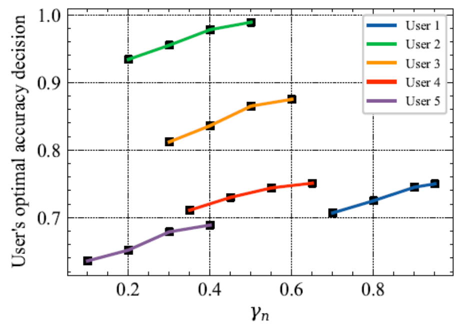
(a) Optimal accuracy decision and utility (b) Utility convergence (c) Accuracy convergence

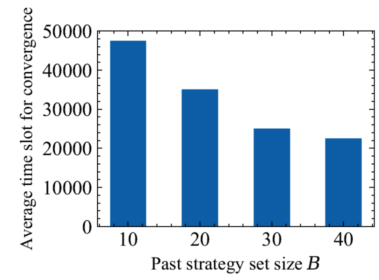
Performance Evaluation

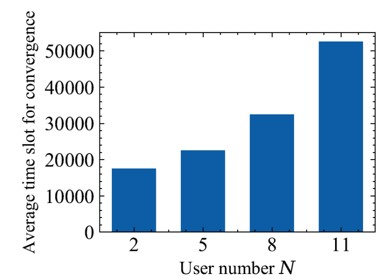
- **Influence of parameters**





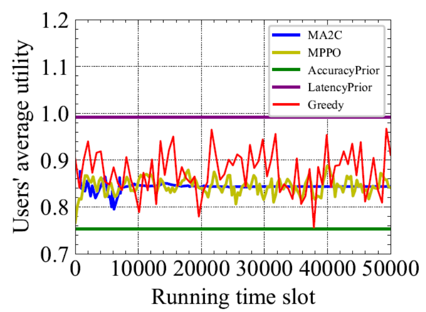




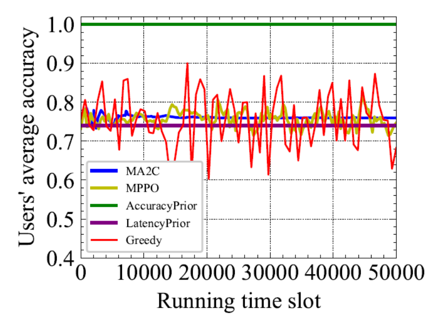


- **Comparing with other baseline approaches**

Our proposed solution: MA2C



(a) Users' average utility



(b) Users' average accuracy

Conclusion

- Study the multi-user edge-assisted video offloading and analyzing problem
- Design utility function based on the video analytics features
- Propose the GT-based algorithm to achieve the Nash equilibrium and the optimal video analytics accuracy
- Propose the RL-based algorithm to tackle the problem without information sharing
- Show that our design has better performance when compared with some other approaches

Thank you!
Q&A