

Orquestração de Multimídia Imersiva Baseada em Aprendizado por Reforço Profundo em Computação Local e de Borda Ciente de Energia

David Galhego¹, Matheus Brito¹, Hugo Santos², Iago Medeiros¹,
Denis Rosário¹, Eduardo Cerqueira¹

¹Universidade Federal do Pará (UFPA) – Belém, PA – Brasil

²Universidade Estadual do Pará (UEPA) – Redenção, PA – Brasil

david.galhego@icen.ufpa.br, matheus.moraes.brito@itec.ufpa.br,
hugo.santos@uepa.br, {iagomedeiros, denis, cerqueira}@ufpa.br

Abstract. *The growing adoption of Immersive Multimedia Systems (IMS) demands ultra-low latencies to prevent cyber-sickness on mobile devices with limited battery and processing capabilities. To mitigate these restrictions, the implementation of Multi-access Edge Computing (MEC) emerges as a viable architecture. In this context, this article analyzes the resource allocation and microservice orchestration problem under energy and delay constraints. Energy-aware orchestration models are evaluated, three of them being based on Deep Reinforcement Learning (DRL), integrating computational, cache, and communication costs. An analysis of the trade-offs between CPU utilization and GPU acceleration is presented, enabled by an action masking technique. Experiments demonstrate that the proposed DRL approach achieves a service acceptance rate of over 95% and reduces mobile energy consumption by up to 30.8%, while maintaining a median end-to-end latency of ≈ 10.6 ms and a high fairness level (Jain's Fairness Index ≈ 0.93).*

Resumo. *A adoção crescente de Sistemas Multimídia Imersivos (IMS) exige latências ultrabaixas para evitar o cyber-sickness em dispositivos móveis com bateria e processamento limitados. Para mitigar essas restrições, a implementação da computação de borda multi-acesso (MEC) surge como uma arquitetura viável. Neste contexto, este artigo analisa o problema de alocação de recursos e orquestração de microsserviços sob restrições de energia e atraso. Avaliam-se modelos de orquestração cientes de energia, sendo três deles baseados em Aprendizado por Reforço Profundo (DRL), integrando custos computacionais, de cache e de comunicação. Apresenta-se uma análise dos trade-offs entre o uso de CPU e a aceleração em GPU, viabilizada por uma técnica de mascaramento de ações. Experimentos demonstram que a abordagem proposta atinge uma taxa de aceitação superior a 95% e reduz o consumo de energia móvel em até 30.8%, mantendo a latência mediana em ≈ 10.6 ms e um alto nível de equidade (Índice de Justiça de Jain ≈ 0.93).*

1. Introdução

A popularização dos Visores Montados na Cabeça (*Head Mounted Displays* - HMDs), também chamados de óculos VR, tem impulsionado a demanda por Sistemas Multimídia Imersivos (IMS) compartilhados e de alta fidelidade. No entanto, esses dispositivos enfrentam restrições severas de bateria e capacidade computacional. Atrasos na

renderização superiores ao limiar de 13 ms frequentemente causam *cyber-sickness*, vulgo enjoo cibernético [Rauschnabel et al. 2022]. Para contornar essa barreira física do hardware local sem sacrificar a imersão do usuário, torna-se necessário descarregar a carga computacional pesada para processamento em nós externos. Nesse cenário, a arquitetura de *Multi-access Edge Computing* (MEC) surge como uma infraestrutura vital, mitigando essas restrições ao fornecer recursos de processamento virtualizados, altamente conectados e heterogêneos (oferecendo tanto CPUs de propósito geral quanto aceleração gráfica dedicada via GPUs) próximos aos usuários finais.

A integração do MEC permite decompor aplicações IMS em cadeias de microsserviços (MSC) [Zhang et al. 2024, Liang et al. 2025]. Otimiza-se o processamento segmentando o serviço: fluxos gerais gerenciam funcionalidades compartilháveis (renderização de fundo e sincronização), enquanto fluxos pessoais focam em dados únicos do usuário (interações e primeiro plano). O desafio analítico e operacional reside em compreender como orquestrar esses microsserviços ao longo dos nós da borda, explorando oportunidades de consolidação nos fluxos gerais e garantindo o processamento isolado dos fluxos pessoais, de modo que a viagem de dados permaneça abaixo do limiar crítico de latência exigido para garantir uma experiência fluida [Santos et al. 2025a].

Apesar dos benefícios evidentes, avaliar a alocação de recursos em cenários IMS revela dinâmicas complexas, especialmente em ambientes multiusuário com alta mobilidade. Orquestradores do estado da arte frequentemente focam na minimização da latência assumindo servidores MEC homogêneos e o uso exclusivo de microsserviços. Essa abordagem falha ao ignorar oportunidades cruciais, como a fusão de fluxos gerais para usuários com alto grau de similaridade de visão (reuso de microsserviços) e a alocação inteligente baseada em aceleradores de hardware (GPUs) para tarefas intensivas. Compreender como a heterogeneidade da rede e o compartilhamento afetam a justiça na latência (*fairness*) e o consumo de energia é essencial para viabilizar sessões colaborativas.

Neste contexto, este artigo analisa o problema de alocação de recursos sob rigorosas restrições de energia e atraso. Propõe-se um modelo de orquestração inteligente, baseado em *Deep Reinforcement Learning* (DRL), para otimizar o roteamento e o posicionamento conjunto de microsserviços. O trabalho realiza uma análise detalhada de *trade-offs* em topologias heterogêneas, investigando o balanço entre o compartilhamento via fusão de fluxos e a consciência de aceleração em hardware, avaliando rigorosamente o impacto na preservação da bateria dos HMDs.

O artigo está organizado da seguinte forma: a Seção 2 apresenta trabalhos relacionados; a Seção 3 detalha a modelagem do sistema e a arquitetura do agente DRL; a Seção 4 discute a avaliação de desempenho; e a Seção 5 apresenta as conclusões.

2. Trabalhos Relacionados

[Brito et al. 2025] avaliaram o orquestrador OSCIM, baseado em Algoritmo Genético (GA), demonstrando ganhos do compartilhamento de microsserviços. Contudo, abordagens meta-heurísticas baseadas em biologia populacional possuem alta complexidade polinomial ($O(N_{gen}P(k \log |V| + C_{fitness}))$), o que resulta em lentidão de convergência e dificuldade inerente em mapear eficientemente clusters heterogêneos de hardware em tempo real sob tráfego dinâmico. Além disso, falham ao não mensurar o esgotamento energético direto dos dispositivos móveis (HMDs). Para superar essas limitações, o modelo proposto neste estudo integra a consciência de aceleração (GPU) e otimiza ativa-

mente a conservação de bateria móvel com decisões em tempo hábil.

Abordagens analíticas baseadas em Aprendizado por Reforço (RL) também ganharam destaque. O sistema Hephaestus [Santos et al. 2025b] introduziu redes neurais para generalizar distribuições em clusters, incorporando GPUs na avaliação. Contudo, focou na distribuição de réplicas, negligenciando a análise do reuso de microsserviços em sessões colaborativas (fusão de fluxos). De forma paralela, [Wang et al. 2025] investigaram o modelo DA-RSPPO, avaliando o balanço entre energia e atraso. Embora comprovem economia nos servidores, o estudo limitou-se à infraestrutura, não analisando o esgotamento da bateria dos usuários.

A literatura evidencia uma lacuna em estudos que integrem decomposição fina de serviços, heterogeneidade de hardware e eficiência de ponta a ponta. De forma complementar, modelos como o de [Dang et al. 2025] propuseram otimizações energéticas em nível de rede, contudo, operam na camada de infraestrutura e não resolvem diretamente a eficiência de renderização no nível da aplicação. Dessa forma, a análise conduzida com a abordagem proposta fornece uma visão holística inédita ao integrar simultaneamente as dinâmicas de reuso, a preservação energética móvel e o suporte direcionado à aceleração por GPU.

3. Modelagem Analítica do Sistema

A análise computacional depende da definição rigorosa do ambiente MEC. Define-se o conjunto de todos os nós como $V = \mathcal{U} \cup \mathcal{M}$ (usuários \mathcal{U} e servidores MEC \mathcal{M}), com unidades de processamento $P = \{\text{CPU}, \text{GPU}\}$. Cada demanda IMS é atendida por uma cadeia Φ_c , composta por microsserviços \mathcal{F}_c . O problema central baseia-se na variável de decisão binária $x_{f,v,p} \in \{0, 1\}$, que assume valor 1 se o microsserviço f for alocado no nó v utilizando a unidade p . As métricas limitantes que orientam essa alocação são os atrasos totais de processamento (D_c^{comp}) e de comunicação (D_c^{comm}).

3.1. Arquitetura da Cadeia de Microsserviços IMS

O sistema IMS é decomposto em duas cadeias paralelas. A cadeia *General* gerencia elementos compartilhados e inclui o *Synchronization Manager* (SM), o *Virtual Object Placement* (VOP) e o *General Purpose Renderer* (GPR) para o cenário de fundo. Simultaneamente, a cadeia *Personal* lida com interações únicas através do *Personalized Purpose Renderer* (PPR), que processa objetos de primeiro plano. Por fim, o *Transcoder* (TC) comprime o vídeo resultante. Essa segmentação permite que o orquestrador identifique que o SM e o GPR podem ser fundidos entre usuários na mesma área, reduzindo a carga computacional no nó MEC sem degradar a experiência individual.

3.2. Definição de Custos e Dinâmicas de Reuso

O modelo incorpora a latência de processamento baseada no fator de aceleração a_p , capturando a discrepância teórica e prática de performance entre CPUs e GPUs [Liu et al. 2021]. O estudo quantifica separadamente a energia gasta por dispositivos móveis (E_c^{mob}) e por servidores (E_c^{mec}). O impacto do reuso é mensurado pela Equação 1: se um microsserviço compatível existe no nó v , a variável $\theta_{f,v,p}$ garante que o modelo reflita um custo operacional inferior.

$$C_{f,v,p}^{\text{comp}} = (1 - \theta_{f,v,p}) \cdot \psi_{f,v,p}^{\text{reuse}} + \theta_{f,v,p} \cdot \psi_{f,v,p}^{\text{new}} \quad (1)$$

As restrições avaliativas exigem que a alocação seja única (Equação 2), que a infraestrutura suporte a demanda de CPU/GPU (Equação 3) e que a latência total respeite o limite de tolerância humana D_c^{\max} (Equação 4).

$$\sum_{v \in V} \sum_{p \in P} x_{f,v,p} = 1 \quad \forall f \in \mathcal{F} \quad (2)$$

$$\sum_{f \in \mathcal{F}} x_{f,v,p} \cdot \mu_{f,v,p} \leq C_{v,p} \quad \forall v \in V, \forall p \in P \quad (3)$$

$$D_c^{\text{total}} \leq D_c^{\max} \quad \forall \Phi_c \in \mathcal{C} \quad (4)$$

3.3. Modelagem do Agente de Orquestração

O orquestrador baseia-se na arquitetura *actor-critic* do algoritmo *Proximal Policy Optimization* (PPO). Para maximizar a estabilidade e prover escalabilidade, a observação de estados formulada no ambiente de simulação mapeia a infraestrutura em vetores normalizados através de redes neurais densas (arquitetura *MlpPolicy* com 2 camadas de 64 neurônios e taxa de aprendizado de 3×10^{-4}). O espaço de observação (s_t) funciona como um dicionário dinâmico que compreende três pilares de dados: o tipo da cadeia SFC (classificando-se as funções virtuais em exclusivas ou compartilháveis por reuso), a proporção de utilização global de banda de rede, e uma matriz detalhada dos recursos disponíveis nos nós válidos $R_{f_k}(m, p, t)$.

A técnica de *Action Masking* $\Xi_t \in \{0, 1\}^{|V|}$ é aplicada nativamente sobre a saída da rede neural para zerar as probabilidades de escolha do espaço de ação discreto para os nós que não possuam recursos suficientes ou que violariam o teto de latência de 13 ms de antemão. Esse mascaramento impede explorações fisicamente impossíveis, acelerando a convergência em topologias estritas [Yuan et al. 2025, Huang and Ontañón 2020].

O treinamento e o direcionamento das ações ocorrem mediante um sistema de recompensas multiobjetivo (Equação 5) cujos hiperparâmetros foram empiricamente ajustados para priorizar limites de tempo e dados. Designaram-se os pesos $\alpha_{\text{lat}} = 5$ e $\alpha_{\text{comm}} = 5$ para aplicar rigor sobre a latência e a banda da infraestrutura (evitando congestionamentos de *backhaul*), e os pesos $\alpha_{\text{comp}} = 1$ e $\alpha_{\text{cache}} = 1$ para controlar a saturação local de processamento. Ademais, para estabilizar o gradiente de recompensa esparsa, definiu-se uma arquitetura baseada em bônus explícitos: uma penalidade rigorosa de -100.0 é aplicada caso a alocação falhe, mitigada por um prêmio imediato de $+100.0$ na aceitação ponta a ponta.

$$r_t = -(\alpha_{\text{comp}} \mathcal{C}^{\text{comp}} + \alpha_{\text{cache}} \mathcal{C}^{\text{cache}} + \alpha_{\text{comm}} \mathcal{C}^{\text{comm}} + \alpha_{\text{lat}} D^{\text{total}}) \quad (5)$$

A complexidade computacional assintótica do agente orquestrador por passo de decisão é matematicamente limitada a $O(FN(T_{sp} + L))$ — ou $O(FN^2 \log N)$ em redes esparsas. Essa eficiência teórica e prática garante que a etapa de inferência do modelo em produção ocorra em tempo hábil para fluxos imersivos (compensando o denso treinamento *offline*), superando gargalos operacionais amplamente reportados em avaliações de meta-heurísticas populacionais clássicas.

4. Análise Experimental e Discussão

Nesta seção, avalia-se o desempenho da abordagem DRL proposta frente às ferramentas DA-RSPPO, Hephaestus e OSCIM, evidenciando seu comportamento sob restrições dinâmicas de recursos e mobilidade.

4.1. Cenário de Avaliação

A análise experimental foi conduzida em um simulador Python3/NetworkX. O cenário replicou o tráfego veicular urbano de Luxemburgo através da extração de dados do SUMO (*Simulation of Urban Mobility*) integrados ao *OpenStreetMap*, justificando realisticamente o uso de usuários movendo-se a velocidades de até 40 km/h. A infraestrutura física modelada consistiu em uma topologia controlada de 35 nós de borda (distribuídos rigorosamente sob a proporção de 1/3 de servidores MEC para 2/3 de roteadores de acesso 5G). Exigiu-se a renderização cooperativa das sessões imersivas a 60 FPS mantendo o teto inflexível de latência em 13 ms. Ademais, o multiplicador da aceleração suportada por GPU no ambiente foi fixado computacionalmente em 17 vezes, de acordo com [Liu et al. 2021]. Para garantir a robustez e a reprodutibilidade dos resultados alcançados pelas redes neurais, foram executadas 33 rodadas de simulação independentes, assegurando um intervalo estrito de confiança estatística de 95%.

Para atestar a eficácia, utilizaram-se métricas de alocação multiobjetivo: Taxa de Aceitação de Serviço (SAR), Uso de GPU por Fluxo (GPU_pF), Consumo de Potência (PC_pF) e Latência de Serviço (SL). Adicionalmente, para quantificar de forma estatisticamente rigorosa a distribuição equitativa dessa latência entre os diferentes usuários operando sob a mesma sessão colaborativa, adotou-se o Índice de Justiça de Jain (JFI), calculado conforme a Equação 6, onde 1 representa a justiça total no atendimento da rede.

$$JFI = \frac{(\sum_{i=1}^n x_i)^2}{n \cdot \sum_{i=1}^n x_i^2} \quad (6)$$

4.2. Discussão de Resultados e Trade-offs Identificados

A Figura 1 demonstra que a Taxa de Aceitação (SAR) da abordagem proposta atinge um valor médio superior a 95%. Este resultado indica que a inteligência no agrupamento fomenta diretamente esse volume de aceitação: o modelo atinge taxas de reuso próximas a 1 instância por fluxo, superando o Hephaestus em até 80%.

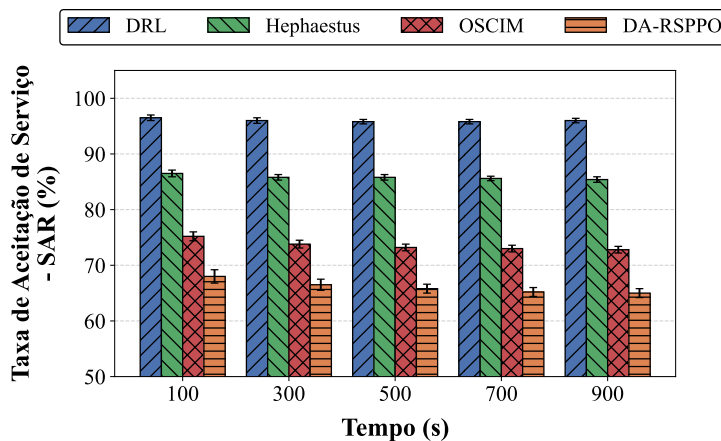


Figura 1. SAR: Taxa de aceitação comparada entre os orquestradores.

A Figura 2 evidencia o *trade-off* de processamento sistêmico detalhado no estudo. Embora o modelo exiba maior uso de CPU local por fluxo (para admitir cadeias complexas que outros descartariam), ele compensa agressivamente na borda. O gráfico detalha o uso de GPU (GPU_{pF}) médio em 2% (até 52% mais econômico que os *baselines*), sugerindo uma alocação que otimiza aceleradores de hardware e mascara nós ineficientes.

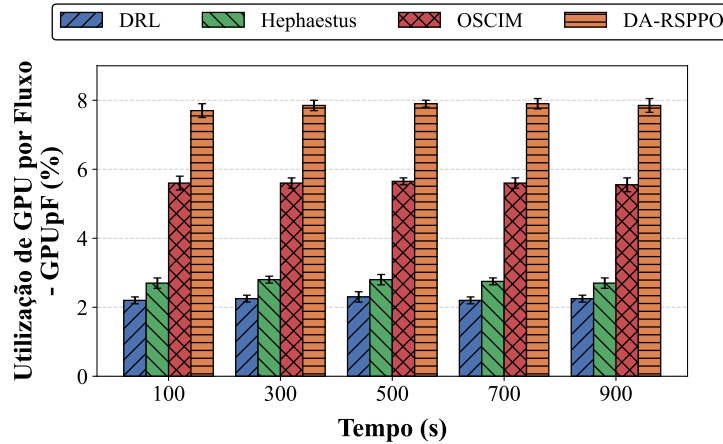


Figura 2. GPU_{pF}: Eficiência e estabilidade na alocação de recursos de GPU.

A Figura 3 mostra a avaliação do Consumo de Potência (PC_{pF}), corroborando a consciência energética da estratégia, que minimiza em 30.8% o dreno de bateria dos usuários frente à abordagem meta-heurística OSCIM. Concomitantemente, a abordagem reduz a banda utilizada em $\approx 46\%$ frente ao DA-RSPPO, evidenciando que a fusão de fluxos evita a saturação do *backhaul*.

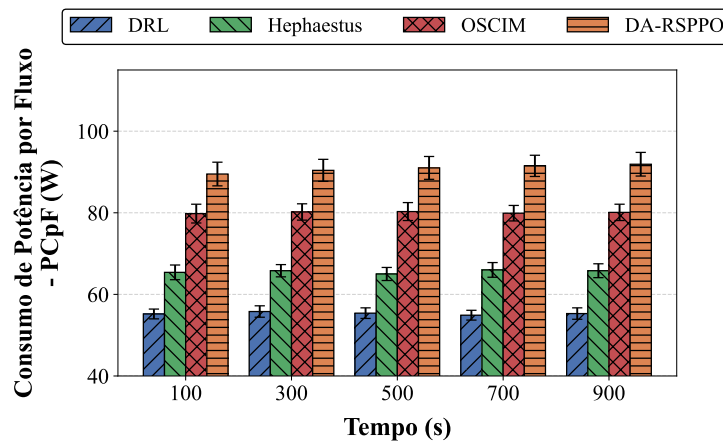


Figura 3. PC_{pF}: Conservação de energia elétrica por fluxo.

A Figura 4 ilustra que a abordagem sustenta uma Latência de Serviço (SL) — a soma dos atrasos de processamento e comunicação — mediana de 10,6 ms, mantendo-se de forma segura abaixo do limiar crítico de 13 ms. A baixa variância indica uma entrega altamente estável. Consequentemente, essa homogeneidade nos atrasos reflete um alto nível de justiça na rede, comprovado por um Índice de Jain (JFI) mediano de 0,93. Isso

confirma estatisticamente que o modelo atua de forma equitativa na mitigação do *cyber-sickness* (enjoo causado pelo conflito sensorial entre o ambiente virtual e o corpo físico) para todos os usuários da sessão colaborativa.

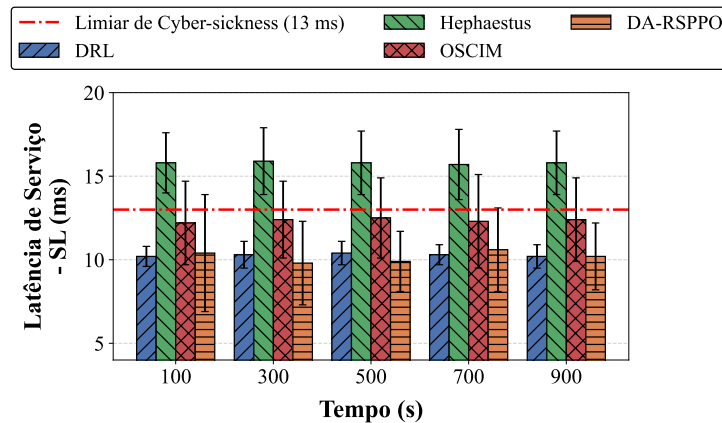


Figura 4. SL: Latência mediana sustentada de forma segura abaixo do limiar de 13 ms.

5. Conclusão

Este estudo conduziu uma análise aprofundada do comportamento de orquestradores baseados em DRL em cenários imersivos (IMS). Observando a dinâmica de um framework provido de mascaramento de ações lógicas, demonstrou-se que é viável interpretar a rede como um ecossistema heterogêneo operando sob restrições críticas.

Os achados evidenciaram que priorizar o reuso de microsserviços gera um *trade-off* favorável: a leve elevação no uso local de CPU compensa ao economizar agressivamente GPU, Banda e Energia Móvel. A abordagem resultou em taxas de aceitação recorde ($> 95\%$), latência de serviço de ≈ 10.6 ms e estabilidade de justiça inter-usuários ($JFI \approx 0.93$), confirmando sua aptidão para prevenir o *cyber-sickness* com decisões de roteamento ultrarrápidas (< 7 ms). Trabalhos futuros focarão em avaliar as variações de processamento em *testbeds* físicos com diferentes arquiteturas de hardware e na integração de modelos perceptuais subjetivos de Qualidade de Experiência (QoE) diretamente nas métricas de recompensa.

Agradecimentos

Este projeto foi apoiado pelo Ministério da Ciência, Tecnologia e Inovações, com recursos da Lei nº 8.248, de 23 de outubro de 1991, no âmbito do PPI-SOFTTEX, coordenado pela Softex e publicado Arquitetura Cognitiva (Fase 3), DOU 01245.003479/2024-10. Esta pesquisa é parte do INCT de Redes de Comunicações Inteligentes e Internet das Coisas (ICoNIoT) financiado pelo CNPq (processo #405940/2022-0) e pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) Código de Financiamento #88887.954253/2024-00. Este trabalho foi financiado pelo Instituto Nacional de Ciência e Tecnologia em Inteligência Artificial Aplicada a Cidades Inteligentes e Sustentáveis na Amazônia Brasileira (INCT IAMazônia - inctiamazonia.org.br) (Processo nº #409001/2024-4).

Referências

- Brito, M., Flexa, R., Santos, H., Vieira, D., Rosário, D., and Cerqueira, E. (2025). An efficient and resilient mechanism for immersive service using microservice chaining. In *2025 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6. IEEE.
- Dang, A. K., Khalifé, H., Sintorn, M., Rovedakis, S., and Secci, S. (2025). Data-driven energy optimization in mobile networks with user experience guarantees. In *IEEE INFOCOM 2025*. IEEE.
- Huang, S. and Ontañón, S. (2020). A closer look at invalid action masking in policy gradient algorithms. *arXiv preprint arXiv:2006.14171*.
- Liang, J., Li, W., and Leung, V. C. (2025). Stateful virtual network function decomposition and deployment with reliability guarantee in edge networks. *IEEE Transactions on Network and Service Management*.
- Liu, X., Yang, J., Zou, C., Chen, Q., Yan, X., Chen, Y., and Cai, C. (2021). Collaborative edge computing with fpga-based cnn accelerators for energy-efficient and time-aware face tracking system. *IEEE Transactions on Computational Social Systems*, 9(1):252–266.
- Rauschnabel, P. A., Felix, R., Hinsch, C., Shahab, H., and Alt, F. (2022). What is xr? towards a framework for augmented and virtual reality. *Computers in Human Behavior*, 133:107289.
- Santos, J., Van Damme, S., Sameri, J., Schwarzmann, S., Wei, Q., Trivisonno, R., De Turck, F., and Vega, M. (2025a). Leveraging user perception for 6g edge-cloud orchestration of networked extended reality. *IEEE Communications Magazine*.
- Santos, J., Zaccarini, M., Poltronieri, F., Tortonesi, M., Stefanelli, C., Di Cicco, N., and De Turck, F. (2025b). Hephaestusforge: Optimal microservice deployment across the compute continuum via reinforcement learning. *Future Generation Computer Systems*, 166:107680.
- Wang, L., Liu, X., Ding, H., Hu, Y., Peng, K., and Hu, M. (2025). Energy-delay-aware joint microservice deployment and request routing with dvfs in edge: a reinforcement learning approach. *IEEE Transactions on Computers*.
- Yuan, M., Yu, Q., Zhang, L., Lu, S., Li, Z., and Pei, F. (2025). Deep reinforcement learning based proximal policy optimization algorithm for dynamic job shop scheduling. *Computers & Operations Research*, page 107149.
- Zhang, X., Huang, Z., Huang, L., and Yang, H. (2024). Resource allocation strategy based on service function chaining in multi-access edge computing network. *Journal of King Saud University-Computer and Information Sciences*, 36:102001.