

边缘环境下推理精度感知的服务部署与模型选择

陈哲毅^{1),2),3)} 池梦莉^{1),2),3)} 张俊杰^{1),2),3)} 林锦晖^{1),2),3)}
闵革勇⁴⁾ 李 颀⁵⁾

¹⁾(福州大学计算机与大数据学院 福州 350116)

²⁾(大数据智能教育部工程研究中心 福州 350002)

³⁾(福建省网络计算与智能信息处理重点实验室(福州大学) 福州 350116)

⁴⁾(埃克塞特大学计算机科学系 埃克塞特 EX4 4QF 英国)

⁵⁾(上海交通大学计算机科学与工程系 上海 200240)

摘 要 随着边缘计算与人工智能技术的快速发展,深度神经网络(Deep Neural Networks,DNNs)已被广泛应用于边缘环境中,为终端用户提供智能化服务。然而,面对有限的边缘资源与多样化用户需求之间的矛盾,如何实现高效的服务部署与模型选择已成为一个挑战性难题。现有研究大多未充分考虑推理精度对服务体验的重要影响,难以有效应对动态变化的系统状态与用户需求,进而导致资源浪费或服务降级。为了解决这一重要挑战,本文提出了一种新颖的边缘环境下推理精度感知的服务部署与模型选择(Service Deployment and Model Selection,SDMS)方法。首先,利用自注意力机制对未来时间隙中的用户请求流量分布进行预测,并结合剪枝搜索策略对服务部署方案进行优化。其次,设计了一种新型的基于深度模仿学习(Deep Imitation Learning,DIL)的引导式深度强化学习(Deep Reinforcement Learning,DRL)框架,通过引入专家经验以加速收敛并快速逼近最佳的模型选择与资源分配决策。基于真实用户通信流量数据和DNN推理数据集,大量实验验证了所提出SDMS方法的有效性。实验结果表明,相比于其他基准方法,SDMS方法在不同场景下均展现出了更加优越的服务收益、延迟与精度满足率,有效实现了多目标之间的平衡。

关键词 边缘计算;服务部署;模型选择;自注意力机制;深度强化学习;深度模仿学习

中图法分类号 TP18

DOI号 10.11897/SP.J.1016.2026.02274

Inference Accuracy-Aware Service Deployment and Model Selection in Edge Environments

CHEN Zhe-Yi^{1),2),3)} CHI Meng-Li^{1),2),3)} ZHANG Jun-jie^{1),2),3)} LIN Jin-Hui^{1),2),3)}
MIN Ge-Yong⁴⁾ LI Jie⁵⁾

¹⁾(College of Computer and Data Science, Fuzhou University, Fuzhou 350116)

²⁾(Engineering Research Center of Big Data Intelligence, Ministry of Education, Fuzhou 350002)

³⁾(Fujian Key Laboratory of Network Computing and Intelligent Information Processing (Fuzhou University), Fuzhou 350116)

⁴⁾(Department of Computer Science, University of Exeter, Exeter EX4 4QF, UK)

⁵⁾(Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240)

Abstract With the rapid advancement of edge computing and artificial intelligence technologies, Deep Neural Networks (DNNs) have been extensively deployed in edge environments to provide

收稿日期:2025-09-12;在线发布日期:2026-03-24。本课题得到国家自然科学基金(62202103)、福建省杰出青年科学基金(2025J010020)、中央引导地方科技发展资金项目(2022L3004)、福厦泉国家自主创新示范区电力物联网平台关键技术及应用协同创新平台项目(2023-P-002)资助。陈哲毅,博士,教授,博士生导师,中国计算机学会(CCF)专业会员,主要研究领域为云边协同计算、资源智能优化。E-mail: z.chen@fzu.edu.cn。池梦莉,硕士研究生,主要研究领域为移动边缘计算、服务部署、深度强化学习。张俊杰(通信作者),博士研究生,主要研究领域为移动边缘计算、模型推理加速、资源优化。E-mail: junjiefzu@qq.com。林锦晖,硕士研究生,主要研究领域为移动边缘计算、服务部署、模型选择。闵革勇(通信作者),博士,教授,博士生导师,主要研究领域为计算机网络、无线通信、并行与分布式计算。E-mail: g.min@exeter.ac.uk。李 颀,博士,教授,博士生导师,IEEE会士,主要研究领域为大数据与人工智能、区块链、边缘计算。

intelligent services to end-users. However, a critical challenge arises due to the inherent contradiction between the limited computational and storage resources of edge servers and the increasingly diverse and dynamic demands of users. Efficient service deployment and model selection in such resource-constrained and demand-heterogeneous scenarios have become a significant problem. Existing studies often neglect the critical impact of inference accuracy on overall service quality, which leads to suboptimal system performance when handling dynamically changing system states and user requirements. This oversight can result in resource underutilization, service degradation, and an inability to satisfy multiple objectives simultaneously. Addressing this challenge requires a holistic approach that jointly considers prediction of user demand, resource constraints, model performance, and service quality requirements. To tackle this problem, this paper proposes a novel inference-accuracy-aware service deployment and model selection framework for edge environments, referred to as Service Deployment and Model Selection (SDMS). The core idea of SDMS is to optimize both the deployment of services across edge nodes and the selection of DNN models for inference tasks to maximize service utility while satisfying latency and accuracy constraints. Specifically, the proposed approach consists of two key components. First, a self-attention mechanism is employed to predict the distribution of user request traffic in future time slots, capturing temporal correlations and dynamic patterns in user demands. This prediction is then combined with a pruning-based search strategy to identify near-optimal service deployment schemes, thereby ensuring that limited edge resources are allocated efficiently to meet expected demand. Second, a novel guidance-based Deep Reinforcement Learning (DRL) framework is designed, which is enhanced by Deep Imitation Learning (DIL). By incorporating expert knowledge and historical experience into the learning process, the DIL-guided DRL agent can significantly accelerate convergence and rapidly approach near-optimal decisions for model selection and resource allocation. This integration of imitation learning and reinforcement learning allows the framework to adaptively balance multiple objectives, including service latency, inference accuracy, and overall system throughput, under dynamically changing conditions. The proposed SDMS framework is extensively evaluated using real-world user traffic traces and DNN inference datasets. A comprehensive set of experiments is conducted to assess its effectiveness in various realistic scenarios, including varying traffic patterns, heterogeneous model characteristics, and different resource availability levels. Experimental results demonstrate that the SDMS method consistently outperforms existing baseline approaches in terms of service utility, latency, and accuracy satisfaction rates. In particular, the framework exhibits a robust ability to maintain high inference accuracy while meeting strict latency requirements, effectively achieving a balance among multiple conflicting objectives. In summary, the SDMS method provides a systematic and effective solution for inference-accuracy-aware service deployment and model selection in edge computing environments. By leveraging predictive modeling, pruning-based optimization, and imitation-guided reinforcement learning, the framework can dynamically adapt to user demand fluctuations and resource constraints, thereby enhancing the quality of service delivered to end users. The proposed approach offers a valuable reference for the design of intelligent edge systems where both computational efficiency and inference accuracy are critical for user satisfaction and overall system performance. The contributions of this work demonstrate the potential of combining advanced AI techniques with edge resource management to address real-world challenges in next-generation intelligent services.

Keywords edge computing; service deployment; model selection; self-attention mechanism; deep reinforcement learning; deep imitation learning

1 引 言

随着人工智能技术的快速发展,深度神经网络(Deep Neural Networks,DNNs)已被广泛用于支持和增强许多新兴的智能应用(如自动驾驶、视频分析和聊天机器人等^[1])。经典的云计算模式已难以高效应对用户对推理速度与隐私保护日益增加的需求^[2]。为了缓解该问题,边缘计算将计算和存储资源下沉至网络边缘,以提高系统响应速度和减轻核心网络压力^[3]。在边缘计算中,边缘服务提供商(Edge Service Provider,ESP)为用户提供 DNN 模型部署以完成服务请求,但仍面临着任务调度、资源分配、成本控制等一系列技术难题^[4-5]。

大部分现有工作^[6-8]聚焦于通过优化调度策略以降低计算密集型任务的延迟与能耗,但忽略了 DNN 推理精度对服务质量的影响。例如,Bi 等人^[9]通过复杂的混合整数非线性规划进行建模,以联合优化服务缓存、计算卸载和资源分配。Zhao 等人^[10]针对无人机辅助边缘系统,利用李雅普诺夫优化联合决策无人机轨迹和任务卸载,以最大化系统吞吐量。Zhou 等人采用深度强化学习(Deep Reinforcement Learning,DRL)来联合优化三层移动云结构中的计算卸载、服务缓存与资源分配策略,以最小化云数据中心的运营成本并保证用户时延要求。然而,现有工作通常将 DNN 推理过程抽象为统一的计算单元,其核心优化目标始终围绕时延、能耗或运营成本等系统级性能指标。本质上,这种处理范式延续了对于传统计算任务的优化思路,忽视了 DNN 推理精度这一关键属性对服务体验的重要影响。这种局限性的根源在于,推理精度通常由模型结构、资源分配策略等多个因素共同决定,其呈现出强耦合且跨任务异质的特性,使得将其融入至现有资源优化框架中变得极具挑战性。

在实际 DNN 应用场景中,推理精度与服务体验具有强相关性^[11]。例如,在自动驾驶中,目标检测精度稍有偏差都可能导致严重的交通事故。此外,为满足多样的用户需求与资源限制,许多 DNN 模型提供了不同规模的架构^[12]。例如,YOLOv8 系列包含了 Nano/Small/Medium/Large/Extra-Large 五个版本,其每个版本在模型复杂度、计算量和推理精度上都具有一定差异。通过合理选择模型配置,能够在满足不同用户需求的基础上,进一步提升用户的服

务体验。然而,如何实现边缘环境下高效的服务部署与模型选择仍面临以下关键挑战:

(1)难以平衡推理精度与延迟进而提升服务体验质量。动态的计算资源与多样的任务属性给精准的体验质量刻画带来了较大挑战。对于用户动态的推理请求,不同规模的网络架构会产生差异化的推理精度与延迟。同时,所分配的计算频率、上传带宽等因素也会影响推理延迟。此外,对于各类型服务,用户通常有着不同的服务等级目标(Service Level Objective,SLO)。例如,对于目标检测类服务,用户倾向于对检测目标的实时感知;对于图像生成类服务,用户倾向于生成更高质量的图像。因此,如何根据服务类型平衡推理精度与延迟以提升体验质量需要被仔细考虑。虽然一些工作^[14-15]考虑了推理精度和时延对体验质量的影响,但它们大多是针对某一特定推理服务而设计的,无法有效扩展至其他推理服务类型,难以满足实际应用对多样化推理服务的动态需求。

(2)难以在内存限制下高效部署服务与模型进而满足用户服务偏好。不同服务与模型的差异化特征(如,内存消耗、推理精度和计算延迟)给为服务选择合适的模型部署方案造成了巨大的挑战。对于边缘环境中的用户,其对各类型服务的请求流量模式可能存在较大差异,且请求流量模式会随着时间的推移发生动态变化。因此,服务部署方案需要能够有效应对这种动态性,同时还需要兼顾不同服务之间的公平性,避免某些请求因无法得到及时响应而造成长距离的云端传输时延。虽然一些工作^[16-17]尝试对不同服务的请求流量模式进行预测,但受限于固有的长距离依赖问题,其无法有效捕获长期序列依赖关系,因此难以准确预测用户请求模式随时间的长期变化趋势。

(3)难以选择最佳模型配置进而最大化服务收益。基于服务与模型部署方案,如何进一步选择最佳模型配置以最大化服务收益是一个关键问题。相比于高精度的深层 DNN 模型,浅层 DNN 模型通常具有更快的推理速度,但也可能出现推理精度的严重下降,尤其在面对复杂的任务输入时。本文通过实验发现,不同规模的模型在各类任务上的性能差异呈现多样性。例如,对于某些任务,深层 DNN 模

型的精度反而不如浅层 DNN 模型。同时,这种差异性随不同的任务输入呈现动态变化(即,对具有不同特征的任务所表现出的性能差异可能会发生变化)。作为一种新兴的智能决策算法,DRL 已被应用于边缘环境下的许多优化问题(如,计算卸载和资源分配等^[18-20]),可作为选择模型配置的一种潜在可行的解决方案。然而,在面对新环境时,经典的 DRL 需要从头训练智能体并逐步优化策略,导致了过度的训练时间与资源开销。此外,现有解决方案通常忽略了模型选择与资源分配决策的问题规模,难以取得快速稳定的收敛。

为了解决上述重要挑战,本文提出了一种新颖的边缘环境下推理精度感知的服务部署与模型选择(Service Deployment and Model Selection, SDMS)方法。本文的主要贡献总结如下:

(1)提出了一种统一的面向边缘环境推理精度感知的服务部署与模型选择系统模型,以应对多样化的服务请求类型和模型规模配置。所提出的系统模型综合考虑了推理延迟、推理精度和内存限制等多个因素,旨在兼顾用户服务质量的同时提升边缘资源的利用效率。

(2)设计了一种基于自注意力机制的主动式服务部署方法。基于历史用户服务请求偏好,利用概率稀疏自注意力机制以预测未来时隙中用户对不同服务的请求流量分布。进一步地,为每种模型的不同属性构建潜在价值函数,并设计了一种高效的剪枝搜索策略以获取和当前用户请求流量分布所匹配的最佳服务部署方案。

(3)设计了一种基于深度模仿学习(Deep Imitation Learning, DIL)引导式 DRL 的模型选择与资源分配方法。通过引入广义优势估计和裁剪目标函数,进而高效选择合适的模型以提升 DNN 推理的服务质量。特别地,通过引入 DIL,所提出的方法能够充分利用专家经验引导 DRL 智能体加速模型收敛并实现探索与利用之间的良好平衡。

(4)基于真实用户通信流量与 DNN 推理数据集,大量实验验证了所提出 SDMS 方法的有效性。结果表明,SDMS 方法能够快速有效地做出服务部署与模型选择决策。与其他基准方法相比,SDMS 方法在不同场景下均展现出了更加优越的服务收益、延迟和精度满足率。

本文其他部分的组织如下。第 2 节回顾并分析了相关工作。第 3 节描述了所提出的系统模型并对优化问题进行了形式化定义。第 4 节详细介绍了所

提出的 SDMS 方法。第 5 节评估了 SDMS 方法并与其他基准方法进行了大量对比实验。第 6 节对本文的工作进行了总结。

2 相关工作

近年来,边缘环境下的服务部署问题受到了学术界和工业界的广泛关注。本节从延迟感知的服务部署与精度感知的服务部署两个不同的研究角度回顾并分析了相关工作。

2.1 延迟感知的服务部署

通过联合优化服务部署与计算卸载等问题,许多现有工作致力于优化边缘系统的整体延迟与资源效率。例如,Zhang 等人^[7]针对多用户多任务边缘场景下的服务缓存、计算卸载和资源分配联合问题,提出了一种基于半正定松弛的近似算法以降低系统的整体延迟。Zhou 等人^[8]针对无人机辅助边缘场景下的计算卸载与服务缓存问题,设计了一种基于 Lyapunov 优化的二维时间尺度方法,旨在降低服务延迟。Zhou 等人^[21]开发了一种基于异步优势 Actor-Critic 的服务缓存与计算卸载策略,以在保证任务延迟约束的同时降低系统资源成本。Li 等人^[22]以最小化服务请求响应延迟为优化目标,利用了一种改进的蚁群算法以求解最优服务部署方案。然而,这些工作通常面向的是单一类型的计算密集型服务,忽略了实际边缘环境中多样化的服务类型与模型配置,因此难以有效适应现实场景中智能应用日益增加的动态需求。

针对上述问题,一些工作通过考虑服务与任务的多样性以使得所提出的解决方案能够更加贴近现实需求。例如,Li 等人^[18]利用贝叶斯优化寻找最佳的云配置和 DNN 部署方案,旨在实现一种自动化的云服务部署。Mao 等人^[23]提出了一种面向通用 DNN 应用程序的服务迁移框架,旨在降低设备端的计算成本和内存使用。Zeng 等人^[24]根据设备计算能力和网络状态对 DNN 推理服务的负载进行动态拆分,以改善系统延迟和能量消耗。Cui 等人^[25]设计了一种基于自适应权重采样机制的 DRL 算法以优化动态 B5G/6G 网络中多类型任务的协同卸载,旨在最小化任务执行时延。虽然这些工作进一步考虑了多样化的服务与模型类型,但它们仅关注推理延迟与能耗等通用指标,忽略了 DNN 推理精度对服务体验的关键影响,因此难以实现推理精度与延迟之间的良好平衡。

2.2 精度感知的服务部署

一些工作从提升 DNN 推理精度的角度出发,研究如何通过优化服务部署与任务调度进而取得推理精度与延迟之间更好的平衡。例如,Wang 等人^[15]提出了一种基于服务编排的多范式部署方案,旨在寻求云边环境下 DNN 推理服务系统准确性、服务规模和部署成本之间更好的平衡。Kang 等人^[26]设计了一种服务质量感知的推理加速控制回路,以动态调整 DNN 网络规模,旨在降低推理延迟和精度损失。Zhang 等人^[27]开发了一种 SLO 感知的渐进式推理调度框架,以在提高边缘集群吞吐量的同时保持推理高精度。Wu 等人^[28]设计了一种面向异构基础设施服务的推理 workflow 调度系统,以选择满足精度目标的最具成本效益的模型,进而在基础设施的不同层级中部署服务。然而,这些工作大多采用的是启发式搜索或基于控制理论的决策算法,难以适应边缘环境的动态性与服务的多样性。

针对上述问题,一些工作引入了新兴的深度强化学习(Deep Reinforcement Learning, DRL)。例如,Fan 等人^[16]针对工业物联网中端到端的 DNN 联合任务推理问题,提出了一种基于深度确定性策略梯度(Deep Deterministic Policy Gradient, DDPG)的多任务多服务场景资源管理方案,旨在降低任务处理延迟和推理损失。Li 等人^[19]设计了一种基于 DRL 的用户端服务部署方法,在仅掌握少量任务和用户属性的情况下,可根据节点状态实现对用户延迟的优化。Yao 等人^[29]通过结合数字孪生(Digital Twin, DT)与边缘计算,提出了一种基于图注意力机制的多智能体 DRL 方法以优化计算卸载与服务缓存策略,旨在最大化系统收益。虽然这些工作展现出了良好的决策效果,但它们大多针对的是小规模的状态和动作空间。随着服务类型与模型规模的增加,上述解决方案面临着探索效率低下甚至难以收敛的严峻问题。此外,现有工作大多忽略了边缘环境下用户对不同服务多变的偏好模式。

3 系统模型与问题定义

3.1 系统模型概览

所提出的面向边缘环境推理精度感知的服务部署与模型选择系统模型如图 1 所示。为了降低用户推理延迟,ESP 租用边缘服务器的计算资源为用户

提供 DNN 推理服务(如知识问答、目标检测、自动驾驶等)。为了兼顾不同用户的需求并为其提供定制化服务,ESP 为其各类型服务提供了多种模型配置(如 YOLOv8 系列中的 YOLOv8N、YOLOv8S 等),这些模型具有不同的网络规模,同时也会产生不同的推理精度与延迟。因此,ESP 需要为用户提供合适的模型配置以进一步提升服务体验。然而,受限于边缘可用资源,ESP 无法在线部署所有的模型。因此,需要合理选择部署部分模型,以在保证服务质量的同时提升边缘资源利用效率。同时,ESP 需要为不同用户设备发起的推理请求分配合适的带宽资源以保证稳定的数据传输。具体而言,所提出系统基于多时隙运行。在每个时隙 $t \in T = \{1, 2, \dots, T\}$, 用户的推理任务 i 记为一个五元组 $\langle d_i, s_i, \eta_i, acc_i, \rho_i \rangle$, 其中 d_i 表示任务大小, $s_i \in S = \{s_1, s_2, \dots, s_J\}$ 表示服务类型, $\eta_i \in \eta = \{\eta_{i,1}, \eta_{i,2}, \dots, \eta_{i,K}\}$ 表示在不同配置下的计算密度, $acc_i \in ACC = \{acc_{i,1}, acc_{i,2}, \dots, acc_{i,k}\}$ 表示在不同配置下的推理精度, ρ_i 表示任务优先级。该优先级由任务的服务质量(Quality-of-Service, QoS)需求动态确定,其用于衡量任务的紧迫性和重要性。任务的计算密度和推理精度会随资源配置的不同而变化。因此,在不同资源配置下的表现将直接影响任务的时延、精度满足率和资源消耗,从而影响服务收益。表 1 列出了主要符号及其定义。

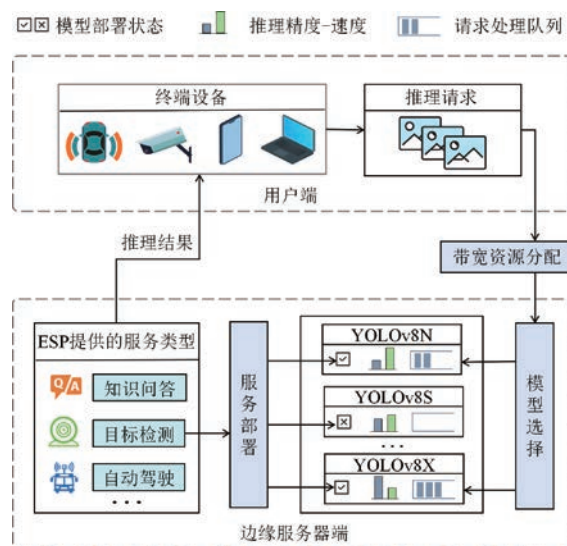


图 1 面向边缘环境精度感知的服务部署与模型选择系统

3.2 通信与计算模型

当任务被上传至边缘服务器后,所提出系统会为其选择一个合适的模型配置以完成推理。若用户请求服务所需的模型未被部署,该请求将被转发至远程云进行处理。若请求服务所需的模型已被部署,任务将直接在边缘服务器上处理。在任务的

表 1 主要符号及其定义

符号	定义
S	服务类型的集合
η	计算密度的集合
ACC	推理精度的集合
ρ_i	任务优先级
J	服务类型的数量
K	配置类型的数量
f^{edge}	边缘服务器的计算能力
$y_{i,k}$	任务 i 的模型配置决策
f^{cloud}	云服务器的计算能力
$x_{j,k}$	模型部署决策
T_i^{total}	处理任务 i 的总延迟
R^t	时隙 t 内的服务收益
$m_{j,k}$	部署服务类型 j 对应的配置 k 所需的内存
M	边缘环境下总的可用内存

处理过程中,需经过上传和执行两个阶段。具体而言,在任务 i 的上传阶段,根据香农信道容量公式^[30],其上传速率定义为

$$r_i = b_i \log_2 \left(1 + \frac{p g_i}{\sigma^2} \right) \quad (1)$$

其中, b_i 表示分配给任务 i 的带宽, p 表示信号功率, g_i 表示信道增益, σ^2 表示信号传输过程中的噪声功率。

因此,任务 i 的上传时延定义为

$$T_i^{\text{up}} = \frac{d_i}{r_i} \quad (2)$$

由于边缘服务器资源有限,任务需在排队中等待前序任务完成推理后方可进行处理,此外,在一个时隙结束时,未完成的任务将被丢弃,避免出现任务堆积,从而保证系统在动态边缘环境中的稳定性和可控性。因此,任务 i 的排队时延可表示为

$$T_i^{\text{que}} = \sum_{q=1}^{|Q_s|} \frac{d_{i,q} \eta_{i,q}}{f^{\text{edge}}} \quad (3)$$

其中, Q_s 表示边缘服务器中当前待处理的队列, $d_{i,q}$ 和 $\eta_{i,q}$ 分别表示队列中第 q 个任务的输入数据大小及其所选模型配置的计算密度, f^{edge} 表示边缘服务器的计算能力(即, CPU 频率)。

当在边缘服务器上执行 DNN 推理时,任务 i 的执行时延定义为

$$T_i^{\text{exe}} = \frac{\sum_{k=1}^K \eta_{i,k} y_{i,k}}{f^{\text{edge}}} \quad (4)$$

其中, $y_{i,k} \in \{0,1\}$ 表示为任务 i 分配的模型配置决策。若任务 i 选择配置 k , $y_{i,k} = 1$; 否则, $y_{i,k} = 0$ 。

因此,在边缘处理任务 i 所需的总时延定义为

$$T_i^{\text{edge}} = T_i^{\text{up}} + T_i^{\text{que}} + T_i^{\text{exe}} \quad (5)$$

当所需模型未被部署在边缘服务器上时,任务将被上传至远程云进行处理。该过程的时延定义为

$$T_i^{\text{cloud}} = \frac{d_i}{r^{\text{cloud}}} + \frac{\sum_{k=1}^K \eta_{i,k} y_{i,k}}{f^{\text{cloud}}} \quad (6)$$

其中, r^{cloud} 表示云服务器的上传速率, f^{cloud} 表示云服务器的计算能力。

因此,处理任务 i 所需的总时延定义为

$$T_i^{\text{total}} = (1 - x_{j,k}) T_i^{\text{edge}} + x_{j,k} T_i^{\text{cloud}} \quad (7)$$

其中, $x_{j,k} \in \{0,1\}$ 表示模型部署决策。若任务 i 所需的服务类型 j 对应的配置 k 已被部署在边缘服务器, $x_{j,k} = 1$; 否则, $x_{j,k} = 0$ 。

对于用户的服务请求,不同模型配置对应的计算密度和推理精度都具有差异性^{[[31-33]]}。根据用户所选择的模型配置,任务 i 的推理精度可表示为

$$Acc_i = \sum_{k=1}^K acc_{i,k} y_{i,k} \quad (8)$$

3.3 收益模型

ESP 服务的目标是在用户可容忍的时延内提供令其满意的推理结果。因此,当任务的总处理时间和推理精度都能满足用户需求时,ESP 可从用户获得收益。因此,将时隙 t 内的单位收益定义为

$$v_i^t = \begin{cases} \Phi, & \text{当 } Acc_i \geq \widehat{Acc}_j \text{ 并且 } T_i^{\text{total}} \leq \hat{T}_j \\ 0, & \text{其他情况} \end{cases} \quad (9)$$

其中, $\widehat{Acc}_j$ 和 $\hat{T}_j$ 分别表示服务 j 的最低容忍精度和最高容忍时延。

当完成优先级越高的用户的任务时,ESP 所能获得的收益越高。在时隙 t 内,服务收益定义为

$$R^t = \sum_{i=1}^m v_i^t \rho_i \quad (10)$$

其中, m 表示在时隙 t 内 ESP 所服务的用户数量。

3.4 问题定义

基于所提出的系统模型,本文的优化目标为在系统内存限制下最大化系统长期收益。该优化问题可形式化表示为

$$\begin{aligned}
 P1: & \max_{x,y,b} \sum_{t=1}^T R^t \\
 \text{s. t.} \quad & C1: x_{j,k} \in \{0,1\}, \forall j, \forall k, \forall t, \\
 & C2: \sum_{j=1}^J \sum_{k=1}^K x_{j,k} m_{j,k} \leq M, \forall j, \forall k, \forall t, \\
 & C3: y_{i,k} \in \{0,1\}, \forall i, \forall k, \forall t, \\
 & C4: \sum_{k=1}^K y_{i,k} = 1, \forall i, \forall t, \\
 & C5: b_i \in [0,1], \forall i, \forall t, \\
 & C6: \sum_{i=1}^I b_i \leq 1, \forall t
 \end{aligned} \quad (11)$$

其中, $m_{j,k}$ 表示部署服务类型 j 对应的配置 k 所需的内存, M 为边缘环境下总的可用内存。C1 表示部署决策的二元性约束, 即模型是否部署在边缘服务器上; C2 表示内存容量约束, 其用于保证部署方案不超过边缘内存容量; C3 表示配置选择的二元性约束, 即是否为服务选择对应的模型配置; C4 表示单任务的唯一配置约束, 即每个任务在一个时隙内最多只能选择一个模型配置进行部署; C5 表示带宽分配的范围约束, 其表示分配给任务的带宽比例。C6 表示带宽总量限制约束, 即为所有任务分配的带宽比例之和不能超过总可用带宽。

4 所提出的 SDMS 方法

4.1 概览

针对问题 P1, 本文提出一种新颖的边缘环境下推理精度感知的服务部署与模型选择方法 SDMS, 旨在通过对多样化服务请求和模型配置的精准分析与匹配, 以最大化服务收益。图 2 概览了所提出的 SDMS 方法, 其主要步骤如算法 1 所示。每隔 T^{deploy} 个时隙, 调用算法 2 获取最新的服务部署方案, 执行并更新服务部署状态(第 2~5 行)。具体而言, 本文

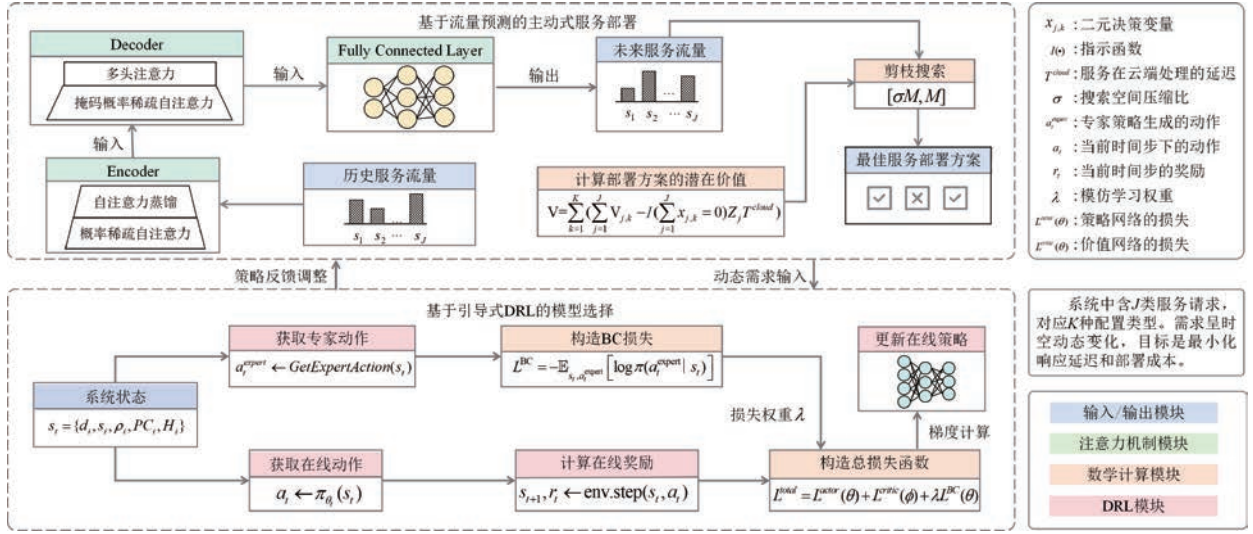


图 2 所提出 SDMS 方法的概览

设计了一种基于自注意力机制的流量预测模型以获取未来时隙的服务流量分布, 并基于此利用剪枝搜索得到每种服务对应的最优模型部署方案。

算法 1. 所提出的 SDMS 方法

输入: 模型集合 V 、时隙数量 T 、服务部署间隔

输出: 服务收益 R'

1. FOR $t = 1, 2, \dots, T$ DO
2. IF $t \% T^{\text{deploy}} = 0$ DO
3. 调用算法 2 获取服务部署方案 p ;
4. 执行服务部署方案并更新服务部署状态 S_t^{deploy} ;
5. END IF
6. FOR $i = 1, 2, \dots, m$ DO
7. 接收用户向边缘服务器发起的推理任务请求;
8. 调用算法 3 生成模型选择与资源分配方案 F^i ;
9. 将任务送入到指定模型中执行推理;
10. 根据任务完成情况计算当前时隙的收益 R' ;
11. 收集任务完成状态与服务流量;
12. END FOR
13. END FOR

在每个时隙 t , 调用算法 3 为推理任务请求选择合适的模型配置与资源分配方案, 并在任务执行完成后收集任务完成状态与服务流量(第 6~12 行)。具体而言, 本文通过将任务的大小、任务在不同配置下的计算密度等属性作为状态信息输入到智能体中进行模型配置的决策。通过离线搜索获取专家动作构造行为克隆损失, 并联合智能体与环境探索过程中的在线损失以更新策略网络, 进而加速智能体对最佳模型配置策略的拟合。

4.2 基于流量预测的主动式服务部署

服务部署旨在边缘内存限制下, 合理部署适合当前任务精度和延迟需求的服务。然而, 不同服务的用户请求流量存在显著差异并且随着时间会发生动态变化。因此, 需要能够为不同服务分别分配合适的内存并根据系统状态变化进行动态调整。同时, 由于不同服务所产生的推理延迟和精度的变化范围可能较大, 需要根据每种服务所包含不同模型的属性进而选择合适的模型。

通过利用历史流量变化特征可对预测未来用户

请求模式进行预测,进而支持服务部署^[34]。为了提升服务部署的稳定性和避免频繁的模式切换,需要有效捕获长期序列依赖以实现精准的流量预测。相比经典的循环神经网络(Recurrent Neural Network, RNN)和卷积神经网络(Convolutional Neural Network, CNN),新兴的 Transformer^[35]在捕捉长期依赖和特征间动态关联性上展现出了突出能力,其能够显式建模不同时间步特征的相关性,从而更准确地理解用户请求服务的全局时空模式。此外,得益于自注意力机制,Transformer 无需调整模型结构即可处理不同时间尺度的输入数据,适用于推理请求流量预测问题。同时,在服务部署过程中需要考虑服务部署的公平性,以降低因某些服务未部署任何配置导致任务转发至云端产生的高时延。具体而言,由于每种模型具备不同的资源需求、内存成本和性能特点,因此在服务部署时需要充分考虑每个节点的能力和负载情况,以确保任务能够被高效分配,在满足内存限制的同时提升服务质量。通过合理的负载均衡和公平性机制,可减少不必要的任务转发,从而有效降低传输时延,优化整体系统性能。此外,每种模型具备不同的潜在价值和内存成本,需要通过高效的服务部署,在满足内存限制的前提下提升系统收益。

为了解决上述问题,本文提出了一种基于流量预测的主动式服务部署方法,其关键步骤如算法 2 所示。首先,初始化候选服务部署方案 p^{best} 及其潜在价值 v^{best} , 并构造编码器输入 X^{en} 和解码器输入 X^{de} (第 1~2 行), 其中 X^{his} 表示历史推理请求流量, X^{input} 表示当前窗口收集到的流量序列, X^0 表示目标序列的时间采样(即,需要预测的流量序列)。经典的自注意力机制在构造编码器时需要计算每个位置与其他所有位置的注意力权重,会导致较高的计算复杂度。为了缓解该问题,受文献[36]启发,本文在构造编码器时引入了概率稀疏自注意力并在层之间使用自注意力蒸馏以降低计算开销。具体而言,从 j 层向第 $j+1$ 层的特征提取定义为

$$X_{t+1}^i = \text{MaxPool}(\text{ELU}(\text{Convld}([\mathbf{X}_t^i]_{\text{AB}}))) \quad (12)$$

其中, $\text{Convld}(\cdot)$ 表示时间序列上的一维卷积, ELU 表示激活函数, $\text{MaxPool}(\cdot)$ 表示最大池化操作, $[\cdot]_{\text{AB}}$ 表示稀疏自注意力。

通过在 Convld 后增加最大池化层可使输入下采样减少到其原始大小的一半,从而将训练过程的计算复杂度从 $O(L^2)$ 降低至 $O(L \log L)$, 其中 L 表

示 X_{input}^t 的长度(第 3 行)。接着,将编码器的输出和解码器的输入送入解码器(第 4 行),其包含一层多头稀疏概率自注意力机制和一层多头注意力机制。随后,将编码器输出送入多层感知机(Multi-layer Perceptron, MLP)以预测所需的流量序列(第 5 行)。接着,根据流量预测结果,迭代搜索所有可行的服务部署方案(第 6~14 行)。在搜索过程中,首先排除内存消耗超过阈值 M 的方案,因为其超过了 P1 中 C2 的限制;进一步,排除内存消耗低于阈值 σM 的方案,使得所有候选的部署方案的内存消耗位于 $[\sigma M, M]$, 其中 σ 为剪枝系数,用于控制搜索效率与部署方案开销之间的平衡(第 7~9 行)。如果一个部署方案剩余很多内存,那么通过部署更多模型可进一步增加其价值。因此,最优服务部署方案往往趋向于选择一组内存之和接近但不超出内存上限的模型。

算法 2. 基于流量预测的主动式服务部署

输入:系统历史推理流量 X^{input}

输出:最佳服务部署方案 p^{best}

1. 初始化候选最佳服务部署方案及其潜在价值:
 $p^{\text{best}} = 0, v^{\text{best}} = 0$;
2. 构造编码器和解码器的输入:
 $X^{\text{en}} \leftarrow X^{\text{his}}, X^{\text{de}} \leftarrow \text{Concat}(X^{\text{input}}, X^0)$;
3. 通过稀疏自注意力机制处理编码器输入:
 $H_1 = \text{Encoder}(X^{\text{en}})$;
4. 将编码器的输出与解码器的输入送入解码器:
 $H_2 = \text{Decoder}(H_1, X^{\text{de}})$;
5. 流量预测: $Z^p = \text{MLP}(H_2)$;
6. FOR $p = 1, 2, \dots, P$ DO
7. IF $\text{Mem}(p) > M$ OR $\text{Mem}(p) < \sigma M$ DO
8. Continue;
9. END IF
10. 使用公式(14)计算部署方案 p 的潜在价值 V ;
11. IF $V > v^{\text{best}}$ DO
12. $p^{\text{best}} = p$;
13. END IF
14. END FOR

对于候选的服务部署方案,本文量化了部署每种模型所具有的潜在价值,其定义为

$$V_{j,k} = Z_j(\delta \overline{\text{Acc}} - \overline{\text{Delay}}) \quad (13)$$

其中, $\overline{\text{Acc}}$ 和 $\overline{\text{Delay}}$ 分别表示模型在测试集中的平均精度和平均延迟, δ 表示权重系数。

接着,为了避免因不均衡的流量分布可能导致的不公平服务部署,本文基于任务传输至云端的时延设计惩罚项。因此,一个服务部署方案的潜在价

值定义为

$$V = \sum_{k=1}^K \left(\sum_{j=1}^J V_{j,k} - I \left(\sum_{j=1}^J x_{j,k} = 0 \right) Z_j T^{cloud} \right) \quad (14)$$

其中, $I(\cdot)$ 表示指示函数。当服务 j 没有部署模型时, $I(\cdot) = 1$; 否则, $I(\cdot) = 0$ 。

最后, 基于每个模型所需消耗的内存和潜在价值进行迭代搜索, 并输出潜在价值最高的最佳服务部署方案(第 10~13 行)。

4.3 基于引导式 DRL 的模型选择与资源分配

针对服务部署方案, 需要设计一种面向多样化模型配置的动态选择方法, 结合实时资源分配策略, 以协同优化不同推理请求的精度和延迟需求。作为一种新兴的智能决策算法, DRL 可作为解决模型选择与资源分配问题的一种潜在可行的方案。然而, 经典的 DRL 在面对新环境时, 通常需要从头开始探索和逐步优化策略。这不仅需要消耗大量训练时间和系统资源, 还可能出现 DRL 智能体难以收敛的情况, 尤其是在面对大规模状态和动作空间时。随着边缘环境下服务类型与模型配置的多样化, DRL 智能体需要拟合的策略复杂度呈指数级增长, 大大增加了合理模型选择与资源分配策略的拟合难度。为了解决该难题, 本文引入了 DIL, 通过学习专家示范行为来训练 DRL 智能体, 使其策略接近专家策略, 避免了可能需要的大量试错性探索。作为 DIL 中一种被广泛应用的监督学习方式, 行为克隆(Behavior Cloning, BC)可直接利用专家知识帮助 DRL 智能体学习状态和动作之间的映射关系。相比于从零开始与环境进行交互, BC 的引入能够帮助 DRL 智能体快速学习到指定领域的专家知识, 从而加速其收敛速度, 并提升 DRL 在资源受限边缘环境中的实用性。

在 DRL 智能体与环境的交互过程中, 相应的状态空间、动作空间和奖励函数定义如下:

(1) 状态空间。包含任务的服务类型、数据大小、可选模型等属性。同时, 为了更准确反映任务的推理密度与复杂度, 将像素量与图像熵加入状态空间。在视觉任务中, 像素量 PC_i 为图像长宽像素的乘积, 图像熵则是衡量图像信息量和不确定性的指标, 更高的图像熵表明图像含有更丰富的信息。根据香农熵^[37], 图像熵定义为

$$H = - \sum_{h=0}^{255} p_h \log_2 p_h \quad (15)$$

其中, p_h 表示第 h 个灰度级像素出现的概率。

因此, 在时隙 t 的状态空间定义为

$$s_t = \{d_i, s_i, \rho_i, PC_i, H_i\} \quad (16)$$

(2) 动作空间。DRL 智能体的目标是根据当前状态空间为每个推理请求选择合适的模型配置和带宽分配以取得推理精度与延迟之间的良好平衡。因此, 在时隙 t 的动作空间定义为

$$a_t = \{y_{i,k}, b_i\} \quad (17)$$

(3) 奖励函数。首先, 奖励函数需考虑最大化系统累积收益(根据公式(10)计算得到)。其次, 若 DRL 智能体所选择的模型未被部署, 则将其任务调度至精度最低的可用模型, 并反馈一个惩罚项 r^{penalty} 。此外, 为了避免先到达的任务抢占剩余资源, 将剩余可用带宽考虑进奖励函数, 记为 r^{res} 。因此, 在时隙 t 的奖励函数定义为

$$r_t = v_i^t + r^{\text{res}} - r^{\text{penalty}} \quad (18)$$

其中,

$$r^{\text{penalty}} = \begin{cases} \Phi, & \text{if } y_{i,k} = 1 \text{ 且 } x_{j,k} = 0 \\ 0, & \text{else} \end{cases} \quad (19)$$

其中, Φ 表示模型未部署导致任务精度下降产生的惩罚系数。

基于上述定义, 本文提出了一种基于引导式 DRL 的模型选择与资源分配方法, 其关键步骤如算法 3 所示。对于每个接收到的推理任务, 根据公式(16)为其构建动作状态输入 s_t 。本文考虑为部分任务构建专家动作, 但与具有明确最优决策规则的任务(如棋类游戏)不同, 模型选择场景的奖励模式具有多维耦合性, 难以直接推导出最优解。为此, 本文设计了一种基于启发式搜索的专家知识获取方法。通过离线模拟环境交互, 动态搜索专家模型选择策略。具体而言, 对于每个任务, 首先构建动作候选集合 $A = \{a_1, a_2, \dots, a_k\}$ (第 7 行), 其中每个动作包括可选择的模型类型、计算节点分配以及资源分配。接着, 综合考虑任务的延迟约束、资源消耗和推理精度, 利用奖励函数 r_t 对候选动作进行评估并选择得分最高的若干候选动作作为专家动作(第 8 行)。最后, 将其以<状态-动作-奖励>三元组形式存入专家经验池, 用于后续指导智能体的训练过程(第 9 行)。随后, 将状态 s_t 输入至 Actor 网络以获取模型选择与资源分配动作(第 10 行)。接收到动作 a_t 后, 环境将执行该动作并反馈奖励与下一个状态, 其将被存入本地经验池用于更新模型参数(第 11~12 行)。由于资源分配决策会产生跨时隙的长期影响, 本文引入折扣因子 γ 以计算当前任务的奖励(第 13 行), 其定义为

算法 3. 基于引导式 DRL 的模型选择与资源分配输入:模型部署状态 $x_{j,k}$ 、任务属性输出:模型选择决策 $y_{i,k}$ 、带宽分配决策 b_i

1. 初始化 Actor 网络、Critic 网络、专家经验池 expertBuffer 和本地经验池 localBuffer ;
2. FOR $epoch = 1, 2, \dots, E$ DO
3. 初始化状态: $s_0 = \text{env.init}()$;
4. FOR $t = 1, 2, \dots, T$ DO
5. FOR $n = 1, 2, \dots, N$ DO
6. IF $n < N^{\text{expert}}$ DO
7. 构建动作候选集合: $A = \{a_1, a_2, \dots, a_k\}$;
8. 选择奖励得分最高的动作作为专家动作:
 $a_{esp}^i = \underset{a_k \in A}{\operatorname{argmax}} r_t(a_k, s_t)$;
9. 存入专家经验池:
 $\text{expertBuffer.Add}(s_t, a_{esp}^i)$;
10. 选择动作: $a_t \leftarrow \pi_{\theta_t}(s_t)$;
11. 执行动作并反馈: $s_{t+1}, r_t \leftarrow \text{env.step}(s_t, a_t)$;
12. 存入本地经验池:
 $\text{localBuffer.Add}(s_t, a_t, r_t, \gamma_t, s_{t+1})$;
13. 计算折扣奖励: $R_t = \sum_{n=1}^N \gamma^n r_{t+n+1}$;
14. 计算优势函数: $\hat{A}(s_t, a_t) = \sum_{k=0}^{\infty} (\gamma \lambda)^k \delta_{t+k}$;
15. 最小化 L^{total} 以优化网络;
16. END IF
17. END FOR
18. END FOR
19. END FOR

$$R_t = \sum_{n=1}^N \gamma^n r_{t+n+1} \quad (20)$$

为了减少环境噪声对梯度估计的影响,本文引入了广义优势估计(Generalized Advantage Estimation, GAE)^[38]作为网络更新的目标(第 14 行),其定义为

$$\hat{A}(s_t, a_t) = \sum_{k=0}^{\infty} (\gamma \lambda)^k \delta_{t+k} \quad (21)$$

$$\delta_t = R_t + \gamma V_{\phi}(s_{t+1}) - V_{\phi}(s_t) \quad (22)$$

其中, λ 表示优势函数的折扣率, δ_t 表示时序差分(Temporal-Difference, TD), V 是状态-值函数。

进一步地,为了提升优势值在策略更新过程中的稳定性,本文通过裁剪目标函数来控制 Actor 网络每次更新的范围,其定义为

$$L^{\text{actor}}(\theta) = E[\min(r(\theta)\hat{A}(s_t, a_t), \text{clip}(r(\theta), 1 - \epsilon, 1 + \epsilon)\hat{A}(s_t, a_t))] \quad (23)$$

其中, $r(\theta)$ 表示新旧策略之间的采样率。通过调整 $r(\theta)$, 可纠正策略更新引起的采样偏差,其定义为

$$r(\theta) = \frac{\pi_{\theta}(a_t | s_t)}{\pi_{\text{old}}(a_t | s_t)} \quad (24)$$

接着,通过最小化 $L^{\text{critic}}(\phi)$ 来优化 Critic 网络,其定义为

$$L^{\text{critic}}(\phi) = E_{\mathcal{M}}(R_t + \beta V_{\pi_{\theta}}^{\pi_{\theta}}(s_{t+1}) - V_{\pi_{\theta}}^{\pi_{\theta}}(s_t))^2 \quad (25)$$

对于动作空间,其行为克隆损失函数定义为

$$L^{\text{BC}} = -\mathbb{E}_{s_t, a_t^{\text{expert}}} [\log \pi(a_t^{\text{expert}} | s_t)] \quad (26)$$

其中, a_t^{expert} 表示专家在状态 s_t 下的动作, $\pi(a_t^{\text{expert}} | s_t)$ 表示在 s_t 下输出 a_t^{expert} 的概率。

通过结合行为克隆损失和 DRL 中 Actor 和 Critic 网络的损失(第 15 行),总损失函数定义为

$$L^{\text{total}} = L^{\text{actor}}(\theta) + L^{\text{critic}}(\phi) + \lambda L^{\text{BC}}(\theta) \quad (27)$$

其中, λ 表示用于控制 BC 与 DRL 相对重要性的权重因子。通过控制 λ 的变化,可在训练初期显著降低策略的探索成本以加速模型收敛。在训练后期,可进一步优化模型性能使其泛化至其他状态空间。

5 性能评估

本节首先介绍了实验设置。接着,通过大量实验全面评估了所提出 SDMS 方法的有效性和优越性。

5.1 实验设置

实验平台。在本实验中,所有实验均在配备 i5-12400F CPU(4.4 GHz)和 NVIDIA GeForce RTX 4070TiS 显卡的工作站上进行。本文使用 Python 构建所提出的系统,并利用 Pytorch 框架实现所提出的 SDMS 方法以及其他对比方法。

推理服务与模型。本文采用了 ImageNet 和 COCO 数据集来模拟系统中推理任务的属性。具体而言,使用 ResNet 在 ImageNet 数据集上执行图像分类任务,采用 Top1 准确率来评估模型精度;使用 YOLOv8 在 COCO 数据集上执行目标检测任务,采用平均精度均值(mean Average Precision, mAP)@0.50 : 0.95 来评估模型精度;使用 DeepLabv3 在 COCO 数据集上执行语义分割任务,采用平均交并比(mean Intersection over Union, mIoU)来评估模型精度。在实验中,随机抽取数据集中的图像 ID 并读取相应任务属性(如数据大小、像素数量和图像熵等)来评估不同模型在推理精度和推理延迟方面的表现。同时,本文评估了多个典型的 DNN 模型结构在不同任务下的表现。对于每种任务,使用了不同的模型进行推理,每种服务类型对应不同的模型

结构。详细信息如表 2 所示。

表 2 不同服务类型所使用模型的详细信息

服务类型	模型结构	内存消耗 (MB)	Top1/mAP /mIoU(%)	平均延迟 (ms)
图像分类	ResNet18	44.69	35.03	2.02
	ResNet34	83.20	45.44	2.15
	ResNet50	97.73	45.21	2.34
	ResNet101	170.65	49.19	3.35
	ResNet152	230.28	52.65	4.41
目标检测	YOLOv8N	12.25	48.45	7.65
	YOLOv8S	42.82	55.47	8.11
	YOLOv8M	100.05	59.96	8.59
	YOLOv8L	168.13	61.59	10.99
	YOLOv8X	262.40	62.22	12.99
语义分割	deeplabv3_ mobilenet_ v3_large	42.26	44.76	18.20
	deeplabv3_ resnet50	762.74	47.45	31.00
	deeplabv3_ resnet101	235.41	48.22	35.70

数据集与系统参数。本文采用米兰真实蜂窝流量数据集^[39]来模拟用户服务请求,其包含三种类型的服务(即消息、呼叫和互联网)。该数据集记录了2个月期间的流量,采样频率为10 min。本文选择了三个区域(即ID=4259、4456和5060)的流量记录,作为三种服务在时隙内的用户请求数。用户上传功率分布在 $[80, 120]$ mW^[20]。信道增益与高斯白噪声功率分别为 10^{-4} W和 10^{-8} W^[40]。对于不同任务,其优先级为分布在 $[1, 3]$ 之间的整数,其最大容忍延迟分别为0.5 s、0.7 s、1.2 s,最低容忍精度分别为50%、45%、45%。时隙数量为100,训练轮次为300,服务部署间隔为10,云传输延迟为0.4 s,边缘服务器的总可用内存为500 MB。在最优策略搜索中,精度-延迟权重系数为0.01,剪枝系数为0.5。

对比方法。为了验证所提出SDMS方法的优越性,本文将其与以下基准方法进行比较:

(1)PredRNN^[41]:采用基于RNN的服务流量预测,并利用门控机制来捕获流量序列中的依赖关系。模型选择与SDMS方法一致。

(2)PPO-TO^[42]:采用近端策略优化(Proximal Policy Optimization, PPO)进行模型选择,为每个服务优先部署延迟最低的模型和精度最高的模型。

(3)DQNM^[21]:采用了深度Q网络(Deep Q-Network, DQN)进行模型选择,为每个服务优先部署延迟最低的模型和精度最高的模型。

(4)Greedy-Delay:采用贪心算法为每个任务选择延迟最低的模型。

(5)Greedy-Acc:采用贪心算法为每个任务选择精度最高的模型。

5.2 实验结果与分析

首先,本文评估了不同YOLOv8模型在COCO数据集样本上的推理延迟和精度。如图3所示,相比于YOLOv8S模型,YOLOv8L模型在大部分样本上都取得了更高的推理延迟。这是因为YOLOv8L模型采用了更复杂的网络结构,因此需要消耗更多的计算资源。然而,YOLOv8L模型的推理精度却并不总是高于YOLOv8S模型,甚至在某些样本上的推理精度反而小于YOLOv8S模型。这可能源于复杂模型在特定样本上的过拟合,或样本本身的特征更符合小模型的归纳偏置,从而导致其在小模型上的表现更优。值得注意的是,无论是推理延迟还是推理精度,YOLOv8L和YOLOv8S模型之间的差异在不同样本上都在不断发生变化,表现出强动态性。这促使本文去关注如何通过实时优化服务部署和模型选择策略,以实现推理延迟和精度之间更好的平衡。

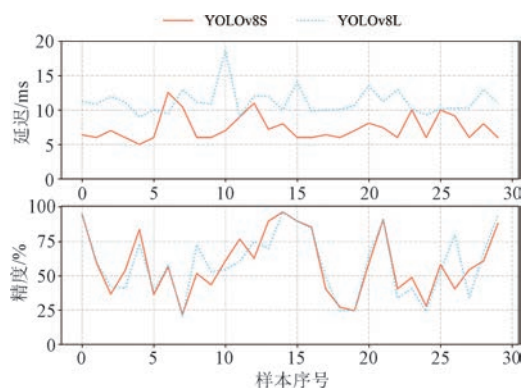


图 3 不同模型在COCO数据集上的推理延迟和精度

接着,本文对比了所提出SDMS与其他方法的收敛性。如图4所示,Greedy-Delay和Greedy-Acc方法采用静态决策,不存在学习的过程。这两种方法的性能不如其他基于动态决策的方法。这是因为它们进行决策的方式单一,未充分考虑不同模型在不同样本上的动态表现,无法有效平衡推理精度和延迟。相比于DQN方法,PPO方法获得了更高的奖励,也展现出了更平稳的收敛曲线。这是因为PPO方法引入了一种剪枝操作,确保新旧策略之间的差异不会过大,解决了DQN方法中Q值高估的问题。通过引入流量预测和剪枝搜索,SDMS和PredRNN方法的性能进一步得到了显著提升。这表明准确的流量预测有助于在动态变化的边缘环境中改善服务和模型部署的合理性,以在有限内存预

算下提升任务成功率。相较于 PredRNN 方法使用 RNN 进行时序预测,SDMS 方法通过引入 Informer 可更加高效地捕获长期时序依赖,取得了比 PredRNN 更好的性能。值得注意的是,相较于 PPO 和 DQN 方法,SDMS 方法通过结合 DIL,取得了更快速且更稳定的收敛。SDMS 方法在约 140 轮训练后达到收敛,最终奖励值稳定在 1100 左右。PPO 方法则在约 270 轮训练后收敛,最终奖励值停留在 890 左右。结果验证了 SDMS 方法的优越性。

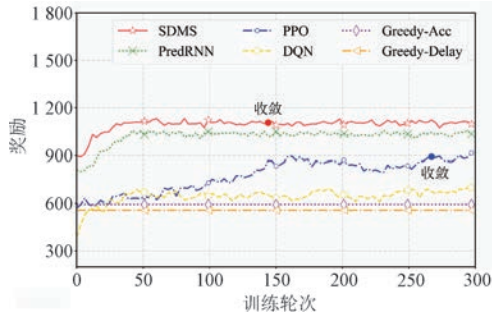


图4 不同方法的收敛性对比

随后,本文对比了所提出 SDMS 与其他方法在不同服务与模型下的精度和延迟满足率。如图 5 所示,对于图像分类和实例分割服务,SDMS 方法均取得了最高的精度满足率,其在目标检测服务上略低于 Greedy-Acc 方法。这是因为 Greedy-Acc 方法总是选择精度最高的模型,使其在不同服务上均表现出了较高的精度满足率,但也导致了较长的推理延迟。相比之下,Greedy-Delay 方法在不同服务上也表现出了较高的延迟满足率,但却暴露出了最低的精度满足率。相较于其他方法,SDMS 方法通过合理的模型部署与配置选择策略,在精度和延迟满足率这两个指标上均取得了优越的性能。结果验证了 SDMS 方法平衡模型推理精度和延迟的有效性。

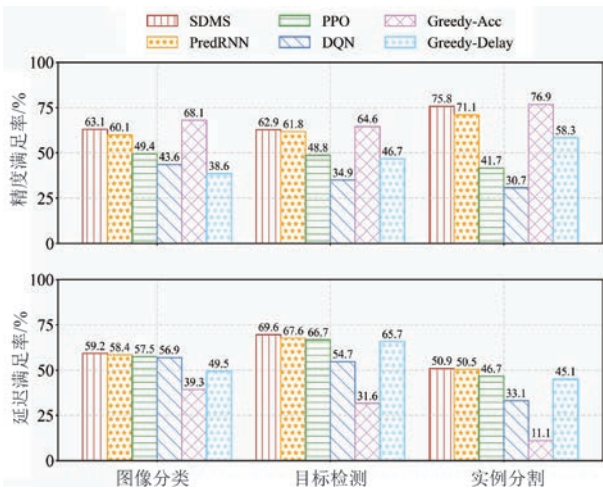


图5 不同方法在不同服务上的精度和延迟满足率对比

为评估 SDMS 方法中各核心组件对整体性能的贡献,本文通过消融实验分别去除了以下核心组件,进而观察性能的变化:

(1)TP(Traffic Prediction):未使用流量预测,将历史时隙的流量作为未来时隙流量的估计。

(2)PS(Pruning Search):未使用剪枝搜索,使用穷尽搜索(Exhaustive Search, ES)以获得最佳服务部署方案。

(3)DRL(Deep Reinforcement Learning):未使用深度强化学习,亦未引入专家策略以引导模型训练过程。

如图 6 所示,当未使用 TP 组件时,SDMS 方法的精度和延迟满足率受到了显著影响。这是因为若缺乏准确的流量预测,SDMS 方法只能依赖历史流量来推测未来流量,巨大的偏差导致了在面对负载波动时的性能下降。当使用 ES 替代 SDMS 中的 PS 组件时,精度和延迟满足率分别平均提升了约 0.5% 和 1.8%。具体而言,精度满足率在图像分类、目标检测和实例分割任务上分别提升了 0.2%、0.5% 和 0.7%,而延迟满足率分别提升了 0.6%、3.2% 和 1.5%。这是因为 ES 通过遍历所有可能的方案以获取最优解,但这严重增加了寻找最优服务部署方案的开销。相比之下,本文所设计的剪枝搜索方案以极低的开销取得了与穷举搜索相当的性能表现,更加适用于时延敏感的推理系统。当未使用 BC 组件时,出现了最显著的精度满足率的下降。通过引入专家策略以引导训练,SDMS 可更加快速地学习到专家行为策略。当移除 DRL 组件,SDMS 方法难以实现高效的训练。结果表明,这三个核心组件从不同方面保证了 SDMS 方法的性能,它们的协同工作确保了 SDMS 方法在动态复杂环境中的高性能和强适应性。

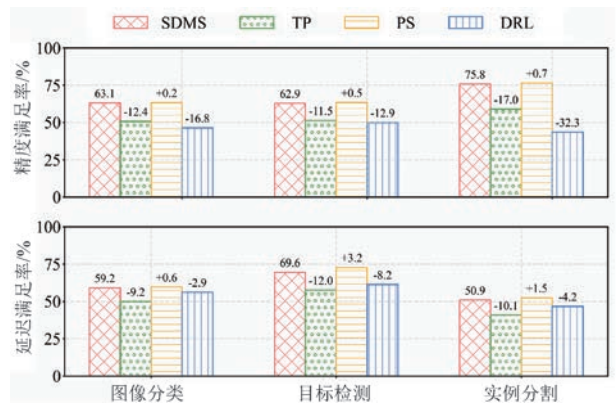


图6 对 SDMS 方法中各核心组件的消融实验

进一步地,本文分析了 SDMS 方法中各核心组件及 ES 所需的执行延迟,以验证 SDMS 方法中各核心组件的高效性。如表 3 所示,ES 的执行延迟达 492.51 ms,占系统总延迟的 43.55%。这是因为 ES 需要遍历所有可能的服务部署方案,导致了极大的计算量和过度的延迟。相比之下,所提出的 PS 组件通过有效过滤大部分价值较低的方案,能够将延迟降低至 12.82 ms,占系统总延迟的 2.01%,仅为 ES 的 2.6%。同时,TP 和 DRL 所需的额外延迟分别仅为 9.65 ms 和 1.87 ms,分别占系统总延迟的 1.51% 和 0.29%。上述实验结果表明,SDMS 方法中的核心组件在系统总延迟中的占比仅为 3.81%,验证了 SDMS 方法中所设计组件的高效性,未引入明显的额外开销,具备良好的可部署性。

表 3 SDMS 方法中各核心组件及 ES 的执行延迟

组件	TP	PS	DRL	ES
执行延迟(ms)	9.65	12.82	1.87	492.51
系统总延迟中占比	1.51%	2.01%	0.29%	43.55%

此外,本文分析了系统在动态流量场景下资源利用率的变化趋势,以评估所提出系统在动态环境下的稳定性与可扩展性。如图 7 所示,随着用户发起请求流量的增加,系统整体的资源利用率呈现上升趋势。类似地,系统整体的资源利用率也随着用户请求流量的下降而下降。结果表明所提出系统能够实时感知负载变化并及时调整服务部署。通过这种自适应的调整机制,所提出系统能够在满足用户需求的同时避免不必要的资源浪费。

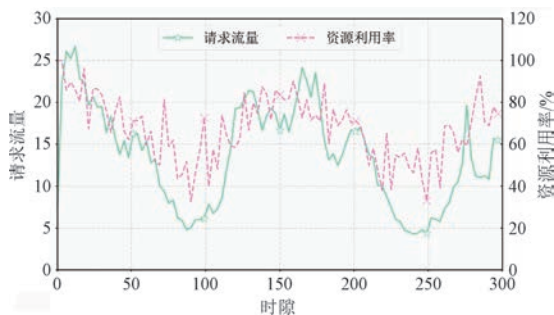


图 7 资源利用率随流量波动的变化趋势

接着,本文对比了所提出 SDMS 和其他方法在不同流量规模下的服务收益。如图 8 所示,不同方法的服务收益随着流量规模的扩大而显著增加。当流量倍数为 $0.5\times$ 时,由于任务数量较少,SDMS 和其他方法之间服务收益的差距较小。随着流量倍数的增加,不同方法服务收益之间的差距也逐渐扩大。具体而言,SDMS 和 PredRNN 方法引入了流量预测技术,其服务收益均高于其他方法。这是因为通

过精确的流量预测可辅助制定更为合理的服务部署方案,从而实现更为均衡的模型选择,以提升系统服务收益。PPO 和 DQN 方法均采用静态的服务部署策略,难以进行合理的模型选择,因此其服务收益低于 SDMS 方法。Greedy-Delay 和 Greedy-Acc 方法则采用了单一的模型选择策略,无法感知不同请求流量对服务部署状态的影响,因此其与 SDMS 方法服务收益之间的差距随着流量规模的扩大而逐渐增大。

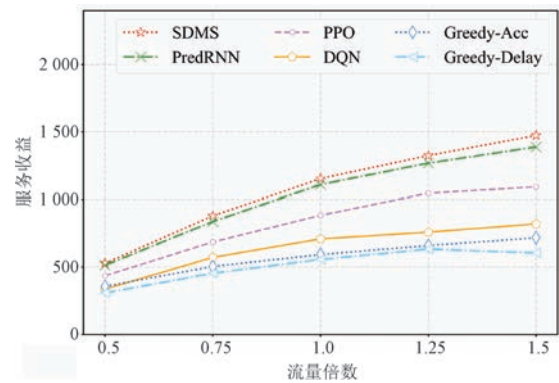


图 8 不同方法在不同流量规模下的服务收益对比

最后,本文对权重系数 δ 和剪枝系数 σ 进行了敏感性分析。 δ 权重系数用于控制服务部署过程中模型精度与推理开销之间的平衡。由于模型精度大多分布在 $[0.4, 0.6]$, 而延迟处于 10 ms 量级, 因此 δ 在平衡系统精度与开销的同时需要对这二者进行量纲对齐。如图 9(a) 所示, 随着 δ 的增加, PPO 方法的服务收益未发生变化, 而 SDMS 和 PredRNN 方法的服务收益则是先增加后减少。这是因为 PPO 方法未考虑动态的服务部署策略, 不合理的 δ 则会由于系统对精度或资源的需求偏差而导致其收益下降。当 δ 为 $1e-2$ 时, SDMS 方法的收益达到最高, 表明此时系统在推理精度与资源开销之间取得了最佳平衡。此外, 本文分析了剪枝系数 σ 对搜索开销与服务收益的影响, 其用于控制搜索空间的剪枝比例。如图 9(b) 所示, 随着 σ 的增加, 剪枝搜索空间逐渐缩小, 搜索开销明显下降。同时, 由于候选方案数量减少, 导致搜索过程中的模型部署质量下降, 系统收益也随之降低。因此, 综合考虑搜索效率与部署质量, 当 σ 为 0.5 时, SDMS 方法能够在服务收益与搜索开销之间取得较优平衡。

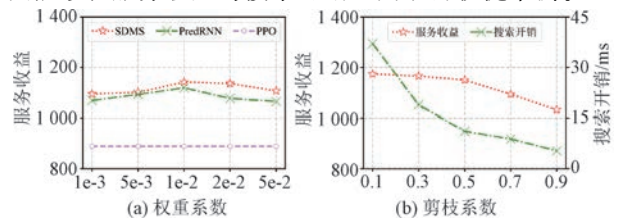


图 9 参数敏感性分析

6 结 论

本文首先将边缘环境下精度感知的服务部署与模型选择形式化定义为一个服务收益最大化问题。接着,通过结合概率稀疏自注意力机制、剪枝搜索、DRL 和 DIL 等先进技术,提出了一种新颖的 SDMS 方法,充分考虑推理精度对服务体验的重要影响,以有效应对动态变化的系统状态与用户需求。基于真实用户通信流量数据集与 DNN 推理数据集,本文通过大量实验验证了所提出 SDMS 方法在提高服务收益、延迟和精度满足率等方面的有效性。结果表明,与其他 5 种基准方法相比,SDMS 方法在不同服务和模型上均表现出了更加优越的延迟和精度性能。与先进的 PredRNN 和 PPO 等方法相比,SDMS 方法也展现出了更好的稳定性与更快速的收敛速度。同时,SDMS 方法在不同推理流量规模下的泛化性能也得到了充分验证。在未来工作中,本文将进一步探索所提出的方法在多节点协同场景下的可扩展性,并考虑更细粒度的任务状态空间以实现推理成本变化的精准捕获。同时,考虑将所提出的方法进一步扩展至自然语言处理等更为多样化的 DNN 推理应用场景中。

参 考 文 献

- [1] Zhang X, Li Y, Wang J, et al. MESON: A mobility-aware dependent task offloading scheme for urban vehicular edge computing. *IEEE Transactions on Mobile Computing*, 2024, 23(5):4259-4272
- [2] Chen Z, Cao M T, et al. Graph-to-sequence deep reinforcement learning based complex task deployment strategy in MEC. *Journal on Communications*, 2024, 45(3):244-257 (in Chinese)
(陈卓,操民涛,周致圆,等. 移动边缘计算中基于图到序列深度强化学习的复杂任务部署策略. *通信学报*, 2024, 45(3):244-257)
- [3] Zhang Y L, Liang Y Z, Yin M J, et al. A survey on computation offloading schemes in mobile edge computing. *Chinese Journal of Computers*, 2021, 44(12):2406-2430 (in Chinese)
(张依琳,梁玉珠,尹沐君,等. 移动边缘计算中计算卸载方案研究综述. *计算机学报*, 2021, 44(12):2406-2430)
- [4] Shang K, He W, Zhang S. Review on security defense technology research in edge computing environment. *Chinese Journal of Electronics*, 2024, 33(1):1-18
- [5] Kuang Z F, Chen Q L, Li L F, et al. A deep reinforcement learning-based task offloading scheduling and resource allocation algorithm for multi-user edge computing. *Chinese Journal of Computers*, 2022, 45(4):812-824 (in Chinese)
(邝祝芳,陈清林,李林峰,等. 基于深度强化学习的多用户边缘计算任务卸载调度与资源分配算法. *计算机学报*, 2022, 45(4):812-824)
- [6] Zhou H, Wang Z, Zheng H, et al. Cost-minimization-oriented computation offloading and service caching in mobile cloud-edge computing: An A3C-based approach. *IEEE Transactions on Network Science and Engineering*, 2023, 10(3):1326-1338
- [7] Zhang G, Zhang S, Zhang W, et al. Joint service caching, computation offloading and resource allocation in mobile edge computing systems. *IEEE Transactions on Wireless Communications*, 2021, 20(8):5288-5300
- [8] Zhou R, Wu X, Tan H, et al. Two time-scale joint service caching and task offloading for UAV-assisted mobile edge computing//*Proceedings of the IEEE INFOCOM Conference on Computer Communications*. London, UK, 2022:1189-1198
- [9] Bi S, Huang L, Zhang Y J A. Joint optimization of service caching placement and computation offloading in mobile edge computing systems. *IEEE Transactions on Wireless Communications*, 2020, 19(7):4947-4963
- [10] Zhao M, Zhang R, He Z, et al. Joint optimization of trajectory, offloading, caching, and migration for UAV-assisted MEC. *IEEE Transactions on Mobile Computing*, 2024, 24(3):1981-1998
- [11] Ogden S S, Guo T. Characterizing the deep neural networks inference performance of mobile applications. *arXiv* 2019. *arXiv preprint arXiv:1909.04783*, 1909
- [12] Bianco S, Cadene R, Celona L, et al. Benchmark analysis of representative deep neural network architectures. *IEEE Access*, 2018, 6:64270-64277
- [13] Liu X, Yu J, Wang J, et al. Resource allocation with edge computing in IoT networks via machine learning. *IEEE Internet of Things Journal*, 2020, 7(4):3415-3426
- [14] Huang Y, Feng B, Cao Y, et al. Collaborative on-demand dynamic deployment via deep reinforcement learning for iov service in multi edge clouds. *Journal of Cloud Computing*, 2023, 12(1):1-18
- [15] Wang L, Ren X, Zhao C, et al. MPDM: A multi-paradigm deployment model for large-scale edge-cloud intelligence. *IEEE Internet of Things Journal*, 2023, 10(10):8773-8785
- [16] Fan W, et al. DNN Deployment, Task offloading, and resource allocation for joint task inference in iioT. *IEEE Transactions on Industrial Informatics*, 2023, 19(2):1634-1646
- [17] Zhang J, Liu Y, Li Z, et al. Forecast-assisted service function chain dynamic deployment for SDN/NFV-enabled cloud management systems. *IEEE Systems Journal*, 2023, 17(3):4371-4382

- [18] Li Y, Han Z, Zhang Q, Li Q, et al. Automating cloud deployment for deep learning inference of real-time online services. *IEEE Conference on Computer Communications*, 2020; 1668-1677
- [19] Li G, et al. An adaptive user service deployment strategy for mobile edge computing. *China Communications*, 2022, 19(10):238-249
- [20] Fan W. Hybrid deep reinforcement learning-based task offloading for D2D-assisted cloud-edge-device collaborative networks. *IEEE Transactions on Mobile Computing*, 2024, 23(12):13455-13471
- [21] Zhou H, Wu T, Zhang H, et al. Incentive-driven deep reinforcement learning for content caching and D2D offloading. *IEEE Journal on Selected Areas in Communications*, 2021, 39(8):2445-2460
- [22] Li C, Zhang Q, Huang C, et al. Optimal service selection and placement based on popularity and server load in multi-access edge computing. *Journal of Network and Systems Management*, 2023, 31(1):15-32
- [23] Mao J, Chen X, Nixon K W, et al. MoDNN: Local distributed mobile computing system for deep neural network//*Proceedings of the Design, Automation & Test in Europe Conference & Exhibition*. Lausanne, Switzerland, 2017:1396-1401
- [24] Zeng L, Chen X, Zhou Z, et al. CoEdge: Cooperative DNN inference with adaptive workload partitioning over heterogeneous edge devices. *IEEE/ACM Transactions on Networking*, 2021, 29(2):595-608
- [25] Cui Y, Li H, Zhang D, et al. Multiagent reinforcement learning-based cooperative multitype task offloading strategy for internet of vehicles in B5G/6G network. *IEEE Internet of Things Journal*, 2023, 10(14):12248-12260
- [26] Kang W. QoS-aware inference acceleration using adaptive depth neural networks. *IEEE Access*, 2024, 12:49329-49340
- [27] Zhang Z, Zhao Y, Liu J, et al. Octopus: SLO-aware progressive inference serving via deep reinforcement learning in multi-tenant edge cluster//*Proceedings of the International Conference on Service-Oriented Computing*. Rome, Italy, 2023:1-18
- [28] Wu Y J, Lentz M, Zhuo D, et al. Serving and optimizing machine learning workflows on heterogeneous infrastructures. *Proceedings of the VLDB Endowment*, 2022, 16(3):406-419
- [29] Yao Z, Xia S, Li Y, et al. Cooperative task offloading and service caching for digital twin edge networks: A graph attention multi-agent reinforcement learning approach. *IEEE Journal on Selected Areas in Communications*, 2023, 41(11):3401-3413
- [30] Tang F, Hofner H, Kato N, et al. A deep reinforcement learning-based dynamic traffic offloading in space-air-ground integrated networks (SAGIN). *IEEE Journal on Selected Areas in Communications*, 2022, 40(1):276-289
- [31] Zhao K, Zhou Z, Jiao L, et al. Taming serverless cold start of cloud model inference with edge computing. *IEEE Transactions on Mobile Computing*, 2023, 23(8):8111-8128
- [32] Liu H, Liu S, Long S, et al. Joint optimization of model deployment for freshness-sensitive task assignment in edge intelligence//*Proceedings of the IEEE INFOCOM Conference on Computer Communications*. Vancouver, Canada, 2024: 1751-1760
- [33] Xu A, Hu Z, Li X, et al. CoMS: Collaborative DNN model selection for heterogeneous edge computing systems. *IEEE Transactions on Vehicular Technology*, 2024, 74(2):3172-3184
- [34] Xiang Z, Yan H, Xu Y, et al. Cost-effective service deployment and balanced traffic management on edge//*Proceedings of the IEEE International Conference on Web Services*. Chicago, USA, 2023:78-83
- [35] Wen Q, Zhou T, Zhang C, et al. Transformers in time series: A survey. *arXiv preprint arXiv:2202.07125*, 2022
- [36] Zhou H, Zhang S, Peng J, et al. Informer: Beyond efficient transformer for long sequence time-series forecasting//*Proceedings of the AAAI Conference on Artificial Intelligence*. Vancouver, Canada, 2021:11106-11115
- [37] Prokopenko M, Einav I. Information thermodynamics of near-equilibrium computation. *Physical Review E*, 2015, 91(6):062143
- [38] Schulman J, Moritz P, Levine S, et al. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015
- [39] Barlacchi G, De Nadai M, Larcher R, et al. A multi-source dataset of urban life in the city of milan and the province of trentino. *Scientific Data*, 2015, 2(1):1-15
- [40] Liu T, Fang L, Zhu Y, et al. A near-optimal approach for online task offloading and resource allocation in edge-cloud orchestrated computing. *IEEE Transactions on Mobile Computing*, 2022, 21(8):2687-2700
- [41] Wang Y, Wu H, Zhang J, et al. Predrnn: A recurrent neural network for spatiotemporal predictive learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022, 45(2):2208-2225
- [42] Xu D, Su X, Wang H, et al. Towards risk-averse edge computing with deep reinforcement learning. *IEEE Transactions on Mobile Computing*, 2023, 23(6):7030-7047



CHEN Zhe-Yi, Ph. D. , professor.

His research interests include cloud-edge collaborative computing and intelligent resource optimization.

CHI Meng-Li, M. S. candidate. Her

research interests include mobile edge computing, service deployment, and deep reinforcement learning.

ZHANG Jun-Jie, Ph. D. candidate. His research inter-

ests include mobile edge computing, model inference acceler-

ation, and resource optimization.

Lin Jin-Hui, M. S. candidate. His research interests include mobile edge computing, service deployment, and model selection.

MIN Ge-Yong, Ph. D. , professor. His research interests include computer networks, wireless communications, and parallel and distributed computing.

LI Jie, Ph. D. , professor, IEEE Fellow. His research interests include big data and artificial intelligence, blockchain, and edge computing.

Background

With the rapid development of edge computing and artificial intelligence technologies, Deep Neural Networks (DNNs) have been widely applied in edge environments to provide more intelligent service quality. However, when facing the contradiction between limited edge resources and diverse user demands, it has become a challenging problem to achieve efficient service deployment and model selection. For latency-aware service deployment, existing studies commonly target a single computation-intensive service, ignoring the diverse service types and model configurations, making them hard to fit in the dynamic demands of intelligent applications in real-world edge environments. Although some studies further consider diverse services and model types, they focus on general indicators such as inference latency and energy consumption, which ignore the key impact of DNN inference accuracy on service experience and cannot achieve a good balance between inference accuracy and latency. For accuracy-aware service deployment, most existing studies use heuristics or control theory, which are difficult to adapt to dynamic edge environments and diverse services. Although some studies introduce Deep Reinforcement Learning (DRL) for making decisions, they target small-scale states and action spaces. With the increase in service types and model scale, they face low exploration efficiency and convergence difficulty.

To address the above important challenges, this paper

proposes an inference accuracy-aware Service Deployment and Model Selection (SDMS) method in edge environments. First, the self-attention mechanism is used to predict the distribution of user request traffic in future time-slots, and then the service deployment plans are optimized by integrating a pruning search strategy. Next, a new guided Deep Reinforcement Learning (DRL) framework based on Deep Imitation Learning (DIL) is designed to rapidly approximate the optimal decisions of model selection and resource allocation by incorporating expert experience. Using real-world user communication traffic and DNN inference datasets, extensive experiments are conducted to validate the effectiveness of the proposed SDMS method. The results show that the SDMS method achieves better service profits, latency, and accuracy satisfaction rates than other benchmark methods across various scenarios. Therefore, this paper offers a scalable and practical solution for intelligent service deployment and model selection in edge systems.

This work was partly supported by the National Natural Science Foundation of China (Grant No. 62202103), the Natural Science Foundation of Fujian Province for Distinguished Young Scholars (Grant No. 2025J010020), the Central Funds Guiding the Local Science and Technology Development (Grant No. 2022L3004), and the Collaborative Innovation Platform Project of Fuzhou City (Grant No. 2023-P-002).