科技创新

基于大模型的城轨云算力资源平台设计研究

赵俊华，杜呈欣

（中国铁道科学研究院集团有限公司电子计算技术研究所，北京 100081）

摘要： 随着城市轨道交通系统智能化程度的提升，传统算力资源分配模式已难以应对多源异构数据处理、实时决策支持等新型需求。提出基于大语言模型（large language models，LLMs）的城轨云算力资源平台架构，构建了融合知识图谱的混合调度算法，通过动态资源分配优化策略提升了算力资源利用率。实验表明，该平台在处理突发客流预测、设备故障诊断等典型场景时，缩短了响应时间，提高了模型推理准确度，为智慧城轨建设提供了新型算力基础设施解决方案。

关键词： 城轨智能化；云算力资源平台；混合调度；城轨大模型

中图分类号： U121 **文献标志码：** A **文章编号：** 1671－1807(2026)02－0019－07

城市轨道交通系统作为现代都市交通主动脉，日均客流量超千万人次的特大型城市已普遍面临三大挑战[1-5]：多源设备数据融合困难（视频监控、传感器网络、票务系统等日均产生PB级数据）；传统专家系统难以适应动态场景（如突发客流、设备连锁故障等）[6-8]；专用计算设备重复建设导致资源浪费[各子系统独立部署图形处理器（graphics processing unit，GPU）服务器]。本文通过构建基于大模型的云算力资源平台，创新性地将模型即服务（model as a service，MaaS）理念引入城轨领域，实现跨模态数据融合分析（视频、文本、时序数据联合处理），动态场景自适应推理（基于强化学习的资源调度），异构算力虚拟化整合（CPU/GPU混合部署）。

1 平台架构设计

1.1 现状分析

城轨交通系统产生的数据量庞大且类型多样，包括但不限于乘客出行数据、列车运行数据、设备状态数据等。这些数据的实时处理和分析对于提高运营效率、保障乘客安全至关重要。然而，传统数据处理方式往往受限于算力瓶颈，无法实现高效的数据分析和决策支持。此外，随着人工智能技术的发展，城轨交通系统对算力的需求更加多样化，不仅需要处理大量结构化数据，还需要处理图像、视频等非结构化数据，这对算力资源的灵活性和扩展性提出了更高要求。既有研究主要集中在三个方向：轨道交通信息化系统；云计算资源调度算法（Kubernetes等通用方案[9-10]）；垂直领域大模型应用（交通大模型TransGPT[11-13]）。但现有方案的局限有：专用硬件部署模式导致资源闲置率高达65%；传统调度算法未考虑大模型推理的突发性特征；行业知识库与通用大模型存在语义鸿沟。本文通过构建城轨领域知识增强型大语言模型（large language models，LLMs），结合任务感知型调度器与多模态融合引擎，构建云-边-端协同算力平台，填补上述研究空白。

1.2 总体架构设计

城轨云算力资源平台采用六层垂直架构，从顶层的用户交互到底层的物理设备形成完整的闭环。每层功能高度内聚，通过标准化接口实现解耦。平台总体架构如图1所示。

（1）用户接口层。与用户直接交互，接收指令并展示结果。集成运营监控中心与车载终端，提供可视化控制面板和实时数据仪表盘。

（2）安全可信层。融合联邦学习框架（纵向数据分割机制）与区块链存证技术，采用可信执行环境（trusted execution environment，TEE）可信执行环境保障敏感数据处理安全。

（3）调度管理层。负责资源调度和性能优化，包含Quality of Service Awareness（QoS）感知混合

收稿日期： 2025-08-16

基金项目： 北京市自然科学基金（L221010）；中国铁道科学研究院重大基金（2024YJ092）

作者简介： 赵俊华（1990—），女，山西汾阳人，硕士，副研究员，研究方向为信息技术；杜呈欣（1980—），男，河南洛阳人，硕士，研究方向为信息技术。

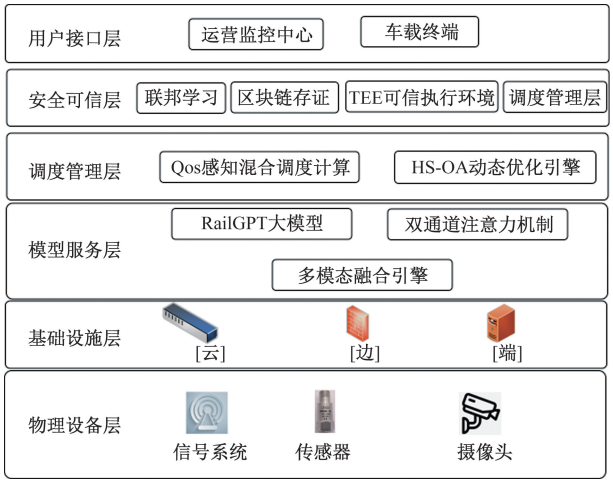


图 1　平台总体架构

调度算法和 Hybrid Scheduling-QoS Awareness (HS-QA)动态优化引擎。QoS 感知混合调度算法中任务 t_i 分配到资源 r_j 的综合评分公式为

$$\text{Score}(t_i,r_j) = \alpha\text{EDD}(t_i) + \beta\text{ACC}(m_k)\gamma\text{Power}(r_j) \tag{1}$$

式中：t_i 为基本调度单元第 i 个任务；r_j 为第 j 个资源，如计算节点、中央处理器(central processing unit，GPU)等；m_k 为第 k 个模型，指任务关联的机器学习模型或性能预测模型；EDD 为任务紧迫度；ACC 为模型精度；Power 为能耗；α、β、γ 分别为权重系数。通过权重系数 α、β、γ 实现多目标资源调度优化，EDD(t_i)越小(任务越紧迫)，评分越高；ACC(m_k)越大(模型精度越高)，评分越高；Power(r_j)越小(资源能耗越低)，评分越高。

(4)模型服务层(MaaS)。提供模型服务处理多模态数据，包含城轨大模型(RailGPT)、双通道注意力机制[8]和多模态融合引擎。开发 RailGPT 基础模型，注入城轨领域知识库(包含 30 万条设备参数、运行规程等结构化数据)。创新设计双通道注意力机制，全局通道捕捉系统级状态特征，局部通道聚焦设备级异常信号。

(5)基础设施层。构建跨地域的“云边端”协同算力网络，中心云部署高性能计算集群处理离线训练任务，边缘节点配置边缘计算设备实现对视频流、传感器数据的实时推理，移动终端设备通过 5G 多接入边缘计算(5G MEC)网络接入系统。

(6)物理设备层。与物理设备交互，采集和反馈数据，包含信号系统、传感器和摄像头。

架构图 1 遵循“基础设施→模型服务→调度决策→安全控制→用户交互”的递进式设计。双向自适应数据流：从用户接口层到物理设备层下发控制指令/策略，从物理设备层到用户接口层反馈实时数据/推理结果，下行控制流遵循调度指令→资源分配→模型加载，上行数据流遵循设备数据→特征提取→模型推理。跨层联动机制：MaaS 层实时向调度层反馈模型负载状态(如 GPU 内存占用率)，安全可信层对模型输入/输出进行可信验证(如区块链存证推理日志)。

2　关键技术

以下关键技术形成了“知识驱动-资源优化-感知增强”的闭环，大模型为多模态融合提供语义先验，资源调度算法保障边缘端实时性，多模态数据反哺模型迭代，最终支撑城轨业务智能化升级。

(1)领域大模型训练优化。针对城轨领域专业性强、知识密集的特点，采用两阶段混合训练策略，解决通用模型与垂直场景适配难题。在预训练阶段，基于大模型引入知识感知注意力机制，通过动态筛选 80 万条通用语料中与城轨相关的上下文(如“信号系统”“接触网”)，结合知识图谱实体嵌入进行联合训练，提升模型对领域术语的隐性理解能力；在微调阶段，设计适配器(Adapter)与低秩矩阵双模块协同机制，前者针对工单分类、故障推理等任务优化参数，后者将 30 万条设备参数、规程知识通过图神经网络编码为结构化向量，并与文本表征对齐，既避免灾难性遗忘，又实现知识逻辑的精准注入。

(2)弹性资源调度算法。面向边缘-云端异构资源动态调度需求，构建基于时空特征预测与混合决策的弹性资源管理体系。设计轻量化 Informer 时序预测模型，以历史负载、任务队列优先级、设备健康状态(如 GPU 显存碎片率)为输入特征，实现未来 1 h 资源需求预测。结合预测结果，提出“离线强化学习＋在线遗传算法”的双层调度策略：离线阶段通过深度强化学习训练全局最优资源分配模型，最大化 GPU 利用率与任务完成率的均衡；在线阶段采用遗传算法快速响应突发负载，支持毫秒级动态重调度。

(3)多模态数据融合。为解决城轨场景中视频、传感器、文本数据的异构融合难题，提出时空对齐与可解释门控融合一体化架构[14-15]。视频流经轻量化三维卷积神经网络提取时空特征，实现每秒 30 帧实时处理并捕获设备运动异常。传感器数据通过多尺度时序卷积网络(temporal convolutional network，TCN)建模设备状态演化，结合自监督异

常检测屏蔽噪声信号。文本日志由领域微调的RailGPT生成语义嵌入,并关联知识图谱构建事件链(如“车门故障-复位失败-系统告警”)。在此基础上,开发动态门控融合网络,基于跨模态注意力机制为各模态分配权重(如视频 $\alpha=0.6$、传感器 $\beta=0.3$、文本 $\gamma=0.1$),并通过知识图谱置信度仲裁解决模态冲突(如传感器误报)。该技术在Jetson AGX边缘节点上实现多模态推理延迟小于80 ms,提升了设备故障检测率。

3 算力网络工作原理及业务流程

3.1 工作原理

基于大模型的城轨云算力网络将算力、网络、存储等资源深度融合和调度,以实现算力资源的高效共享和灵活分配。用户可按需获取,无须关心算力资源的具体位置和物理形态,算力网络的工作原理如图2所示。

(1)资源感知与建模。算力网络中的各个节点(包括算力节点、网络节点等)会实时收集自身的资

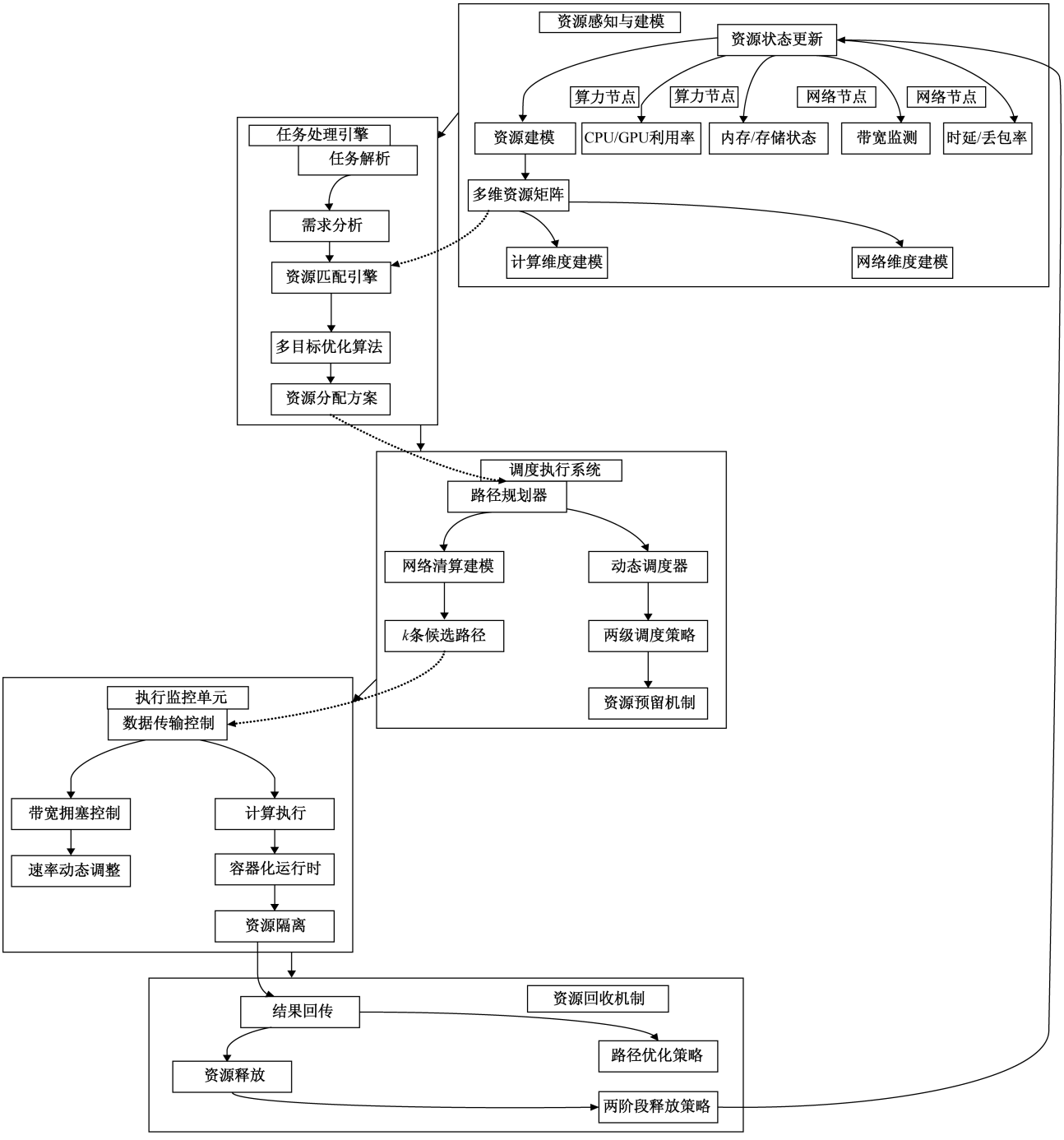


图2 算力网络工作原理

源信息，如算力节点的CPU、GPU性能、内存使用情况、存储容量等；网络节点的带宽、延迟、丢包率等。利用这些收集到的信息，通过特定的算法和模型对整个网络中的资源进行建模、形成一个全局的资源视图，以便清晰地了解网络中各种资源的分布和状态。

(2)任务解析与需求匹配。当用户提交一个计算任务时，算力网络首先会对任务进行解析，分析任务的类型(如人工智能训练、数据处理、科学计算等)、计算量大小、数据量大小、对网络带宽和延迟的要求等。然后根据任务的需求，在全局资源视图中寻找最合适的算力资源和网络资源组合，以满足任务的执行要求。

(3)路径选择与调度。确定了所需的算力资源后，算力网络会根据网络拓扑结构和当前的网络状态，选择一条最优的数据传输路径，将任务数据从用户端传输到选定的算力节点。同时，根据算力节点的负载情况和任务的优先级，对任务进行合理调度，确保任务能够在合适的时间在算力节点上开始执行。

(4)数据传输与计算执行。按照选定的路径，数据通过网络传输到算力节点。在传输过程中，网络会根据实际情况动态调整传输参数，如带宽分配等，以保证数据传输的稳定性和效率。算力节点接收到数据后，根据任务要求进行计算处理，利用自身的计算资源(如CPU、GPU等)对数据进行运算，完成任务的计算部分。

(5)结果反馈与资源释放。计算完成后，算力节点将计算结果通过网络传输回用户端。同时，算力网络会对本次任务所占用的算力资源和网络资源进行释放，使其可以被其他任务重新使用，以提高资源的利用率。

3.2 业务流程

基于大模型的城轨云算力资源平台的业务流程如图3所示。

(1)用户需求提交。用户通过特定的接口或平台，向算力网络提交计算任务请求，包括任务的详细描述、所需的算力资源类型和数量、数据输入输出要求等信息。

(2)需求审核与评估。城轨云算力资源平台接收到请求后，对用户需求进行审核，检查需求的合理性和合法性。同时，对任务进行评估，分析任务的复杂性、计算量、数据量等，为后续的资源分配和调度提供依据。

(3)资源分配与调度。根据任务评估结果，结合全局资源视图，城轨云算力资源平台进行资源分配，确定为该任务提供算力的具体节点和所需的网络资源。制定详细的调度计划，包括任务启动时间、数据传输时间等，确保任务能够高效执行。

(4)任务执行与监控。按照调度计划，任务数据被传输到指定的算力节点，算力节点开始执行任务。在任务执行过程中，算力网络的监控系统会实时监测任务的执行状态，包括算力节点的运行情况、网络传输情况等，及时发现并处理可能出现的问题，如节点故障、网络拥塞等。

(5)结果验证与交付。任务执行完成后，对计算结果进行验证，确保结果的准确性和完整性。将验证后的结果通过网络交付给用户，用户可以在自己的终端设备上获取和使用计算结果。

(6)计费与结算。根据用户使用的算力资源和网络资源的量，按照事先约定的计费标准，对用户进行计费。完成费用的结算，用户可以通过各种支付方式支付使用算力网络的费用。

(7)业务反馈与优化。收集用户对本次业务的反馈意见，了解用户的满意度和需求改进点。根据用户反馈和业务运行数据，对算力网络的资源配置、调度策略等进行优化，以提高算力网络的服务质量和性能，为后续的业务提供更好的支持。

4 实验分析

在基于典型云平台试点架构与工作负载构建的实验室高仿真环境中，搭建硬件基础设施采用三级分布式架构，即中心云节点层部署8台高性能计算服务器以支持大模型训练与远端渲染任务、边缘节点配置20台边缘计算设备用于实时视频分析、车载终端配置多模态计算集成高精度定位模组。软件栈采用Linux操作系统，基于Docker容器引擎与Kubernetes集群管理实现容器化部署以支持服务动态扩展与资源高效调度，基于深度学习实现模型高效训练与推理。

实验数据集模拟大型城市的近三年运营数据，日均数据量1.2 TB，具体包括：客流数据(闸机通行记录，峰值采样频率1 Hz)；设备状态数据(列车牵引系统、CBTC信号设备的32维传感器时序指标)；故障日志(8.7万条带标注的维修工单文本)。数据预处理阶段，视频流经目标检测模型生成乘客密度热力图，传感器数据通过滑动窗口(60 s窗口＋10 s

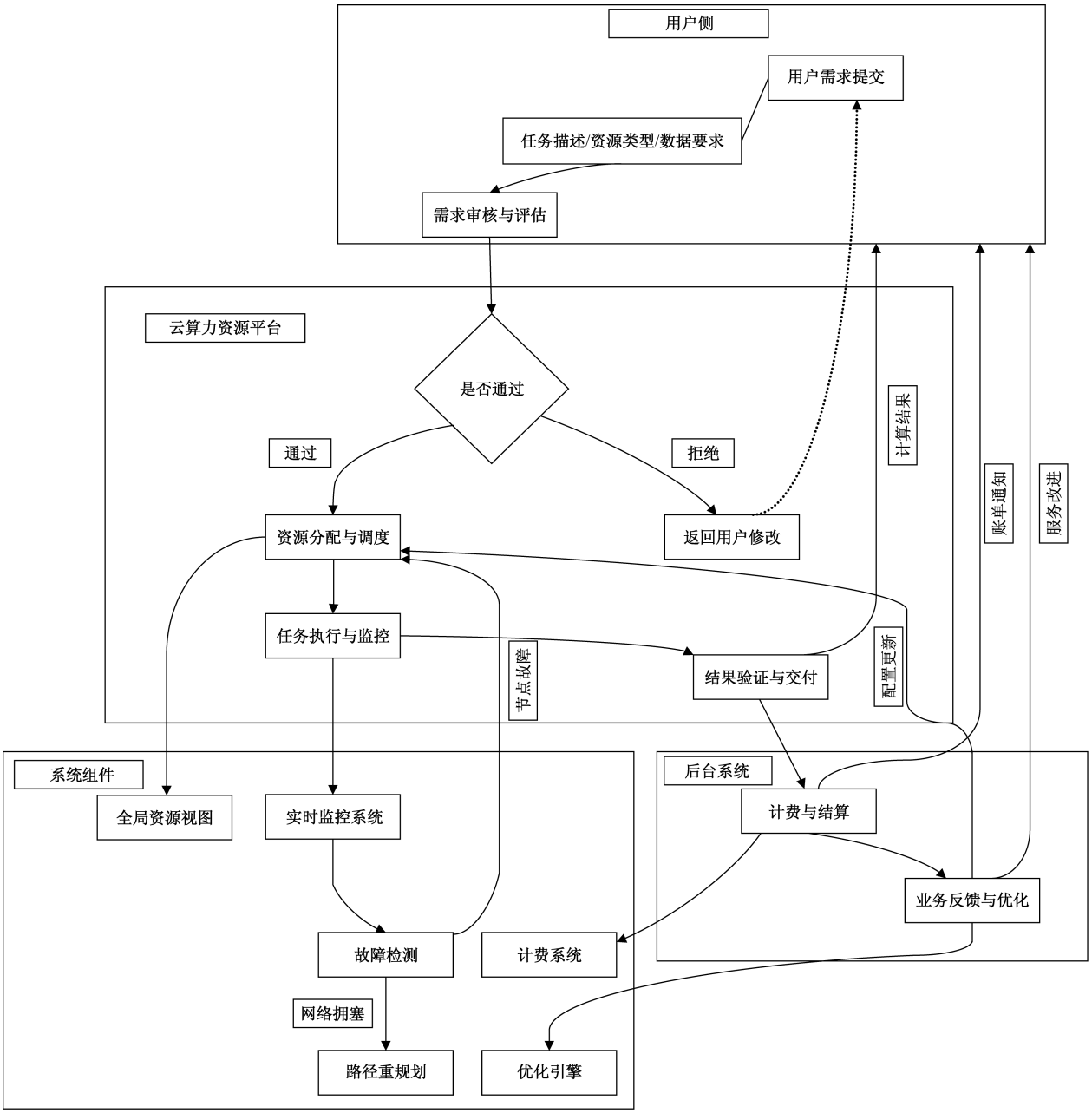


图 3　算力资源平台的业务流程

步长)提取时频特征,文本数据则由 RailGPT 完成实体识别与结构化映射。

选取的基线模型有:Kubernetes 默认调度器(基于资源请求量的静态分配)[9-10];TransGPT(通用交通大模型,未注入城轨领域知识)[12-13];传统专家系统(基于规则推理的故障诊断引擎)[7-8]。

评估指标涵盖资源利用率(resource utilization rate, RUR)、端到端延迟及加权 F_1 值。

(1)资源利用率(RUR):

$$\mathrm{RUR}=\frac{\sum_{j=1}^{n}\mathrm{Active_Time}(r_j)}{n\mathrm{Total_Time}}\times 100\% \qquad (2)$$

式中:n 为计算节点总数(如 A100 服务器、边缘设备等),个;Total_Time 为实验总时长(s),取 24 h (86 400 s);Active_Time(r_j)为节点 r_j 处于任务执行状态的时间(即 GPU/CPU 利用率≥80%的时段),通过 Prometheus 监控系统采样获取(采样间隔 1 s)。资源利用率反映平台在实验周期内整体计算资源的有效利用比例,值越高表明资源闲置越少。

(2)响应延迟(end-to-end latency):从任务提交到返回结果的时间间隔。

(3)模型准确率:采用加权 F_1 值表示,计算公式为

$$F_1 = 2\frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}\times 100\% \quad (3)$$

式中：精确率为

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}}\times 100\% \quad (4)$$

召回率为

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}}\times 100\% \quad (5)$$

式中：TP（true positive）为正确预测的正例数；FP（false positive）为错误预测的正例数；FN（false negative）为漏检的正例数。

表 1 展示了资源调度性能对比结果。

在资源调度性能方面，本文提出的 HS-QA 算法实现 89.2%的资源利用率，较 Kubernetes（61.6%）与 TransGPT（68.4%）分别提升 27.6%与 20.8%。高负载时段（早高峰 8:00—9:00）的 GPU 利用率稳定在 85%以上，突发任务响应延迟标准差仅为 0.3 s。

表 2 展示了模型推理能力对比结果。

模型推理能力测试显示，RailGPT 在突发客流预测（F_1=93.8%）、设备故障诊断（F_1=92.4%）及多模态异常检测（F_1=89.7%）三大场景中均显著优于基线模型，其中故障诊断准确率较 TransGPT 提升 15.3%。图 4 给出了平台在城轨通信设备状态统计中的应用效果。

表 1　资源调度性能对比结果

指标	本文方法	Kubernetes	TransGPT	传统专家系统
资源利用率（RUR）/%	89.2	61.6	68.4	53.1
平均响应延迟/s	1.8	3.1	2.7	4.9
任务排队成功率/%	98.5	82.3	89.7	71.6

表 2　模型推理能力对比结果

场景	RailGPT-F_1	TransGPT-F_1	传统专家系统
突发客流预测/%	93.8	84.5	—
设备故障诊断/%	92.4	77.1	68.9
多模态异常检测/%	89.7	72.6	—

5　结论

本文构建的云算力资源平台通过大模型与云原生技术的深度融合，实现了算力资源的高效利用与智能化服务创新，有效解决了城轨系统智能化转型中的算力瓶颈问题。实证表明，该平台在故障预测、乘客服务等多场景中具备显著优势。未来研究将聚焦于城轨数字孪生与元宇宙技术的深度集成、面向 6G 通信的算力-网络协同优化等前沿方向，进一步推动城轨系统的数字化转型，为乘客提供更加便捷、安全、舒适的出行体验。

参考文献

[1] 李美，邱烨，刘德坤，等. 超大线网标准城轨云及共享数据平台分析[J]. 交通科技与管理，2024，5(8)：9-11.

[2] 宋程程，陈艳艳，陈宁，等. 城市轨道交通车站客流状态采集范围[J]. 科学技术与工程，2021，21(27)：11836-11842.

[3] 姚世峰. 智慧城市轨道交通云平台应用方案研究[J]. 铁路计算机应用，2020，29(11)：9-13.

[4] 孙然然，陈家旭，李晓璐，等. 面向运营安全的城市轨道交通线网性能综合评价[J]. 科学技术与工程，2020，20(36)：15067-15073.

[5] 陈基漓，李海军，谢晓兰. 基于时间卷积和长短期记忆网络的短期云资源预测模型[J]. 科学技术与工程，2025，25(7)：2856-2864.

[6] 李瑞，李平，马小宁，等. 5G 环境下铁路视频图像智能化应用云边协同技术架构研究[J]. 铁道运输与经济，2022，44(5)：57-63.

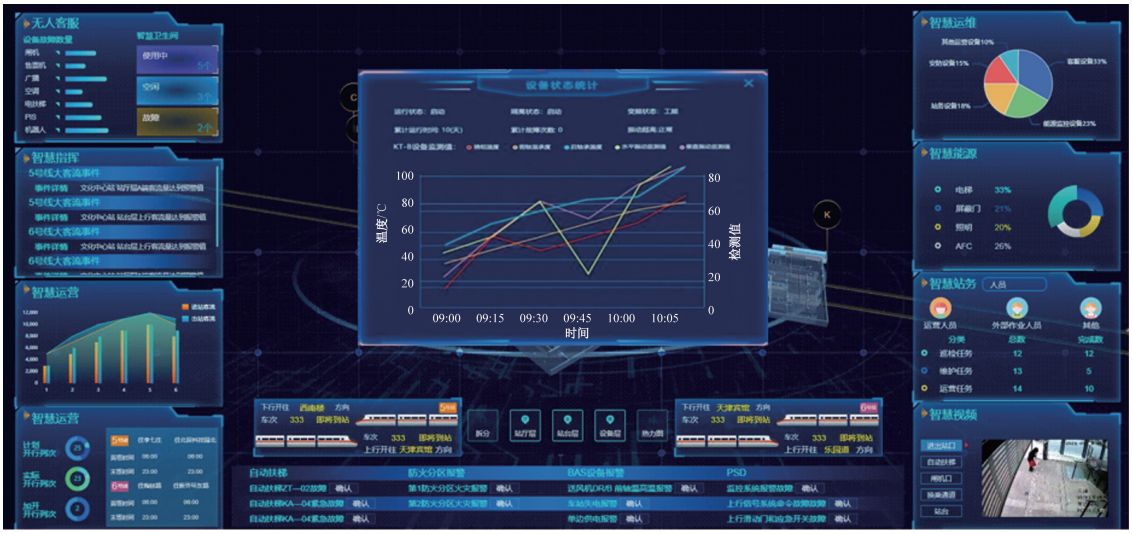


图 4　设备统计展示效果

[7] 刘雯. 基于推理机的机械臂故障诊断专家系统优化设计[J]. 液压气动与密封，2023，43(2)：3-6.
[8] 崔鸣石，邬雪阳，朱宏伟，等. 基于知识图谱的电力通信设备故障智能诊断方法[J]. 科技和产业，2024，24(5)：212-221.
[9] 高荣，谢晓兰，刘亚荣，等. Kubernetes 资源调度策略[J]. 计算机工程与设计，2024，45(6)：1896-1902.
[10] 刘阳. 基于 Kubernetes 的机器学习云平台架构的设计与实现[J]. 信息与电脑，2025，37(1)：53-55.
[11] 黄霁. 大模型 AI 在城市轨道交通中的应用探讨[J]. 智能建筑电气技术，2024，18(2)：25-31.
[12] 肖建力，邱雪，张扬，等. 交通大模型综述[J]. 交通运输工程学报，2025，25(1)：8-28.
[13] 张博宇. 基于 AI 大模型的城市轨道交通智慧运维与智慧车站建设研究[J]. 人民公交，2025(6)：103-105.
[14] 苏易礵，李卫军，李贯峰，等. 基于门控特征融合的双通道文本分类模型[J/OL]. 计算机工程与应用，1-15. [2025-08-01]. https://link. cnki. net/urlid/11. 2127. tp. 20250225. 1637. 018.
[15] 王燕，陈燕燕，刘晶晶，等. 基于 Transformer 和门控融合机制的图像去雾算法[J]. 计算机系统应用，2025，34(2)：1-10.

Design of Cloud Computing Resource Platform for Urban Rail Transit Based on Large Language Models

ZHAO Junhua，DU Chengxin

(Institute of Computing Technology，China Academy of Railway Sciences Corporation Limited，Beijing 100081，China)

Abstract：With the improvement of the intelligence level of urban rail transit systems，the traditional computing power resource allocation mode has been difficult to cope with new demands such as multi-source heterogeneous data processing and real-time decision support. An architecture of urban rail cloud computing power resource platform based on large language Models (LLMs) was proposed，a hybrid scheduling algorithm integrating knowledge graph was constructed，and the utilization rate of computing power resources through a dynamic resource allocation optimization strategy was improved. Experiments show that when dealing with typical scenarios such as sudden passenger flow prediction and equipment fault diagnosis，this platform shortens the response time and improves the accuracy of model reasoning. A new type of computing power infrastructure solution is provided for the construction of smart urban rail transit.

Keywords：intelligent urban rail transit；cloud computing power resource platform；hybrid scheduling；RailGPT