工程与应用

# 算网大脑与边缘计算的智能化协同调度研究

安颖[1]，闫亚旗[1]，王东[1]，汪涛[1]，刘申易[2]
（1. 中国铁塔股份有限公司，北京 100080；
2. 中国移动集团设计院有限公司，北京 100080）

**摘　要：**在数字时代信息科技不断发展的进程中，以算力网络及边缘计算为核心的计算范式，已于诸多领域得到了应用与认可。通过整合异构异地的计算资源，算网大脑可实现高效的资源调度及任务分配。对算网大脑与边缘计算之间的协同方式展开了探索，剖析当前及新兴的技术手段，提出了独创性的算法体系，其具备主客观多要素指标、动态阈值及赋权、双尺度决策与三级资源调度五大特征。通过该算法的落地执行，可有效减少业务总时延，提升资源利用率，加强对业务可靠性的保障。

**关键词：**算力网络；算网大脑；边缘计算
**中图分类号：**TP393.0
**文献标志码：**A
**doi:** 10.11959/j.issn.1000−0801.DXKX250463

# Research on intelligent integrated collaborative scheduling of computing network brain and edge computing

An Ying[1], Yan Yaqi[1], Wang Dong[1], Wang Tao[1], Liu Shenyi[2]
1. China Tower Co., Ltd., Beijing 100080, China
2. China Mobile Group Design Institute Co., Ltd., Beijing 100080, China

**Abstract:** Amid the ongoing advancement of information technology in the digital era, computing paradigms centered on computing power network and edge computing have been widely applied and recognized across numerous fields. By integrating heterogeneous and distributed computing resources, the computing network brain enables efficient resource scheduling and task allocation. The collaborative mechanisms between the computing network brain and edge computing were explored, both current and emerging technical approaches were examined, and an innovative algorithmic system was proposed. This system was characterized by five key features: multi-dimensional subjective and objective metrics, dynamic threshold adjustment and weighting, dual-scale decision-making, and intelligent hierarchical scheduling strategies. The implementation of this algorithm effectively reduces overall service latency, enhances resource utilization, and improves operational reliability.

**Key words:** computing power network, computing network brain, edge computing

收稿日期：2025−07−21；修回日期：2026−03−30
通信作者：安颖，anying3@chinatower.com.cn

## 0　引言

近年来，伴随着人工智能（AI）、物联网（IoT）和大数据技术的飞速进步，对于计算技术的需求正呈现快速增长态势。鉴于数据处理规模巨大，尤其在需要极低时延和极高带宽的应用情境下，传统集中式云计算架构在性能与成本方面面临着制约。边缘计算[1]技术将部分数据处理和计算过程移到数据产生点附近，也就是“边缘”，这种做法有助于提高服务响应速度，从而解决本地实时性需求与云端密集计算能力的矛盾。同时，边缘算力通过“私有云”“专属云”的部署方式有效满足了各方机构及客户对于数据主权、数据交易越来越高的需求，增强了对个人隐私的保护、数据价值的确权以及系统的安全防护[2]。

在此背景下，算力网络的理念应运而生，通过引入算力路由算法，借助网络技术连接分散的计算及数据资源，整合目前的通算、智算、超算及未来的量算等异构算力为一体的计算网络，并依托“云边端网”协同的机制[3-4]，将复杂计算任务细化并分配给分布式多域网络中的不同资源节点进行并行处理，如此便极大地提升了整体性能及反应速度。但另一方面，异构异地异网部署的边缘算力也带来了资源分布碎片化的局面，导致多方资源难以有效协同，总体利用率难以提高的问题日益凸显。而算网大脑[5]作为一种创新性的智能计算资源调度解决方案，主要致力于高效地管理及调配异地异构计算资源，从而进一步优化整体计算资源的利用率。尤其在未来 6G 网络背景下，算网大脑的智能资源编排技术也为跨域算力协同提供了新的技术路径[6]。

算网大脑，顾名思义，是一套将算力网络与人工智能技术融为一体的高级计算系统。此系统采用了智能算法和大数据分析工具，可以实时监控资源分配并做出最优选择，以满足不断增长的计算和数据处理需求。王琛等[5]在 2024 年提出了算网大脑的智能化水平分级方法，为智能的定义及演进奠定了基础。但是，算网大脑由于系统自身的复杂特性，研发维护管理的开销成本相对来说更大，意味着相关资源管理及调度算法的优化是一项持续的工作。陈星延等[7]在 2022 年提出了异构资源协同调度的系统性方案，研究构建了广义图结构的通用服务模型，通过双虚拟队列动态表征计算与传输负载，但在弱网环境下仍存在边缘节点资源利用率波动较大的问题。孙滔等[8]在 2021 年提出了 DTN 技术可将边缘节点的资源预测误差降低至 3% 以内，并支持跨域算力的纳秒级调度，但是在动态网络故障恢复方面仍存在短板，当网络切片发生中断时，孪生模型的同步更新时延可能导致调度策略失效。

## 1　算网大脑和边缘计算协同调度研究

算网大脑与边缘计算相结合[9-10]的应用模式，能够有效结合两者的优势，同时规避各自的不足之处，从而实现更加高效的计算能力和数据处理能力。在此协同框架下，边缘计算主要承担在数据产生点附近执行即时数据处理及局部分析决策任务，以此减少时延并节约带宽资源，避免大量原始数据在云端排队的拥塞情况。此外，边缘设备还能先行完成基本的数据筛选和预处理工作，只将有价值或已处理过的数据传输给算网大脑进行更深入细致的解析和战略规划。因此，算网大脑能够依靠其强劲的中心化计算功能，有效地执行各种复杂的算法计算和建模训练流程，整合从各边缘节点收集的信息，从而优化整个系统决策的品质和广泛性。基于这种深度协作，算网大脑与边缘计算携手共同建立了一个灵活可扩展、高效安全的新型计算方法体系。该体系不仅具备实时数据评估、智能决策支持、资源最优配置和安全信息等多方面能力，而且大大提高了整个计算架构的智能水平和响应迅速性。

### 1.1 架构设计

算网大脑和边缘计算智能化协同的架构通过层次化设计，实现了高效的数据处理、分析、全局资源调度和智能决策。脑边智能化协同架构如图1所示。

终端节点包含各种用户设备，如个人计算机（PC）、手机、智能穿戴设备、虚拟现实设备、物联网设备等。基础设施层提供各种算力资源，包括通算、智算、超算等，可满足容器、虚拟化、裸金属等多样性算力需求。边缘计算节点实时收集基础设施层和终端信息，在本地进行初步分析、过滤和局部决策。经过处理的数据通过安全的传输网络发送至算网大脑，后者运用强大的计算能力进行深度分析和决策。最终，算网大脑将智能决策反馈给资源节点，形成闭环反馈机制，促进系统的优化与提升。这种协同架构不仅提升了处理性能，降低了时延，还提升了系统的整体智能化水平。

### 1.2 关键算法研究

边缘节点设备具备一定程度的计算能力，可以执行数据预处理和计算任务[11]，以执行算网大脑下发的算法。借助数据的预处理过程，可减轻算网大脑运算负载，降低数据传输时延。具体可分为数据采集、数据清洗、数据聚合和数据压缩[12]4个部分。算网大脑可调用云端算力资源，并依据海量的历史数据训练异常检测模型，训练完成后，把该模型部署到边缘计算节点上，边缘节点借助训练好的模型开展智能推理以及模型微调工作，可以实时识别异常数据和潜在故障，从而及时触发预警和故障预防机制，并且可以将优化后的模型返回到云端进行节点间的共享与模型层面的迭代优化。为实现上述目标，提出了一种基于多要素指标的动态决策算法，该算法结合主客观指标进行阈值标定及动态赋权，以生成判定函数，之后运用两个时空尺度（微观/时间点、宏观/时间轴）的模型来判定下发的资源调度策略，随后将调度策略的运行状态反馈给算网大脑，以此实现调度策略层面的优化。算法关键技术点可分为如下5个部分。

#### 1.2.1 构建主客观多要素指标

算法的核心约束条件包括客观性指标和主观性指标。其中，客观性指标主要是硬件负载情况，包括算力资源满足情况$C$、存储资源满足情况$S$、网络资源满足情况$N$；主观性指标主要是用户满意度情况，包括计算速度$C_i$、服务响应时间$N_i$、数据安全满意度$S_i$等。边缘计算多要素判断如图2所示。

具体计算式如下：

$$C=\begin{cases}(1-\dfrac{C_f}{C_z})*\min(\dfrac{C_S}{C_x},1), & a*C_x\leqslant C_s\\ 0, \text{其他}\end{cases} \quad (1)$$

其中，$C$表征算力资源的空闲程度，$C_f$代表当前边缘节点下所有资源节点算力负载值（可通过中

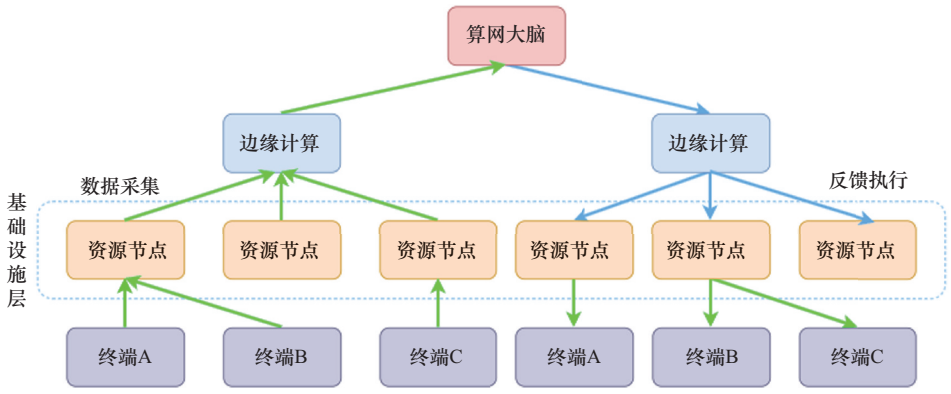



图1 脑边智能化协同架构

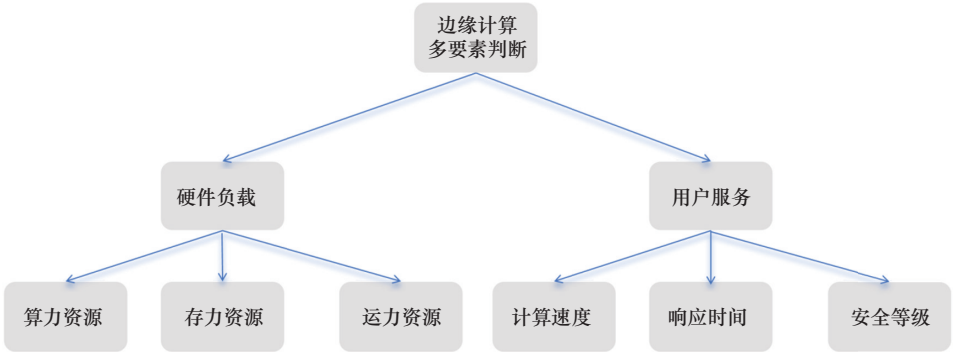



图2　边缘计算多要素判断

央处理器（CPU）、内存等负载情况综合评定），$C_z$代表设定的算力负载阈值，$C_s$代表总算力供给，$C_x$代表算力总需求。$a$为性能或用户满意度显著劣化时的门限值，表征资源调度的弹性，对算力资源一般取经验值为0.6，后续可根据实际运行情况迭代调整。

$$S=\begin{cases}\left(1-\dfrac{S_f}{S_z}\right), a*S_x\leqslant S_s\\ 0, \text{其他}\end{cases} \tag{2}$$

其中，$S$表征存储资源的空闲程度，$S_f$代表当前边缘节点下所有资源节点存储负载值（可通过存储容量、并发访问、冗余备份等负载情况综合评定），$S_z$代表设定的存储负载阈值，$S_s$代表总存力供给，$S_x$代表总存力需求。由于存力资源分配没有弹性，$a$对于存力资源一般取定值为1。

$$N=\begin{cases}\max\left(0,1-\dfrac{N_f}{N_z}\right)*\min\left(\dfrac{N_s}{N_x},1\right), a*N_x\leqslant N_s\\ 0,\text{其他}\end{cases} \tag{3}$$

其中，$N$表征网络资源的空闲程度，$N_f$代表当前边缘节点下网络的负载度，$N_z$代表设定的网络负载阈值（可通过带宽、时延、吞吐量、丢包量等负载情况综合评定），$N_s$代表总运力，$N_x$代表总运力需求。由于运力资源一般要求轻载甚至专线承载，调度弹性较低，$a$对于运力资源一般取经验值为0.8，后续可根据实际运行情况迭代调整。

$$C_i=\min\left(\frac{C_n}{C_p},1\right) \tag{4}$$

其中，$C_p$代表承诺给用户的计算速度，$C_n$代表当前边缘节点的计算速度。

$$N_i=\min\left(\frac{N_p}{N_n},1\right) \tag{5}$$

其中，$N_p$代表承诺给用户的服务响应时间上限，$N_n$代表当前服务响应时间。

$$S_i=\min\left(\frac{S_n}{S_p},1\right) \tag{6}$$

其中，$S_p$代表承诺给用户的安全等级积分，$S_n$代表当前安全等级积分。

### 1.2.2　动态阈值设定

上述指标的阈值如$C_z$、$N_z$等若作为固定静态指标则难以精确衡量边缘节点时空分布的差异性，如季节性的闲时和忙时，或工作日的峰谷值，也无法契合节点的业务流量突变场景，即当算力需求、网络时延需求陡增时，静态指标会存在滞后、误判或者频繁过载的情况。针对这些情况，提出融合长短期记忆（LSTM）网络时序建模与突变适应机制的动态阈值设定方法，以实现阈值的精准适配。

运用3层LSTM网络，其结构包含输入层、隐藏层以及输出层，借助门控机制来捕捉长短期依赖[13]，如节假日、工作日与周末的负载周期差异以及突发多并发任务导致的短期波动等。输出层为$t$时刻的阈值预测值$\mathrm{LSTM}_C(t)$，该值可作为动态阈值的基础值。以算力负载为例，对$T$个周期内的算力负载值进行如下定义：

$$\{C_f(t-1), C_f(t-2), C_f(t-3), ..., C_f(t-T)\} \quad (7)$$

归一化处理各个$k$时刻的算力负载为：

$$C_f^*(k) = C_f(k)/C_{\max} \quad (8)$$

其中，$C_f(k)$为$k$时刻算力负载，$C_{\max}$为边缘节点最大算力负载。

针对突变情况，引入突变检测与动态修正模块，解决LSTM对历史趋势的依赖导致的滞后问题，定义突变函数为：

$$MC(t) = \begin{cases} MC(t-1)+1 & , \text{abs}[C_f(t) - C_f(t-T, t-1)] > 3\sigma C(t-T, t-1) \\ 0 & , \text{其他} \end{cases} \quad (9)$$

$$M(t) = \begin{cases} 1, & MC(t) \geqslant q \\ 0, & \text{其他} \end{cases} \quad (10)$$

其中，$C_f(t-T, t-1)$为$t-T$至$t-1$周期的$C_f$均值，$\sigma C(t-T, t-1)$为算力的历史负载标准差，衡量负载的正常波动范围。利用$MC(t)$函数记录突变连续发生的次数，当连续$q$个周期发生偏离度高的波动时，一般可以认为当前系统发生了突变，此时对$M(t)$函数赋值为1。针对不同场景，采样周期根据参数敏感度取不同值，针对时延带宽敏感型任务，单个采样周期时长取值为10~100 ms，针对普通业务场景，往往可以取值为1~30 s。$q$为突变连续发生次数，系统噪声或瞬时任务可能导致单次负载跳变（如偶然的大文件传输），但不一定代表系统负载持续提升。因此通过设置$q$值，可过滤指标抖动，提升检测准确性以避免出现单次误判的情况。当$M(t)$函数值为1即系统检测到发生突变时，会对基础预测值进行校准修正，此时算力负载阈值表达式为：

$$C_z(t) = \text{LSTM}_C(t)*[1 - \alpha * M(t)] \quad (11)$$

其中，$\alpha$为修正系数，其作用是在突变发生时让阈值下调$\alpha$，即可快速触发预警。

同理设定存力及网络负载阈值。边缘节点在单位周期内采集样本，涵盖算力负载、存力负载、网络负载等方面，然后按一定比例划分训练集及验证集。边缘节点会将阈值$C_z(t)$及其实际运行效果上传至云端，云端通过增量学习的方式更新LSTM参数，使模型逐步学习突变模式，使得后期减少对$\alpha$修正系数的依赖。

LSTM动态阈值方法，通过时序建模捕捉上下文指标的周期性规律，同时依靠突变适应机制对突发异常作出响应，解决了传统固定阈值的适配局限问题。相较于传统静态阈值取定方式而言，该方法使算力负载阈值的预警准确率提升15%，服务响应时间阈值的误判率降低20%。

1.2.3 指标的动态赋权

为了尽可能客观并精确地衡量各个指标对于决策判定结果的贡献度，采用熵权法对其赋权。核心理念是，变化范围大的指标携带更多信息量，故赋以更高权重；反之变化范围小的指标赋以较低权重。在每个边缘节点处，单位周期内按最小元粒度采集6项基础指标形成数据矩阵，之后根据熵权法计算各指标权重，输出权重向量$\boldsymbol{W_i}$，并会依据不同场景下的任务类型对各指标权重进行动态调整，使得指标更好地适配业务类型。举例来说，对于时延敏感任务，会将响应时间的权重提升至原值的1.5倍，而对于计算密集任务，将算力负载的权重提升至原值的1.5倍。熵权法赋值权重向量如算法1所示。

**算法1** 熵权法赋值权重向量

（1） 初始化

def entropy_weight(data)

（2） 数据归一化

normalized = (data-data. min())/(data. max()-data.min()) #用数据减去其最小值，再除以数据最大值与最小值的差值

（3） 计算概率密度

*p*=normalized/sum(normalized, axis=0) # 概率密度

（4） 计算熵值

*k*=1/log(data.shape[0])

entropy=-*k**sum(*p**log(*p*+1e-10),axis=0) #熵值，避免log0

（5） 计算权重

info=1-entropy #熵值与信息量负相关，熵值越大，信息量越小，权重越低

（6） weights=info/sum(info)

（7） 输出权重

return weights

1.2.4　双尺度决策模型

基于上述过程，可得到决策判断模型如下：

$$\mathrm{PDJ}(x)=\sum_{i=1}^{6}\boldsymbol{W_i}*[\boldsymbol{C},\boldsymbol{S},\boldsymbol{N},\boldsymbol{C_i},\boldsymbol{S_i},\boldsymbol{N_i}] \quad (12)$$

其中，$x$为某算力节点，当PDJ值≥0.8时代表当前策略基本满足硬件资源负载需求和用户需求，当PDJ值为0.6~0.8时代表资源需要重点关注，发出预警信号，当PDJ值＜0.6时代表当前边缘节点负载已无法满足用户需求，需立即调度算力。边缘计算会将判断结果（通过$C,S,N,C_i,N_i,S_i$各项值可以判断是哪些方面不能满足需求）和相关运行数据上传至算网大脑。由于PDJ决策模型是多参数综合判断的结果，某些极端场景下，可能存在某一指标极端劣化而其余指标（包括用户满意度指标）仍然表现出较高可用性的情况。例如，某节点出现网络故障导致数据丢包率由0.5%飙升至30%，其余算力、存储等参数仍处于良好的状态，此时该系统已无法满足节点后续任务处理工作，但PDJ可能仍表现出无须立即调度的结果。针对这种情况，增加校验机制，即单一指标极端劣化时，可直接触发决策判断流程，输出需判断是否调度的决策因子。之后由运维人员调用相关AI工具分析该单一劣化指标是否会对当前及未来业务造成影响，以决策是否发起调度流程。

原初PDJ判断模型会通过线性表达输出决策，以判断是否需要将算力调度至边缘节点，然而该模型难以处理指标之间的复杂关联，在特定场景下的表现往往不尽如人意。例如，在算力完全充足但响应时间不稳定的情形下，线性表达决策结果为无须调度或者频繁调度，因此存在误判情况，导致最终用户的体验急剧下降。另一方面，PDJ模型是基于单个节点的当前时刻进行计算，无法体现大规模宏观尺度及未来的演化趋势。对此，在原决策模型微观/时间点尺度的基础上进行如下优化。

引入集成学习模型，通过历史数据的学习以及推理，输出宏观/时间轴尺度判断结果，以更好地预测资源需求的变化趋势。数据采集在原指标的基础上给予扩展，在一个周期内采样原6项指标，并新增4项指标，定义特征向量矩阵为：

$$\boldsymbol{E}(x_1,x_2,\cdots,x_n)=$$
$$[\boldsymbol{C},\boldsymbol{S},\boldsymbol{N},\boldsymbol{C_i},\boldsymbol{N_i},\boldsymbol{S_i},\Delta C_i,\Delta N_i,\boldsymbol{C}*\boldsymbol{C_i},\boldsymbol{N}*\boldsymbol{N_i}] \quad (13)$$

其中，$x_1,x_2,\cdots,x_n$为$n$个算力节点，$\boldsymbol{C}$、$\boldsymbol{S}$、$\boldsymbol{N}$分别为各节点的算力、存储、网络资源满足度向量；$\boldsymbol{C_i}$、$\boldsymbol{N_i}$、$\boldsymbol{S_i}$分别为计算速度、响应时间、安全等级的用户满意度向量；$\Delta C_i$、$\Delta N_i$为计算速度、响应时间的变化率，分别定义为如下值：

$$\Delta C_i=\frac{\boldsymbol{C_i}(t)-\boldsymbol{C_i}(t-1)}{\boldsymbol{C_i}(t-1)} \quad (14)$$

$$\Delta N_i=\frac{\boldsymbol{N_i}(t)-\boldsymbol{N_i}(t-1)}{\boldsymbol{N_i}(t-1)} \quad (15)$$

其中，$\boldsymbol{C_i}(t)$、$\boldsymbol{C_i}(t-1)$分别为本时刻计算速度以及上一周期计算速度，$\boldsymbol{N_i}(t)$、$\boldsymbol{N_i}(t-1)$分别为本时刻响应时间以及上一周期响应时间。$\boldsymbol{C}*\boldsymbol{C_i}$、$\boldsymbol{N}*\boldsymbol{N_i}$分别为算力资源、网络资源耦合度。通过新增变化率以及耦合度指标，可综合反映资源变化过程的深层规律。核心理念是，以变化率指标衡量时间维度的不稳定性及不确定性，以耦合度指标对资源节点之间即资源的总体宏观使用情况进行评估，以作为后续资源预测优化提升的闭环输入。

算力节点按周期采集以上10类指标，进行归

一化处理后输入学习模型。借助多决策树的集成计算，模型逐层拆分特征空间，针对各类指标细分场景，形成可覆盖各类场景的判断规则网络，从而能够自适应不同时段、不同业务等场景。其核心逻辑可表达为：

$$F(X)=\text{vote}[f_1(x),f_2(x),\cdots,f_k(x)] \quad (16)$$

其中，$f_k(x)$为第$k$棵决策树对特征集$X$的判断结果，对应无须调度、立即调度、需要预警3类结果，$k$为决策树数量，vote[]为投票机制，选取占多数树的判断结果作为最终输出。

模型最终输出无须调度、立即调度、需要预警3类结果，并附带决策置信度。当置信度高于设定阈值时，直接采用该决策树集成模型（宏观/时间轴尺度）结果；反之，则调用微观/时间点尺度（PDJ）模型进行二次校验，确保低置信度场景下的决策稳定性。同时，模型基于熵权法输出各指标对模型的贡献度，以辅助判断资源调度的合理性。常规场景下，优先使用决策树集成模型，利用其非线性拟合能力提升判断精度；在边缘节点算力有限或数据量不足时，自动切换至PDJ线性模型，保障基础决策能力；关键业务场景下，两者结果需交叉验证，仅当结论一致时执行调度，降低误判风险。两类模型的协同互补，能够更有效地完成边缘节点的资源调度，为算网大脑提供更贴合实际的技术支撑。

1.2.5 三级资源调度执行

基于上述决策结果，云端与边缘计算的协同可以采用三级资源调度机制，时延敏感型任务就近调度、计算密集型任务云端卸载、普通任务多目标优化，并集成动态负载均衡功能，持续进行智能策略更新。三级资源调度执行程序如算法2所示。

**算法2** 三级资源调度执行程序

（1） #初始化资源拓扑

topology←build_topology(resources) # 包含算力/带宽/位置

task_queue←prioritize_tasks(tasks) #基于服务水平协议分级对任务进行优先级排序

（2） 多要素决策主循环

while task_queue.not_empty():

current_task←task_queue.pop()

（3） #要素1: 时延敏感型判断

if current_task.latency<50 ms #严格时延要求

node←find_qualified_nearest_edge_node (topology,task_requirements) # 选择满足资源需求的最近边缘节点，即调度前判断待选节点的资源满足情况以及距离等

（4） #要素2: 计算密集型判断

else if current_task.compute>10 TFLOPS

if check_cloud_connection(): #网络状态检测

node←select_cloud_server() #选择云端资源

else

node←greedy_select(topology, 'max_compute') #凭借贪心选择算法在拓扑结构中选取有本地最大化算力的节点

（5） #要素3: 多目标优化

else

node←PSO_optimizer(objectives=[latency, energy,cost],constraints=current_task.requi-rements) #优化能耗及成本、考虑约束条件

end while

（6） #执行任务并更新拓扑

execute_task(node,current_task)

update_topology(node) #对节点执行更新操作，动态调整资源状态

（7） #周期性的全局再平衡

if system_load()>0.8 #负载阈值判断

rebalance_withGT_GAOA() #依据博弈论对系统整体进行优化

（8） end procedure

仿真实验设置输入时延敏感任务、计算密集任务、普通任务比例为3∶3∶4，涉及200个边缘节点，节点间网络时延为2~10 ms，模拟省内部署资源场景，针对以上不同场景，采样频率分别设置为20 Hz、5 Hz、1 Hz，仿真时长设置为1 000 s，信道丢包率设置为1%~5%。根据在CloudSim平台的仿真结果，本文模型相较于常用ARIMA预测模型性能显著提升。性能指标对比见表1。

### 1.3　算网大脑的运行流程

算网大脑的运行流程分为业务开通态和常时运行态。在业务开通态下，算网大脑根据多要素设计原则基于深度学习平台对业务进行建模处理。依据业务的算力、存储、带宽、时延需求，全局算力、存储、网络的能力和负载情况，基于历史数据构建业务开通资源调度模型，再判断是将业务下放到最优的算力资源节点抑或由云端承载。

在常时运行态下，算网大脑需要综合来自多个边缘节点的数据，进行深度分析和建模，优化全局资源调度策略及进行各项任务的智能决策。无论是边缘计算节点从基础设施层收集的数据还是全局任务运行数据都具有时间属性，如何挖掘潜藏的时间信息是算网大脑实现动态调度和智能决策的关键，LSTM、循环神经网络（RNN）[14]相对于其他深度学习模型可以学习到各项数据在时间长度上的上下文信息。

在业务开通态下，算网大脑将业务开通部署策略发送到边缘设备后，边缘设备执行指令将业务部署到具体的资源节点，在运行一个或多个周期后，采集设备收集相关运行数据，由边缘设备进行数据处理判断用户对部署策略的满意度及硬件资源的负载情况，将处理后的数据和判断结果反馈至算网大脑，如部署策略能满足用户需求和硬件负载要求，算网大脑则发送维持指令，如不满足，算网大脑根据相关运行数据进行策略调整和算法优化后重新发送至边缘设备执行指令，并重复此过程，直到部署策略满足用户需求和硬件负载要求。

在常时运行态下，由采集设备在固定周期内收集全局硬件运行数据和用户满意度数据，由边缘设备进行数据处理和局部决策后反馈至算网大脑，算网大脑对全局边缘数据进行分析，针对需要优化的策略和算法进行更新，将更新信息发送至边缘设备执行。执行反馈流程如图3所示。

总之，结合AI技术应用，基于算网大脑构建云边协同的算力网络，实现算力效率提升、算力灵活配给、算力感知调度、算力确定承载以及算网全程可视。最终实现业务的用户体验最优化、资源利用率最优化、网络效率最优化。

## 2　算网大脑和边缘计算协同实践

算网大脑与边缘计算协同[15]在智能制造、智慧城市交通治理[16]、智能网联汽车服务、医疗健康应用等多个业务场景中实现深度融合，形成

表1　性能指标对比

| 性能指标 | 本文模型 | 对比模型（ARIMA） | 提升幅度（相对值） |
|---|---|---|---|
| 预测误差（MAE） | 9.4 | 12.5 | 24.80% |
| 预测误差（RMSE） | 10.9 | 18.3 | 40.44% |
| 吞吐量/(Mbit·s$^{-1}$) | 27.3 | 21.3 | 28.17% |
| 平均计算时延/ms | 14.8 | 21.5 | 31.16% |
| 能量消耗/(mAh·h$^{-1}$) | 46.2 | 68.7 | 32.75% |
| 预测稳定性（方差） | 5.5 | 11.8 | 53.39% |

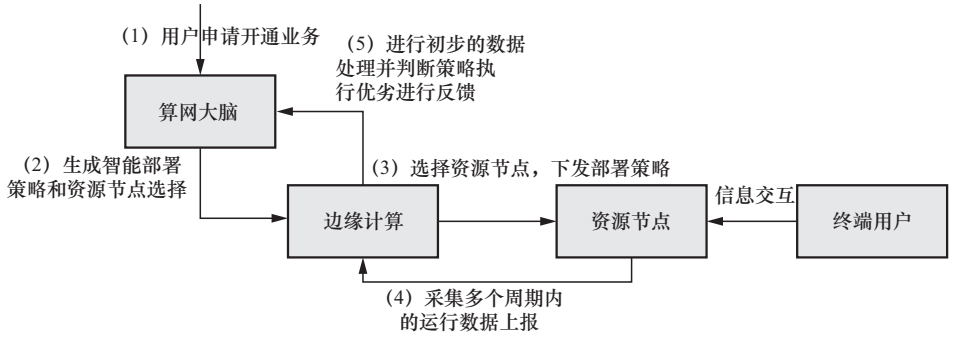



图3 执行反馈流程

“算网大脑”+“云–边–端”协同一体化服务能力。

针对智能制造场景，主要可满足制造产线AI质检以及设备故障预警等核心需求。边缘层执行数据采集、清洗等预处理，采集设备振动、温度数据并提取时域峰值等特征，将异常数据传输至算网大脑。同时基于轻量化模型完成指标的基础检测，通过PDJ模型可判断边缘算力是否满足网络时延等需求。依托算网大脑预测产线负载波动，以时延敏感就近、复杂任务云端的原则将边缘任务卸载至云–边分布式算力集群。同时运用联邦学习训练故障预测模型，结合PDJ值调度边缘节点加大异常设备数据采集频率，从而提升故障检测灵敏度。

而针对智慧城市交通场景，边缘层通过构建路侧带宽及时延等网络指标、电子控制单元（ECU）负载等算力指标等多要素输入，可本地化完成交通流统计，进而通过PDJ值快速判断路口算力是否过载，并将交通流数据传输至算网大脑。算网大脑以动态赋权方式灵活调整交通流数据、信号配时等指标权重，可实现每5 min更新区域内信号灯方案，大幅提升车辆通过率。遇交通事故时，可触发动态阈值突变修正，并及时同步数据至导航系统，指引后续车辆规避事故路段。

针对未来的智能网联汽车场景，重点满足车–边–云低时延传输与多车协同避障需求。边缘层部署在路侧，对车辆上传的激光雷达数据、视频流进行压缩与特征提取，再将处理后的车辆核心数据上传算网大脑。算网大脑将重要任务、困难任务卸载至云端集群。利用集成学习模型优化全局调度方案，遇节点故障或团雾等突发情况时，立即通过PDJ模型调用备用节点，保障协同避障不中断。

以医疗健康行业应用中的远程影像诊断为例，描述算网大脑和边缘计算协同的具体实践。远程影像诊断系统功能架构如图4所示。

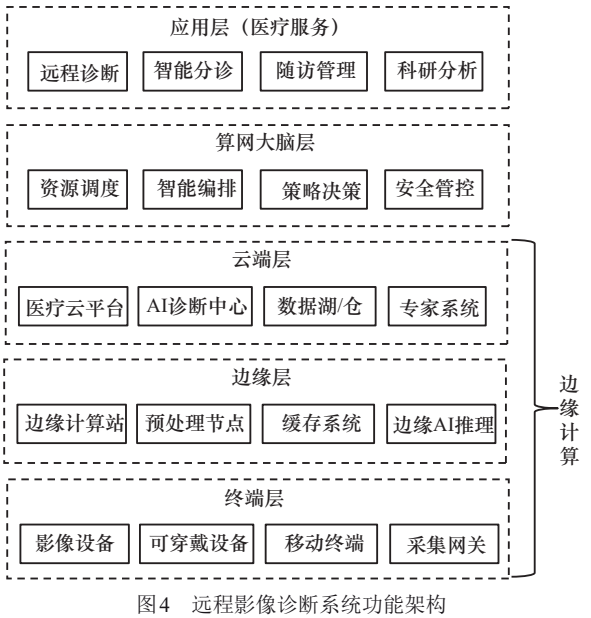



图4 远程影像诊断系统功能架构

远程影像诊断中算网大脑与边缘计算的协同流程如下。

（1）边缘侧数据采集与预处理

本地医疗机构通过CT/MRI设备生成原始影

像数据后，边缘计算节点就近完成图像降噪、格式标准化、AI辅助预处理等操作，并通过资源动态调度与协同，根据网络状态动态调整传输策略，保障高优先级任务的服务质量（QoS）。

（2）资源动态调度与AI辅助分析

算网大脑基于任务优先级（如急诊标记）调度算力资源及医疗资源，实现算力资源及医疗资源的弹性扩容以及跨机构协同诊断与决策。

① 边缘层实时推理，边缘节点搭载的轻量化AI模型，如TinyML，可完成初步的病灶检测工作，可支持在断网场景下进行应急诊断。

② 复杂分析如肿瘤恶性度评估，会触发算网大脑调度云端资源进行三维影像重建与多模态影像融合分析，通过特征对齐技术及联邦计算技术关联不同模态数据，以提升病灶定位精度。

③ 算网大脑借助如V2V协议之类的相关协议来构建跨域算力网络，以此匹配专家资源，专家会利用视频会诊系统给出诊断意见，而诊断意见会和AI分析结果自动整合成为结构化报告，之后进行加密并回传至边缘节点。

（3）治疗跟踪与模型迭代

边缘节点持续收集患者后续治疗产生的数据，如术后影像、病理结果等，借助联邦学习的方式更新AI模型参数，以提升病灶预测的精确程度。

远程影像诊断流程如图5所示。

在算网大脑层，通过部署基于强化学习的智能资源调度算法，实现端到端时延最小化的目标。在边缘计算层，采用自适应任务卸载算法，根据实时网络状况和边缘算力动态决定任务执行位置。

以急诊脑卒中患者的头部CT影像快速诊断任务为例，具体地展示算网大脑在接收到一个急诊任务时的资源调度决策过程。

（1）急诊任务特征与决策前提

急诊任务基础信息如下。

① 任务类型：针对急诊脑卒中患者的头部，展开CT影像快速诊断工作，需识别早期缺血灶，

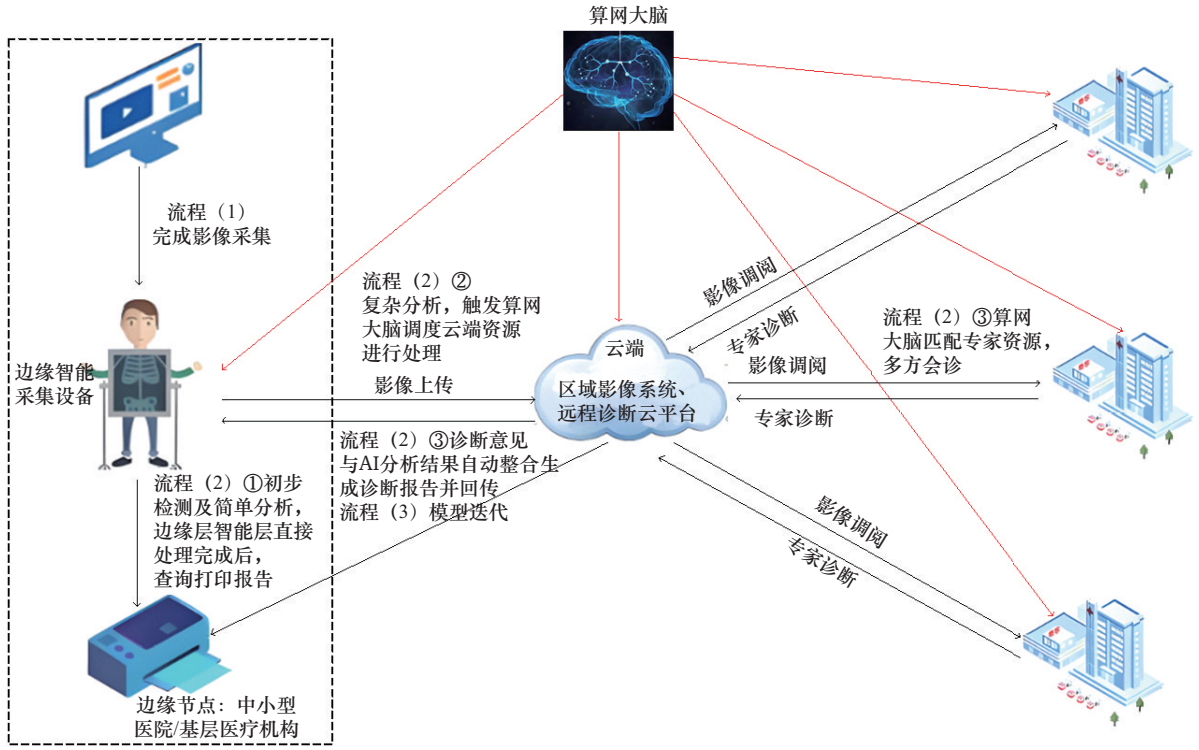



图5　远程影像诊断流程

属于“时延敏感型”，最大容忍时延需小于或等于15 s。

② 数据特征：CT影像格式为DICOM，原始大小为512 MB，边缘侧已完成预处理工作，压缩至128 MB，同时去除患者隐私标识。

③ 算力需求：需图形处理器（GPU）加速推理。

算网大脑实时感知的资源节点属性见表2。

各备选算力节点在数据加密、访问控制、合规审计等安全维度，均满足医疗行业数据安全规范，故安全性不作为本次备选节点间资源调度的输入指标。

其中比较重要的一点是，边缘节点网络传输时延较低，云端节点处理时延较低。

（2）算网大脑的动态决策逻辑（多因素加权评分）

算网大脑通过内置决策引擎，基于强化学习+规则引擎构建，对节点进行评分，具体步骤如下。

**步骤1**：映射算法主客观指标。资源节点指标见表3。

**步骤2**：对候选节点进行PDJ($x$)= $\sum_{i=1}^{6} \boldsymbol{W}_i$* [$\boldsymbol{C}, \boldsymbol{S}, \boldsymbol{N}, \boldsymbol{C}_i, \boldsymbol{S}_i, \boldsymbol{N}_i$]判别。

Edge-A的PDJ计算结果为PDJ=0.2×0.3($\boldsymbol{C}$)+0.1×0.7($\boldsymbol{S}$)+0.1×1.0($\boldsymbol{N}$)+0.2×1.0($\boldsymbol{C}_i$)+0.3×0.43($\boldsymbol{N}_i$)+0.1×1.0($\boldsymbol{S}_i$)≈0.66。

Edge-B的PDJ计算结果为PDJ=0.2×0.7($\boldsymbol{C}$)+0.1×0.8($\boldsymbol{S}$)+0.1×0.9($\boldsymbol{N}$)+0.2×1.0($\boldsymbol{C}_i$)+0.3×1.0($\boldsymbol{N}_i$)+0.1×1.0($\boldsymbol{S}_i$)≈0.91。

Edge-C的PDJ计算结果为PDJ=0.2×0.15($\boldsymbol{C}$)+0.1×0.65($\boldsymbol{S}$)+0.1×0.8($\boldsymbol{N}$)+0.2×1.0($\boldsymbol{C}_i$)+0.3×0.83($\boldsymbol{N}_i$)+0.1×1.0($\boldsymbol{S}_i$)≈0.72。

Cloud-X的PDJ计算结果为PDJ=0.2×0.6($\boldsymbol{C}$)+0.1×0.8($\boldsymbol{S}$)+0.1×0.85($\boldsymbol{N}$)+0.2×1.0($\boldsymbol{C}_i$)+0.3×1.0($\boldsymbol{N}_i$)+0.1×1.0($\boldsymbol{S}_i$)≈0.89。

根据上述结果，得出Edge-B的PDJ分数最高，即其负载冗余更优，且网络时延更低，因此可以作出决策，选择Edge-B作为本次算力调度节点。

（3）动态调整：节点状态突变时的切换机制

若Edge-B在任务调度后突发负载飙升，GPU利用率从30%升至90%，处理时延从8 s增至22 s，总时延=12 ms+22 s=22.012 s＞15 s，算网大脑将

**表2 资源节点属性**

| 节点类型 | 节点标识 | 物理位置 | 当前负载（CPU/GPU利用率） | 网络状态 | 存储利用率 | 处理CT影像数/（张·min$^{-1}$） | 预估处理时延 |
|---|---|---|---|---|---|---|---|
| 就近边缘节点 | Edge-A | 医院急诊楼机房 | CPU 70% | 带宽1 Gbit/s，时延5 ms，丢包率0 | 30% | ≈10 | 35 s（CPU处理，不满足时延） |
| 区域边缘节点 | Edge-B | 城市边缘数据中心 | GPU 30%（空闲算力充足） | 带宽10 Gbit/s，时延12 ms，丢包率0 | 20% | ≈60 | 8 s（GPU加速推理） |
| 区域边缘节点 | Edge-C | 邻区边缘机房 | GPU 85%（负载较高） | 带宽5 Gbit/s，时延20 ms，丢包率1% | 35% | ≈30 | 18 s（负载高导致推理变慢） |
| 云端数据中心 | Cloud-X | 省政务云集群 | GPU 40%（资源充足） | 带宽100 Gbit/s，时延80 ms，丢包率0.5% | 20% | ≈100 | 5 s（云端GPU算力强） |

**表3 资源节点指标**

| 节点标识 | 剩余算力冗余度$\boldsymbol{C}$ | 剩余存储冗余度$\boldsymbol{S}$ | 网络质量$\boldsymbol{N}$ | 计算速度$\boldsymbol{C}_i$ | 响应时间$\boldsymbol{N}_i$ | 安全$\boldsymbol{S}_i$ |
|---|---|---|---|---|---|---|
| Edge-A | 0.3 | 0.7 | 1.0 | 1.0 | 0.43 | 1.0 |
| Edge-B | 0.7 | 0.8 | 0.9 | 1.0 | 1.0 | 1.0 |
| Edge-C | 0.15 | 0.65 | 0.8 | 1.0 | 0.83 | 1.0 |
| Cloud-X | 0.6 | 0.8 | 0.85 | 1.0 | 1.0 | 1.0 |

触发实时重决策：

① 检测到Edge-B时延超标，立即激活备选节点Cloud-X；

② 重新计算Cloud-X总时延：80 ms+5 s=5.08 s≤15 s，满足要求。

切换路径至Cloud-X，任务中断时间＜100 ms，其中，边缘与云端的数据同步由算网大脑通过预缓存机制完成。

（4）主要结论

传统模式下急诊影像的处理单一依赖边缘端，端到端总时延包括传输时延、处理时延以及资源排队时长等，因此多任务并发下任务处理时长会平均大于60 s。而在此决策模式下，可动态实时感知云–边多个节点的资源情况，从而充分利用各节点算力，最终可将急诊影像诊断的端到端时延控制在10 s内，较传统模式，总时延缩短80%以上，考虑每个案例都是生命攸关，为急诊救治争取了宝贵的时间，因此具备极大社会及经济价值。

总之，在远程影像诊断中，算网大脑结合边缘计算的模式通过算网大脑的智能资源调度能力与边缘侧的本地化处理优势，相比传统医疗影像处理模式，在效率、可靠性、资源利用等方面实现了多维度升级。算网大脑与边缘计算协同架构优势见表4。

## 3　结束语

本文研究着重剖析算网大脑与边缘计算之间的智能协作模式，构建了融合边缘计算局部智能分析及实时反馈功能的整体架构，提出了实现资源动态调度的关键算法及实现流程。借助这一机制，算网大脑加快了数据处理速度和响应速率，还可高效分配资源并提升系统的整体安全系数。边缘设备有能力快速处理大量信息，此设计使算网大脑可以获取高品质的数据输入，从而增强了决策准确性和响应速度。后续研究将探索更先进和更高效的算法模型，以强化两者之间的协作深度，未来，通过将空间计算技术、元宇宙/XR以及多种具身智能技术（如无人机、机器人和自动驾驶技术），整合进终端AI应用，期望“脑边端一体化”的概念得到实际应用，从而在智慧城市、智能制造和智能生活等多领域实现更广泛的创新、落地与成长。在这种协同模式中，算网大脑和边缘计算将为应对复杂环境带来的挑战奠定稳固的基石，并推动计算网络时代向智能化的方向发展，助力社会在数字信息时代的下一次变革。

**表4　算网大脑与边缘计算协同架构优势**

| 对比维度 | 传统医疗影像 | 算网大脑与边缘计算相结合的远程影像诊断 | 核心优点体现 |
|---|---|---|---|
| 数据传输效率 | 原始影像全量上传（如CT/DICOM文件可达数百MB至数GB） | 边缘侧预处理（压缩、降噪、关键帧提取）+算网大脑动态分配带宽，仅传输有效数据（压缩率30%~70%） | 减少60%~90%带宽占用，避免网络拥堵 |
| 实时性响应 | 依赖云端集中处理，端到端诊断时延（含传输+分析）通常>5 min（偏远地区>30 min） | 边缘侧本地完成初筛（如病灶定位<100 ms）+算网大脑调度就近算力节点加速分析，端到端时延<1 min（急诊场景<10 s） | 诊断实时性提升80%以上，急诊响应速度提升5~30倍 |
| 算力资源利用率 | 云端算力固定配置，高峰时段易过载，低谷时段闲置率>40% | 动态调度边缘节点与云端算力（如将70%基层影像任务分配至边缘侧，30%复杂任务调度至云端），资源利用率提升至85%以上 | 算力资源利用率提升100%~150%，降低硬件闲置成本 |
| 网络可靠性 | 网络中断时无法处理数据，诊断流程停滞 | 算网大脑实时感知网络状态，自动切换边缘侧离线处理+多路径冗余传输，断网时本地缓存结果，网络恢复后同步，业务中断率<0.1% | 保障99.9%业务连续性，尤其适合偏远地区 |
| 数据安全性 | 全量数据上传云端，隐私泄露风险较高（如患者影像信息） | 敏感数据（如个人标识）边缘侧脱敏后再传输 | 降低80%以上数据泄露风险 |
| AI辅助效率 | 云端AI模型需等待全量数据，推理耗时较长 | 边缘侧轻量化AI模型先做初筛（如病灶定位），云端再精判 | 初筛效率提升3~5倍，减少云端计算压力 |

## 参考文献：


[1] 施巍松, 孙辉, 曹杰, 等. 边缘计算: 万物互联时代新型计算模型[J]. 计算机研究与发展, 2017, 54(5): 907-924.
Shi W S, Sun H, Cao J, et al. Edge computing-an emerging computing model for the Internet of everything era[J]. Journal of Computer Research and Development, 2017, 54(5): 907-924.

[2] 邱勤, 徐天妮, 张智杰, 等. 算力网络安全应用需求与关键技术研究[J]. 信息技术与标准化, 2022(11): 19-24, 33.
Qiu Q, Xu T N, Zhang Z J, et al. Research on security application requirements and key technologies of computing force network[J]. Information Technology & Standardization, 2022(11): 19-24, 33.

[3] 张宏科, 权伟, 刘康. 算力网络研究与探索[J]. 中兴通讯技术, 2023, 29(1): 1-5.
Zhang H K, Quan W, Liu K. Research and exploration of computing power network[J]. ZTE Technology Journal, 2023, 29(1): 1-5.

[4] Li X, Zhang Y, Chen W. A survey on computing power network and edge computing: architecture and challenges[J]. IEEE Transactions on Cloud Computing, 2023, 11(4): 567-582.

[5] 王琛, 赵鹏, 唐国华, 等. 算网大脑智能化水平分级方法研究[J]. 电信科学, 2024, 40(8): 149-161.
Wang C, Zhao P, Tang G H, et al. Research on intelligent level of the brain of computility network[J]. Telecommunications Science, 2024, 40(8): 149-161.

[6] Wang H, Chen Z. Intelligent resource orchestration in computing-network integrated brain for 6G networks[C]//Proceedings of the IEEE INFOCOM 2024. Piscataway: IEEE Press, 2024: 1-10.

[7] 陈星延, 张雪松, 谢志龙, 等. 面向“云–边–端”算力系统的计算和传输联合优化方法[J]. 计算机研究与发展, 2023, 60(4): 719-734.
Chen X Y, Zhang X S, Xie Z L, et al. A computing and transmission integrated optimization method for cloud-edge-end computing first system[J]. Journal of Computer Research and Development, 2023, 60(4): 719-734.

[8] 孙滔, 周铖, 段晓东, 等. 数字孪生网络(DTN): 概念、架构及关键技术[J]. 自动化学报, 2021, 47(3): 569-582.
Sun T, Zhou C, Duan X D, et al. Digital twin network(DTN): concepts, architecture, and key technologies[J]. Acta Automatica Sinica, 2021, 47(3): 569-582.

[9] 许斌, 赵云凯, 朱剑鸣, 等. 移动边缘计算不确定性任务持续卸载及资源分配方法[J]. 软件学报, 2024, 35(3): 1466-1484.
Xu B, Zhao Y K, Zhu J M, et al. Continuous offloading and resource allocation method of uncertain tasks in mobile edge computing[J]. Journal of Software, 2024, 35(3): 1466-1484.

[10] Xu R, Li Y. Low-latency computing power scheduling in edge networks via reinforcement learning[J]. IEEE Internet of Things Journal, 2023, 10(15): 13245-13258.

[11] 白文超, 卢先领. 边缘计算中动态服务器部署与任务卸载联合优化算法[J]. 计算机应用研究, 2025, 42(6): 1830-1837.
Bai W C, Lu X L. Joint optimization algorithm for dynamic server deployment and task offloading in edge computing[J]. Application Research of Computers, 2025, 42(6): 1830-1837.

[12] 王曰芬, 章成志, 张蓓蓓, 等. 数据清洗研究综述[J]. 现代图书情报技术, 2007(12): 50-56.
Wang Y F, Zhang C Z, Zhang B B, et al. A survey of data cleaning[J]. New Technology of Library and Information Service, 2007(12): 50-56.

[13] 杨世强, 杨江涛, 李卓, 等. 基于LSTM神经网络的人体动作识别[J]. 图学学报, 2021, 42(2): 174-181.
Yang S Q, Yang J T, Li Z, et al. Human action recognition based on LSTM neural network[J]. Journal of Graphics, 2021, 42(2): 174-181.

[14] Nallapati R, Zhai F F, Zhou B W. SummaRuNNer: a recurrent neural network based sequence model for extractive summarization of documents[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Menlo Park, CA: AAAI Press, 2017, 31(1).

[15] 张承, 刘少华, 程盟珂. 算力网络中的任务调度和协同优化策略分析[J]. 电子技术, 2024, 53(8): 108-109.
Zhang C, Liu S H, Cheng M K. Analysis of task scheduling and collaborative optimization strategies in computing power networks[J]. Electronic Technology, 2024, 53(8): 108-109.

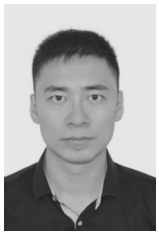
[16] 王聪, 马兴宇, 李旭, 等. 边缘计算在智慧交通系统中的应用研究[J]. 交通科技与管理, 2025, 6(6): 22-24.
Wang C, Ma X Y, Li X, et al. Research on the application of edge computing in intelligent transportation system[J]. Traffic Technology and Management, 2025, 6(6): 22-24.


[作者简介]

**安颖**（1992–），女，中国铁塔股份有限公司工程师，主要研究方向为边缘计算、算力网络。

**闫亚旗**（1988– ），男，中国铁塔股份有限公司高级工程师，主要研究方向为物联网、边缘计算、算力网络相关技术及产品创新。

**王东**（1994– ），男，现就职于中国铁塔股份有限公司，主要研究方向为边缘计算、算力网络和分布式计算。



**汪涛**（1988– ），男，中国铁塔股份有限公司技术经理，主要研究方向为人工智能算力网络架构设计、边缘计算云平台架构设计。

**刘申易**（1980– ），男，中国移动集团设计院有限公司咨询师，主要研究方向为算力网络工程设计。