

基于强化学习的车联网任务卸载策略

王哲¹ 邹经^{1*} 葛丽娜¹

WANG Zhe ZOU Jing GE Lina

摘要

随着边缘计算在车联网中的研究与应用, 计算任务能够从车辆卸载到路边边缘服务器以降低系统服务延迟。然而, 随着车辆的移动, 连续变化的环境可能导致传统的任务卸载策略失效或者算法复杂程度上升。如何预测车辆的移动轨迹并制定合理的任务卸载策略是亟待解决的关键问题, 传统的计算卸载方法无法在车联网中直接应用。基于多层次计算网络构建车联网任务卸载系统, 引入马尔科夫移动模型描述车辆移动轨迹, 并求解最优迁移路径问题; 为此提出基于受约束的马尔科夫模型的求解方法并设计了基于强化学习的演员网络原始-对偶深度确定性策略梯度(actor-network primal-dual deep deterministic policy gradient, ANPD-DDPG)算法以实现最优解的学习。在仿真实验中将所提方法与现有研究进行对比, 同时分析了 ANPD-DDPG 算法在收敛速度与系统延迟方面的特征, 在系统延时方面, 所提方法约降低 33%。

关键词

车联网; 边缘计算; 任务卸载; 强化学习

doi: 10.3969/j.issn.1672-9528.2025.06.018

0 引言

近年来, 随着车载算力与高速通信技术的进步, 车辆的智能化水平快速提升, 加速了车联网(internet of vehicle, IoV)的构建与发展^[1]。与车辆本地计算相比较, IoV 需要应对大量计算密集和延时敏感型任务^[2], 迫使传统的中央式云计算向移动边缘计算(mobile edge computing, MEC)迈进, 在更靠近道路的边缘设备中为车辆提供较本地更多资源的网络服务^[3-4]。IoV 的服务质量(quality of service, QoS)^[5]成为车辆行驶决策的核心要素, 因此需要考虑车辆的任务卸载。

文献[6]针对移动计算卸载进行了综述, 分析了传统启发式计算卸载策略和在线学习型计算卸载策略各自的优势及适用场景, 并指出由于车辆移动位置与速度的不确定性可能导致传统卸载策略失效, 需要考虑动态变化的车辆及环境。基于此, 文献[7]将移动任务卸载模拟为马尔可夫决策(markov decision process, MDP)过程, 根据边缘服务器状态、车辆移动轨迹和任务缓冲队列来设计任务卸载策略, 并以带宽约束下的系统延迟最小化为目标, 利用一维搜索算法求解最优卸载决策。与此同时, 文献[8]在能量收集的 MEC 系统中通过

研究卸载问题, 以系统延迟时间和任务失败成本作为优化目标, 提出了基于李雅普诺夫优化的动态在线卸载算法, 该算法负载度较低, 并且在卸载时考虑到中央处理器权重、车辆所处的位置和传输过程中的能量损耗, 无需考虑任务类型、无线信道状态和能量收集进程等信息, 提升了算法的在线性能和鲁棒性。文献[9]从资源分配角度考虑 MEC 系统卸载问题, 以最小化系统时延约束下的能耗为目标, 通过证明最优解存在阈值特征, 结合车辆移动轨迹和车辆的计算能耗对卸载优先级进行决策, 优先级高于阈值的任务进行完全卸载, 低于阈值的任务部分卸载。同样以能耗为优化目标, 文献[10]联合考虑车辆与边缘服务器的综合能耗, 对服务器选择、带宽资源分配、车辆轨迹预测和计算资源分配进行联合优化, 提出受特定应用约束的混合非线性规划问题(mixed integer nonlinear program, MINLP)并设计贪婪启发式算法求解。文献[11]基于车辆的移动性和异构性提出了车联网下 MEC 系统的计算卸载方案, 设计了深度强化学习自足管理框架, 通过建立 MDP 模型和深度强化学习算法求解。值得一提的是, 上述研究涉及的环境数据均是瞬时量, 连续的环境变化可能导致瞬时量参与到策略中的误差增大, 继而导致策略失效或算法复杂程度上升, 影响车联网的 QoS 需求。

本文针对以上不足, 提出一种基于强化学习的车联网任务卸载策略, 联合考虑任务卸载比例与边缘资源约束以实现自适应卸载。首先, 构建以车辆、路边边缘服务器和中央云服务器为系统的多层计算框架, 建立车联网系统模型; 然

1. 广西民族大学人工智能学院 广西南宁 530006

[基金项目] 广西自然科学基金资助项目(2024GXNSFAA010111); 广西民族大学“相思湖青年学者创新团队”项目(2023GXUNXSHQN02); 广西民族大学引进人才科研启动项目(2019KJQD17)

后,在系统计算资源与时延联合约束下,提出系统任务时延最小化问题,通过受约束的马尔科夫决策过程(constrained markov decision process, CMDP)进行建模求解最小化时延问题;继而设计了基于强化学习的演员网络原始-对偶深度确定性策略梯度(actor-network primal-dual deep deterministic policy gradient, ANPD-DDPG)算法以实现系统策略;最后,通过所设计的仿真实验验证本文所提出方法与所设计算法在降低系统成本与提升性能增益方面的优势。

1 系统模型和问题描述

1.1 系统模型

以车辆、路边边缘服务器(roadside edge server, RES)和中央云服务器为系统的多层计算框架如图1所示。

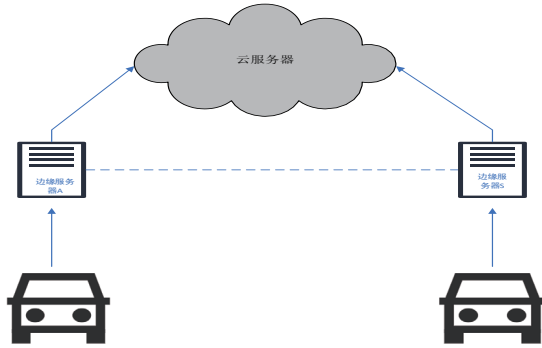


图1 多层次计算卸载系统网络

假定系统中边缘服务器数量为 $s \in \{1, 2, \dots, S\}$ 、车辆数量为 $u \in \{1, 2, \dots, U\}$ 。所有车辆都在 $t=1$ 时刻进入系统,并在时间 $t=T$ 时刻退出。

为描述车辆的移动特征,引入马尔可夫移动模型,二元指示器变量 $i_{u,s}^t$ 表示车辆 u 在时间 t 是否位于服务器 s 。将变量 $i_{u,s}^t = p[i_{u,s}^t]$, 以指示车辆 u 在时间 t 处于服务器 s 的概率; $p[i_{u,s}^t]$ 取值为 $[0,1]$ 中的连续变量。

1.2 通讯模型

整个通信周期 T 被划分为 t 个周期,并且车辆以一定的速度移动。任务一部分卸载到 RES。另一部分由车辆自身进行计算。从车辆 u 到 s 的传输速率为:

$$R_{s,u} = C \log_2(1 + r_{s,u}) \quad (1)$$

式中: C 表示的是系统的通信带宽; $r_{s,u}$ 表示的是 u 到 s 的信噪比,其公式可以被表示为:

$$r_{s,u} = \frac{p_u g_{s,u}(t)}{\sum_{n \neq u} p_u g_{s,n}(t) + b_u(t) P_{\text{NLOS}} + \sigma_s^2} \quad (2)$$

式中: σ_s^2 表示 s 所收到的加性高斯白噪声(additive white gaussian noise, AWGN),其分布为 $\mathcal{N}(0, \sigma_s^2)$; p_u 表示 u 的发送功率; $b_u(t) \in \{0,1\}$ 表示 u 到 s 是否存在障碍物^[12]; P_{NLOS} 表示 u 到 s 之间的非视距传输损失(non-line-of-sight

transmission loss, NLOSTL); $g_{s,u}(t)$ 表示 u 到 s 在周期 t 的信道增益。

1.3 计算模型

在系统模型中,系统的延时包含传输延时和计算延时。计算延迟包含本地延时和 RES 层延时。车辆将部分任务卸载到 RES 来执行的计算。 $\tau_{s,u}(t)$ 表示卸载任务卸载到 RES 的卸载比率; $\tau_u(t)$ 表示任务在车辆本地执行比例, $\tau_{s,u}(t) + \tau_u(t) = 1$ 。

(1) 本地计算延时:在周期 t 内车辆 u 的本地计算延迟为:

$$D_{\text{local},u}(t) = \tau_u(t) M_u(t) l f_u \quad (3)$$

式中: $M_u(t)$ 表示 u 任务的数据量大小; l 表示处理每单位字节 CPU 所需周期数; f_u 表示车辆 u 的算力。

(2) RES 层计算延时: RES 层的传输延时包含两部分,一部分是车辆 u 将计算任务上传到服务器 s 的延迟可以表示为:

$$D_{s,u}^{\text{transform}}(t) = \frac{\tau_{s,u}(t) M_s(t)}{R_{s,u}} \quad (4)$$

式中: $M_s(t)$ 是 s 所接受的任务量大小。

服务器将计算结果传输车辆的时延,可以表示为:

$$D_{s,u}^{\text{download}}(t) = \frac{M_u^{\text{processed}}(t)}{R_{s,u}} \quad (5)$$

式中: $M_u^{\text{processed}}(t)$ 表示 s 响应 u 的数据量大小。

RES 层的计算延迟可以表示为:

$$D_{s,u}^{\text{calculate}}(t) = \frac{\tau_{s,u}(t) M_s(t) s}{f_s} \quad (6)$$

式中: f_s 为服务器 s 的计算能力。

车辆 u 将任务卸载的服务器 s 并且接收到最终结果的任务延迟时间可以表示为:

$$D_{s,u}(t) = D_{s,u}^{\text{transform}}(t) + D_{s,u}^{\text{download}}(t) + D_{s,u}^{\text{calculate}}(t) \quad (7)$$

基于上述模型,优化问题总结为:在多层次计算网络系统场景中,为最小化任务延迟,联合优化车辆移动轨迹预测、任务卸载比例和任务迁移等问题。

最终该问题可以表示为:

$$\min_{\{\tau_{s,u}(t), \tau_u(t)\}} = \sum_{t=1}^T \sum_{u=1}^U \sum_{s=1}^S \{D_{\text{local},u}(t), D_{s,u}(t)\} \quad (8)$$

s.t. C1: $i_{u,s}^t \in [0,1], \forall u, s, t$

$$C2: \sum_{t=1}^T \sum_{s=1}^S \tau_{s,u}(t) M_s(t) + \sum_{t=1}^T \tau_u(t) M_u(t) = M$$

$$C3: b_u(t) \in \{0,1\}, \forall t, u$$

$$C4: \tau_{s,u}(t) + \tau_u(t) = 1, \tau_{s,u}(t), \tau_u(t) \in [0,1]$$

约束条件 C1 表示每个时间段车辆只能选择一个 RES 来进行任务计算和卸载。约束条件 C2 确保所有任务计算在整个通信时间段内完成。约束条件 C3 表示车辆 u 和服务器 s

之间无线电信道在一个时间段内的拥塞情况。约束条件 C4 表示约束了可进行任务卸载的比例边界。

2 ANPD-DDPG 算法

2.1 MDP 模型

2.1.1 状态空间

在所描述的场景下, RES 可以作为传感器, 用于获取环境的实时信息。在每个时间周期内, 系统状态由车辆位置信息、障碍物信息、任务信息、任务迁移信息和迁移成本信息组成。该时间段的系统状态可以表达为:

$$S_t = \{l_{u,s}^t, b_u(t), M_u(t), M_{s,u}(t)\} \quad (9)$$

式中: $l_{u,s}^t$ 表示车辆 u 在时间周期 t 时刻所处位置; $b_u(t)$ 表示车辆 u 和服务 s 之间的拥塞状况; $M_u(t)$ 表示在时间周期 t 车辆 u 本地计算任务的大小; $M_{s,u}(t)$ 则表示在时间周期 t 车辆 u 卸载到服务器 s 的任务大小。

2.1.2 行为空间

在时间周期 t 中的行动集可以表示为 A_t 。包括车辆的位置, 车辆向 RES 卸载任务的比例。每个行动空间中的行动描述为:

$$A_t = \{l_{u,s}^t, \tau_{s,u}(t)\} \quad (10)$$

式中: $\tau_{s,u}(t)$ 表示当前时刻任务卸载比例。

2.1.3 奖励函数

一个准确的奖励函数可以用来帮助卸载策略的制定, 而目标函数通常被用作机器学习过程中的奖励函数。目标函数的定义在式 (18) 中, 因此将奖励函数设定为:

$$R_t = R(S_t, A_t) = -D^{\text{system}}(t) \quad (11)$$

式中: $D^{\text{system}}(t)$ 是时间周期 t 时刻的系统总延时, 描述为:

$$D^{\text{system}}(t) = \sum_{k=1}^K \max\{D_{\text{local},u}(t), D_{s,u}(t)\} \quad (12)$$

2.2 受约束的马尔科夫决策过程

为满足策略 π 下的长期折现 (discounted reward, DR) [13] 奖励, 提出了一种 CMDP 过程。在 CMDP 中, 对长期折现成本的约束被增加到 MDP 中, 成本函数 $x_1, x_2, \dots, x_n$ 被添加到传统的 MDP 中。每个 x_n 是从转移元组到成本的映射。CMDP 的目标是选择策略 π 以最大化式 (13) 的长期奖励, 同时满足 $x_n(\pi) \leq u_n$ 的约束, $n \in \{1, 2, \dots, N\}$, u_n 是相应的限制。策略方式表示为:

$$\pi^* = \arg \max R(\pi) \quad (13)$$

$$\text{s.t. } x_n(\pi) \leq u_n, \quad \forall n \in [1, 2, \dots, N]$$

2.3 ANPD-DDPG 算法

ANPD-DDPG 从在线网络复制到目标网络, 不同于 DQN, 后者直接将 Q 在线网络的参数复制到 Q 目标网络。

ANPD-DDPG 选择软更新, 这意味着每次只更新一小部分参数。可以分别表达为:

$$\theta^* \leftarrow \tau \theta_c + (1 - \tau) \theta \quad (14)$$

$$\omega_r^* \leftarrow \tau \omega_r + (1 - \tau) \omega_r \quad (15)$$

$$\omega_c^* \leftarrow \tau \omega_c + (1 - \tau) \omega_c \quad (16)$$

式中: τ 通常表示为小值的更新系数。同时, 在学习过程中增加随机性以增加学习的覆盖范围, 则最终与环境交互的动作表达式为:

$$A_t = \pi_\theta(S_t) \quad (17)$$

为最小化损失函数, 在评论家网络中使用优化器来更新 Q 值。与 DQN 类似, 奖励评论家网络的损失函数可以表示为均方差的形式:

$$J(\omega_r) = \frac{1}{m} \sum_t (y_t - Q_r(\Phi(S_t), A_t, \omega_r))^2 \quad (18)$$

类似的成本评论家网络损失函数:

$$J(\omega_c) = \frac{1}{m} \sum_t (y_t - Q_c(\Phi(S_t), A_t, \omega_c))^2 \quad (19)$$

ANPD-DDPG 执行确定性策略, 因此演员网络的损失函数可以表示为:

$$\nabla_{\lambda}(\theta, \lambda) = \frac{1}{m} \sum_t \nabla_{\pi} [Q_c(S_t, A_t, \omega_c)]_{s=S_t, A=\pi(A_t)} - \lambda Q_c(S_t, A_t, \omega_c)_{s=S_t, A=\pi(A_t)} \quad (20)$$

如对于相同的状态输出了两个不同的动作 A_1 和 A_2 , 则来自评论家在线网络的两个反馈 Q 值分别为 Q_1 和 Q_2 。 $Q_1 > Q_2$ 意味着采取动作 A_1 可以获得比 A_2 更多的奖励。根据策略梯度, 应该增加 A_1 的发生概率并减少 A_2 的发生概率, 这意味着演员网络希望尽可能获得较大的 Q 值。因此, 代理获得的反馈 Q 值越小, 其损失就越大。因此, 损失函数的计算方式为:

$$J(\theta) = -\frac{1}{m} \sum_t (Q_r(S_t, A_t, \omega_r) - \lambda Q_c(S_t, A_t, \omega_c)) \quad (21)$$

ANPD-DDPG 算法的流程如算法 1 所示。

算法 1 ANPD-DDPG 算法

- (1) 输入: 折现系数 ω ; 学习率 α_1, α_2 ; 软更新因子 τ ; 训练周期 T ;
- (2) 初始化: 初始目标 R_0 和最大能耗 C_p ;
- (3) 生成初始任务流 M_u ;
- (4) 预处理 M_u , 并且获取处理后的任务流 M_u^* ;
- (5) 初始化一个容量为 M 的经验池 D ;
- (6) 初始化主网络评论家参数 θ 和演员家的网络参数 ω_r ;
- (7) 初始化目标网络的评论家参数 θ_1 和演员家网络参数 ω_c ;
- (8) 初始化通信链路;
- (9) for $t = 1, 2, \dots, T$ do;
- (10) 重置并且从 CECN 函数中获取初始状态 S_0, A_0 和 R_0 ;

- (11) for $h = 1, 2, \dots, H$ do;
- (12) 状态归一化 $S_t \rightarrow S_t^*$;
- (13) 获取行动集 $A_t = u(S_t^* | \theta)$;
- (14) 执行行动集 A_t 并且获取奖励函数 R_t 和下一个状态 S_{t+1} ;
- (15) if 当前经验池 D 容量小于 M 将当前 S_t 、 A_t 、 R_t 、 S_{t+1} 的快照存入经验池中;
- (16))else;
- (17) 将 S_t 、 A_t 、 R_t 、 S_{t+1} 提出之前存在经验池 D 中的快照;
- (18) 设置 $y_t = R_t + \gamma Q_t(S_{t+1} + A_{t+1}, \omega_r^*)$ 、 $z_t = R_t + \gamma Q_t(S_{t+1} + A_{t+1}, \omega_c^*)$;
- (19) 根据式 (15) (16), 更新最小损失函数的参数 ω_r 和 ω_c ;
- (20) 通过式 (14) 更新 θ ;
- (21) 基于式 (17) 更新演员家策略;
- (22) 更新对偶梯度的对偶变量:
- $$\nabla_{\lambda}(\theta, \lambda) = \frac{1}{m} \sum_i^m \nabla_{\lambda} [Q_i(S_n, A_n, \omega_r)]_{\lambda=\lambda_i} - u$$
- (23) 软更新: ω_r^* 、 ω_c^* 、 θ^* 、 θ_1^* ;
- (24) end if;
- (25) end for;
- (26) 输出: 基于式 (21) 获得演员家网络。

3 仿真实验

通过进行一系列仿真实验, 验证所提出的多层次网络架构下的 ANPD-DDPG 算法的算法性能。

3.1 环境设置

在仿真实验系统中, 设置了正方形区域的多层次计算架构, 有一个均匀随机分布区域和一台云服务器, 该区域均匀随机分布边缘服务。实验中的所有车辆只与 RES 通信, 所有 RES 都连接到云服务器, 如图 1 所示。车辆的初始位置从均匀分布中随机抽取。

表 1 显示了用于模拟的马尔可夫模型对车辆位置预测的准确性。正如数据展示的那样, 时间周期越大的位置预测准确性越低。同时服务器越多会使车辆位置的预测变得更加困难。

表 1 移动模型预测距离车辆 U 最近的服务器平均概率

时间周期	1	5	10	20
5 台服务器	0.87	0.62	0.49	0.26
10 台服务器	0.80	0.49	0.39	0.14
20 台服务器	0.71	0.40	0.25	0.13

3.2 算法收敛性评估

图 2 描述了提出的 ANPD-DDPG 算法的收敛性, 可以看出在 0~200 次迭代系统延时开始下降, 并且在 200 次迭代左右后达到收敛稳定。刚开始的时延波动是因为拉格朗日因子

的初始值与最优差距较大, 算法中动作的选择使得时延急速下降, 随着迭代次数的增多, 系统逐渐逼近最优时延性能。

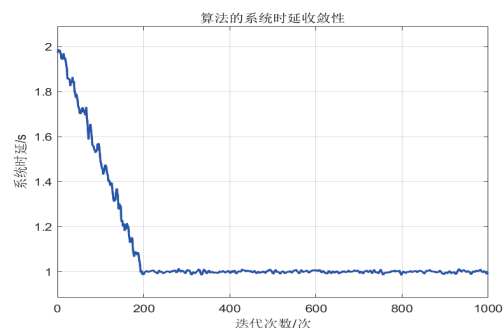


图 2 算法收敛性

3.3 比较不同的迁移方法

将 ANPD-DDPG 算法和 3 种不同的迁移方法进行比较。朴素方法 (native) [14] 在不进行迁移的情况下选择最接近的可用服务器, 以最小化成本, 这类似于 SDN/NFV 放置 [15] 优化; 短视 (myopic) [16] 方法在每个时间步将任务迁移到最近的可行服务器, 这类似于反应性迁移框架 [17]; 云方法 [18] 将需要迁移的任务都放置到云服务器, 而不考虑资源约束, 类似于中央处理的做法。通过比较验证了车辆轨迹预测的价值。

如图 3 所示, 比较了在资源有限和资源充足的情况下不同迁移方案实现的任务迁移成本。与有限资源相比, 所有计划生成方法在充足资源下都能降低成本, 因为每个卸载任务都可以实现低延迟的放置。ANPD-DDPG 算法方法由于任务迁移次数较少而明显优于另外 3 种算法。特别是在有限资源下, ANPD-DDPG 算法相对于朴素方法节省了 33% 的成本, 相对于短视方法节省了 18% 的成本。在有限资源下, ANPD-DDPG 算法优于云方式, 而在充足资源下, 由于卸载任务都可以充分卸载到各个 RES 上, 其性能是等效的。

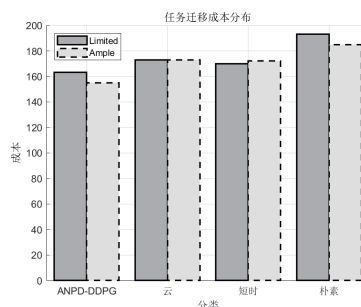


图 3 不同资源下迁移成本

3.4 算法的效率

图 4 展示了 ANPD-DDPG 算法与其他基线算法之间的性能结果。迭代总次数为 1000, 在增加迭代次数时, DQN、DDPG 和 ANPD-DDPG 都能够收敛。结果表明, AN-

PD-DDPG 效用明显优于另外两个算法, 因 3 种算法均包含一个策略网络和一个目标网络。双网络结构可以通过隔离训练数据之间的相关性来找到最佳的动作策略。与 DQN 算法不同, ANPD-DDPG 算法可以输出连续的动作, 在选择动作空间方面更有优势。结果显示, ANPD-DDPG 算法收敛后的结果明显优于两者的延迟结果。

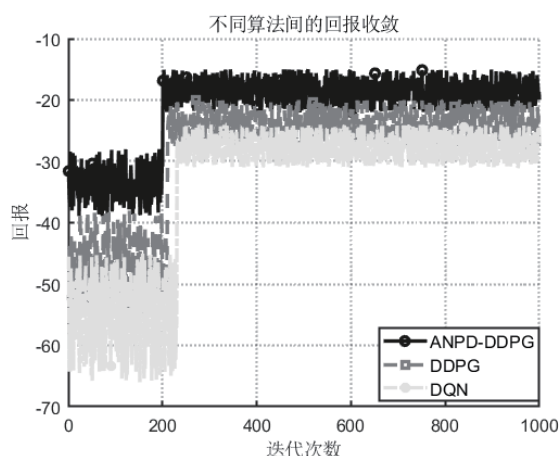


图 4 不同算法的回报收敛率

4 总结

文中提出了一种应用于任务卸载和迁移的多层次计算框架, 通过马尔科夫移动模型来进行车辆轨迹预测, 解决了传统机器学习进行轨迹预测时大量源于车辆的历史数据造成的带宽和时延压力。并提出 ANPD-DDPG 算法来制定合理的迁移计划, 通过分析 ANPD-DDPG 算法与其他 3 种迁移方案相比, ANPD-DDPG 在降低任务迁移成本和提升系统响应时间方面都有不错的表现。然而, 文中考虑的是一个 RES 为一辆车进行服务器的场景。未来, 可以研究如何同时为多个边缘节点上的多辆车通过服务器来进行任务卸载决策。

参考文献:

[1] DAI Y Y, XU D, MAHARJAN S, et al. Joint load balancing and offloading in vehicular edge computing and networks[J]. IEEE internet of things journal, 2018, 6(3): 4377-4387.

[2] TAMANI N, BRIK B, LAGRAA N, et al. On link stability metric and fuzzy quantification for service selection in mobile vehicular cloud[J]. IEEE transactions on intelligent transportation systems, 2019, 21(5): 2050-2062.

[3] YANG C, LIU Y, CHEN X, et al. Efficient mobility-aware task offloading for vehicular edge computing networks[J]. IEEE access, 2019, 7: 26652-26664.

[4] SORKHOH I, EBRAHIMI D, ATALLAH R, et al. Workload scheduling in vehicular networks with edge cloud

capabilities[J]. IEEE transactions on vehicular technology, 2019, 68(9): 8472-8486.

[5] WANG L, JIAO L, LI J, et al. MOERA: mobility-agnostic online resource allocation for edge computing[J]. IEEE transactions on mobile computing, 2018, 18(8): 1843-1856.

[6] 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 [J]. 计算机学报, 2021, 44(12): 2406-2430.

[7] LIU J, MAO Y Y, ZHANG J, et al. Delay-optimal computation task scheduling for mobile-edge computing systems[C]//2016 IEEE International Symposium on Information Theory (ISIT). Piscataway: IEEE, 2016: 1451-1455.

[8] YOU C S, HUANG K B. Multiuser resource allocation for mobile edge computation offloading[C/OL]//2016 IEEE Global Communications Conference (GLOBECOM). Piscataway: IEEE, 2016[2024-10-11]. <https://ieeexplore.ieee.org/document/7842016>. DOI: 10.1109/GLOCOM.2016.7842016.

[9] ZHAO P T, TIAN H, QIN C, et al. Energy-saving offloading by jointly allocating radio and computational resources for mobile edge computing[J]. IEEE access, 2017, 5: 11255-11268.

[10] WANG S Q, URGANKAR R, ZAFER M, et al. Dynamic service migration in mobile edge computing based on Markov decision process[J]. IEEE/ACM transactions on networking, 2019, 27(3): 1272-1288.

[11] WANG J, HU J, MIN G Y, et al. Online service migration in edge computing with incomplete information: a deep recurrent actor-critic method[J]. IEEE transactions on mobile computing, 2023, 22(1): 6663-6675.

[12] WANG S G, GUO Y, ZHANG N, et al. Delay-aware microservice coordination in mobile edge computing: a reinforcement learning approach[J]. IEEE transactions on mobile computing, 2021, 20(3): 939-951.

[13] LABRIJI I, MENEGHELLO F, CECCHINATO D, et al. Mobility aware and dynamic migration of MEC services for the internet of vehicles[J]. IEEE transactions on network and service management, 2021, 18(1): 570-584.

[14] OU-YANG T, ZHOU Z, CHEN X. Follow me at the edge: mobility-aware dynamic service placement for mobile edge computing[J]. IEEE journal on selected areas in communications, 2018, 36(10): 2333-2345.

[15] GEBRIE H, FAROOQ H, IMRAN A. What machine learning predictor performs best for mobility prediction in cellular networks?[C/OL]//2019 IEEE International Conference on Communications Workshops (ICC Workshops). Piscataway: IEEE, 2019[2024-06-23]. <https://ieeexplore.ieee.org/abstract/document/8756972>. DOI: 10.1109/ICCW.2019.87569720.

基于反向传播算法的网络安全态势感知方法

杜秋阳¹
DU Qiuyang

摘要

高级持续性网络威胁具有的复杂、隐蔽和持久等特征,网络安全态势影响因素无法有效量化,使得网络正向传播无法深入提取高层次特征,导致网络安全态势感知精度下降现象。对此,文章提出了基于反向传播算法的网络安全态势感知方法。首先,采用卡尔曼滤波器和自回归滤波技术处理时间序列数据;然后,考虑网络漏洞严重性和网络受攻击的影响程度作为态势影响因素进行量化分析;最后,将态势影响因素作为卷积神经网络模型的输入信息,结合反向传播算法构建残差卷积神经网络模型,将态势影响因素作为输入信息 ResNet 通过多层卷积操作,从复杂的网络流量数据中提取高层次特征,识别攻击行为的细微模式,并通过反向传播优化,ResNet 不断调整模型参数,实现网络安全态势感知。实验结果显示:采用反向传播算法的网络安全态势感知技术,有效提升了检测精度,减少了误差,并优化了模型的性能表现。

关键词

态势感知;攻击检测;网络安全;网络防护;反向传播算法

doi: 10.3969/j.issn.1672-9528.2025.06.019

0 引言

随着物联网和数字化的迅速发展,网络规模持续扩大,拓扑结构愈发复杂,网络安全管理的挑战性和难度也随之加剧。导致难以及时探测恶意行为,也无法全面掌控网络安全态势。相比之下,网络安全态势感知技术致力于捕捉并分析那些能影响网络状态变化的安全因素,进行理解、展示,并预测短期内的安全发展趋势。该技术通过辨识、理解和预测网络中的各种活动,为网络安全管理提供了更为全面和深入的视角。

文献[1]运用 Marchenko-Pastur 模型和单环定律,对网络安全态势的特征值在随机矩阵中的分步实施了谱分析,揭示了网络在稳定与异常状态下的变换规律,从而实现了电网调度系统的网络安全态势感知。然而, Marchenko-Pastur 模型和单环定律适用于高维随机矩阵的谱分析,在低维数据或非随机数据中会出现失效现象。文献[2]则借助网络流量探测器和入侵检测探测器监测流量,分别捕获了流量的基础特征和恶意活动特征,利用属性提炼技术从各探测器中提取核心属性数据作为输入,判断了各种攻击类型,进而把握了整个网络的安全态势。然而,依赖探测器捕获的基础特征和恶

1. 湖北民族大学智能科学与工程学院 湖北恩施 445000

[16] AL-SHAERY A M, AHMED S G, ALJASSMI H, et al. Open dataset for predicting pilgrim activities for crowd management during hajj using wearable sensors[J]. IEEE access, 2024,12:72828 - 72846.

[17] KIM T, SATHYANARAYANA S D, CHEN S Q, et al. MoDEMS: optimizing edge computing migrations for user mobility[C/OL]//IEEE INFOCOM 2022 - IEEE Conference on Computer Communications. Piscataway: IEEE, 2022[2024-05-25]. <https://ieeexplore.ieee.org/document/9796680>. DOI:10.1109/INFOCOM48880.2022.9796680.

[18] SUN X, ANSARI N. EdgeIoT: mobile edge computing for

the internet of things[J]. IEEE communications magazine, 2016, 54(12): 22-29.

【作者简介】

王哲(1994—),男,河南南阳人,博士,副教授、硕士生导师,研究方向:计算机网络、智能通信与智能算法。

邹经(1994—),通信作者(email: 645163640@qq.com),男,江西宜春人,硕士研究生,研究方向:计算机网络。

葛丽娜(1969—),女,广西环江人,博士,教授、硕士生导师,研究方向:计算机网络与信息安全。

(收稿日期: 2024-12-24 修回日期: 2025-04-29)