计算机科学
COMPUTER SCIENCE

基于深度强化学习的安全感知服务功能链部署方法

朱子怡, 张建辉, 曾俊杰, 张洪源

引用本文

朱子怡, 张建辉, 曾俊杰, 张洪源. 基于深度强化学习的安全感知服务功能链部署方法[J]. 计算机科学, 2025, 52(10): 404-411.

ZHU Ziyi, ZHANG Jianhui, ZENG Junjieand ZHANG Hongyuan. Security-aware Service Function Chain Deployment Method Based on Deep ReinforcementLearning [J]. Computer Science, 2025, 52(10): 404-411.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.240800015

基于深度强化学习的安全感知服务功能链部署方法

朱子怡[1] 张建辉[1,2] 曾俊杰[1] 张洪源[1]
1 郑州大学网络空间安全学院 郑州 450000
2 嵩山实验室 郑州 450000
(zhuziyi@gs.zzu.edu.cn)

摘 要 服务功能链作为提升网络资源利用率的关键技术,其结合深度强化学习能够实现灵活且安全的部署。然而,如何有效地部署具有安全需求的服务功能链,同时最大化长期平均收益是服务功能链面临的一个重要的挑战。为此,提出了一种基于深度强化学习的安全感知服务功能链部署方法(DRL-SASFCD)。首先,提出了一种安全感知机制,用于评估物理网络节点的可信度,并引入安全需求指数感知SFC对安全性的需求。其次,利用图注意力网络和序列到序列模型提取底层物理网络信息以及服务功能链请求序列信息的相关特征,并依据这些特征生成服务功能链部署策略。最后,采用近端策略的优化方法来优化策略和训练网络参数,通过限制新旧策略之间的更新幅度,避免策略更新过程中的剧烈波动,从而提高安全策略优化效率。仿真实验结果表明,DRL-SASFCD在考虑服务功能链部署安全需求的同时,与现有方法相比,在部署接受率、长期平均收益以及长期平均收益成本比3个方面均有所提高。

关键词:服务功能链;虚拟网络功能;深度强化学习;安全;部署收益
中图分类号 TP393

Security-aware Service Function Chain Deployment Method Based on Deep Reinforcement Learning

ZHU Ziyi[1],ZHANG Jianhui[1,2],ZENG Junjie[1] and ZHANG Hongyuan[1]
1 College of Cyberspace Security,Zhengzhou University,Zhengzhou 450000,China
2 Songshan Laboratory,Zhengzhou 450000,China

Abstract As a key technology to improve the utilization of network resources,service function chain combined with deep reinforcement learning makes it possible to achieve flexible and secure deployment. However,how to effectively deploy service function chains with security requirements while maximizing long-term average revenue is an important challenge it faces. This paper proposes a deployment method for security-aware service function chain based on deep reinforcement learning(DRL-SASFCD). Firstly,a security-aware mechanism is proposed to evaluate the credibility of physical network nodes,and a security requirement index is introduced to perceive the security requirements of SFC. Secondly,this method utilizes graph attention network and sequence to sequence models to extract relevant features of underlying physical network information and service function chain request sequence information. It generates service function chain deployment strategies based on these features. Finally,the proximal policy optimization method is adopted to optimize the policy and training network parameters. By limiting the update amplitude between the new and old policies,the drastic fluctuations during the policy update process are avoided,thereby improving the efficiency of security policy optimization. The simulation results show that DRL-SASFCD can improve the deployment acceptance rate,long-term average revenue and long-term average revenue-cost ratio compared with the existing methods while considering the security requirements of service function chain deployment.

Keywords Service function chain,Virtual network function,Deep reinforcement learning,Security,Deployment revenue

1 引言

随着云计算、物联网和5G等技术的快速发展,网络服务的需求变得日益复杂和多样化。传统的网络架构已经无法有效满足多样化的需求,因此网络功能虚拟化[1-2](Network Function Virtualization,NFV)技术应运而生,它提供可扩展

到稿日期:2024-08-02 返修日期:2024-11-10
基金项目:国家重点研发计划(2022YFB2901304);河南省重大科技专项(221100210900)
This work was supported by the National Key Research and Development Program of China(2022YFB2901304) and Major Science and Technology Program of Henan Province(221100210900).
通信作者:张建辉(ndsczjh@163.com)

性、灵活性和敏捷性的服务。与传统网络的中间件不同，NFV 技术将网络功能从特定硬件中解耦，可以使用虚拟网络功能（Virtual Network Function，VNF）灵活发放网络服务并按需为用户提供服务[3]，VNF 是将防火墙、负载均衡器、入侵检测系统等传统网络功能以软件形式实现的网络功能。服务功能链（Service Function Chain，SFC）是 NFV 中的典型应用，通常由一个有序的 VNF 序列组成，如图 1 所示。SFC 部署[4]则根据特定的业务需求和约束条件，将 VNF 动态地部署在网络中，按顺序连接并分配网络资源，从而为用户提供所需的网络服务。

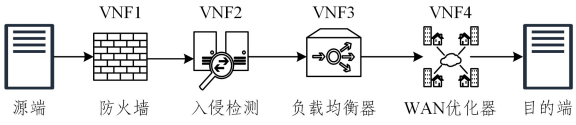

图 1 服务功能链的示例

Fig. 1 Example of service function chain

NFV 为网络架构带来了灵活性的同时也带来了一些安全威胁[5]。由于 NFV 依赖于软件，因此它比传统的硬件网络功能更容易受到攻击或被操纵，若承载 VNF 的服务器被潜在的攻击者攻破，则部署在该服务器上的虚拟机将面临安全威胁，攻击者可以发起旁路攻击和拒绝访问攻击来破坏整个网络。除了节点，攻击者也会通过重放和中间人攻击来影响物理链路，因此在服务功能链部署中考虑安全需求是很有必要的。

目前，已有许多国内外研究人员对 SFC 部署问题进行研究，并且已有研究证明 SFC 部署问题属于 NP-hard 问题[6]。文献[7]将 SFC 部署问题形式化为整数线性规划（Integer Linear Programming，ILP）模型，提出一种基于优先级驱动的加权算法来实现 SFC 的最优部署。然而，传统的 ILP 方法会产生较高的计算复杂度，针对这一问题，文献[8-9]设计了启发式算法来实现 SFC 部署。文献[8]提出一种基于动态规划的启发式算法，利用 Viterbi 算法在多阶段图中找到近似最优路径，并降低了 SFC 部署成本，从而提高收益。文献[9]提出一种两阶段启发式算法，以最大化收益为目标来求解 SFC 的动态部署问题。传统的启发式方法能在较短的时间内找到一个可行解，但它过于依赖先验知识，且容易陷入局部最优解。

深度强化学习[10]（Deep Reinforcement Learning，DRL）是一种结合了深度学习和强化学习的算法。它利用深度神经网络来表示强化学习中的策略或价值函数，能够在高维状态空间中进行有效的学习和决策。目前已有研究[11-14]表明了其在解决 SFC 部署问题时的有效性。文献[15]提出一种基于改进深度强化学习的 VNF 部署优化算法，以同时优化部署成本和端到端时延为目标实现了 SFC 部署。文献[16]通过改进双深度 Q 网络提高学习效率，进而实现 SFC 的有效部署。文献[17]采用深度确定性策略梯度（Deep Deterministic Policy Gradient，DDPG）算法，以平均映射率为评价标准对 SFC 请求进行部署，然而 DDPG 对 SFC 部署问题的离散动作空间求解会导致策略更新不稳定。文献[18]使用图卷积网络（Graph Convolutional Network，GCN）提取网络特征，采用异步优势演员评论家（Asynchronous Advantage Actor-Critic，A3C）网络训练深度神经网络，从而实现 SFC 收益最大化，但 GCN 无法通过动态调整权重来反映不同节点的重要性。然而，上述研究均没有考虑 SFC 部署中的安全需求。对此，文献[19]提出一种动态安全服务链组合机制，通过整数线性规划和启发式算法解决了安全服务链组合中的优化问题。文献[20]在 SFC 部署中考虑了物理网络的节点和链路安全性，并证明增强用户的虚拟网络安全性不会降低接受率和对资源的有效利用。文献[21]考虑物理链路和虚拟链路设置安全级别，提出了一种基于路径的信任感知服务功能链嵌入算法。文献[22]引入安全服务的需求，通过设置安全等级并提出一种基于自注意力机制的双重深度 Q 网络算法来保证不同 VNF 的安全部署。

然而，以上研究仅考虑到了安全需求或 SFC 部署收益，无法应用于同时要求安全需求和部署收益的场景。为解决以上问题，本文以考虑安全需求的同时最大化长期平均收益为目标，提出一种基于深度强化学习的安全感知 SFC 部署方法。本文的贡献如下：

1）针对 SFC 部署中的安全威胁，设置安全约束并提出一种安全评估机制来评估物理网络节点的可信度，同时引入安全需求指数以实现感知 SFC 安全性。

2）将 SFC 部署问题建模为马尔可夫决策过程（Markov Decision Process，MDP），提出一种基于深度强化学习的安全感知 SFC 部署方法（DRL-SASFCD），通过图注意力网络（Graph Attention Network，GAT）来考虑链路特征，从而更全面地提取物理特征，并通过近端策略优化（Proximal Policy Optimization，PPO）算法，以最大化长期平均收益为目标优化部署策略。

3）设计仿真实验并与现有算法进行对比，实验结果表明，所提方法能够在考虑安全需求的同时，提高 SFC 部署接受率和长期平均收益。

2 问题建模

2.1 系统建模

1）物理网络。将物理网络抽象化为一个无向图，其中 $N_p=\{n_i \mid i=1,2,\cdots,|N_p|\}$ 表示物理网络中的节点集合，$L_p=\{l_i \mid i=1,2,\cdots,|L_p|\}$ 表示物理链路的集合，$|N_p|$ 和 $|L_p|$ 表示物理节点和物理链路的数量，$C_p^{\mathrm{cpu}}(n_i)$，$C_p^{\mathrm{mem}}(n_i)$，$Sl(n_i)$ 和 $Sdl(n_i)$ 分别表示物理节点 n_i 可用的 CPU 资源量、内存资源量、安全级别和安全需求级别，$B_p(l_i)$ 和 $Sl(l_i)$ 分别表示物理链路 l_i 的带宽资源和安全级别。

2）SFC 请求。每个 SFC 请求由以特定顺序连接的多个 VNFs 组成。为了更好地反映不同网络功能的安全需求，VNF 根据其安全需求被划分为普通 VNF 和安全 VNF 两类。普通 VNF（如负载均衡器、流量监控器等）主要关注性能需求，这类 VNF 通常对系统的计算资源（如 CPU、内存等）和带宽有较高的要求，但对安全的需求相对较低。在 SFC 的实际部署中，普通 VNF 会优先选择资源丰富的物理节点和链路进行映射，以确保部署收益最大化。安全 VNF（即防火墙，入侵检测和深度数据包检测等）则侧重于安全需求，这类 VNF 的主要目标是保证网络的安全性、数据的完整性以及防御外部攻击，对物理节点和链路的安全等级有较高的要求，因此在

部署过程中，安全 VNF 会优先选择具有高安全等级的物理节点和链路，以确保网络的安全性。本文将 SFC 请求分为安全敏感性 SFC 和资源敏感性 SFC 两类。安全敏感性 SFC 主要由安全性 VNF 组成，这类 SFC 对安全等级的需求较高，但对计算资源的需求相对较低。资源敏感性 SFC 主要由普通 VNF 组成，这类 SFC 对计算资源的需求较高，而对安全等级的需求较低。

用$\mathbb{S}$表示 SFC 请求的集合，将第 g 条 SFC 请求 f_g 建模为一个有向图 $G_v^g=\{V_g,E_g\}$，其中 $V_g=\{v_j\mid j=1,2,\cdots,|V_g|\}$ 和 $E_g=\{e_j\mid j=1,2,\cdots,|E_g|\}$ 分别表示 SFC 请求 f_g 的 VNF 的集合和虚拟链路集合。对于 SFC 请求 f_g，VNF v_i 和 VNF v_j 之间的虚拟链路表示为 e_{ij}，$C_g^{\mathrm{cpu}}(v_j)$ 和 $C_g^{\mathrm{mem}}(v_j)$ 表示 VNF v_j 对物理节点的 CPU 资源量和内存资源量的需求，$Sl(v_j)$ 和 $Sdl(v_j)$ 分别表示虚拟节点的安全级别和安全需求级别，$B_g(e_j)$ 和 $Sdl(e_j)$ 分别表示虚拟链路 e_j 的带宽资源需求和安全需求级别。

2.2 约束条件

每个 VNF 只能部署在一个物理节点上。

$$\sum_{n_i\in N_p}x^g(n_i,v_j)=1,\forall v_j\in V_g \tag{1}$$

其中，二进制 $x^g(n_i,v_j)=1$ 表示 SFC 请求 f_g 中的一个 VNF v_j 映射到物理网络中的节点 n_i 上，否则为 0。

1)流量守恒约束。SFC 流量守恒约束确保流入节点的流量之和等于从节点流出的流量之和。

$$\sum_{v_j\in V_g}y^g(l_{ij},e_{ij})-\sum_{v_j\in V_g}y^g(l_{ji},e_{ij})=x^g(n_i,v_i)-x^g(n_i,v_j) \tag{2}$$

2)容量约束。对于 SFC 请求 f_g 来说，CPU 和内存的消耗不能超过服务器节点上的可用资源。约束条件为：

$$\sum_{f_g\in\mathbb{S}}\sum_{v_j\in V_g}C_g^{\mathrm{cpu}}(v_j)\cdot x^g(n_i,v_j)\leqslant C_p^{\mathrm{cpu}}(n_j),\forall n_i\in N_p \tag{3}$$

$$\sum_{f_g\in\mathbb{S}}\sum_{v_j\in V_g}C_g^{\mathrm{mem}}(v_j)\cdot x^g(n_i,v_j)\leqslant C_p^{\mathrm{mem}}(n_j),\forall n_i\in N_p \tag{4}$$

3)带宽约束。映射到一条链路上的所有 SFC 请求的流量需求之和不超过该链路的带宽容量。二进制 $y^g(l_i,e_j)$ 表示虚拟链路 e_j 部署到物理链路 l_i 成功或失败，若部署成功，则 $y^g(l_i,e_j)$ 等于 1，反之为 0。本文将链路带宽约束表述为：

$$\sum_{f_g\in\mathbb{S}}\sum_{e_j\in E_g}B_g(e_j)\cdot y^g(l_i,e_j)\leqslant B_p(l_i),\forall l_i\in L_p \tag{5}$$

4)安全约束。本文设置 3 个安全约束：

$$x^g(n_i,v_j)\cdot Sdl(v_j)\leqslant Sl(n_i),n_i\in N_p,v_j\in V_g \tag{6}$$

$$x^g(n_i,v_j)\cdot Sdl(n_i)\leqslant Sl(v_j),n_i\in N_p,v_j\in V_g \tag{7}$$

$$y^g(l_i,e_j)\cdot Sdl(l_i)\leqslant Sl(e_j),l_i\in L_p,e_j\in E_g \tag{8}$$

式(6)保证 VNF 部署在安全级别比 VNF 的安全需求级别高的物理节点上。式(7)保证部署在物理节点上的 VNF 的安全级别不低于物理节点的安全需求级别。式(8)保证部署虚拟链路的物理链路的安全级别不低于虚拟链路的安全需求级别。

2.3 安全感知机制

本文研究的安全感知能力指在部署 SFC 的过程中，实时评估网络的安全状态，并根据请求是否为安全敏感性 SFC 以及是否满足 VNF 和虚拟链路的安全需求，做出合适的部署决策，从而降低对物理网络和虚拟资源施加的风险，并减少对额外策略的需求。安全感知从物理网络安全性和 SFC 请求安全性两方面来体现。

首先，引入物理节点安全评估机制。物理节点 n_i 的可信度 $Tru(n_i)$ 基于节点自身的安全级别 $Sl(n_i)$ 和参与 SFC 部署的次数来评估，计算式如下：

$$Tru(n_i)=\beta\cdot Sl(n_i)+\gamma\cdot\frac{cou(n_i)}{Succ(G^v)} \tag{9}$$

其中，$Succ(G^v)$ 表示服务功能链部署成功的数量，β 和 γ 是固定设置的评估权重，$cou(n_i)$ 表示节点 n_i 参与服务功能链部署的次数。$cou(n_i)$ 值越高，说明参与部署的频率越高，节点面临各种网络环境的机会就越多，其系统资源、安全等级匹配能力等都会逐渐优化，意外错误和安全等级不匹配的可能性减少，从而越可信。因此，参与部署频率较高的节点通常会被网络部署策略更优先地选择，形成一个正反馈机制，从而承载更高安全级别的任务并保证长期的网络安全和稳定。

其次，提出安全需求指数 SSI 来感知请求是否为安全敏感性 SFC，安全需求指数 SSI 的计算式如下：

$$SSI=\frac{1}{|V_g|}\sum_{j=1}^{V_g}Sdl(v_j) \tag{10}$$

当 SSI 大于阈值 ϑ 时，认为该请求为安全敏感性 SFC，否则为资源敏感性 SFC。SFC f_g 部署的安全性 $TS(f_g)$ 的计算式如下：

$$TS(f_g)=\sum_{j=1}^{|V_g|}\sum_{i=1}^{|N_p|}x^g(n_i,v_j)\cdot Tru(n_i)+\sum_{u=1}^{|E_g|}\sum_{m=1}^{|L_p|}y^g(l_m,e_u)\cdot SL(l_m) \tag{11}$$

将 $TS(f_g)$ 作为奖励函数的组成部分，当安全敏感性 SFC 被部署到高安全等级的节点或链路上时，能够通过更高的安全奖励值获得正向反馈，实时更新物理节点的信任度 $Tru(n_i)$，促使未来继续选择这些高安全性资源。由此，算法在部署过程中不仅注重长期平均收益，还能始终保持对安全需求的敏感性。

2.4 优化目标

本文的目标是在 SFC 部署时，保证 SFC 成功部署并考虑安全需求的同时，最大化长期平均收益。成功部署一个 SFC 请求 f_g 获得的收益 $Rev(f_g)$ 的计算式如下：

$$Rev(f_g)=t_r\cdot\sum_{j=1}^{|V_g|}C_g(v_j)\cdot Sdl(v_j)+t_r\cdot\sum_{i=1}^{|E_g|}BW(e_i)\cdot Sdl(e_i) \tag{12}$$

其中，$C_g(v_j)=C_g^{\mathrm{cpu}}(v_j)+C_g^{\mathrm{mem}}(v_j)$，$t_r$ 表示 SFC 请求的时间。部署一条 SFC 请求 f_g 的成本如式(13)所示：

$$Cost(f_g)=t_r\cdot\sum_{u=1}^{|V_g|}\sum_{m=1}^{|N_p|}x^g(n_m,v_u)\cdot C_g(v_u)\cdot Tru(n_m)+t_r\cdot\sum_{i=1}^{|E_g|}\sum_{j=1}^{|L_p|}y^g(l_j,e_i)\cdot BW(e_i)\cdot SL(l_i) \tag{13}$$

长期平均收益即成功部署的 SFC 请求获得的利润与总服务时间 T 的比率，计算方法如式(14)所示：

$$A_{Rev_t}(f_g)=\lim_{T\to\infty}\frac{\int_{t=0}^{T}Rev(f_g)}{T} \tag{14}$$

则优化目标表示为：

$$\begin{aligned}&\max\ A_{Rev_t}(f_g)\\&\text{s.t. 式(1)}-\text{式(8)}\end{aligned} \tag{15}$$

3 算法设计

DRL-SASFCD是一种基于深度强化学习的SFC部署方法，该算法结合GAT和seq2seq架构，生成SFC请求的部署策略，并通过PPO算法使学习代理与环境进行迭代交互，从而实现SFC的最优部署。DRL-SASFCD框架如图2所示。

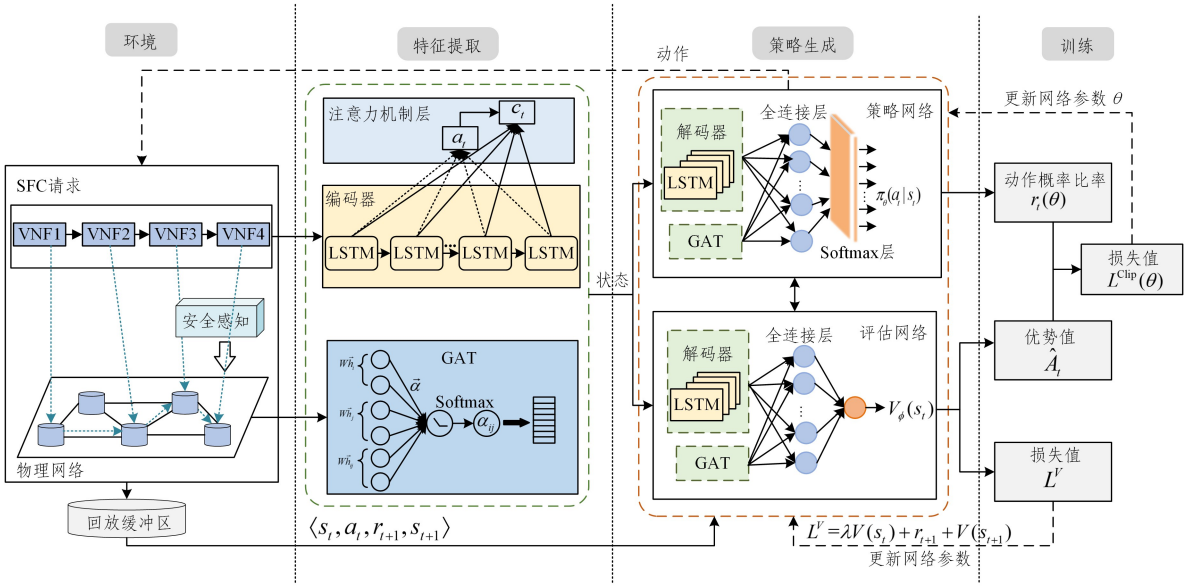

图2 DRL-SASFCD框架

Fig. 2 Architecture of DRL-SASFCD

3.1 马尔可夫决策过程

为处理由SFC部署引起的实时网络状态变化，将SFC部署问题建模为MDP。在DRL模型中，MDP可以用一个三元组$(S_t, \mathbb{A}_t, R_t)$来描述，分别代表状态空间、动作空间和奖励。

1）状态空间。状态空间S_t包括物理网络的特征s_t^p和当前SFC请求的特征s_t^v，$s_t^p=\{\boldsymbol{C}_p^{\text{cpu}}, \boldsymbol{C}_p^{\text{mem}}, \boldsymbol{Tru}, BW_{\text{sum}}\}$，$\boldsymbol{C}_p^{\text{cpu}}$表示物理节点的可用计算资源向量，$\boldsymbol{C}_p^{\text{mem}}$表示物理节点的可用存储资源向量，$\boldsymbol{Tru}$表示物理节点的信任度向量，$BW_{\text{sum}}$表示连接在物理节点上的链路带宽之和。$s_t^v=\{C_g^{\text{cpu}}, C_g^{\text{mem}}, Sdl, B_g\}$，该特征包括VNF v_j计算资源需求量C_g^{cpu}、存储资源需求量C_g^{mem}、安全需求级别Sdl以及虚拟链路e_j的带宽需求量B_g。综上所述，状态空间S_t可以表示为$S_t=\{s_t^p, s_t^v\}$。

2）动作空间。动作空间$\mathbb{A}_t$表示DRL代理在每个时间步t下为每个VNF选择一个物理节点进行部署的决策，当最后一个VNF被放置或者当前VNF的放置违反了约束，动作$a_t=\bar{\zeta}$结束当前SFC请求部署。因此动作空间$\mathbb{A}_t$表示为$\mathbb{A}_t=\{\bar{\zeta}\}\cup\{n_1, n_2, \cdots, n_{|N_p|}\}$。

3）奖励。本文的奖励以长期平均收益为优化目标，奖励函数如式(16)所示：

$$R_t=\begin{cases}\omega\cdot TS(f_g)+\varphi\cdot\dfrac{Rev(f_g)}{Cost(f_g)}, & \text{SFC部署成功}\\ -\tau\cdot\left(\omega\cdot TS(f_g)+\varphi\cdot\dfrac{Rev(f_g)}{Cost(f_g)}\right), & \text{其他}\end{cases}\tag{16}$$

其中，τ为奖励系数，ω和φ分别是安全权重和收益权重，初始值均为0.5。当不同SFC部署时，奖励函数通过Sigmoid函数动态调整权重ω和φ，权重ω的调整公式如下：

$$\omega=\frac{1}{(1+\mathrm{e}^{-k(SSI-\vartheta)})}\tag{17}$$

其中，k是控制曲线陡峭度的参数。k越大，曲线越陡，权重的变化越敏感，本文设$k=2$，$\varphi=1-\omega$。

当安全VNF引入到MDP中时，状态空间S_t需要更加细致地表达物理节点的安全等级、链路的安全性等。这意味着MDP的状态描述不仅包含物理节点的计算和存储资源，还包括这些节点和链路的安全性信息。当部署安全敏感性SFC时，通过计算安全需求指数SSI和阈值ϑ，动态提高安全权重ω，使奖励的计算不仅依赖于收益和成本等指标，更依赖于部署的安全性。如果安全VNF被部署到低安全性的物理节点或链路上，则奖励会降低。

3.2 特征提取

许多图神经网络模型假设相邻节点对中心节点表示的重要程度相同，或在聚集过程中预先确定。本文采用GAT[23]提取物理网络特征。GAT是一种提供注意力机制以有效地传播来自前一层的节点信息的神经网络，它以不同重要程度的相邻节点作为关注分数来更新节点表示。

GAT的输入是各个节点的特征组合$h=\{\vec{h}_1, \vec{h}_2, \cdots, \vec{h}_N\}$，$\vec{h}_i\in\mathbb{R}^F$，其中$N$是节点的数量，$F$是特征维度。此外，本文在GAT输入时考虑到边特征$\vec{h}_{ij}\in\mathbb{R}^{F_e}$。GAT首先对每个节点应用一个线性变换权重矩阵$\boldsymbol{\Theta}\in\mathbb{R}^{F'\times F}$，然后使用单层前馈神经网络计算连接节点之间的注意力权重$\alpha\in\mathbb{R}^{2F'}$，$F_e$和$F'$分别是边特征维度和线性变换后节点的特征维度。假设节点n_j是节点n_i的邻居，那么注意力系数e_{ij}就表示节点n_i在聚合邻居节点n_j的特征时赋予的权重，计算式如式(18)所示：

$$e_{ij}=LeakReLU(\vec{a}^{\mathrm{T}}\cdot\boldsymbol{\Theta}[\vec{h}_i\,\|\,\vec{h}_j\,\|\,\vec{h}_{ij}])\tag{18}$$

其中，T表示矩阵转置，$\|$表示连接，$\vec{a}$是权重向量。通过式(19)对注意力系数进行归一化，GAT将节点的特征与其相邻节点的特征进行聚合，节点n_i更新后的特征表示$\vec{h}_i'$的计算式如式(20)所示：

$$\alpha_{ij}=\frac{\exp(e_{ij})}{\sum_{k\in\mathcal{N}_i}\exp(e_{ij})} \tag{19}$$

$$\vec{h'}_i=\sigma\left(\sum_{j\in\mathcal{N}_i}\alpha_{ij}\Theta(\vec{h}_j\parallel\vec{h}_{ij})\right) \tag{20}$$

其中，$\mathcal{N}_i$ 是节点 n_i 的所有邻居节点，σ 是非线性激活函数，得到节点的新特征 $h'=\{\vec{h'}_1,\vec{h'}_2,\cdots,\vec{h'}_N\}$，$\vec{h'}_i\in\mathbb{R}^{F'}$。

3.3 策略生成及训练

SFC 请求的到达和处理具有明显的时序依赖性，历史请求和当前网络状态会影响未来的服务请求和网络状态。本文采用 seq2seq 模型构建策略网络，该模型是一个编码器-解码器神经网络结构，它擅长处理序列数据，能够捕捉到 SFC 请求的时序模式和依赖关系，从而为策略网络提供更准确的上下文信息。seq2seq 模型负责学习全局最优观察序列到动作序列的映射，并预测新的观察序列对应的动作序列。编码器的输入是当前服务功能链请求的状态 s_t^v，采用多层长短期记忆网络(Long Short-Term Memory，LSTM)将输入序列编码为上下文向量 $\boldsymbol{c}_t$。LSTM 通过式(21)输出当前时间隙 t 的隐藏状态 h_t。

$$h_t=LSTM(s_t^v,h_{t-1}) \tag{21}$$

为了充分利用历史请求信息，通过注意力机制层使编码器能够动态聚合过去的多个时间隙的隐藏状态，通过式(22)生成上下文向量 $\boldsymbol{c}_t$。

$$\boldsymbol{c}_t=\sum_{i=1}^{t}\rho_i h_t \tag{22}$$

其中，ρ_i 表示注意力权重，通过式(23)进行计算。

$$\rho_i=\frac{\exp(\mu(h_i,h_t))}{\sum_{j=1}^{t}\exp(\mu(h_j,h_t))} \tag{23}$$

其中，$\mu(h_i,h_t)=h_i^T h_t$ 表示每个隐藏状态对当前时刻 t 的相关性。

解码器根据编码器生成的上下文向量 $\boldsymbol{c}_t$，结合代理的当前策略生成一组动作概率，用于指导 SFC 的部署。解码器同样由多层 LSTM 单元组成，输出每个节点的值并使用 softmax 函数确定节点选择概率分布。为了优化策略并利用采样经验来训练网络参数，采用 PPO 算法更新策略网络的参数，策略网络和评估网络分别输出状态 s_t 下选择不同动作的概率和价值 $V_\phi(s_t)$。在状态 s_t 处，从策略 $\pi_\theta(a_t\mid s_t)$ 采样动作 a_t，执行动作后获得奖励 r_t，更新状态为 s_{t+1}，收集一组数据 (s_t,a_t,r_t,s_{t+1})。

为了提高训练的稳定性和效率，PPO 算法引入一个 clip 机制。PPO 算法的目标函数为：

$$L^{\mathrm{CLIP}}(\theta)=\hat{\mathbb{E}}_t\left[\min(r_t(\theta)\hat{A}_t,\mathrm{clip}(r_t(\theta),1-\epsilon,1+\epsilon)\hat{A}_t)\right] \tag{24}$$

其中，clip 函数用于限制 π_θ 的范围，$r_t(\theta)$ 是采样策略与目标策略的比值，其计算式为：

$$r_t(\theta)=\frac{\pi_\theta(a_t\mid s_t)}{\pi_{\theta_{\mathrm{old}}}(a_t\mid s_t)} \tag{25}$$

其中，$\pi_{\theta_{\mathrm{old}}}(a_t\mid s_t)$ 固定为样本数据，不断更新 $\pi_\theta(a_t\mid s_t)$ 来训练网络。使用优势函数来评价一个动作相较于平均水平的好坏，优势函数的估计值 $\hat{A}_t$ 为：

$$\hat{A}_t=-V_\phi(s_t)+r_t+\gamma r_{t+1}+\cdots+\gamma^{T-t+1}r_{T-1}+\gamma^{T-t}V_\phi(s_T) \tag{26}$$

使用梯度上升方法来更新策略参数 θ，策略网络参数优化的更新规则为：

$$\theta'\leftarrow\theta+\alpha\nabla_\theta^{\mathrm{LCLIP}}(\theta) \tag{27}$$

其中，α 是训练的学习率。经过 t_{update} 次的样本学习后，目标策略网络将网络的参数分配给采样策略网络，即 $\theta_{\mathrm{old}}\leftarrow\theta$。使用新的策略参数重复上述过程，到达迭代次数时停止，具体步骤如算法 1 所示。

算法 1 DRL-SASFCD 方法

输入：物理网络拓扑 $G_p=(N_p,L_p)$，SFC 请求 $G_g^g=\{V_g,E_g\}$

输出：SFC 部署方案

1. 初始化：初始化网络参数 θ
2. for epoch < epochNum do
3. for SFC $=f_1,f_2,\cdots$, do
4. 通过编码器对 SFC 请求的状态进行编码
5. for VNF $=v_1,v_2,\cdots$, do
GAT 提取物理网络特征；
6. 解码器生成动作概率分布，从 $\pi_\theta(a_t\mid s_t)$ 采样 a_t；
7. 执行动作 a_t；
8. if v_1 放置成功 then
9. 映射链路；
10. if 链路映射未成功 then
11. 撤销先前所有操作；
12. end if
13. else
14. 撤销先前所有操作；
15. end if
16. 得到 s_{t+1} 和 r_t，并将其保存在缓冲区
17. 根据式(26)计算优势估计 $\hat{A}_t$
18. for $t_{\mathrm{update}}=1,2,\cdots$, do
19. 由式(24)计算策略目标函数 $L^{\mathrm{CLIP}}(\theta)$
20. 根据式(27)更新策略网络参数 θ
21. end for
22. $\theta_{\mathrm{old}}\leftarrow\theta$；
23. $s_t\leftarrow s_{t+1}$；
24. end for
25. end for
26. ++ epoch；
27. end while

3.4 时间复杂度分析

本文对 DRL-SASFCD 算法的时间复杂度进行分析。假设物理网络的节点数量为 N_p，特征维度为 F，SFC 请求的总数为 M，N_v 和 L_v 分别表示成功部署的 VNF 数量和虚拟链路数量。对于每一个到达的 SFC 请求，计算特征矩阵的时间复杂度为 $O(N_pF)$。通过 seq2seq 模型生成部署策略，这一过程的时间复杂度为 $O(N_v)$。VNF 部署后，虚拟链路的部署时间复杂度为 $O(L_v)$，每个 SFC 成功部署的时间复杂度为 $O(N_pF+N_v+L_v)$。因此，所有 SFC 成功部署的时间复杂度为 $O(M(N_pF+N_v+L_v))$。

4 实验环境及结果

4.1 实验环境

为了验证 DRL-SASFCD 方法对安全服务功能链部署结

果的性能，在一台 Intel(R) Core(TM) i5-10500 CPU 和 8 GB 内存的主机上基于 Python 进行仿真实验，整个模型结构采用 TensorFlow 搭建。使用 Waxman-Salam 模型[24]随机生成一个具有 100 个节点和 400 条链路的物理网络拓扑。在每一轮次中，随机生成 1 000 条 SFC 请求，每条 SFC 请求由普通 VNF 和安全 VNF 组成，普通 VNF 和安全 VNF 数量均服从[1,5]上的均匀分布，安全需求级别分别服从[1,3]和[4,6]上的均匀分布，具体网络仿真参数如表 1 所列。本文通过 SFC 仿真参数计算出 SSI 期望值为 3.5；阈值设置为比期望值高的值，因此 $\vartheta=4$；算法注重长期的部署收益而不是只关注短期的收益，因此奖励系数 τ 取 0.9；裁剪参数 $\varepsilon=0.2$，防止学习策略更新过度；学习率取损失熵较低情况下的 0.001 能够保证模型在学习过程中的稳定性，同时具备足够的探索性，不至于过快陷入局部最优。

表 1 网络仿真参数

Table 1 Simulation parameters of network

仿真参数	值
物理节点的计算资源、内存资源	[50,100]
物理链路的带宽资源	[50,100]
物理节点的安全级别、安全需求级别	[1,3]
物理链路的安全级别	[1,3]
SFC 请求到达过程	泊松分布 $\lambda=0.04$
每条 SFC 请求中普通 VNF 数量	[1,5]
每条 SFC 请求中安全 VNF 数量	[1,5]
每个普通 VNF 需求的计算资源、内存资源	[2,30]
每个安全 VNF 需求的计算资源、内存资源	[2,15]
每个普通 VNF 的安全需求级别	[1,3]
每个安全 VNF 的安全需求级别	[4,6]
每条虚拟链路需求的带宽资源	[2,30]
每条虚拟链路的安全需求级别	[1,3]
阈值 ϑ	4
策略网络学习率	0.001
评估网络学习率	0.001
PPO 的裁剪参数 ε	0.2
奖励系数	0.9

4.2 对比算法

为了评估 DRL-SASFCD 的有效性和性能，本文将 SFC 部署接受率、SFC 部署成本和 SFC 部署长期平均收益作为评价指标，对所提算法进行仿真实验，将其与首次适应算法(First-Fit，FF)，A3C-GCN[18]，SA-VNE[25]和 NP-SVNE[26]这 4 种方法进行对比。其中 FF 是一种传统启发式算法，通过顺序遍历物理节点，寻找第一个满足需求的节点进行部署，直至完成整条 SFC 的部署。A3C-GCN 是一种基于深度强化学习的方法，通过 GCN 来探索物理网络特征并使用 A3C 算法并行训练策略网络。SA-VNE 考虑到节点的安全等级和安全需求等级，使用信息熵 TOPSIS 方法评估底层节点的重要性，从而为每个虚拟节点选择最合适的底层节点。NP-SVNE 是一种启发式算法，它考虑的是 SFC 请求的安全等级而不是节点的安全需求等级，在节点映射阶段选择节点映射函数最大的物理节点进行部署。

4.3 评估指标

本文通过 SFC 请求接受率、长期平均收益和长期平均收益成本比来衡量算法的性能。

1)SFC 请求接受率：该指标表示 SFC 请求部署的成功率，定义为：

$$Accept_R=\frac{\sum_{t=0}^{T}SFCR^{\text{success}}}{\sum_{t=0}^{T}SFC^{\text{total}}} \tag{28}$$

其中，$SFCR^{\text{success}}$表示成功部署 SFC 请求的数量，SFC^{total}表示 SFC 请求的总数量。

2)长期平均收益：长期平均收益越高，代表资源利用率越高，计算方法如式(14)所示。

3)长期平均收益成本比：长期平均收入成本比值越高，代表资源利用率越高，计算式如式(29)所示：

$$Rev2Cost=\lim_{T\to\infty}\frac{\int_{t=0}^{T}Rev(f_g)}{\int_{t=0}^{T}Cost(f_g)} \tag{29}$$

4.4 结果分析

实验进行了 140 轮的训练，对 1 000 个 SFC 请求进行处理，损失的变化趋势如图 3 所示。随着训练阶段的进行，损失逐渐减少并趋于稳定，表明本文提出的 DRL-SASFCD 对虚拟环境具有良好的适应能力。

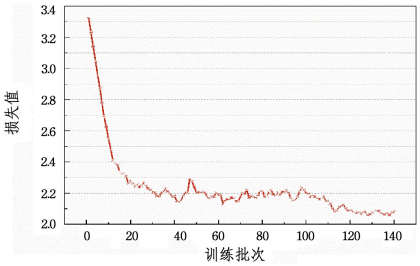

图 3 训练损失值

Fig. 3 Loss value of training

图 4 给出了 DRL-SASFCD 算法与 4 种对比方法在 SFC 请求接受率方面的对比。图中 5 种方法的 SFC 请求部署接受率随着时间的增加呈下降趋势，这是因为物理网络资源有限，随着越来越多的 SFC 请求到达，SFC 部署成功后会消耗物理网络的资源，进而导致 SFC 请求接受率下降，在约16 000个时间单元后趋于稳定。所提方法部署接受率约为 84.2%，相较于 FF，A3C-GCN，SA-VNE 和 NP-SVNE 分别提高了 18.5%，14.8%，6.1%和 17.5%，FF 方法不具备全局优化能力，在 SFC 到达早期时，FF 的部署接受率高，当请求数量增加时，接受率会迅速下降。DRL-SASFCD 评估物理节点的可信度，当 SFC 部署时，安全需求级别低的 VNF 会优先部署到低安全级别的物理节点，从而降低安全需求高的 VNF 部署失败的概率，提高 SFC 部署接受率。

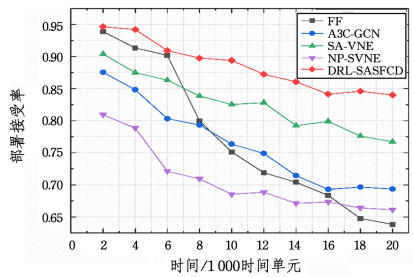

图 4 SFC 部署接受率

Fig. 4 Acceptance rate of SFC deployments

图 5 给出了本文方法与 4 种对比方法在 SFC 部署长期平均收益方面的对比。在前期阶段，5 种方法都能保持较高的长期平均收益，这是因为物理网络资源丰富，SFC 部署成功率高，但随着时间的增加，SFC 接受率下降，长期平均收益也显著下降。与 GCN-A3C 相比，本文提出的 DRL-SASFC 的长期平均收益提高了 5.04%，这是因为 DRL-SASFC 在采用 GAT 时考虑到链路特征并根据邻居节点的重要性自适应地调整特征聚合，从而得到更全面的特征属性，同时引入剪切函数 clip 对策略网络的更新加以限制，所以对网络状态变化的适应性更好。而 SA-VNE 没有考虑物理节点和链路的安全级别对 SFC 部署的影响，FF 和 NP-SVNE 易造成物理资源碎片化，因此 DRL-SASFC 在部署长期平均收益方面相较于 FF，SA-VNE 和 NP-SVNE 分别提高了 15.59%，5.82% 和 13.66%。

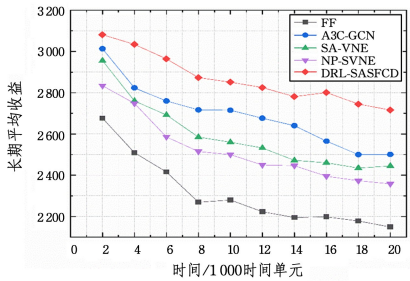

图 5 SFC 部署长期平均收益

Fig. 5 Long-term average revenue of SFC deployments

图 6 给出了本文方法与 4 种对比方法在 SFC 部署长期平均收益成本比方面的对比。随着时间的增加，5 种方法的平均收益成本比均有所下降，这是因为物理网络的资源随着越来越多的 SFC 请求到来而逐渐减少。FF 算法是贪心算法，会导致资源碎片化，因此平均收益成本比较低。DRL-SASFCD 的平均收益成本比高于 GCN-A3C，这是因为 DRL-SASFCD 能更好地学习节点的关系，且 PPO 通过重复利用样本，提高了样本利用效率，在同样的数据量下能更快地优化策略。

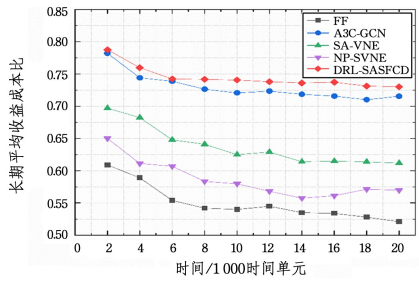

图 6 SFC 部署长期平均收益成本比

Fig. 6 Long-term average revenue-cost ratio of SFC deployments

为进一步对比不同方法在实际运行中的时间，本文以运行时间为指标进行了实验。结果如图 7 所示，5 种算法的运行时间随着 SFC 请求数量的增多均呈上升趋势。FF 在前期部署较快，随着越来越多请求的到来，时间增速变大，其性能变得最差。A3C-GCN 与 DRL-SASFCD 相比运行时间更短，但后者有较高的部署接受率且是以较低时间水平部署。DRL-SASFCD 的运行时间较 FF，SA-VNE 和 NP-SVNE 分别提升了 38.7%，19.1% 和 28.7%，说明本文方法在大量 SFC 请求部署的求解时间上是可接受的。

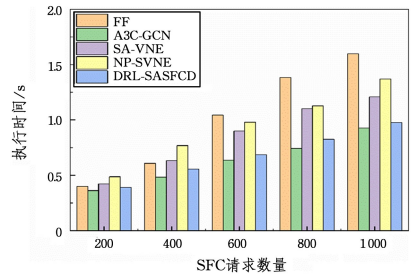

图 7 SFC 部署运行时间

Fig. 7 Running time of SFC deployment

结束语 本文研究了具有安全需求的 SFC 部署问题，以最大化部署平均收益为目标，将问题建模为 MDP，并提出一种基于深度强化学习的安全感知 SFC 部署方法。此外，引入安全评估机制来计算物理网络节点的可信度。并在此基础上通过 GAT 和 seq2seq 模型提取物理网络的特征和 SFC 请求的有序信息，生成 SFC 部署策略，采用 PPO 对策略进行优化，以最大化长期平均收益。实验结果表明，所提算法能够在考虑安全需求的同时，提高部署接受率和长期平均收益。本文方法在 SFC 部署时未全面考虑分支式 SFC 以及 VNF 配置信息的机密性和不可篡改性。在未来工作中，将研究分支式 SFC 请求是如何基于端到端加密技术实现可信部署的。

参 考 文 献

[1] YANG S, LI F, TRAJANOVSKI S, et al. Recent advances of resource allocation in network function virtualization[J]. IEEE Transactions on Parallel and Distributed Systems, 2020, 32(2): 295-314.

[2] MATENCIO E A, WANG Q, CALERO J M A. SliceNetVSwitch: Definition, design and implementation of 5G multi-tenant network slicing in software data paths[J]. IEEE Transactions on Network and Service Management, 2020, 17(4): 2212-2225.

[3] TANG L, WANG K, ZHANG Y, et al. Service function chain anomaly detection based on distributed generative adversarial network in network slicing scenario[J] Journal of Electronics & Information Technology, 2023, 45(1): 262-271.

[4] RUI L L, CHEN S Y, WANG S Y, et al. SFC Orchestration Method for Edge Cloud and Central Cloud Collaboration: QoS and Energy Consumption Joint Optimization Combined With Reputation Assessment[J]. IEEE Transactions on Parallel and Distributed Systems, 2023, 34(10): 2735-2748.

[5] COELHO R W, LEONARDO E J, MARTIMIANO L A F, et al. A survey of the characteristics of SDN, NFV and information security in IoT and 5G networks[J]. Revista Brasileira de Computação Aplicada, 2023, 15(3): 96-105.

[6] LI B, CHENG B, LIU X, et al. Joint Resource Optimization and Delay-Aware Virtual Network Function Migration in Data Center Networks [J]. IEEE Transactions on Network and Service Management, 2021, 18(3): 2960-2974.

[7] ZHANG Q X, XIAO Y K, LIU F, et al. Joint Optimization of Chain Placement and Request Scheduling for Network Function

Virtualization[C]// 2017 IEEE 37th International Conference on Distributed Computing Systems(ICDCS). IEEE,2017:731-741.

[8] BARI F,CHOWDHURY S R,AHMED R,et al. Orchestrating virtualized network functions[J]. IEEE Transactions on Network and Service Management,2016,13(4):725-739.

[9] HUANG H,JIANG J,YANG Y K,et al. Online Service Function Chain Orchestration Method for Profit Maximization[J]. Computer Science,2023,50(6):66-73.

[10] LIU H T,DING S D,WANG S Y,et al. Multi-objective optimization service function chain placement algorithm based on reinforcement learning[J]. Journal of Network and Systems Management,2022,30(4):58-83.

[11] HUANG Z W,ZHONG W J,LI D G,et al. Delay Constrained SFC Orchestration for Edge Intelligence-Enabled IIoT: A DRL Approach[J]. Journal of Network and Systems Management,2023,31(3):53-79.

[12] XU H S,FAN G L,SUN L B,et al. Dynamic SFC placement scheme with parallelized SFCs and reuse of initialized VNFs: An A3C-based DRL approach[J]. Journal of King Saud University-Computer and Information Sciences,2023,35(6):101577.

[13] JEONG E D,YOO J H,HONG J W K. SFC Consolidation: Energy-aware SFC Management using Deep Reinforcement Learning[C]// 2024 IEEE Network Operations and Management Symposium. IEEE,2024:1-5.

[14] RAN J,WANG W K,HU H F. Dynamic Service Function Chain Deployment and Readjustment Method Based on Deep Reinforcement Learning[J]. Sensors,2023,23(6):3054.

[15] TANG L,HE L Q,LIAN Q Y,et al. Virtual Network Function Placement Optimization Algorithm Based on Improve Deep Reinforcement Learning[J]. Journal of Electronics & Information Technology,2021,43(6):1724-1732.

[16] LIU D H,WEI D E,XUAN H J,et al. Improved double deep Q network algorithm for service function chain deployment[J]. Journal of Xidian University,2024,51(1):52-59.

[17] HUANG W W,LI S,WANG S N,et al. An Improved Adaptive Service Function Chain Mapping Method Based on Deep Reinforcement Learning[J]. Electronics,2023,12(6):1307-1325.

[18] WANG T F,FAN Q L,LI X H,et al. Drl-sfcp: Adaptive service function chains placement with deep reinforcement learning [C]// IEEE International Conference on Communications(ICC 2021). 2021:1-6.

[19] LIU Y C,LU Y,QIAO W X,et al. A dynamic composition mechanism of security service chaining oriented to SDN/NFV-enabled networks[J]. IEEE Access,2018,6:53918-53929.

[20] ALALUNA M,FERROLHO L,FIGUEIRA J R,et al. Secure multi-cloud virtual network embedding[C]// Computer Communications. 2020:252-265.

[21] TORKZABAN N,BARAS J S. Trust-aware service function chain embedding: A path-based approach[C]// 2020 IEEE Conference on Network Function Virtualization and Software Defined Networks(NFV-SDN). IEEE,2020:31-36.

[22] ZHANG P Y,WANG C,JIANG C X,et al. Resource management and security scheme of ICPSs and IoT based on VNE algorithm[J]. IEEE Internet of Things Journal,2021,9(22):22071-22080.

[23] VELICKOVIC P,CUCURULL G,CASANOVA A,et al. Graph attention networks[C]// IRLR 2018. 2018.

[24] YAN Z X,GE J G,WU Y L, et al. Automatic virtual network embedding: A deep reinforcement learning approach with graph convolutional networks[J]. IEEE Journal on Selected Areas in Communications,2020,38(6):1040-1057.

[25] ZHANG P Y,LI H S,NI Y J,et al. Security aware virtual network embedding algorithm using information entropy TOPSIS [J]. Journal of Network and Systems Management, 2020, 28(1):35-57.

[26] LIU X B,WANG B H,LIU S Q,et al. Heuristic algorithm for secure virtual network embedding [J]. Systems Engineering and Electronic,2018,40(3):676-681.

ZHU Ziyi, born in 2001, postgraduate. Her main research interests include cyberspace security and service function chain orchestration.

ZHANG Jianhui, born in 1977, Ph.D, associate researcher, master supervisor. His main research interests include new network architecture, network routing technology, network data analysis and security control.

(责任编辑:何杨)