动态对抗不确定性下的网络系统防御深度强化学习

Ashutosh Dutta[1], **Samrat Chatterjee**[1],**Arnab Bhattacharya**[1],**Mahantesh Halappanavar**[1]太平洋西北国家实验室，巴特尔大道902号，华盛顿州里奇兰，邮编99354，美国

arXiv:2302.01595v1 [cs.LG] 3 Feb 2023

摘要

在现实世界中开发自主网络系统防御策略和行动建议颇具挑战，其中包括刻画系统状态不确定性和攻防动态。我们提出一种数据驱动的深度强化学习（DRL）框架，用于学习主动式、具备上下文感知能力的防御对策。这些对策能够动态适应不断演变的对抗行为，同时最大限度地减少网络系统运行损失。我们针对不同类型、技能水平和持久性各异的对手，制定了包含多种防护态势的动态防御优化问题。我们开发了定制仿真环境，并设计实验，系统地评估四种无模型深度强化学习算法应对真实多阶段攻击序列的性能。结果表明，深度强化学习算法能够在多阶段攻击特征和系统不确定性下实现主动式网络防御。

引言

在真实的动态环境中，为网络系统防御提供决策支持是一项具有挑战性的研究问题，其中包括对系统状态不确定性的动态特征进行刻画，以及纳入攻击者与防御者之间的动态交互。网络防御方通常受到资源约束（例如人力、时间和成本），随着系统演变，无论是否受到自适应对手的攻击影响，都必须动态更新策略与战术。防御者往往可能并未察觉系统失陷，因此必须采取主动式策略来维持关键任务运行。过去十年间，越来越多的文献聚焦于博弈论方法在网络安全中的应用（Roy et al. 2010; Liang and Xiao 2012）。这些方法通常涉及在非合作环境下，基于马尔可夫决策过程（MDP）模型进行资源分配优化。网络系统攻击与防御建模方法还包括：1）用于分析系统可靠性及攻击结果依赖关系的概率方法（Hu et al. 2017; Chen, Xu, and Shi 2018）；2）贝叶斯网络（Frigault et al. 2008; Shin et al. 2015）；以及3）故障树/决策树（Fovino, Masera, and DeCian 2009）。尽管基于攻击图的方法（Poolsappasit, Dewri, and Ray 2011）对于识别系统漏洞和已知漏洞利用很有价值，但在现实环境中，防御者通常只能获得关于系统属性和攻击目标的不充分且不完备的信息。

强化学习（RL）方法的最新进展推动了部分信息情境下部分可观测随机博弈（POSG）的发展（MacDermed、Isbell 和 Weiss 2011；Sutton 和 Barto 2018）。此外，网络系统状态空间建模也引发了人们对将部分可观测随机博弈用于网络安全问题的兴趣（Ramuhalli 等，2013；Chatterjee 等，2016；Tipireddy 等，2017）。然而，部分可观测随机博弈通常属于一般性建模框架，且往往难以求解。此外，网络决策支持领域的最先进方法还包括采用部分可观测马尔可夫决策过程（POMDP）模型及分布式部分可观测马尔可夫决策过程，来解决各种问题，例如：1）网络风险评估（Carin、Cybenko 和 Hughes 2008）；2）渗透测试中的不确定性（Sarraute、Buffet 和 Hoffmann 2012）；3）网络任务时间分配（Caulfield 和 Fielder 2015）；4）数据渗漏检测（McCarthy 等，2016）；以及 5）有效的欺骗资源分配（Islam 等，2021）。这些问题大多只关注特定攻击（或攻击类型）、采用有限的防御者行动空间，并假设系统模型已知。近期针对网络安全的深度强化学习（深度强化学习）方法，主要着眼于优化网络运行，或应对不同威胁类型的网络防御（Nguyen 和 Reddi 2021；Dutta、Al-Shaer 和 Chatterjee 2021 ）。尽管相关进展大多集中于无模型深度强化学习方法，但仍亟需进一步研究在多样化对抗环境下训练的不同深度强化学习算法在网络防御中的作用。

本文重点探讨深度强化学习在针对具有策略性的多阶段对手优化网络安全防御规划方面的适用性。具体而言，深度强化学习防御智能体的目标是通过学习网络特征和多阶段攻击行为，计算适应情境的防御计划，同时尽量减少对正常系统运行的影响。一般而言，网络系统的动态特性取决于许多相互关联的因素（例如流量、网络利用率），这些因素可能因意料之外的

版权所有 © 2023年，人工智能促进协会（www.aaai.org）。保留所有权利。

物理链路故障、传感器错误等原因而随时间呈现不确定的行为。另一方面，隐蔽的网络对手可能会根据当前网络状况和正在发生的攻击影响来调整其策略（即战术和技术）。例如，多阶段攻击可以通过部署多个软件进程（例如应用程序编程接口（API）调用）来传播，从而在不引起怀疑的情况下执行攻击动作，而非依赖单个进程。然而，防御者未必总能阻止某个进程，因为该进程可能承担着维护由相互依赖的进程支撑的关键运行的作用。因此，防御规划需要考虑攻击在传播不同阶段中各类攻击动作产生的累积影响。图 1 展示了一个基于 MITRE ATT&CK 框架（MITRE 2022）的多阶段攻击传播模板，本研究利用该模板评估多种基于深度强化学习的防御机制。在一次攻防交互中，对手可以从侦察/初始访问 战术中的任意技术开始；若到达影响/数据渗出 战术中的任意技术，则视为获胜。根据防御动作，攻击者可能会放弃攻击（即转入攻击终止状态，或由防御者获胜），也可能继续行动并进入下一阶段。

本文的主要贡献是系统评估在多种对抗不确定性下训练的多种网络防御深度强化学习算法的性能。下一节将简要介绍本研究采用的深度强化学习算法。接下来，我们将介绍所提出的自主网络防御框架和定制仿真环境。随后是实验结果与讨论，以及结论。

背景

网络防御深度强化学习智能体会计算一种策略，用于在当前网络空间状态下推荐最优动作。在时间步序列 t，智能体会在当前状态 s_t 执行动作 a_t，并获得奖励 $R(s_t, a_t)$，用于更新策略。深度强化学习智能体的目标之一是在探索（即执行随机动作以了解其后果）和利用（即基于先前探索获得的知识执行最优动作）之间取得平衡。本研究采用 ϵ- 贪婪方法，以概率 ϵ 执行随机动作。值得注意的是，我们的 ϵ 会随时间推移而衰减。在本研究中，我们评估了四种不同的深度强化学习方法：

深度 Q 网络 (DQN) 深度 Q 网络是一种无模型的深度强化学习方法，可最大化贝尔曼方程：

$$Q_t(s_t,a_t) = Q_t(s_t,a_t) + \alpha(R_t(s_t,a_t) + \gamma \max_a Q_t(s_{t+1},a) - Q_t(s_t,a_t)) \quad (1)$$

其中，$Q_t(s_t, a_t)$ 是近似 Q 值（即期望累积奖励），$R_t(s_t, a_t)$ 是在 t 时的奖励（r_t），而 α 是学习率。在每个 t，智能体执行一个动作，并将（s_t，a_t，r_t，s_{t+1}）存储在经验回放缓冲区中。每经过 n 步，智能体就从缓冲区中随机抽取批次，并使用公式 1 计算损失。在此，我们使用神经网络作为 Q 函数 $Q_t(s_t, a_t)$ 的通用函数逼近器（Riedmiller 2005；Sutton 和 Barto 2018年）。因此，通过随机梯度下降最小化公式 2 中的损失，以更新神经网络参数。扩展方法包括双重深度Q网络方法，该方法使用两个不同的神经网络预测 $Q_t(s_{t+1}, a)$ 和 $Q_t(s_t, a)$，从而使 Q 函数的更新更加稳定。

$$Loss = R_t(s_t,a_t) + \gamma \max_a Q_t(s_{t+1},a) - Q_t(s_t,a_t). \quad (2)$$

演员-评论家 演员-评论家方法结合了策略和价值迭代方法，使用以下组件：

- 评论家 负责策略评估，并使用深度神经网络（DNN）估计 Q 值。评论家根据公式 2 中的损失更新 DNN 的参数，并将计算得到的梯度发送给演员。
- 演员 使用 DNN 为评论家的当前状态推荐最优动作，其参数根据从评论家接收的梯度进行更新。具体而言，演员搜索能够最大化期望累积奖励的 DNN 最优参数 θ^*（即 DNN 的权重）（Sutton 和 Barto，2018）：

$$J(\pi_\theta) = E_{\pi_\theta}[\sum_{t=1}^{T} G(s_t,a_t)] \quad (3)$$

其中，T 是决策轮次的数量，而 $G(s_t, a_t)$ 是总累积奖励。DNN 参数使用以下方程进行更新：

$$\theta = \theta + \alpha \nabla J(\theta) \quad (4)$$

其中，α 是学习率。在公式 4 中，$\nabla J(\theta)$ 是按如下方式推导的梯度下降项（Sutton 和 Barto2018年）：

$$\begin{aligned} \nabla J(\pi_\theta) &= \nabla E_{\pi_\theta}[\sum_{t=1}^{T} G(s_t,a_t)] \\ &= \nabla E_{\pi_\theta}(\sum_{t=1}^{T} \nabla_\theta log \pi_\theta(a_t|s_t) G(s_t,a_t)) \end{aligned} \quad (5)$$

其中，$\pi(a_t|s_t)$ 是在当前策略参数 θ 下，于状态 s_t 执行动作 a_t 的概率。我们实现了以下三种不同的演员-评论家算法变体。

– 优势演员-评论家方法 *(A2C)*该方法使用优势函数 $A(s_t, a_t)$，其定义如下：

$$A_(s_t,a_t) = G(s_t,a_t) - V(s_t) \quad (6)$$

以替代公式 5 中的 $G(s_t, a_t)$。优势函数可降低高方差，使策略网络更加稳定。请注意，公式 6 中的 $V(s_t)$ 是在 s_t 处取得的基线值。

–异步优势演员-评论家方法 *(A3C)*：A3C 与 A2C 的不同之处在于，演员会在任意评论家完成梯度更新后立即异步更新策略参数，而不会等待所有评论家都完成更新。

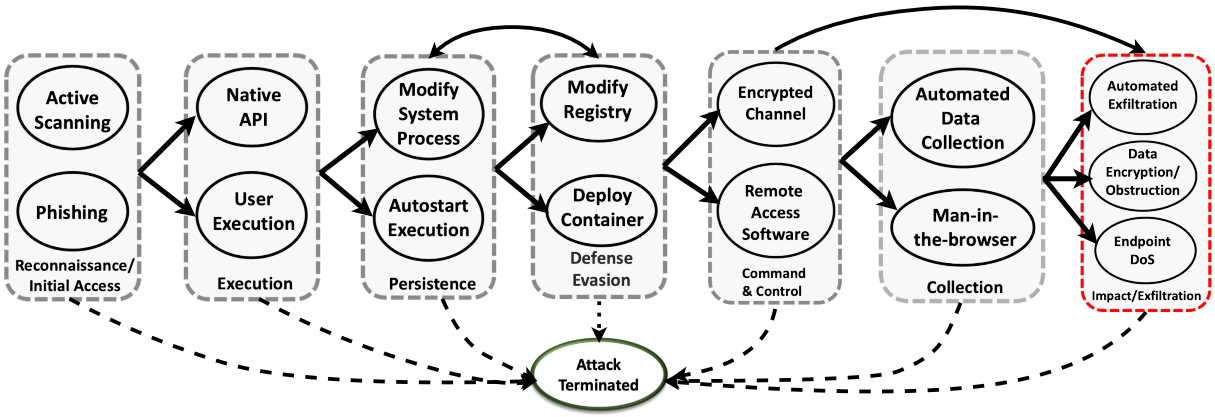


图 1：以 MITRE ATT&CK 战术和技术表示的多阶段攻击传播。（注：攻击战术与技术之间的有向边表示，攻击者在实现该攻击战术的目标后，可能会尝试采用该技术。双向箭头表示防御规避 可能先于 持久化。）

–近端策略优化 *(PPO)* 当新策略的偏离超出区域 $[1-e, 1+e]$（e 称为裁剪参数）时，此方法会对其进行裁剪，以避免与先前评估的策略产生过大偏差。这可能有助于应对传感器噪声和误差。PPO 优化以下裁剪后的代理目标函数， $L^{CLIP}(\theta)$ ：

$$L_t^{CLIP}(\theta) = \mathbb{E}[\min\{r_t(\theta)A_t, \text{clip}\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{old}(a_t|s_t)}, 1-e, 1+e\right)A_t\}] \quad (7)$$

其中， π_θ 和 π_{old} 分别表示新的和旧的随机策略。

自主网络防御框架

图2展示了我们提出的自主网络防御框架。该框架的核心是我们开发的定制 OpenAI Gym（Brockman 等人，2016年）仿真环境。在每个时间序列中，深度强化学习防御智能体执行一个防御动作，并观察攻击位置 和奖励 作为反馈。每个回合（即多个时间序列）对应一条攻击路径（见图 1），图2中的红色节点表示当前攻击位置。接下来，我们将介绍框架中的对手模型和防御模型。

对手模型对手的目标是依次从侦察/初始访问战术推进到目标战术影响/数据外泄（见图 1）。如果对手成功执行某项技术，就会使用基于逻辑的模型来实现特定战术的目标。例如，在攻击启动后，对手可通过成功执行主动扫描（例如，扫描以发现存在漏洞的机器或服务）或网络钓鱼（例如，向用户发送恶意链接）中的任一项，进入侦察/初始访问战术。在任何时候，攻击位置都是最后成功执行的攻击技术。图 1 中的攻击图满足单调性性质，这意味着对手一旦成功到达某个特定攻击位置，就绝不会放弃该位置。

从侦察/初始访问位置移动到影响/数据外泄有许多可能的链/路径，攻击者可以根据

攻击策略。我们假设，对手可以根据对先前攻击影响以及网络/系统状况的观测来改变其策略。因此，他们可以通过发现已部署的对策来规避或击败任何静态防御策略。

要成功执行一项技术，对手需要利用至少一个现存且尚未被发现，或因可用性、安全性或稳定性要求而无法修补的漏洞。为此，对手可以遵循以下逻辑，采取不同的流程。如果由于对策或缺乏专业知识，对手无法利用任何相关漏洞，则攻击尝试失败。无论采用何种战术/技术，如果对手多次尝试执行下一项攻击技术，均因防御行动成功而失败，就会转移到攻击终止 状态。

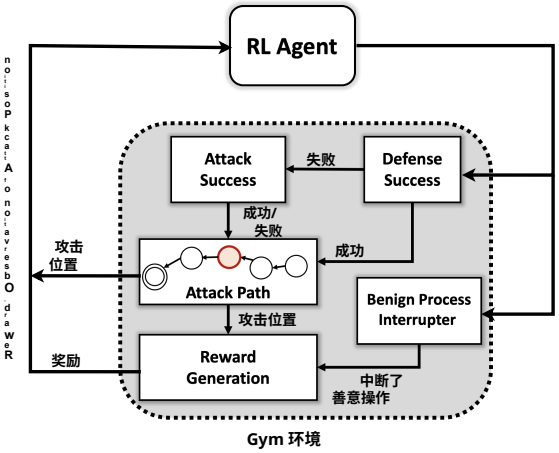


图2：自主网络防御框架。

如果对手多次尝试执行下一项攻击技术，均因防御行动成功而失败，那么攻击会终止。因此，攻击有两种结果：（i）如果对手能够在影响/数据外泄状态下成功执行任意技术，则对手获胜；（ii）如果对手转移到攻击终止 状态，则对手失败。需要注意的是，对手的策略隐含在系统状态演化的描述中。

防御模型 防御者（深度强化学习智能体）的主要目标是主动式地阻止对手进入影响/渗出战术阶段，同时尽量减少因中断良性操作而造成的损失。为确定最优防御，智能体需要推断当前攻击位置并预测下一步攻击动作。然而，缺乏有关系统复杂性和对抗行为的领域专属信息，导致防御者面临以下不确定性，使其无法预先建立系统动力学模型:

- 下一步攻击技术不确定：防御智能体无法预测下一步攻击技术，原因有二。首先，防御智能体并不具备图 1 所示攻击图的先验知识，因为这需要领域专属的真实世界攻击序列，而这些序列很难获取。其次，对手并不总是遵循相同的策略（攻击序列）。例如，在用户执行之后，对手可能会尝试修改注册表 或修改系统进程。

- 下一步攻击流程不确定：由于缺乏领域数据，防御智能体不知道下一步攻击流程（即实施某种攻击技术所需的攻击动作）及其发生的可能性。

- 观测不完善且不完整：部署的警报机制可能无法确定对手当前的位置，原因有二：(1) 进程的可观测性有限；(2) 从观测到攻击技术的映射存在不确定性。

在我们的深度强化学习实验中，我们假设每个回合开始时，对手都处于初始访问位置；防御者则利用监控 API 调用的警报系统来了解当前的攻击位置。如前所述，这些警报信息不仅因未能观测到所有 API 调用而不完整，而且由于将聚合后的 API 调用映射到 MITRE 战术和技术时可能出现错误，因而并不完美。此外，防御者只能部分了解攻击位置，却不知道哪个进程具有恶意，因此需要学习由一系列攻击动作支配的对手动态攻击策略。

防御优化问题

我们使用序贯决策过程（SDP）（Braziunas 2003）框架来构建网络防御者的优化问题，其中下一步攻击技术取决于当前攻击位置和防御动作。需要指出的是，本研究并未学习对手的最优策略；相反，我们针对对应不同攻击行为的各种对手策略训练防御智能体。我们的SDP模型是一个包含四个参数的元组：(S，A，R，γ)，其中，S 是网络系统状态空间，A 是防御动作空间，R 是奖励函数，γ 是折扣因子。智能体仅了解 S、A和 γ，它们定义了智能体与环境的交互。尽管智能体并不了解 R 的数学形式。SDP模型的目标是针对任意当前状态 $s \in S$，计算使有限攻击时域内累积奖励最大化的最优动作 $a^* \in A$。

在每个时间 t开始时，智能体会根据当前环境状态 $s_t \in S$执行防御动作 $a_t \in A$，并接收来自环境的反馈。反馈包括作为观测的当前攻击位置 $s_{t+1} \in S$，以及作为奖励的防御收益 r_t。 a_t 的收益取决于其是否有效阻止了 a_t 发动下一次攻击技术。因此，如果 a_t 能迫使攻击者留在先前的位置（$s_{t+1} = s_t$）或移动到攻击终止节点，那么它就是在 s_t 处有效的防御动作。由此，该模型还通过学习特定环境条件下防御动作的有效性，将预期的攻击行为隐式地纳入决策模型。接下来，我们将详细介绍优化模型的各个要素。

状态空间（S）我们的状态空间 S由17个状态组成，其中每个状态 $s \in S$都是表示一个独特攻击位置的稀疏向量。因此，任意时间t 的当前状态都指定了对手在 t所处的位置，防御智能体据此为该时间序列选择最优防御动作。我们考虑三种状态：（1）对应于图1中每种攻击技术的15个攻击技术状态；（2）攻击已发起状态；以及（3）攻击终止状态。初始状态为攻击已发起状态；如果对手因失败或被检测到而中止攻击，则会转移到攻击终止状态。此外，对手成功执行相关攻击技术后会转移到新状态。我们考虑两种类型的目标状态:

- 攻击目标状态：对手在达到以下任一状态时获胜：(1) 自动化数据窃取、(2) 数据加密/破坏或 (3) 端点拒绝服务（*DoS*）状态，属于影响/数据窃取战术；

- 防御目标状态：如果对手移动到攻击终止状态，防御者获胜。

防御动作空间（A）防御者的动作空间A具有三种不同的运行模式:
1. 不活跃：防御者保持沉默，不执行任何操作；2. 响应式：防御者移除调用或执行与当前攻击位置或上一个攻击动作相关动作的所有进程；3. 主动式：防御者阻止特定的一组 API 调用或操作，以防止下一个攻击动作。

防御者保持不活跃，不进行战略推理，或避免终止关键良性进程。而对于响应式和主动式防御方法，良性操作都可能被中断。我们只考虑一种响应式动作，并假设它可以移除与上一次攻击动作相关的所有进程（无论该攻击是否成功）。对于主动防御，我们考虑 21 种不同的防御动作

这些动作会阻止特定的一组 API 调用和操作，或采用特定方法／措施来防止特定攻击技术。例如，一种主动防御动作“限制注册表权限”会阻止更改某些键，从而防止对手执行自动启动执行（持久化战术）或修改注册表（防御规避战术）。另一种主动防御动作，例如限制文件和目录权限（仅针对特定的一组敏感文件），也可以阻止自动启动执行。然而，不同防御动作的成功概率各异，在防御有效性（即阻止攻击动作）和预期误报率（即终止良性操作）方面均有所不同。因此，防御智能体必须综合考虑防御有效性和预期误报率，选择最佳动作。此外，本研究还假设防御者每个时间步只能执行一个动作。因此，对于主动防御方法，防御者的决策模型需要理解对手接下来可能采取的行动，以执行有效的缓解动作。如果对手因无法执行新的攻击技术，或被检测并移除，而进入攻击终止状态，则防御者获胜。

奖励函数（R）我们考虑以下针对防御智能体的奖励函数：

$$R = -p_g(s) \times I_g - \mathcal{I}_v \times I_g - C_f \tag{8}$$

其中，$\mathcal{I}_v = -1$ 表示防御者获胜，否则为 0。注意，$p_g(s)$ 表示对手从状态 s到达影响/数据外泄战术的概率，而 I_g 表示一次成功的攻击执行所造成的影响/损失。公式 8 由三项组成：(1) $p_g(s) \times I_g$ 量化状态 s 下由于攻击者到达目标的概率（$p_g(s)$）所产生的风险；(2) $I_v \times I_g$ 量化当对手获胜或失败时对防御智能体施加的惩罚或激励；(3) C_f 表示在 A中执行任意防御动作的成本。成本 C_f 取决于中断良性操作所造成的总损失，以及防御实施成本或运营成本。我们假设所有缓解措施的防御实施成本均相同，并且不活跃动作的成本为零。

实验

为实现我们的框架并解决网络防御优化问题，我们制定了实验计划，以评估四种深度强化学习方法（即 DQN、A2C、A3C 和 PPO）的性能。下文将介绍实验设置、训练与测试场景、对手类型以及仿真环境。

实验设置

我们的实验使用 Python 3.7（Van Rossum 和 Drake Jr, 1995）设计，并使用 *RLlib*库实现深度强化学习算法。我们使用一台戴尔 Alienware 计算机进行实验模拟，其配置为 16 核 3GHz Intel Core i7-5960X 处理器、64GB RAM 和三块 4GB NVIDIA GM200 显卡。尽管 MITRE ATT&CK 框架包含 11 种战术和更多技术，我们的实验仅考虑 7 种战术和 15 种技术。我们的防御动作空间 A 由 23 项缓解措施组成，其中包括 21 项主动措施。表1列出了我们所有实验使用的学习参数。

参数名称	值
熵系数	0.05
初始探索概率	1.0
最终探索概率	0.04
探索延迟周期	300,000
工作器数量	4
轨迹片段长度	12
批量大小	48
PPO裁剪值	0.4
训练轮数	100
每个训练轮次的步数	25000

表1：用于训练和测试仿真实验的参数。（注：工作器数量 指并行进程的数量，而探索延迟周期 指将探索概率从1.0衰减至0.04所需的训练步数。）

训练与测试场景

我们生成了从初始攻击位置（即初始攻击状态）到最后一个攻击战术（即影响/数据外泄 战术）中任意状态的所有可能的不同攻击路径。我们使用 80% 的攻击路径进行训练，20% 的攻击路径进行测试；每条攻击传播路径都是对手为实现其目标而执行的一组独特攻击技术序列。

在训练期间，每个独特的回合包含一条攻击传播路径；当防御者或对手获胜或失败时，该回合结束。如果对手成功执行攻击路径中的所有攻击技术，以满足影响/数据外泄战术，则对手在该回合中获胜，或防御者失败。然而，如果对手由于在一条路径中至少失败 n 次而转移到攻击终止 状态，则对手失败，或防御者获胜。

对手类型

在实验中，我们根据两个攻击参数确定对手类型：(1) 技能，以及 (2) 持久化。攻击技能越高，由于利用漏洞的能力更强，执行下一项攻击技术的成功率也越高。我们将 ρ 定义为攻击技能参数，表示利用漏洞的攻击成功率。我们假设所有漏洞具有相同的可利用性（即影响和复杂度）。另一方面，攻击持久性越高，表明对手不会放弃其目标，或者即使多次尝试失败，也不容易被检测到。我们将 τ 定义为攻击持续性参数，表示进入攻击终止状态（即防御方获胜）之前允许失败的尝试次数。通过调整 ρ 和τ，我们考虑三种不同的攻击配置/策略：(1) 攻击配置文件 1（Av_1）：$\rho = 0.75$ 和 $\tau = 4$；(2) 攻击配置文件 2（Av_2）：$\rho = 0.85$ 和 $\tau = 5$；(3) 攻击配置文件 3（Av_3）：$\rho = 0.95$ 和 $\tau = 7$。例如，$Av_2(\rho = 0.85,\ \tau = 5)$表示配置文件 2 的对手利用特定漏洞的成功概率为 85%，并且能够容忍

每回合最多失败 5 次。因此，攻击配置文件 3 代表最老练的对手，而攻击配置文件 1 代表天真的对手。需要注意的是，如果对手先前利用特定攻击技术的尝试失败，他们可能会更改攻击流程。

仿真环境

我们开发了一个定制的 OpenAI Gym 仿真环境（目前尚未公开），用于图2所示的自主网络防御框架。深度强化学习防御智能体不了解环境和攻击行为。防御智能体接收关于下一攻击位置的观测以及奖励反馈。随后，智能体根据近期观测确定当前攻击位置。我们假设，系统警报能够分别以 85%、75% 和 65% 的准确率对应到特定攻击位置，即 Av_1、Av_2 和 Av_3，这表明随着复杂程度提高，对手具有更强的隐蔽性。

在仿真环境中，每个回合开始时会选定一条攻击传播路径，并将环境设置为初始攻击位置。在每个时间步开始时，防御成功 模块根据攻击流程与防御行动之间的关联，确定当前防御动作是否阻止了当前攻击技术。另一个模块良性进程中断器 将因近期防御动作而中断的良性进程 / 操作列表发送给奖励生成 模块。我们假设使用幂律分布函数生成被中断的良性操作数量，该数量在接近最后一个攻击战术时逐渐减少。如果防御动作成功，对手将停留在当前位置。否则，如果防御动作失败，攻击成功 模块将根据攻击剖面的攻击技能（ρ）确定当前攻击技术是否成功。如果攻击成功，对手将移动到下一个位置；否则，其停留在当前位置或移动到攻击终止状态 （回合结束）。奖励根据攻击位置、被中断的良性操作及其他因素，使用公式 8 生成。在时间步结束时，奖励和下一次观测将发送给深度强化学习防御智能体，用于更新和改进其策略。

评估

我们使用防御胜率 （DWR）指标评估不同深度强化学习算法的性能。该指标用于衡量单个批次中防御者获胜的回合所占比例，每个批次包含 200 个回合。利用 DWR 比率，我们分析：(i) 深度强化学习算法在训练期间的收敛情况；(ii) 训练好的防御模型在测试期间对未见过的攻击序列的表现。我们针对三种攻击配置（如第 节所述）评估全部四种深度强化学习算法。

超参数优化与训练

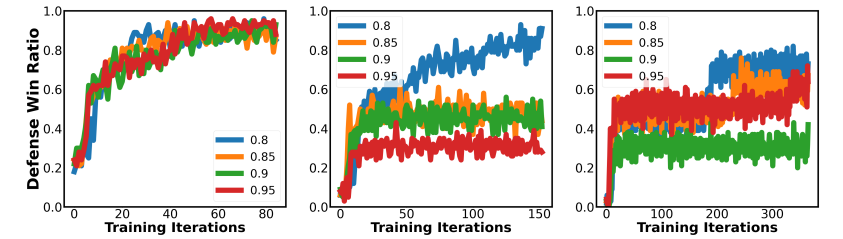
为简洁起见，本文仅通过改变两个超参数：折扣因子（γ）和学习率（α），使用 DWR 指标展示 DQN 和 A2C 的训练性能。在以下各节中，每个图均有两行三列；各列对应三种攻击特征，各行对应超参数（第一行为 γ，第二行为 α）。在本研究中，对于所有深度强化学习算法，我们均采用具有 2 个隐藏层的全连接神经网络，每个隐藏层包含 256 个神经元。在所有情况下，我们均使用 双曲正切函数作为激活函数。

A2C 图 3a 展示了 A2C 对不同 γ取值的敏感性，同时将 α 固定为 0.005。面对攻击配置文件 1 时，所有 A2C 实例均在 40 次迭代（40×200 回合）内收敛至 DWR 值 0.95。面对攻击配置文件 2 时，$\gamma = 0.8$ 的 A2C 实例取得最高的 DWR 比率，而其他所有折扣因子都会导致性能较差。对于攻击配置文件 3，$\gamma = 0.8$ 达到最高 DWR 值 0.8，但需要 200 次迭代。因此，攻击复杂度不仅会延长收敛时间，还会降低防御智能体的成功率。对所有攻击配置取平均后，A2C 在 $\gamma = 0.8$时表现最佳，这表明防御智能体必须在当前奖励与潜在未来收益之间取得平衡。图 3b 展示了 A2C（最优 γ设为 0.8）对不同 α取值的敏感性。我们观察到，面对攻击配置文件 1 时，算法性能没有显著变化。有趣的是，面对攻击配置文件 2 和 3 时，采用 $\alpha = 0.0005$ 的 A2C 表现最佳；这可能是因为当 α取值较大时，算法会陷入局部极小值。因此，在测试实验中，我们将 A2C 的 $\alpha = 0.005$和 $\gamma = 0.8$ 设为相应取值。

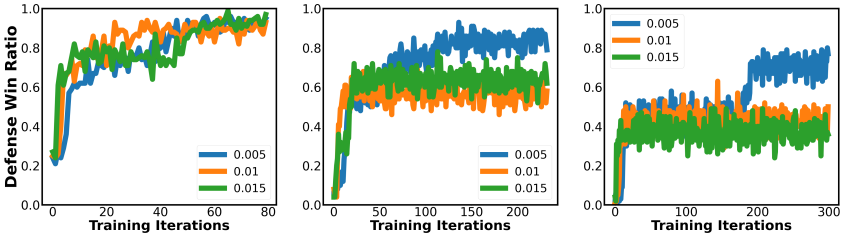
DQN 我们为 DQN 采用了与 A2C 相同的训练方法。从图 4a 可以看出，DQN 在 10 次迭代内收敛到最优策略，速度远快于 A2C；事实上，它的表现优于我们测试的所有深度强化学习算法。此外，对于每种攻击剖面，DWR 都高于其他深度强化学习方法，这表明 DQN 在具有离散状态与动作的环境中表现更出色。此外，图 4b 显示，DQN 的性能不会随着 α 的变化而显著改变。在测试中，我们为 DQN 设置了 $\gamma = 0.8$ 和 $\alpha = 0.01$ 。

测试结果

我们讨论深度强化学习算法在面对三种攻击配置时的测试表现，这些测试使用了训练期间未使用过的攻击序列。图 5 展示了在不同阶段防御对手的累积结果。此处，每个阶段对应一种不同的对手战术。战术编号：*0* 表示侦察之前的初始攻击位置，战术编号：*6* 表示收集，而图 1 中的其他所有战术则从战术编号：*1* 到战术编号：*5*依次编号。请注意，图 5 未显示影响/数据外泄（战术编号 7），因为攻击者一旦到达该状态，防御者就会落败。表 2 报告了在相应攻击战术处被阻止的攻击序列数量。例如，战术编号：*3* 对应的列表示在防御规避阶段被阻止的攻击数量；其他列也同理。平均奖励 列表示相对于最佳奖励（即在战术编号：0 处阻止所有攻击时所获得的奖励）所达到的奖励水平。

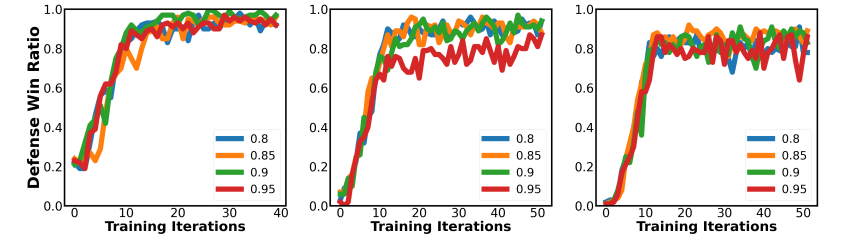


（a） γ 取不同值时的收敛情况（α 设为 0.005）。

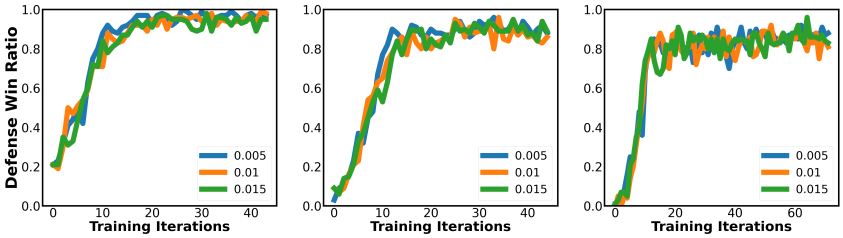


（b） α 取不同值时的收敛情况（γ 设为优化值 0.8）。

图 3：敏感性　　A2C 在不同 γ 和 α 取值下对攻击剖面 Av_1、Av_2 和 Av_3 的敏感性（从左到右）　　.



（a） γ 取不同值时的收敛情况（α 设为 0.005）。



（b） α 取不同值时的收敛情况（γ 设为优化值 0.8）。

图 4：DQN 对不同 γ 和 α 值的敏感性，在攻击特征 Av_1、Av_2 和 Av_3 下（从左到右）

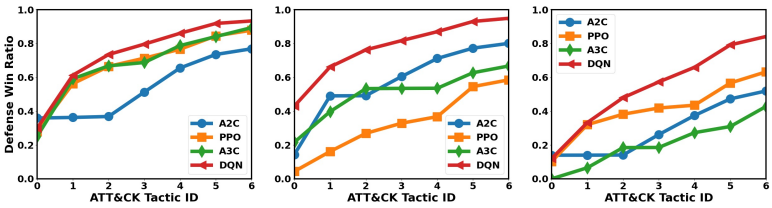


图 5：DWR 在测 试中针对三种攻击配置 Av_1、Av_2 和 Av_3 的表现（从左到右　　t).

防御深度强化学习智能体的目标不仅是阻止对手进入影响/数据外泄 状态，还要尽可能早地阻止攻击进程。在面对所有攻击特征时，我们观察到，DQN 在尽早阻止对手方面表现最佳，可能。如表 2 和图 5 所示，在防御规避 （战术 ID 3）中，DQN 分别在 79.6%、82% 和 57.3% 的情况下阻止了三种攻击配置，同时相应的防御成功率分别达到 93.3%、95% 和 84.1%。面对 Av_3 时，DQN 的成功率较低

DRL 算法	攻击配置文件 1 （%）			攻击配置文件 2 （以%计）			攻击配置文件 3 （%）		
	战术 编号：3	战术 编号：6	Mean 奖励	战术 编号：3	战术 编号：6	Mean 奖励	战术 编号：3	战术 编号：6	Mean 奖励
A2C	51.3	76.9	56.7	60.5	80	59.8	26.2	51.9	27.11
PPO	71.3	87.9	70.3	32.9	58.5	36.8	41.9	63.2	40
A3C	68.8	89.4	72	53.5	66.7	48	18.5	42.8	18.5
DQN	79.6	93.3	77	82	95	77.2	57.3	84.1	60.7

表 2：不同深度强化学习算法的测试性能比较。

于其他攻击配置，因为老练的对手几乎不会在利用漏洞时失手。此外，Av_3 具有持久性，即使尝试失败也不会轻易放弃，这导致防御智能体的 DWR值较低。请注意，其他深度强化学习算法并未始终能有效应对所有三种攻击配置。事实上，针对 Av_3 表现次佳的算法成功率约为 50%。这可能是因为演员-评论家方法通常需要比 DQN 更多的训练样本，才能取得更高的测试准确率。

结论

在网络系统防御中应用深度强化学习方法具有良好前景，尤其是在动态对抗性不确定性和系统状态信息有限的情况下。评估在多样化对抗环境下训练的多种深度强化学习算法，是实现实用型自主网络防御解决方案的重要一步。我们的实验表明，黑箱深度强化学习算法能够在具有不同技能和持久化水平的多阶段攻击特征下进行有效训练，并在对抗环境中取得良好的防御结果。然而，在使用黑箱深度强化学习时，仍有一些实际挑战需要进一步解决（Dulac-Arnold、Mankowitz 和 Hester，2019）：（i）黑箱深度强化学习策略的可解释性；（ii）对对抗噪声和数据投毒的脆弱性；以及（iii）大型状态-动作空间中的收敛性。未来的工作将包括：针对动态环境，为分布式多智能体防御系统开发基于深度强化学习的迁移学习方法。

致谢

本研究得到了美国能源部高级科学计算研究办公室“复杂系统数据驱动决策控制（DnC2S）”项目的支持。本研究部分得到了太平洋西北国家实验室（PNNL）“科学中的人工推理数学”（MARS）倡议的支持，该倡议隶属于实验室指导研究与开发（LDRD）计划。PNNL 是由 Battelle 运营的多项目国家实验室，根据合同 DE-AC05-76RL01830 为美国能源部提供服务。

参考文献

Braziunas, D. 2003. 部分可观测马尔可夫决策过程求解方法。https://www.techfak.uni-bielefeld.de/~skopp/Lehre/STdKI_SS10/POMDP solution.pdf。（最后访问日期：2022年11月9日）。Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; and Zaremba, W. 2016. OpenAI Gym。*arXiv* 预印本 *arXiv:1606.01540*。Carin, L.; Cybenko, G.; and Hughes, J. 2008. 网络安全策略：查询方法。《*Computer*》，41(8)：20–26。Caulfield, T.; and Fielder, A. 2015. 优化网络防御的时间分配。《网络安全期刊》，1(1)：37–51。Chatterjee, S.; Halappanavar, M.; Tipireddy, R.; and Oster, M. 2016. 网络防御应用中的博弈论与不确定性量化。《*SIAM*新闻》，49(6)。Chen, D.; Xu, M.; and Shi, W. 2018年. 利用早期预警机制防御网络系统。《可靠性工程与系统安全》，169：224–234。Dulac-Arnold, G.; Mankowitz, D.; and Hester, T. 2019.《现实世界强化学习的挑战》。发表于第*36*届国际机器学习会议（*ICML*）论文集。arXiv:1904.12901。Dutta, A.; Al-Shaer, E.; and Chatterjee, S. 2021年. 面向自主网络防御的约束满足驱动强化学习。发表于第*1*届自主智能网络防御智能体国际会议（*AICA*）论文集。arXiv:2104.08994。Fovino, I. N.; Masera, M.; and De Cian, A. 2009. 将网络攻击纳入故障树。《可靠性工程与系统安全》，94(9)：1394–1402。Frigault, M.; Wang, L.; Singhal, A.; and Jajodia, S. 2008. 使用动态贝叶斯网络测量网络安全。发表于第*4*届*ACM*保护质量研讨会论文集，23–30。Hu, X.; Xu, M.; Xu, S.; and Zhao, P. 2017. 针对目标的多重网络攻击：观测误差与相关结果的特征分析及优化。《可靠性工程与系统安全》，159：119–133。

Islam, M. M.; Dutta, A.; Sajid, M. S. I.; Al-Shaer, E.; Wei, J.; and Farhang, S. 2021年。CHIMERA：恶意软件欺骗的自主规划与编排。载于*2021 IEEE*通信与网络安全会议（*CNS*），173–181。IEEE。

Liang,E.; Liaw,R.; Nishihara, R.; Moritz, P.; Fox, R.; Goldberg, K.; Gonzalez, J.; Jordan, M.; and Stoica, I. 2018年. RLlib：分布式强化学习的抽象。

发表于国际机器学习会议，3053– 3062。PMLR。Liang, X.; and Xiao, Y. 2012. 网络安全博弈论。*IEEE*通信调查与教程，15(1)：472–486。MacDermed, L.; Isbell, C.; and Weiss, L. 2011. 用于多智能体强化学习的不完全信息马尔可夫博弈。发表于第二十五届*AAAI*人工智能会议研讨会。McCarthy, S. M.; Sinha, A.; Tambe, M.; and Manadhata, P. 2016. 数据外泄检测与防护：通过虚拟分布式部分可观测马尔可夫决策过程构建更安全的网络。发表于安全决策与博弈论国际会议，39–61。Springer。MITRE。2022。MITRE ATT&CK。https://attack.mitre.org/。（最后访问日期：2022年11月9日）。Nguyen, T. T.; and Reddi, V. J. 2021年。网络安全深度强化学习。*IEEE*神经网络与学习系统汇刊，1–17。Poolsappasit, N.; Dewri, R.; and Ray, I. 2011. 使用贝叶斯攻击图进行动态安全风险管理。*IEEE*可靠与安全计算汇刊，9(1)：61–74。Ramuhalli, P.; Halappanavar, M.; Coble, J.; and Dixit, M. 2013. 探索受损网络系统自主重构理论。发表于*2013*年*IEEE*国土安全技术国际会议（*HST*），577–583。IEEE。Riedmiller, M. 2005. 神经拟合Q迭代——数据高效神经强化学习方法的初步实践。发表于第*16*届欧洲机器学习会议（*ECML*）论文集，葡萄牙波尔图，*2005*年*10*月*3*日至*7*日。论文集*16*，317–328。Springer。Roy, S.; Ellis, C.; Shiva, S.; Dasgupta, D.; Shandilya, V.; and Wu, Q. 2010. 博弈论在网络安全中的应用综述。发表于*2010*年第*43*届夏威夷国际系统科学会议，1–10。IEEE。Sarraute, C.; Buffet, O.; and Hoffmann, J. 2012. 部分可观测马尔可夫决策过程让黑客更胜一筹：将不确定性纳入渗透测试。发表于第二十六届*AAAI*人工智能会议。Shin, J.; Son, H.; Heo, G.; et al. 2015. 使用贝叶斯网络开发网络安全风险模型。《可靠性工程与系统安全》，134：208–217。Sutton, R. S.; and Barto, A. G. 2018年。强化学习：导论。麻省理工学院出版社。Tipireddy, R.; Chatterjee, S.; Paulson, P.; Oster, M.; and Halappanavar, M. 2017. 用于网络安全决策支持的以智能体为中心的方法：部分可观测性下的决策支持。发表于*2017*年*IEEE*国土安全技术国际研讨会（*HST*），1–6。IEEE。Van Rossum, G.; and Drake J r, F. L. 1995. *Python*参考手册。阿姆斯特丹数学与计算机科学中心。