基于微分博弈的容器网络安全防护方法研究

温佳坤[1,2]，袭龙[3]，曹源[1]，孙永奎[1]，张舒铭[2]
（1.北京交通大学自动化与智能学院，北京 100044；2.北京全路通信信号研究设计院集团有限公司，北京 100070；
3.中国核电工程有限公司，北京 100840）

摘　要：基于云边端协同的客运服务系统网络面临严峻的横向移动攻击风险。为确保此类关键信息基础设施的安全稳定运行，提出了一种融合微隔离与微分博弈的动态自适应防护方案。首先，通过分布式采集微服务间的调用数据，构建微服务全局流量视图。然后，使用扩散卷积循环神经网络（DCRNN）进行流量异常检测，对微服务远程过程调用（RPC）流量在时间序列上的时空依赖性进行建模，从而实现流量的异常检测。最后，将攻防对抗建模为非零和微分博弈，以系统安全、资源成本等为准则动态求解最优微隔离策略。实验表明，所提方法能有效识别并遏制横向移动攻击，同时实现安全防护与客运服务核心业务性能的动态平衡，为云边端协同的客运服务系统设计提供了理论依据。

关键词：云边端协同；横向移动；微分博弈；动态微隔离
中图分类号：U28
文献标志码：A
DOI：10.11959/j.issn.1000–436x.2026023

Research on container network security protection method based on differential game

Wen Jiakun[1,2], Xi Long[3], Cao Yuan[1], Sun Yongkui[1], Zhang Shuming[2]
1. School of Automation and Intelligence, Beijing Jiaotong University, Beijing 100044, China
2. CRSC Research & Design Institute Group Co., Ltd., Beijing 100070, China
3. China Nuclear Power Engineering Co., Ltd., Beijing 100840, China

Abstract: The network of cloud-edge-end collaborative passenger service systems faced severe risks of lateral movement attacks. To ensure the secure and stable operation of such critical information infrastructure, a dynamic adaptive protection scheme integrating micro-segmentation and differential game was proposed. First, by distributedly collecting call data between microservices, a global traffic view of microservices was constructed. Then, using a diffusion convolutional recurrent neural network (DCRNN)-based traffic anomaly detection method, the spatiotemporal dependencies of microservices remote procedure call (RPC) traffic on time series were modeled, thereby achieving traffic prediction and anomaly detection. Finally, the offensive-defensive confrontation was modeled as a non-zero-sum differential game, dynamically solving the optimal micro-segmentation strategy based on the criteria of systematic security, business continuity, and resource cost optimization. Simulations and experiments demonstrate that the proposed method can effectively identify and curb lateral movement attacks, while achieving a dynamic balance between security protection and the core business performance of passenger services. This provides a theoretical basis for the cybersecurity design of a cloud-edge-end collaborative passenger transportation service system.

Keywords: cloud-edge-end collaboration, lateral movement, differential game, dynamic micro-segmentation

收稿日期：2025–12–03；**修回日期：**2026–01–19
通信作者：温佳坤，23115037@bjtu.edu.cn
基金项目：国家自然科学基金资助项目（No.U2368202, No.U2468203）
Foundation Items：The National Natural Science Foundation of China (No.U2368202, No.U2468203)

0 引言

近年来，我国高速铁路迅猛发展，经过十几年大规模的高铁建设，无论是在运行速度上，还是在运营里程上都已处于世界前列[1-2]。高铁的快速发展对高铁客运服务水平和安全保障能力也提出了更高的要求[3]。尽管现有的列车客运管理信息系统为高铁客运服务提供了重要支撑，但随着高铁列车客运服务的多样化，如车载计算资源不足难以应对大量数据和业务的处理任务、隐私泄露风险大难以保证旅客信息安全、客运服务管理信息各子系统之间信息壁垒高筑、高效安全的协同服务难以形成等，现有系统性能和客运服务质量提升需求之间的矛盾日益凸显。边缘计算的分布式架构、靠近终端和低延时响应能力为解决列车客运服务中车载终端算力不足的问题提供了可能。同时云边融合的资源协同架构也为打破客运服务信息壁垒，实现列车客运服务管理技术变革提供了新的解决方案[4]。然而，高铁列车具有高速移动特征、客运服务业务涉及数据隐私性等特点，构建适用于高铁列车客运服务的云边端协同管理信息系统面临诸多挑战[5]。

在云边端协同客运服务系统中，云端负责全局票务、大数据分析与资源调度；边缘节点处理区域性的实时业务；终端设备则负责数据采集与交互[6]。这一架构广泛采用 Kubernetes 与容器技术，以实现应用的快速迭代与弹性伸缩[7]。然而，该架构也引入了新的安全挑战：攻击者一旦通过某个边缘节点或终端设备侵入系统，即可利用容器网络间宽松的通信策略，在云、边、端之间进行横向移动，如从一个人证核验终端渗透至边缘计算节点的调度数据库，其后果不堪设想[8]。传统的边界防火墙在此场景下已然失效，而静态的微隔离策略又难以适应客运流量波动、业务优先级动态变化等复杂情况[9]。

在横向渗透攻击的研究中，Chen 等[10]对包含横向渗透的高级持续性威胁（advanced persistent threat, APT）攻击的特点进行了分析，描述了此类攻击的特点和攻击模型，分析了在实施攻击中常见的技术，并进一步介绍了一些有助于缓解此类攻击的对策。Greco 等[11-12]演示了如何基于扩展的有限状态机对横向渗透攻击进行分析，使其提出的方法能够尽可能准确地检测到横向渗透攻击的存在。Bohara 等[13]提出了一种基于目标系统安全状态图的建模方法，以实现对横向渗透攻击的有效检测。由于横向渗透攻击的攻击者通常会建立一个命令和控制通道来进行远程控制，实现在目标系统中的横向移动，以此提升攻击者控制系统的权限，进而获取敏感数据。因此，他们在分析横向渗透攻击特征的基础上，提出了利用多种异常检测技术来识别被攻击者入侵主机的方法。Fawaz 等[14]提出了一个分布式数据融合的框架，以实现对横向渗透攻击的有效检测，该方法使用主机级进程通信图来推断网络连接状况，通过将连接状况聚合到系统范围的主机通信图中，该框架可以推断系统中可能存在的横向渗透攻击。Noureddine 等[15]提出了一种针对企业网络中横向渗透攻击的检测方法，通过将系统建模为一个网络服务图，并使用网络监视信息对网络服务图进行标记，以达到对横向渗透攻击的自动响应。但上述研究都缺乏与如何保障既有任务执行效率的相关研究。

本文主要研究工作如下。

1) 建立了基于分布式追踪的微服务远程过程调用（remote procedure call, RPC）流量数据获取方法。并基于这些数据构建微服务全局流量视图，将微服务节点到其相邻节点的应用程序流量与扩散过程相关联，可以更准确地追踪和了解微服务应用程序的整体性能和行为，以便进行后续的流量预测和异常检测。

2) 提出了基于扩散卷积循环神经网络（diffusion convolutional recurrent neural network, DCRNN）的流量异常检测方法，使用扩散卷积和循环神经网络（recurrent neural network, RNN）的结合，同时捕捉微服务之间的空间依赖关系和时间依赖性，从而实现更准确的流量预测；通过预测误差定义阈值，以检测未来 RPC 流量中的异常情况。

3) 将微分博弈引入微隔离策略管理，通过定义融合安全、性能和成本的多目标支付函数，求解防御方的最优策略，能够在持续对抗中自适应地调整隔离策略，从而在有效遏制横向移动攻击的同时，保障关键业务的连续性与系统性能。

1 容器网络安全防护整体设计

容器网络安全防护方法构建一个以动态决策

为核心的智能微隔离模型，如图 1 所示。该模型以微服务为最小访问控制粒度，旨在对应用层东西向恶意流量进行有效隔离和阻断。首先，模型通过采集 RPC 流量数据，预测未来的流量活动并检测异常流量。在关键的响应环节，该模型引入微分博弈理论，将阻断策略的选择过程形式化为一个动态优化问题。具体而言，它将防御方与攻击方的对抗建模为一类连续时间的非零和微分博弈。通过定义一个融合了安全收益、业务性能损耗与策略执行成本的支付函数，模型能够求解出在特定时刻针对特定威胁的最优防御策略，而非依赖静态的预定义规则。

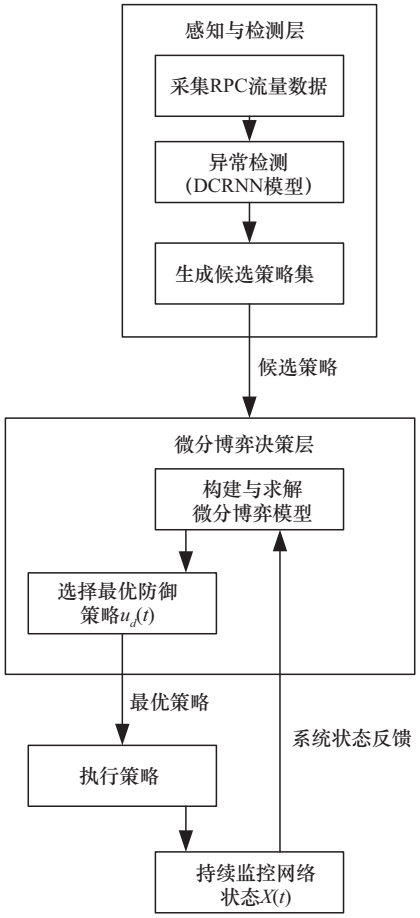


图1　容器网络安全防护结构

最终，系统根据此微分博弈理论优化结果，动态生成、下发并执行相应的微隔离策略。这种基于微分博弈的智能决策机制，使系统能够在复杂对抗环境中实现精准响应，从而更有效地检测、响应和阻断容器网络中的横向移动攻击，最大程度地保障正常业务的连续性与稳定性。

2　分布式流量数据采集

分布式追踪[16]是一种通过追踪与记录跨微服务的请求调用全过程，进而构建调用链路的关键技术。本节详细阐述了其实现流程及流量控制模块的设计。该流程包含 3 个核心步骤：首先，利用分布式追踪技术捕获微服务间的 RPC 通信，并组装为 RPC 流量数据集；然后，将该数据集映射为一个带权有向图（称为微服务调用关系图），从而形式化表征微服务间的交互关系；最终，按固定时间窗口对 RPC 流量数据进行分段统计与聚合，输出依时间排列的流量矩阵序列。此流程为后续的流量时空依赖性建模与异常检测提供了不可或缺的数据支撑。

2.1　流量控制

在基于微服务的容器网络架构中，单次客户端服务调用通常涉及多个服务与中间件之间的复杂交互，形成完整的调用链路。分布式追踪技术通过记录各服务调用的起止时间、请求 ID 等关键信息，将其串联为完整的调用链路，从而实现对请求处理全过程的可观测性。具体而言，该技术使用跨度表征单个服务调用的时间范围和上下文信息，并通过组合多个跨度构建追踪链路，以反映包括子操作在内的整体调用拓扑与性能表现。借助分布式追踪技术，系统能够有效识别性能瓶颈与异常行为，辅助问题定位与系统优化[17]。

尽管分布式追踪技术已取得一定进展，但现有工具在实现方式上仍存在局限。根据设计理念，主流方案可分为基于软件开发工具包、探针和代理的三类实现方式[18]。然而，这些工具通常需要对应用程序代码、系统镜像或运行环境进行侵入式改造，或在追踪信息的准确性与完整性方面存在不足。

为此，本文设计了一种低侵入的流量控制模块，作为一种基于数据收集的分布式追踪实现方案。该方案通过内置的分布式追踪单元采集微服务间 RPC 流量数据，并据此构建微服务调用关系图与时序流量矩阵，为后续建模与分析提供数据支持。与其他方案相比，该方案不需要修改微服务应用代码或配置，能够在保证追踪信息准确与完整的前提下，以低侵入方式为流量预测与异常检测提供可靠数据基础。流量控制模块结构如图 2 所示。

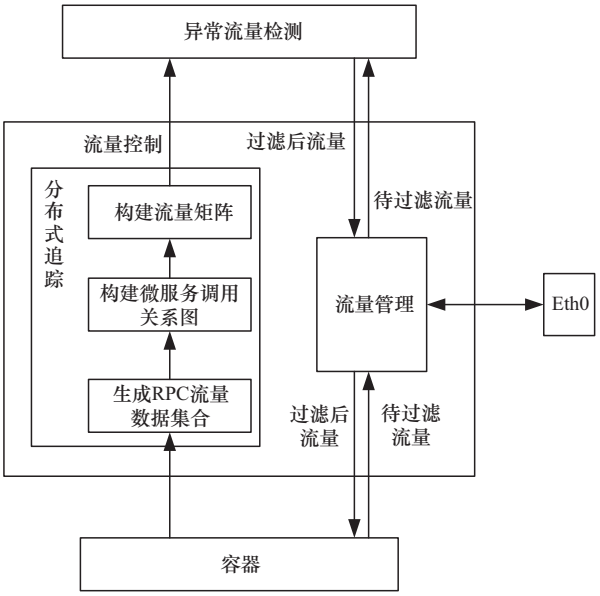


图2 流量控制模块结构

2.2 RPC流量数据集生成

当用户发起请求时，应用程序通常需要协调多个微服务共同完成一系列任务，以实现复杂的业务逻辑。这些微服务之间通过RPC进行协作与数据交换。在生产环境中，此类交互形成了一系列RPC流量数据，构成了微服务架构中功能模块间通信的基础[19]。

如图2所示，在分布式追踪的初始阶段，流量控制模块中的分布式追踪单元负责采集微服务间的RPC元数据，包括调用起止时间、源服务与目标服务等数据。这些数据被整合为微服务RPC流量数据集，为后续分析与建模提供支持。通过对该数据集进行深入分析，能够揭示应用程序内部组件间的交互模式、依赖关系和执行序列，从而有效应对微服务架构在复杂性与可观测性方面所面临的挑战。

2.3 微服务调用关系图

为建模微服务间的调用关系，可将RPC流量数据抽象为带权有向图，即微服务调用关系图。在该图中，每个节点表示一个由源RPC与目的RPC构成的调用关系对。若两个节点具有相同的源RPC或目的RPC，则它们之间建立一条边，并根据其关系强度赋予相应权重。

该带权有向图可形式化表示为$G=(N, E, \boldsymbol{A})$，其中，N表示所有调用关系对节点的集合，E表示节点间依共享关系所构成的边集，$\boldsymbol{A} \in R^{N \times N}$表示带权邻接矩阵，用于量化节点间的邻接强度。

边的权重赋值规则如下：若两节点共享同一源RPC或目的RPC，其边权重设为0.5，反映该类关系在多个节点间共享的局部关联性；若某节点的源RPC与另一节点的目的RPC相同，则两者间存在直接依赖，边权重设为1，表示完整的依赖关系。

通过构建带权邻接矩阵，可实现对微服务间调用关系的结构化建模与量化分析，为后续图结构学习与异常检测提供基础。

2.4 流量矩阵生成

为量化微服务间的交互行为，需对微服务调用关系图中的节点属性进行建模。每个节点具有多个属性（如请求数量、处理时间等），可表示为$N \times M$的矩阵，其中N为图中节点数量，M为属性维度。在流量预测任务中，聚焦于调用关系对的执行次数这一表征应用流量的关键属性，即源RPC至目的RPC的调用频次。

基于已采集的RPC流量数据集，可通过时间维度聚合来构建流量矩阵。具体流程如下。

1) 将时序数据划分为T个等长时间步。

2) 统计每个时间步内各调用关系对的执行次数，生成对应的流量向量。

3) 基于微服务调用关系图G，得到连续T个时间步的流量矩阵序列$[\boldsymbol{X}_{t-T+1}, \cdots, \boldsymbol{X}_t]$。

第t个时间步的流量矩阵如式(1)所示。

$$\boldsymbol{X}_t = \left(N_{1,t}, N_{2,t}, \cdots, N_{n,t}\right) \tag{1}$$

其中，$N_{i,t}$表示第i个节点在第t个时间步的执行次数。该流量矩阵序列有效表征了微服务应用的动态流量特征，为后续时空依赖建模与异常检测提供了关键数据基础。

3 基于扩散卷积神经网络的异常流量检测

异常流量检测旨在通过分析网络流量数据，识别出隐含在正常业务流量中的异常模式。本节首先阐述流量预测的基本概念与目标，继而提出了一种基于DCRNN的异常检测方法。该方法通过融合扩散卷积与RNN[20]，能够同时捕捉微服务间流量的空间依赖性与时间动态特性。基于历史流量数据与微服务交互关系，DCRNN模型可预测未来时段的流量变化趋势。为提升模型性能，训练过程中采用平均绝对误差（mean absolute error, MAE）作为损失函数，以优化模型预测精度。最后，通过设定预测误差的阈值，可对后续RPC流量进行异常判定，

从而实现对潜在异常流量的有效识别。

3.1　流量预测

流量预测[21]指利用历史数据与预测模型，对未来流量趋势进行预估的数据分析技术。其核心是构建一个预测函数 $h()$。具体而言，在给定微服务调用关系图 G 及 T 个历史流量矩阵的条件下，该函数可在第 t 个时间步，将历史流量矩阵序列 $[\boldsymbol{X}_{t-T+1}, \cdots, \boldsymbol{X}_t]$ 映射为未来 T' 个时间步的流量矩阵序列 $[\boldsymbol{X}_{t+1}, \cdots, \boldsymbol{X}_{t+T'}]$，如式(2)所示。基于此函数建立的模型，能够根据流量矩阵的时间序列实现对未来应用流量的精确预测。

$$\left[\boldsymbol{X}^{(t-T'+1)},\cdots,\boldsymbol{X}^{(t)};G\right]\xrightarrow{h(\cdot)}\left[\boldsymbol{X}^{(t+1)},\cdots,\boldsymbol{X}^{(t+T)}\right] \quad (2)$$

3.2　时空依赖性建模

在微服务架构中，服务间通过复杂的调用关系构成图结构，各服务的调用量可视为节点上的流量，其传播过程可建模为图上的扩散行为。为刻画流量传播中的空间依赖关系，本文采用扩散卷积操作对RPC流量数据进行建模，构建基于DCRNN的预测模型，使其能够在微服务调用关系图上学习节点间的结构关系与流量传播特征，从而有效捕捉RPC流量的空间模式，支撑流量异常检测等任务。

具体而言，该扩散过程可形式化描述为在微服务调用关系图 G 上的随机游走。每个节点依据特定概率分布向其邻接节点传递流量，从而建立起流量动态与图结构的显式关联。进一步，将节点间流量建模为加权分布，其传播过程可通过无限步随机游走在图上进行刻画，并利用流量矩阵 $\boldsymbol{X}$ 表征其分布状态。为充分提取流量特征，模型同时引入反向扩散过程，构建双向传播机制，使其能够同时感知上游与下游的流量影响，从而在训练过程中兼顾正向与反向的时空依赖关系。

在给定图结构 G 与流量矩阵 $\boldsymbol{X}$ 的条件下，扩散卷积操作可表示为

$$\boldsymbol{W}_G\boldsymbol{X}=\sum_{d=0}^{K-1}\left(\boldsymbol{W}_O\left(\boldsymbol{D}_O^{-1}\boldsymbol{A}\right)^d+\boldsymbol{W}_I\left(\boldsymbol{D}_I^{-1}\boldsymbol{A}^{\mathrm{T}}\right)^d\right)\boldsymbol{X} \quad (3)$$

其中，K 为扩散过程的最大步数，用于限制随机游走深度，防止无限扩散；$\boldsymbol{A}$ 为基于微服务调用关系图构建的带权邻接矩阵，表示节点间的连接强度；$\boldsymbol{D}_O$ 与 $\boldsymbol{D}_I$ 分别为入度与出度对角矩阵，用于实现节点归一化，赋予模型双向学习能力；$\boldsymbol{D}_O^{-1}$ 与 $\boldsymbol{D}_I^{-1}\boldsymbol{A}^{\mathrm{T}}$ 分别为扩散过程与反向扩散过程的转移矩阵，共同构建双向传播路径；W_O 与 W_I 为作用于双向扩散过程的可学习滤波器，用于提取不同传播方向上的特征。该卷积操作通过将流量矩阵 $\boldsymbol{X}$ 与带权邻接矩阵 $\boldsymbol{A}$ 进行双向扩散与线性变换，生成新的特征矩阵，有效捕捉节点与其多阶邻居之间的空间依赖关系。所得特征矩阵融合了流量在图中传播的结构信息，可用于后续RPC流量预测与异常判别。扩散卷积方法通过建立流量动态与图结构之间的显式关联，系统建模节点间的空间依赖关系。

DCRNN模型通过融合扩散卷积操作与门控循环单元（gated recurrent unit, GRU）[22]，构建了一种能够同时捕捉微服务流量时空依赖关系的深度学习架构。该模型采用编码器–解码器框架，以时序RPC流量矩阵作为输入，通过编码器提取特征并压缩为固定长度的状态向量，再由解码器基于该向量生成未来时间步的流量预测，为后续异常检测任务提供关键数据支撑。

具体而言，编码器将历史流量矩阵序列映射为隐含状态表示，通过多层神经网络提取时序特征；解码器则利用该状态向量，结合自回归机制，逐步生成未来多个时间步的流量预测。整个模型构成一个端到端的预测系统，有效建模微服务流量的时间演化规律。

模型核心采用扩散卷积门控循环单元（diffusion convolutional gated recurrent unit, DCGRU）作为基础构建模块。与传统门控循环单元相比，DCGRU以前向与反向图扩散卷积替代标准矩阵乘法，使其能够同时处理时序依赖与图结构空间关系。DCGRU包含4个关键组件：重置门 r_t、更新门 u_t、细胞状态 c_t 和隐藏状态 h_t，其计算过程分别如式(4)所示。

$$\begin{aligned}
r_t &= \sigma\left(W_r*_{\mathcal{G}}\left[X_t,h_{t-1}\right]+b_r\right)\\
\mu_t &= \sigma\left(W_\mu*_{\mathcal{G}}\left[X_t,h_{t-1}\right]+b_\mu\right)\\
c_t &= \tanh\left(W_c*_{\mathcal{G}}\left[X_t,\left(r_t\odot h_{t-1}\right)\right]+b_c\right)\\
h_t &= \mu_t\odot h_{t-1}+\left(1-\mu_t\right)\odot c_t
\end{aligned} \quad (4)$$

其中，$*_{\mathcal{G}}$ 表示图扩散卷积操作，$\odot$ 表示逐元素乘法。重置门 r_t 控制历史状态的遗忘程度，更新门 u_t 调节新信息的融入比例，两者协同实现对信息流的精细控制。细胞状态 c_t 作为候选状态，使用tanh激活函数确保数值稳定性；隐藏状态 h_t 综合历史与当前信息，形成单元输出。

可学习参数 W_r、W_u 和 W_c 作为扩散卷积滤波器，通过训练自适应优化，增强模型对复杂时空模式的捕捉能力。如图3所示，DCRNN模型通过堆叠多个DCGRU层构建编码器–解码器架构，每层均包含ReLU激活函数以增强非线性表达能力。编码器将输入序列编码为状态向量，解码器利用该向量进行多步预测。模型通过反向传播算法进行端到端训练，实现对RPC流量时空动态的精确建模。

3.3　模型训练及异常检测

通过上述时空建模机制，DCRNN模型能够有效捕捉微服务流量的时间依赖性，并利用带权邻接矩阵 $\boldsymbol{A}$ 刻画节点间的空间关联。这种融合时空特征的建模方式，可更全面地描述流量数据的演化规律，从而显著提升流量预测精度。该架构不仅为微服务流量预测提供了有效解决方案，也对后续的异常检测任务具有重要价值。

如图3所示，DCRNN模型的训练过程如下。首先，将带权邻接矩阵 $\boldsymbol{A}$ 和流量矩阵 $\boldsymbol{X}$ 的时序数据输入编码器。编码器由多层DCGRU组成的循环神经网络构成，通过逐层处理提取输入数据的时空特征，并输出最终隐藏状态。该过程使模型能够自主学习流量网络中的空间动态特性。随后，利用编码器的输出状态初始化解码器。解码器采用相同的DCGRU网络结构，负责预测未来时间步的流量矩阵。

模型采用时间反向传播算法进行端到端训练，通过最小化MAE来优化参数，如式(5)所示。

$$\mathrm{MAE}=\frac{1}{s}\sum_{i=1}^{s}\left|y_i-\hat{y}_i\right| \tag{5}$$

其中，s 为样本数量，y_i 为真实值，$\hat{y}_i$ 为预测值。模型通过迭代训练不断调整参数，使预测值逐渐逼近真实值。随着训练轮次的增加，模型预测精度持续提升，最终输出未来 T 个时间步的流量矩阵预测结果。

在测试阶段，已完成训练的DCRNN模型可直接用于未知流量矩阵的预测。此时该模型将基于学习到的权重矩阵和参数进行前向计算，输出未来时段的流量预测。在实际部署中，该模型可根据实时采集的流量数据持续进行监测与预测，为动态调整流量控制策略和提升网络运营效率提供数据支撑。

在获得DCRNN模型的RPC流量预测结果后，需通过异常检测识别各节点的异常流量波动。本文采用基于预测误差阈值的方法进行判定。预测误差定义为某一时间步流量矩阵的真实值与预测值之间对应元素绝对差的平均值。设 X_i^t 表示节点 i 在第 t 个时间步的实际流量值，则其预测误差如式(6)所示。

$$E_i^t=\left|X_i^t-\hat{X}_i^t\right| \tag{6}$$

为设定每个节点 i 的异常判别阈值，依据其历史误差数据计算均值 μ_i 与标准差 σ_i，进而确定误差的合理波动范围，即上下限阈值 $H_i=\mu_i\pm\lambda\sigma_i$（$\lambda$ 为可调参数，通常取2或3）。若某时间步的预测误差 E_i^t 超出该范围，则判定节点 i 在该时间步存在异常流量。

4　基于微分博弈的微隔离策略管理

本节首先阐述了引入微分博弈理论的必要性，

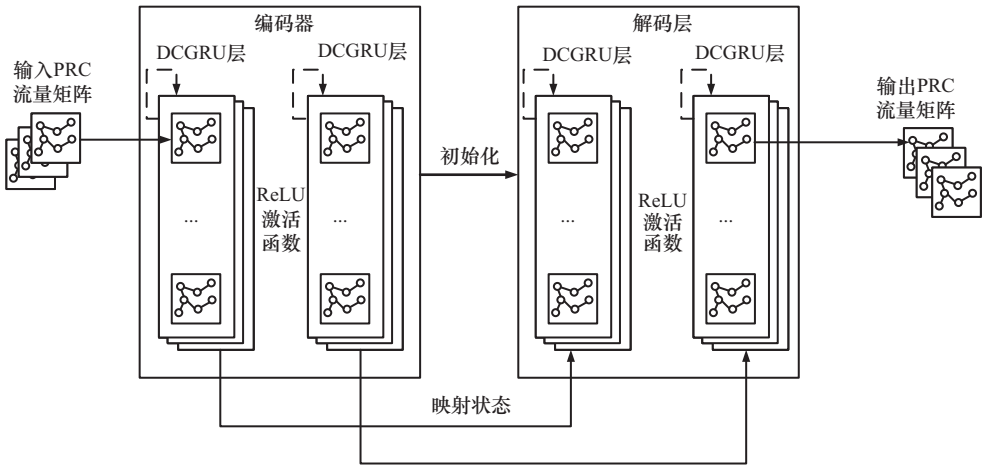


图3　DCRNN模型系统结构

并给出了基于微分博弈的策略管理模块化设计。该方法将防御方与攻击方的对抗过程建模为一类连续时间的非零和微分博弈，通过求解动态最优策略，在有效隔离异常流量的同时，保障关键业务的连续性，从而实现安全防护与任务执行的精细平衡。

4.1　微分博弈策略管理模块定义

本节基于软件定义的思想对微隔离策略管理方法进行了模块化设计，并将微分博弈决策引擎作为核心组件。该方法将微隔离策略的控制和执行逻辑分离，实现了独立管理和动态优化。相较于传统基于硬件的隔离方式，该方法具有更好的灵活性、自适应性和动态平衡能力，工作流程如图4所示，共涉及4个模块：流量异常监测模块、微分博弈模块、策略管理模块和节点代理模块。

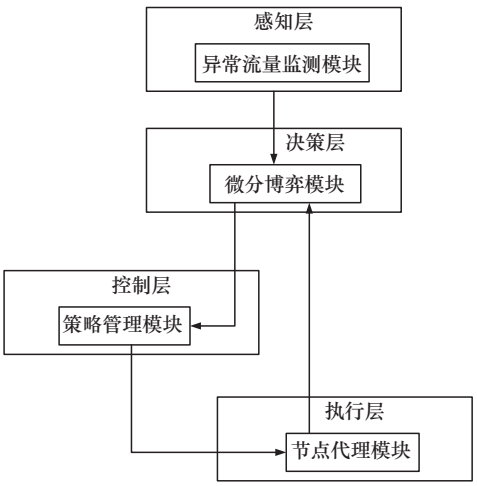


图4　微分博弈的策略管理模块工作流程

如图4所示，其工作流程如下。

1) 感知：流量异常检测模块由异常检测结果（如异常节点、流量强度）与实时采集的系统状态信息共同构成博弈的状态空间$X(t)$。

2) 智能决策：微分博弈决策引擎作为新增的核心组件，接收状态信息，并求解最优防御策略$u_d(t)$。

3) 控制与执行：策略管理模块将博弈输出的最优防御策略$u_d(t)$转换为具体的微隔离规则，并下发至节点代理模块执行。

4) 反馈：节点代理模块将策略执行效果（如是否成功阻断、对业务时延的影响）反馈回微分博弈决策引擎，形成闭环优化。

4.2　微分博弈模型定义

微隔离策略管理微分博弈模型可以形式化定义为六元组$\varsigma=\left(\tau,X,U_d,U_a,F,J\right)$，具体如下。

1) τ为连续时间域，表示微分博弈进行的时间范围。

2) X为状态空间，定义为$X(t)=[S(t),P(t)]$，其中$S(t)$为安全状态向量，包括各微服务对的异常流量指数、活跃攻击路径数量等，$P(t)$为性能状态向量，包括关键业务链路的通信时延、请求成功率等。

3) U_d为防御者策略集。防御者（即微隔离系统）可采取的动作$\mu_d(t)\in U_d$包括μ_d^1：对特定微服务对实施完全阻断；μ_d^2：对特定微服务对进行流量限速（如限制为每秒N次请求）；μ_d^3：提升检测灵敏度，以捕获更细微的异常行为；μ_d^4：仅记录审计告警，而不进行实质性阻断。

4) U_a为攻击者策略集。攻击者可采取的动作$\mu_a(t)\in U_a$包括μ_a^1：攻击强度，如发送恶意流量的频率和数量；μ_a^2：横向移动路径，选择下一个要攻击的微服务目标；μ_a^3：隐匿技术，如降低攻击速率以伪装成正常流量。

5) F为状态方程。描述了系统状态$X(t)$随时间演化的动力学过程，可表示为

$$\frac{\mathrm{d}X(t)}{\mathrm{d}t}=F\left(X(T),\mu_d(t),\mu_a(t),t\right) \tag{7}$$

式(7)量化了攻防动作对系统状态的影响。例如，防御者加强隔离μ_d会降低异常流量$S(t)$，但可能增加通信时延$P(t)$；攻击者增强攻势μ_a会提升$S(t)$并可能降低$P(t)$。

6) J_d和J_a为支付函数。防御者和攻击者分别致力于优化其自身的长期收益。

4.3　支付函数与动态优化目标

为量化防御与攻击双方的收益，实现系统安全、业务性能与资源成本的动态平衡，本文构建了非零和微分博弈模型中的支付函数。防御者的支付函数以多目标综合优化为导向，通过动态调节安全、性能和成本之间的权重，实现对抗环境下的自适应策略优化。攻击者的支付函数则侧重于衡量其攻击收益与被检测风险的权衡。

防御者的支付函数J_d被设计为在时间范围$[0,T]$上积分，旨在最大化其综合收益。

$$J_d(u_d,u_a)=\int_0^T\left[\alpha(t)E(S(t))-\beta(t)R(P(t))-\gamma(t)C(u_d(t))\right]\mathrm{d}t \tag{8}$$

其中，$E(S(t))$为安全收益函数，与异常流量的成功遏制程度正相关，可表示为

$$E(S(t))=\sum_{i\in N}\left[1-\frac{S_i(t)}{S_{\max}}\right]^+ \tag{9}$$

其中，$S_i(t)$为微服务对节点i的异常流量指数，$S_{\max}$为其上限值，$[\cdot]^+$表示取正值部分，确保收益非负，N为微服务调用关系对的总数。

$R(P(t))$为性能惩罚函数，与防御策略导致的业务性能损耗正相关，可表示为

$$R(P(t))=\sum_{i\in L}\left[\frac{D_j(t)-D_{j,0}}{D_{j,0}}\right]^+ \tag{10}$$

其中，$D_j(t)$为关键业务链路j的通信时延，$D_{j,0}$为基准时延，w_j为业务权重，L为系统中定义的关键业务链路总数。

$C(u_d(t))$为成本函数，反映执行防御策略的资源开销，包括计算、存储与网络开销，可形式化为

$$C(u_d(t))=\lambda_d\|u_d(t)\| \tag{11}$$

其中，λ_d为单位策略执行成本系数。

$\alpha(t)$、$\beta(t)$和$\gamma(t)$为动态权重系数，可根据系统实时状态与外部威胁态势自适应调整。

当检测到持续攻击或异常流量激增时，$\alpha(t)$自动升高，强调安全优先；当系统负载高或业务时延超出阈值时，$\beta(t)$升高，侧重性能保障；当资源紧张或策略执行开销过大时，$\gamma(t)$升高，抑制过度防御。

相应地攻击者的支付函数J_a可定义为

$$J_a(u_a,u_d)=\int_0^T\left[\delta A(u_a(t))-\eta T(S(t))\right]\mathrm{d}t \tag{12}$$

其中，$A(u_a(t))$为攻击收益函数，反映攻击行为对系统的破坏程度或信息窃取效果；$\sigma(S(t))$为暴露风险函数，与系统异常检测强度$S(t)$正相关；δ,η为攻击者偏好系数，反映其对攻击收益与隐匿性的权衡。

防御者的目标为在攻击者策略u_a给定的条件下，求解最优防御策略u_d^*，如式(13)所示。

$$u_d^*=\arg\max_{u_d\in U_d}J_d(u_d,u_a) \tag{13}$$

该优化问题通过哈密尔顿–雅可比–贝尔曼方程进行求解，并结合实时状态反馈实现策略的动态更新。该方法使系统能够在安全防护、业务连续性与资源效率之间实现自适应平衡，适用于高铁等复杂业务环境中的动态防御场景。

5 仿真分析

5.1 仿真实验环境

本节实验基于一个自主搭建的Kubernetes集群环境，该集群采用Kubernetes作为容器编排平台，Docker作为容器运行时。集群架构包含一个Master节点与两个Node节点，分别部署于3台Fedora操作系统的主机上，并安装了Cilium等必要的第三方组件。

在组件部署方面，流量异常检测与策略管理模块集中于Master节点，而每个Node节点则分别部署一个流量控制模块与一个节点代理模块。在实现机制方面，依托eBPF技术及Cilium网络插件，模拟实现了流量控制模块中的分布式追踪与流量管理功能，以及策略管理模块中的策略生成、配置与下发等功能。此外，基于配备四卡AMD GPU的独立服务器完成DCRNN模型的训练，该模型进一步用于模拟流量异常检测模块中的流量预测与异常检测功能。

本文实验平台的软件环境基于Fedora 42操作系统，其内核版本为5.14.10-300。容器化环境采用Docker 24.0作为运行时，并使用Kubernetes 1.31.10进行容器编排与管理。网络与安全策略通过Cilium 1.17.0实现。

硬件平台配置如下：系统搭载两颗Intel Xeon Gold 6336Y处理器，每颗处理器拥有24个核心，主频为2.40 GHz，共计48个物理计算核心。内存容量为32 GB，存储硬盘为500 GB。网络方面配备了X520SR2网卡。此外，平台还集成了4块NVIDIA L20GPU，用于加速计算任务。

为验证微隔离策略对异常流量的控制效果，本文在Kubernetes集群中构建了包含系统组件、第三方服务及业务容器（如Nginx、MySQL等）的复合微服务环境。这些容器内集成了多个微服务，通过相互协作构成完整的应用程序逻辑。

在实验过程中，通过应用程序编程接口请求与HTTP工作负载生成器模拟正常与异常RPC调用流量，并配合恶意脚本注入攻击流量，以真实再现横

向移动等网络攻击行为。在此基础上，利用分布式追踪技术持续采集5分钟流量数据，共抽样获取约25 000次RPC调用的相关信息，形成带时间戳的原始数据集。

该数据集进一步转换为包含38个调用关系对（节点）的带权有向图，并构建其带权邻接矩阵。随后，按固定时间间隔对RPC流量数据进行分段聚合处理，生成时序分布的节点流量统计序列。最终，将处理后的数据划分为训练集与测试集，分别用于模型训练和评估横向移动攻击在容器网络中的表现。

5.2　实验结果分析

如图5所示，DCRNN模型的训练与验证损失曲线反映了模型的收敛过程。图5表明，在训练初期，训练与验证损失均处于较高水平；随着训练周期增加，两者呈现同步下降趋势，并在第30个周期后逐渐趋于平稳；在最终训练阶段，两条损失曲线的差异维持在0.1以内，且均收敛至稳定区间。结果表明，DCRNN模型已达到良好收敛状态，具备对新数据进行有效预测的能力。

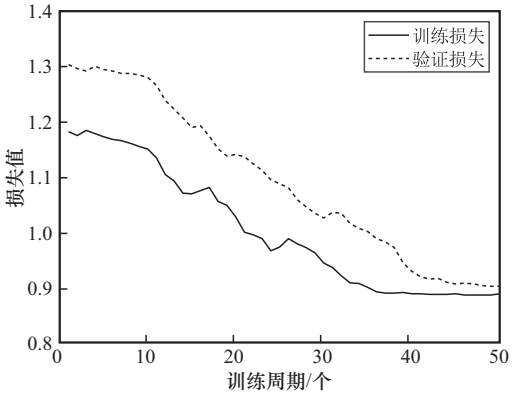


图5　DCRNN模型的训练与验证损失曲线

为验证本文方法的有效性及合理性，在UNSWNB15数据集上与DE-GNN[22]和HRNN[23]方法进行对比实验，其精确率、准确率和F1-score指标对比结果如图6所示。

由图6可知，本文方法在准确率、精确率和F1-score这3个指标上均表现最佳。具体而言，在准确率方面，本文方法达到94.33%，比DE-GNN方法的91.91%和HRNN方法的92.14%分别高2.42%和2.19%；在精确率方面，本文方法也达到最高值96.77%，分别领先DE-GNN和HRNN方法3.32%和1.57%；在F1-score方面，本文方法达到95.41%，优于DE-GNN和HRNN方法1.42%和1.44%。

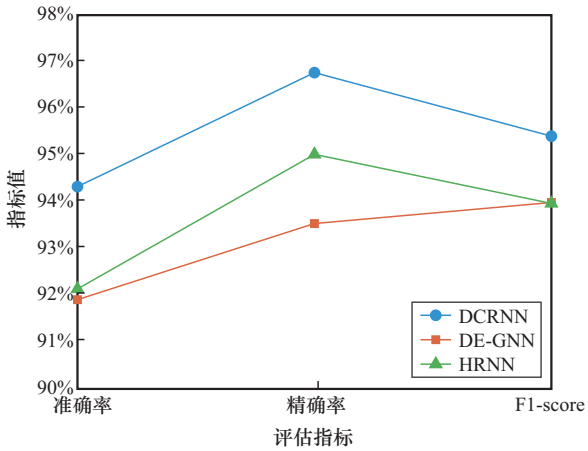


图6　不同方法在各项指标上的性能对比

表1进一步列出了各方法的误报率，其中本文方法为3.36%，显著低于DE-GNN方法的12.71%和HRNN方法的8.39%。综合以上结果，本文方法能更有效地识别网络异常流量，验证了其有效性与合理性。

表1　　不同方法误报率对比

方法	误报率
DCRNN（本文方法）	3.36%
DE-GNN	12.71%
HRNN	8.39%

为有效验证异常检测机制的实用性，本文通过脚本模拟了包括暴力密码破解与分布式拒绝服务（distributed denial of service, DDoS）攻击在内的多种横向移动攻击，并将攻击流量混合于正常微服务RPC流量中，以构建贴近真实的攻击场景。以下分别阐述两种攻击的检测效果。

1) 暴力密码破解攻击检测

暴力密码破解[24]是一种通过系统化尝试多种密码组合以获取访问权限的常见攻击方式。在仿真环境中，通过自动化脚本持续发送大量含不同用户名与密码的登录请求，从而生成异常流量并注入正常业务数据流中。若攻击者借此获得某一容器或微服务的权限，即可进一步实施横向渗透攻击。

图7展示了基于DCRNN的流量异常检测方法对暴力密码破解攻击的检测效果。实验随机选取

8 个 RPC 调用关系对节点进行分析，其中包括正常业务节点与受攻击节点。结果显示，正常业务节点（节点 3、5、6、7）的预测误差均处于较低水平，且未超出设定阈值；受攻击节点（节点 1、2、4、8）的预测误差显著偏高，并突破阈值界限。结果表明，本文方法能够有效识别混杂于正常业务流量中的暴力密码破解行为，验证了其检测能力与可行性。

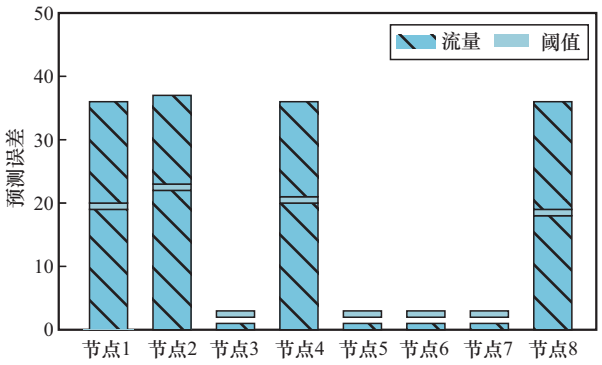


图 7　暴力密码破解攻击的检测效果

2) 分布式拒绝服务攻击检测

DDoS 攻击通过控制多个分布式主机（僵尸节点）向目标服务发起海量请求，旨在耗尽其带宽或计算资源，从而导致服务不可用。相较于传统拒绝服务（denial of service, DoS）攻击，DDoS 攻击因其攻击源分散、流量形态复杂而更难识别与缓解。实验中通过脚本模拟生成大量虚假请求并注入集群内部应用程序，造成微服务 RPC 调用流量激增，干扰正常业务处理。攻击者一旦侵入集群内某一容器或微服务，便可借此手法横向扩散，进一步侵占集群资源并破坏业务连续性。

DDoS 攻击的检测效果如图 8 所示。实验随机选取 8 个 RPC 调用关系对节点，包括正常业务节点与受攻击节点。结果显示，正常业务节点（节点 2、3、6、7）的预测误差始终低于设定阈值；受攻击节点（节点 1、4、5、8）的预测误差显著升高并超出阈值范围。结果表明，本文方法能够有效检测出混杂在正常业务流量中的 DDoS 攻击流量，进一步验证了该方法的适用性与有效性。

综上，基于 DCRNN 的流量异常检测方法能够准确识别隐藏在正常业务流量中的异常 RPC 流量，包括暴力密码破解与 DDoS 等多种攻击形式，验证了其在复杂容器网络环境下的检测能力。此外，该方法所输出的检测结果为后续微隔离策略的精准制定与动态响应提供了可靠依据。

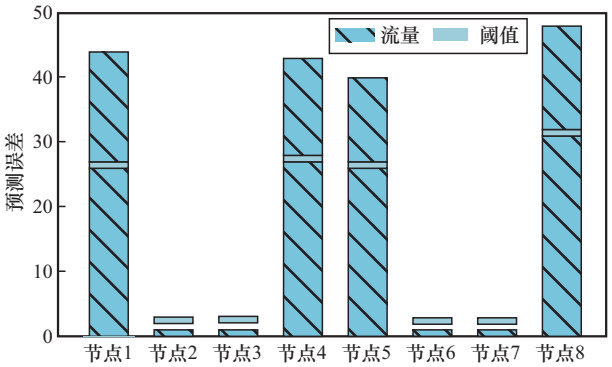


图 8　分布式拒绝服务攻击的检测效果

基于微分博弈的微隔离策略管理方法，通过智能决策机制将异常检测结果转化为动态防御策略，并高效下发至执行节点，实现对异常流量的精准隔离与阻断，形成针对横向移动攻击的闭环主动防御体系。实验结果表明，流量异常检测模块具备可靠的检测能力，为策略生成提供了精准的数据基础。

为验证所提方法的有效性，本文复现了前述攻击场景，并将其与传统防火墙方法[25]及微隔离方法[26]进行对比分析结果如图 9 所示，由图 9 可知，随着初始被攻击率的增加，本文方法的攻破比率呈次线性增长，而未出现传统防火墙方法及微隔离方法的线性扩散态势。这得益于微分博弈机制驱动的动态策略生成与执行，能够在检测到异常后快速阻断横向传播路径。尽管在检测响应时延期间攻击可能存在少量扩散，导致防护效果略低于纯检测能力上限，但仍显著优于传统静态防护方法。

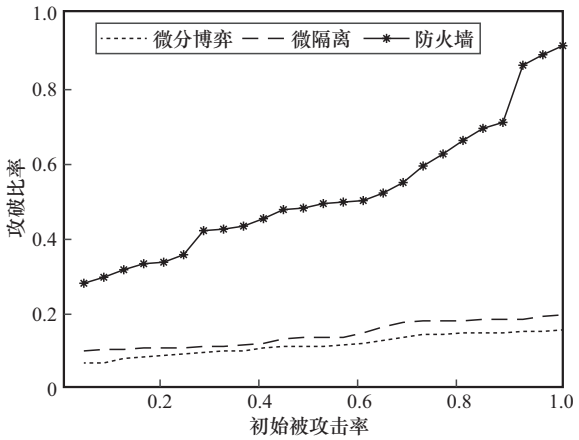


图 9　横向移动攻击的防护效果

图 10 进一步对比了系统资源开销。在持续 10 min、不同传输带宽的负载测试中（集群 CPU 负载按 30 s 间隔采样平均），本文方法相较传统防火墙方法展现出更低的 CPU 资源占用率。这体现了基于软件定义架构的轻量化优势和微分博弈模型在制定策略时对性能损耗的主动权衡。

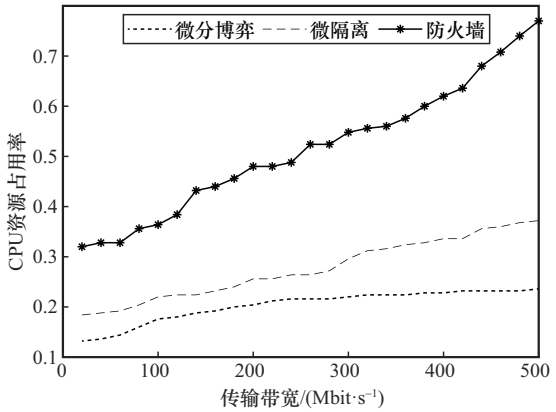


图 10　CPU 资源占用率

图 11 进一步对比了系统任务达成率。随着初始被攻击率的增加，本文方法相较传统防火墙方法展现出更高的任务达成率。这体现了微分博弈模型在制定策略时对任务达成率的主动权衡。

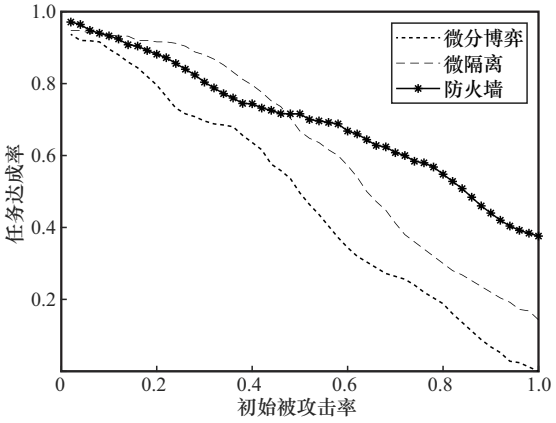


图 11　任务达成率

综上所述，本文提出的基于微分博弈的微隔离策略管理方法通过构建“感知–决策–执行–优化”的动态防御闭环，在保证低资源开销的前提下，实现对横向移动攻击的有效遏制。该方法通过动态优化安全投入与性能损耗的平衡点，显著提升了容器网络在对抗复杂攻击时的安全性与可靠性。

6　结束语

本文提出了一种融合微隔离与微分博弈的动态自适应防护方法。首先，通过分布式采集各级微服务间的调用数据，构建微服务全局流量视图；使用基于 DCRNN 的流量异常检测方法，对微服务 RPC 流量在时间序列上的空间和时间依赖性进行建模，从而实现流量预测和异常检测。然后，将攻防对抗建模为非零和微分博弈，以系统化安全、业务连续性和资源成本多目标最优为准则，动态求解最优微隔离策略。最后，通过仿真分析得出以下结论。

1) DCRNN 模型在 30 周期内达到良好收敛状态，具备对新数据进行有效预测的能力。

2) 基于 DCRNN 的流量异常检测方法能够准确识别隐藏在正常业务流量中的异常 RPC 流量，包括暴力密码破解与 DDoS 等多种攻击形式。

3) 基于微分博弈的微隔离策略管理方法，通过构建“感知–决策–执行–优化”的动态防御闭环，动态优化安全投入与性能损耗的平衡点，显著提升了容器网络在对抗复杂攻击时的安全性与可靠性。

由于作者能力和篇幅有限，本文未对任务执行各阶段效率进行研究，因此后续将开展基于模型轻量化的各阶段任务执行效率研究，以提升该方法的应用价值。

参考文献：

[1] Cao Y, Wen J K, Hobiny A, et al. Parameter-varying artificial potential field control of virtual coupling system with nonlinear dynamics[J]. Fractals, 2022, 30(2): 2240099.

[2] Cao Y, Li P, Zhang Y Z. Parallel processing algorithm for railway signal fault diagnosis data based on cloud computing[J]. Future Generation Computer Systems, 2018, 88: 279-283.

[3] 罗潇, 刘悦. 轨道交通车载端到端语音合成[J]. 机车电传动, 2023(6): 122-128.
Luo X, Liu Y. An end-to-end text-to-speech system for vehicle-mounted devices[J]. Electric Drive for Locomotives, 2023(6): 122-128.

[4] 贺佳. 基于三维点云与图像融合的轨道交通场景行人检测方法[J]. 机车电传动, 2024(3): 146-155.
He J. Pedestrian detection method in rail transit scenes based on fusion of 3D point clouds and images[J]. Electric Drive for Locomotives, 2024(3): 146-155.

[5] Zhao B, Xiao C B, Zhang Y, et al. Assessment of recommendation trust for access control in open networks[J]. Cluster Computing, 2019, 22(1): 565-571.

[6] 李佳曦. 基于容器技术的云化平台安全风险与应对分析[J]. 信息通信技术, 2020, 14(6): 26-31, 38.
Li J X. Security risks and countermeasures analysis of cloud platform based on container technology[J]. Information and Communications

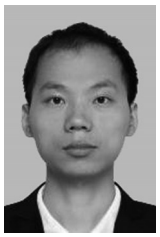
Technologies, 2020, 14(6): 26-31, 38.
[7] Jiang W H, Li Z. Vulnerability analysis and security research of docker container[C]//Proceedings of the 2020 IEEE 3rd International Conference on Information Systems and Computer Aided Education (ICISCAE). Piscataway: IEEE Press, 2020: 354-357.
[8] Zerouali A, Mens T, Roover C D. On the usage of JavaScript, Python and Ruby packages in docker hub images[J]. Science of Computer Programming, 2021, 207: 102653.
[9] Liu P Y, Ji S L, Fu L R, et al. Understanding the security risks of docker hub[C]//Computer Security-ESORICS 2020. Berlin: Springer, 2020: 257-276.
[10] Chen P, Desmet L, Huygens C. A study on advanced persistent threats[C]//IFIP International Conference on Communications and Multimedia Security. Berlin: Springer, 2014: 63-72.
[11] Greco A, Pecoraro G, Caponi A, et al. Advanced widespread behavioral probes against lateral movements[J]. International Journal for Information Security Research, 2016: 651-659.
[12] Greco A, Caponi A, Bianchi G. Facing lateral movements using widespread behavioral probes[C]//Proceedings of the 2016 11th International Conference for Internet Technology and Secured Transactions (ICITST). Piscataway: IEEE Press, 2016: 159-160.
[13] Bohara A, Noureddine M A, Fawaz A, et al. An unsupervised multi-detector approach for identifying malicious lateral movement[C]//Proceedings of the 2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS). Piscataway: IEEE Press, 2017: 224-233.
[14] Fawaz A, Bohara A, Cheh C, et al. Lateral movement detection using distributed data fusion[C]//Proceedings of the 2016 IEEE 35th Symposium on Reliable Distributed Systems (SRDS). Piscataway: IEEE Press, 2016: 21-30.
[15] Noureddine M A, Fawaz A, Sanders W H, et al. A game-theoretic approach to respond to attacker lateral movement[C]//Decision and Game Theory for Security. Berlin: Springer, 2016: 294-313.
[16] Parker A, Spoonhower D, Mace J, et al. Distributed tracing in practice: instrumenting, analyzing, and debugging microservices[M]. California: O'Reilly Media, 2020.
[17] Logrippo L. Multi-level models for data security in networks and in the Internet of things[J]. Journal of Information Security and Applications, 2021, 58: 102778.
[18] Kularatna I M, Rajapaksha U U S. Development of agent-based centralized tool for analyzing and managing security-enhanced linux policies using WebSocket protocol[C]//Proceedings of the 2022 2nd International Conference on Advanced Research in Computing (ICARC). Piscataway: IEEE Press, 2022: 302-307.
[19] Wazan A S, Chadwick D W, Venant R, et al. RootAsRole: a security module to manage the administrative privileges for Linux[J]. Computers & Security, 2022: 102983.
[20] Robinson P. Survey of crosschain communications protocols[J]. Computer Networks, 2021, 200: 108488.
[21] Jiang W W, Luo J Y. Graph neural network for traffic forecasting: a survey[J]. Expert Systems with Applications, 2022, 207: 117921.
[22] Han X B, Xu G Z, Zhang M, et al. DE-GNN: dual embedding with graph neural network for fine-grained encrypted traffic classification[J]. Computer Networks, 2024, 245: 110372.
[23] Yang Z, Ma Z T, Zhao W B, et al. HRNN: hypergraph recurrent neural network for network intrusion detection[J]. Journal of Grid Computing, 2024, 22(2): 52.
[24] Liu X Y, Li N, Guo J, et al. Multistep-ahead prediction of ocean SSTA based on hybrid empirical mode decomposition and gated recurrent unit model[J]. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2022, 15: 7525-7538.
[25] Wang X, Zhao T. Design and research of firewall system of communication department based on network information technology[J]. Journal of Physics: Conference Series, 2021, 2074(1): 012044.
[26] 娄天宇 . 基于多层访问控制的 Kubernetes 集群安全防护[D]. 南京: 东南大学, 2023.
LOU T Y. Security protection of kubernetes cluster based on multi-layer access control[D]. Nanjing: Southeast University, 2023.

[作者简介]

温佳坤（1996– ），男，河北唐山人，北京交通大学博士生、北京全路通信信号研究设计院集团有限公司工程师，主要研究方向为云平台安全保障、高速铁路列车运行控制等。

袭龙（1987– ），男，吉林四平人，中国核电工程有限公司工程师，主要研究方向为安全环保、通信工程等。

曹源（1982– ），男，河南开封人，博士，北京交通大学教授、博士生导师，主要研究方向为列车运行控制系统健康管理。



孙永奎（1993– ），男，河南永城人，博士，北京交通大学副教授，主要研究方向为列车运行控制系统故障诊断、云平台安全保障。

张舒铭（1996– ），男，辽宁沈阳人，北京全路通信信号研究设计院集团有限公司工程师，主要研究方向为安全云平台、高速铁路信号系统安全保障。