软件学报 ISSN 1000-9825, CODEN RUXUEW
2026,37(6):2671−2690 [doi: 10.13328/j.cnki.jos.007565] [CSTR: 32375.14.jos.007565]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

可信数据增强的 QoS 感知云 API 推荐系统投毒攻击持续防御*

陈 真[1,3], 范 爽[1], 余建强[1], 徐悦甡[2], 檀泽宇[1], 尤殿龙[1], 申利民[1]

[1](燕山大学 信息科学与工程学院, 河北 秦皇岛 066004)

[2](西安电子科技大学 计算机科学与技术学院, 陕西 西安 710071)

[3](河北省计算机虚拟技术与系统集成重点实验室 (燕山大学), 河北 秦皇岛 066004)

通信作者: 陈真, E-mail: zhenchen@ysu.edu.cn

摘 要: 服务质量 (quality of service, QoS) 感知云 API 推荐系统在解决云 API 过载问题、差异化云 API 性能和实现高质量云 API 选择中具有重要作用. 但由于网络环境的开放性和云 API 的货币属性, 推荐系统易受到投毒攻击, 从而导致推荐结果偏离公平性和可信性. 现有防御方法主要采用“检测防御”策略, 即在模型训练前通过检测算法滤除恶意用户来缓解攻击影响, 但受限于检测算法性能, 不可避免地会出现无法将恶意用户全部滤除的情形. 为此, 从“以攻学防”的视角提出一种基于可信数据增强的 QoS 感知云 API 推荐系统投毒攻击持续防御方法. 首先构建基于可信数据增强的投毒攻击防御框架, 通过生成高质量可信用户数据并参与模型训练来增强推荐系统的鲁棒性. 其次, 设计基于扩散模型的可信用户生成算法. 采用迭代去噪的方式学习真实云 API 的 QoS 数据分布, 生成高质量的可信用户向量, 消解投毒攻击数据对训练模型的影响. 最后, 基于真实云 API 的 QoS 数据集进行大量实验, 利用 3 类 11 种推荐算法全面评估所提防御方法的有效性和普适性. 实验结果表明, 所提出的基于可信数据增强的投毒攻击持续防御框架是有效的, 生成的可信用户可显著提高云 API 推荐系统的鲁棒性.

关键词: 推荐系统; 云 API; 投毒攻击; 持续防御; 数据增强; 扩散模型

中图法分类号: TP393

中文引用格式: 陈真, 范爽, 余建强, 徐悦甡, 檀泽宇, 尤殿龙, 申利民. 可信数据增强的QoS感知云API推荐系统投毒攻击持续防御. 软件学报, 2026, 37(6): 2671–2690. http://www.jos.org.cn/1000-9825/7565.htm

英文引用格式: Chen Z, Fan S, Yu JQ, Xu YS, Tan ZY, You DL, Shen LM. Continuous Defense Against Poisoning Attack with Trusted Data Augmentation for QoS-aware Cloud API Recommendation System. Ruan Jian Xue Bao/Journal of Software, 2026, 37(6): 2671–2690 (in Chinese). http://www.jos.org.cn/1000-9825/7565.htm

Continuous Defense Against Poisoning Attack with Trusted Data Augmentation for QoS-aware Cloud API Recommendation System

CHEN Zhen[1,3], FAN Shuang[1], YU Jian-Qiang[1], XU Yue-Shen[2], TAN Ze-Yu[1], YOU Dian-Long[1], SHEN Li-Min[1]

[1](School of Information Science and Engineering, Yanshan University, Qinhuangdao 066004, China)

[2](School of Computer Science and Technology, Xidian University, Xi’an 710071, China)

[3](Key Laboratory for Computer Virtual Technology and System Integration of Hebei Province (Yanshan University), Qinhuangdao 066004, China)

Abstract: Quality of service (QoS)-aware cloud API recommendation systems play an important role in solving cloud API overload problems, differentiating cloud API performance, and achieving high-quality cloud API selection. However, due to the openness of the network environment and the monetary nature of cloud APIs, recommendation systems are susceptible to poisoning attacks, which causes the recommendation results to deviate from fairness and credibility. Existing defense methods against poisoning attacks mainly adopt the

* 基金项目: 国家自然科学基金 (62102348, 62276226); 中央引导地方科技发展资金 (236Z7725G, 236Z0103G); 河北省自然科学基金 (F2022203012); 河北省创新能力提升计划 (22567626H)
收稿时间: 2024-11-14; 修改时间: 2025-02-02, 2025-05-06, 2025-10-05; 采用时间: 2025-10-29; jos 在线出版时间: 2026-01-26
CNKI 网络首发时间: 2026-01-27

“detection and defense” strategy, which utilizes detection algorithms to filter out malicious users before model training to mitigate the influence of the attacks. However, due to the performance limitations of detection algorithms, it is inevitable that malicious users cannot be completely filtered out. To this end, this study proposes a continuous defense method against poisoning attacks on the QoS-aware cloud API recommendation system from a “learning to defense by attacks” perspective with trusted data augmentation. First, this study establishes a defense framework against poisoning attacks based on trusted data augmentation and enhances the robustness of the recommendation system by generating high-quality trusted user data for model training. Second, the study designs a trusted user generation algorithm based on the diffusion model, which employs iterative denoising to learn real-world QoS data distribution related to cloud APIs and generate high-quality trusted user vectors, thus mitigating the influence of data subjected to poisoning attacks on training models. Finally, extensive experiments are conducted based on real-world cloud API QoS datasets, and 11 recommendation algorithms from three categories are utilized to comprehensively evaluate the effectiveness and universality of the proposed defense method. Experimental results indicate that the proposed framework of continuous defense against poisoning attacks based on trusted data augmentation is effective, and the generated trusted user can significantly improve the robustness of the cloud API recommendation system.

Key words: recommendation system; cloud API; poisoning attack; continuous defense; data augmentation; diffusion model

应用程序编程接口 (application programming interface, API) 是一组预先定义的网络接口, 通过互联网或者私有云网络为应用程序、开发人员提供访问一组数据和计算资源的服务能力[1]. 进入软件定义一切的云时代, 软件逐渐成为经济社会发展的基础设施, 云 API 也逐渐成为当今世界软件驱动创新的基础元素. 云 API 凭借网构化、跨平台和可扩展等优势将迄今为止一直隐藏在企业与组织背后的数据、算法和算力等核心资产便捷地提供给合作伙伴和对其感兴趣的开发人员, 有力促进了云 API 提供者和使用者的双赢[2,3]. 一方面, 云 API 提供者利用云 API 对外开放其竞品资源和释放数据要素价值. 当前, 国内外科技巨头如阿里、谷歌和 OpenAI 等纷纷布局云 API 的生产和治理, 以汇聚其生态中的合作伙伴, 挖掘新的价值源泉[4]. 例如, 阿里采用云 API 提供在线支付服务, 谷歌利用云 API 共享地图服务, OpenAI 基于云 API 开放 GPT 大语言模型应用. 另一方面, 由于云 API 成功解决了不同编程语言和开发平台导致的软件开发巴别塔难题, 开发人员只需关心云 API 对外暴露的接口, 无需重写底层实现代码, 吸引了众多开发人员采用开放网络中的云 API 作为数字胶水, 便捷地将数据、服务和应用紧密地联系在一起, 营造了优质的用户体验. 可见, 云 API 作为应用程序、人工智能算法和物联网设备之间数据交换、能力复制和服务交付的最佳载体, 已成长为当今面向服务软件开发与运行不可或缺的新型数字基础设施.

如今, 随着企业和组织数字化转型与业务创新对快速接入云端服务以及开放自身竞品资源的日渐重视, 网络中可用的云 API 数目和开发人员数目急剧增长[5]. 例如, 全球最大的云 API 开放平台 RapidAPI 目前支持超过 40k 个 API, 超过 12 000 个 API 发布者, 超过 20 万的月活跃订阅者, 数百万的注册开发者, 以及每月超过 5B 的 API 请求, 并且云 API 数目也正在以每年 30% 以上的增速增长, 由此导致了日益严重的云 API 过载问题. 尽管国内外云 API 开放平台如聚合数据、RapidAPI 和 APIList 纷纷通过构建云 API 仓库, 提供基于关键字匹配、类别浏览等功能来提升云 API 的检索效率, 但功能高度同质化的云 API 严重阻碍了云 API 的选择及其推广应用. 例如, RapidAPI 中仅提供天气预报功能的云 API 数量已高达 117. 这些对于领域知识相对匮乏的用户而言, 快速识别并选择满足其个性化需求的云 API 是极其困难的, 而且云 API 的反复搜索也将大大影响软件的开发效率.

为了解决云 API 功能同质化问题, 研究人员引入服务质量 (quality of service, QoS) 的概念, 用来描述云 API 非功能侧的属性特征, 如响应时间、吞吐量和可靠性等, 进而利用 QoS 数据评估云 API 在特定方面的质量信息[6]. 由于 QoS 能够有效差异化功能相似云 API 之间的性能, 以及推荐系统在解决信息过载问题方面的显著优势, 研究者们随之提出了将 QoS 感知云 API 推荐系统应用于解决功能同质化的高质量云 API 选择难题[7,8]. 值得注意的是, 面对海量云 API, 单一用户只使用过其中有限的云 API, 对大部分云 API 的服务质量是未知的, 而测试所有候选云 API 的服务质量将耗费巨大的时间、资源和费用开销, 因此 QoS 感知云 API 推荐系统能够有效运行的一个基本前提是实现准确的云 API QoS 预测, 以支持 QoS 感知的个性化云 API 推荐.

针对 QoS 感知云 API 推荐任务, 研究人员提出了一系列服务质量预测方法, 如基于协同过滤、基于矩阵分解和基于深度学习的预测方法, 以保障高质量云 API 的选择. 协同过滤是最早的云 API QoS 预测技术之一, 其主要通过计算相似性、识别相似近邻和协同预测这 3 个步骤实现 QoS 个性化预测[9]. 矩阵分解技术则从用户-云 API

交互矩阵中可用的交互数据来学习用户与云 API 的潜在特征表示, 然后利用简单的点积运算进行预测[10]. 基于深度学习的预测方法利用训练数据学习用户与云 API 的潜在特征, 并对用户和云 API 之间的非线性关系进行建模来学习用户的个性化偏好[11]. 尽管现有云 API 服务质量预测方法利用情境信息增强、正则化技术和嵌入特征的交互学习缓解了困扰研究人员已久的数据稀疏性和冷启动问题, 有效提升了推荐结果的准确性和多样性, 但尚未注意到来自恶意用户的数据投毒攻击对服务质量感知云 API 推荐系统的影响.

目前, 服务质量感知云 API 推荐系统面临严峻的数据投毒攻击威胁. 这主要基于以下 3 个基本事实. (1) 云 API 的货币属性. 随着云 API 货币化技术的日趋成熟, 云 API 逐渐被赋予货币属性. 更多的云 API 调用意味着可以给服务提供商带来更高的经济效益, 因此云 API 的货币属性使得广泛使用的服务质量感知云 API 推荐系统成为攻击者重点关注的目标. (2) 网络环境的开放性. 在开放的网络环境中, 攻击者很容易雇佣一些用户作为攻击者, 提供虚假的云 API QoS 反馈数据注入推荐系统的训练集中 (见图 1(b)), 使推荐结果遵循攻击者的意愿, 生成有偏差的云 API 推荐结果. (3) 推荐系统的脆弱性. 已有实验数据表明, 在推荐系统训练数据中注入小规模的恶意用户数据就能使推荐结果产生偏差, 提高目标云 API 的可见性和利用率[12]. 例如, 研究人员在实验环境中发现, 诸如 YouTube、Amazon 和 LinkedIn 等平台推荐系统的性能易受投毒攻击的影响[13]. 因此, 针对服务质量感知云 API 推荐系统设计有效的数据投毒攻击防御方法, 构建可信的云 API 推荐系统成为 API 经济健康发展中亟需解决的现实问题.

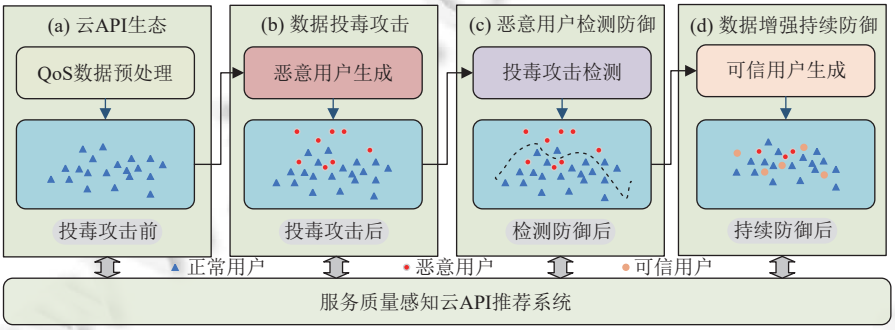


图 1 服务质量感知云 API 推荐系统投毒攻击与防御示意图

现有针对推荐系统的投毒攻击防御方法主要采用“检测防御”策略, 即在模型训练之前利用检测算法检测并滤除恶意用户 (见图 1(c)). 其基本假设是: 恶意用户通常带有特定目的, 恶意用户行为与正常用户行为存在显著差异. 据此, 研究人员提出了基于监督学习的恶意用户检测算法, 例如利用 DegSim 和 RDMA 等人工设计的特征, 在大量有标记的数据集上训练分类器, 以区分正常用户和恶意用户[14]. 然而, 鉴于实际应用中仅有少量有标记数据可用, 而大部分数据未标记, 基于无监督学习的检测算法逐渐成为研究焦点. 这类算法通过聚类、关联规则等方法自主分析数据特征, 以检测恶意用户[15]. 随后, 研究者提出基于半监督学习的检测算法, 旨在利用有限的有标记数据提升检测准确性[16].

尽管基于检测防御策略的方法可以过滤掉部分恶意用户, 降低投毒攻击对推荐系统的影响, 但仍然存在以下不足. (1) 天然的数据稀疏性限制了恶意用户检测的准确率. 现有投毒攻击检测算法都是基于用户与云 API 交互数据来提取用户特征, 并将其视为一个二分类问题来处理. 但目前的攻击方法在保证攻击有效性的同时也会学习真实用户的数据分布来隐藏恶意用户的身份[17]. 因此, 在用户-云 API 交互数据天然高度稀疏的情况下, 传统检测算法很难通过少量的交互数据来准确提取用户特征, 进而导致恶意用户检测的准确率下降. (2) 难以回避的漏检恶意用户使得投毒攻击仍未消除. 相关研究表明, 即使仅注入占用户总量 1% 的恶意用户, 也能对推荐系统的性能产生显著影响[12]. 因此, 即便是小比例的漏检恶意用户, 也依旧能对推荐系统产生攻击效果. 此外, 现有检测算法不仅可能漏检部分恶意用户, 还可能误将正常用户误判为恶意用户, 进一步加剧数据稀疏性问题. 因此, 针对现有检测防御方法难以完全滤除恶意用户这一挑战, 有必要探究不同于传统基于“检测防御”思想的“持续防御”方法, 进一步解决恶意用户投毒攻击的影响 (见图 1(d)).

针对上述问题, 本文提出一种基于可信数据增强的 QoS 感知云 API 推荐系统投毒攻击持续防御方法. 首先, 我们从“以攻学防”的视角, 构建基于可信数据增强的投毒攻击持续防御框架. 该框架通过向被污染的数据集注入可信用户来实现持续防御. 其中, 引入可信用户旨在使数据驱动的推荐系统能够尽可能准确地学习到真实用户和云 API 的特征表示, 从而保证推荐结果的准确性和推荐系统抵御恶意攻击的能力. 其次, 在可信用户生成方面, 提出启发式、基于深度学习以及扩散模型的可信用户生成方法. 其中, 启发式方法利用用户和云 API 历史交互 QoS 数据的统计特征来生成可信用户数据. 基于深度学习的方法通过建模用户和云 API 交互关系来生成可信用户数据. 扩散模型采用迭代去噪的方式学习真实 QoS 数据分布来生成高质量可信用户数据. 最后, 我们在真实世界云 API QoS 数据集上进行大量实验, 并在 11 个经典基线推荐算法上评估本文所提方法的有效性.

综上所述, 本文的主要贡献如下.

(1) 建立基于可信数据增强的投毒攻击持续防御框架. 在投毒攻击已成功的情况下, 通过注入可信数据来消解投毒攻击的影响和增强推荐系统的鲁棒性.

(2) 给出面向推荐系统的投毒攻击问题, 基于可信用户的持续防御问题形式化定义. 引入防御强度和防御规模两种防御策略, 灵活应对不同方式的数据投毒攻击, 提升防御效果.

(3) 提出基于扩散模型的可信用户生成算法. 利用扩散模型在捕捉复杂交互关系方面的优势, 以迭代去噪的方式学习用户与云 API 之间的真实交互, 生成高质量可信用户数据.

(4) 在真实数据集和 11 种推荐算法上进行大量实验, 结果表明利用可信用户可有效消解数据投毒攻击的影响并能够提升推荐系统的鲁棒性. 混合攻击、防御规模和防御强度下的实验进一步验证了所提方法的有效性.

本文将在第 1 节对相关工作进行回顾和总结. 第 2 节详细阐述面向 QoS 感知云 API 推荐系统基于可信数据增强的持续防御框架并给出相关定义. 第 3 节详细介绍基于扩散模型的可信用户生成算法. 第 4 节利用真实云 API QoS 数据集进行实验并分析结果. 最后, 对全文进行总结并对今后的工作进行展望.

1 相关工作

1.1 QoS 感知的云 API 推荐

根据使用技术的不同, 现有 QoS 感知的云 API 推荐方法可被分为 3 类: 基于协同过滤的、基于矩阵分解的和基于深度学习的云 API 推荐方法.

基于协同过滤的推荐方法根据计算方式的不同可细分为基于用户的协同过滤 (user-based collaborative filtering, UCF) 和基于云 API 的协同过滤 (API-based collaborative filtering, ACF)[18]. UCF 的基本思想是利用历史交互数据计算用户之间的相似性, 找到与目标用户偏好相似的其他用户, 然后根据这些相似用户的 QoS 来协同预测未知云 API 的 QoS, 最后根据预测的 QoS 完成高质量云 API 推荐. 不同于 UCF, ACF 根据历史交互数据计算云 API 之间的相似性, 然后把与用户调用过的相类似的云 API 推荐给用户. 进一步, 研究人员提出了混合协同过滤推荐算法[19], 结合上下文信息感知的协同过滤推荐算法[20]来提高推荐结果的准确性. 尽管基于协同过滤推荐算法易于实现和结构易于解释, 但其性能严重依赖于用户和云 API 之间可用的交互数据, 因此在数据稀疏及冷启动场景下, 基于协同过滤的推荐方法将难以有效找到准确的相似邻居, 从而降低了推荐结果的准确性.

基于矩阵分解的推荐方法将用户与云 API 交互的高维稀疏数据投影至两个低维稠密矩阵, 即“用户隐特征矩阵”和“API 隐特征矩阵”, 以此挖掘两者的潜在特征. 然后, 利用用户和云 API 隐特征向量之间的内积运算进行 QoS 预测. 针对大规模数据集, 基于矩阵分解的推荐算法训练需要大量计算资源和时间成本, FunkSVD 算法通过融入随机梯度下降策略与正则项, 既有效缓解了计算耗时过大问题, 又避免了过拟合, 增强了模型的泛化能力[21]. BiasSVD 通过引入偏差项, 更精准地捕捉了交互信息及个体差异[22]. SVD++结合了显式反馈与隐式反馈, 从而更准确地捕获用户偏好[23]. 矩阵分解技术通过低秩近似、隐向量表示及全局拟合策略, 有效应对了推荐系统中的数据稀疏挑战, 但基于矩阵分解的推荐方法通过简单的内积运算难以充分拟合用户与云 API 间复杂非线性的交互关系, 从而限制了其性能表现.

基于深度学习的推荐方法以用户和云 API 静态属性和交互数据作为输入特征, 借助深度学习模型强大的特征表示学习和交互关系挖掘能力, 能有效对用户与云 API 间复杂的非线性交互关系进行建模, 进而实现精准的云 API 推荐. 因子分解机 (factorization machine, FM) 模型[24]将云 API 服务质量预测问题转化为回归任务, 其核心原理是将特征映射至向量空间, 并利用特征向量间的内积来表征特征间的相互影响. 随着深度学习技术的不断进步, 特征提取方式已经变得日益多样化. Zhang 等人[25]提出了一种基于深度学习的 QoS 预测模型, 该模型通过多阶段多尺度特征融合和个体评估来提高预测准确性. 模型首先利用非负矩阵分解提取全局和个体特征, 即使是在冷启动情况下, 也可以通过这些特征来进行一定程度的预测. 然后通过计算用户与服务间的距离相似性获得局部特征. 最终, 模型通过融合这些多尺度特征并结合个体评估, 实现更精准的 QoS 预测.

综上所述, 当前 QoS 感知的云 API 推荐算法研究主要集中于通过解决数据稀疏性、冷启动及特征表示和交互学习等挑战提升推荐精度, 忽视了推荐算法自身的鲁棒性. OWASP 2023 年度报告揭示了云 API 面临的 10 大安全问题, 这些问题可能导致账户控制权丢失、数据泄露等严重后果[26]. 这也让针对云 API 推荐系统的投毒攻击引起广泛关注. 一旦将存在安全隐患的云 API 推荐给开发者, 将会给其实际应用带来严重风险, 进而阻碍云 API 经济的健康发展.

1.2 数据投毒攻击与检测

投毒攻击是指攻击者向推荐系统的训练集中注入恶意用户数据, 影响特定目标云 API 的 QoS 数据分布, 导致推荐系统学习到的特征表示发生偏离, 进而产生有偏的云 API 推荐[27]. 在推荐系统领域, 现有针对投毒攻击的防御方法主要采用“检测防御”策略, 过滤恶意用户数据来降低数据投毒攻击对推荐系统的影响. 现有投毒攻击检测防御方法通常可以分为有监督的检测算法、无监督的检测算法、半监督检测算法.

有监督检测算法的效果高度依赖于数据特征, 这促使研究者关注数据投毒攻击的特征与行为模式. Chirita 等人[28]引入了平均评分偏离度 (RDMA) 特征量化用户向量的差异性. RDMA 通过计算用户属性值, 并与预设阈值比较, 以识别恶意用户. Zhou 等人[29]提出 DL-DRA 算法, 采用双三次插值降低评级矩阵的稀疏性, 并应用结构化深度学习网络进行检测. 充分依托无监督学习无需大量标签数据的优势, 研究者提出了多种无监督检测算法. 例如, Mehta 等人[30]深入分析恶意用户, 并提出了基于 PLSA 软聚类和 PSA 变量选择的无监督算法. 尽管这些算法性能良好, 但需预知部分攻击者信息, 限制了其实用性. 为此, Zhang 等人[31]提出了结合主成分分析 (principal components analysis, PCA) 和数据复杂性的无监督检测方法, 首先利用 PCA 算法筛选可疑用户, 再通过数据复杂性进一步区分真实用户与攻击用户. 该方法相较于单独使用 PCA 算法, 性能更优, 且无需事先确定攻击者数量. 此外, 半监督检测算法因为能利用少量带标签数据提高检测准确度而受到关注. Wu 等人[32]提出了一种半监督检测算法, 该算法融合了朴素贝叶斯分类器和基于选定度量的增强期望最大化方法. Chen 等人[33]利用对抗生成网络实现半监督检测算法. 首先使用干净数据生成合成数据, 以扩充训练数据集; 然后使用带梯度惩罚的条件 Wasserstein 生成对抗网络 (conditional Wasserstein generative adversarial network with gradient penalty, cWGAN-GP) 来训练模仿模型; 最后通过设置检测阈值来识别恶意数据. 目前, 针对服务质量感知云 API 推荐系统的投毒攻击防御研究处于初始阶段, 但研究思路仍采用“检测防御”策略. 例如, 陈真等人[34]提出 NQI-Detector 算法, 从邻域特征、服务质量深度特征及解释特征等多个维度提取用户行为特征, 并结合网格搜索优化恶意用户检测器, 以提升高维稀疏环境下恶意用户的检测效果.

当前, 针对推荐系统的投毒攻击防御方法研究大多聚焦于优化用户数据特征提取, 以提升检测算法精度. 但由于检测算法自身性能的限制, 恶意用户数据漏检是难以避免的, 从而使投毒攻击问题尚未有效解决. 本文从“以攻学防”视角提出基于可信数据增强的持续防御方法, 在投毒攻击已成功的情况下, 通过注入可信数据来降低攻击的影响和增强推荐系统的鲁棒性.

2 基于可信数据增强的持续防御框架

图 2 所示为基于可信数据增强的云 API 推荐系统数据投毒攻击持续防御框架, 包括以下 3 个阶段.

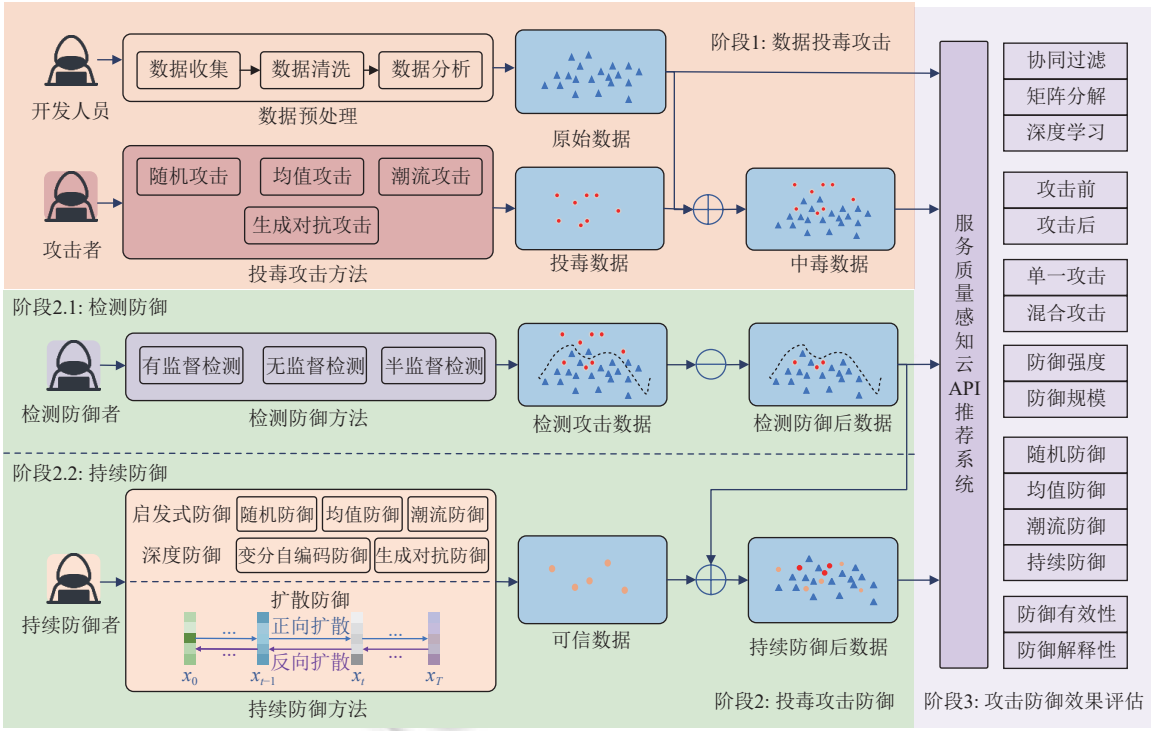


图 2 基于可信数据增强的持续防御框架

(1) 数据投毒攻击. 使用攻击方法如随机攻击、均值攻击、潮流攻击和生成对抗攻击生成恶意用户, 并将这些恶意用户数据注入原始数据集中, 构造被污染的中毒数据.

(2) 投毒攻击防御. 采用两种防御策略进行防御. 1) 检测视角的检测防御, 采用有监督、无监督或半监督的检测算法对中毒数据集中的恶意用户进行检测和剔除. 考虑到现有检测算法在实际应用中存在漏检和误检的现象, 提出在检测防御的基础上进行持续防御. 2) 以攻学防视角的持续防御. 引入可信用户的概念, 并设计可信用户生成算法, 如基于启发式、基于深度学习模型和基于扩散模型的可信用户生成算法, 生成可信用户数据消解恶意数据对模型的影响.

(3) 攻击防御效果评估. 分别将原始数据、中毒数据、检测防御后数据和持续防御后数据作为服务质量感知云 API 推荐系统输入, 从不同推荐系统下的攻击与防御分析、攻击前后分析、单一与混合攻击、防御强度与防御规模影响和防御方法影响等多个维度评估投毒攻击防御方法的有效性和解释性.

2.1 数据投毒攻击

攻击者通过雇佣用户随意提交虚假的 QoS 数据来攻击推荐系统. 然而, 此类攻击数据易被检测算法识别并过滤掉, 故提升恶意用户的隐蔽性对于投毒攻击尤为重要. 目前, 攻击者在构建恶意用户数据时, 在真实用户与云 API 交互数据的基础上, 采用均值攻击、潮流攻击等多种投毒攻击方法生成恶意用户数据. 随后, 将生成的恶意用户数据注入原始数据集中, 完成对云 API 生态原始数据的投毒. 一般地, 数据投毒攻击过程在形式上可表示为:

$$R(U)\oplus R(V)\Rightarrow R(U,V) \tag{1}$$

其中, $\oplus$ 表示数据注入操作, U 和 V 分别表示真实用户和恶意用户集合. $R(U)$ 和 $R(V)$ 分别表示真实用户和恶意用户与云 API 的交互数据集. $R(U,V)$ 表示中毒后的用户-云 API 服务质量数据集.

2.2 可信数据增强的持续防御

2.2.1 可信用户定义

一旦服务质量感知云 API 推荐系统使用被污染的数据训练推荐模型, 攻击者便能够破坏推荐结果, 甚至能够

操纵推荐系统推荐对其有利的云 API. 尽管现有检测算法能够滤除部分恶意用户数据, 但由于用户-云 API 服务质量数据的高度稀疏性, 恶意用户的漏检是难以避免的, 使得投毒攻击仍然存在. 本文提出基于可信数据增强的投毒攻击持续防御方法. 通过注入可信用户来给推荐系统提供更多可用的交互数据, 使其学习到更准确的真实用户与云 API 的特征表示, 消解恶意用户的影响.

定义 1. 可信用户可表示为一个三元组:

$$\tilde{A}_w = (A^S, A^F, A^\phi) \tag{2}$$

其中, $\tilde{A}_w$ 表示可信用户 w 的用户画像, A^S 表示选择填充云 API 集合, A^F 表示随机填充云 API 集合, A^S 和 A^F 都是为了更好地拟合真实数据特征, 提高可信用户的真实性. A^ϕ 是空白云 API 集合, 即用户未调用过的云 API 集合.

由定义 1 可知, 可信用户在本质上是一组用户关于云 API 的 QoS 反馈数据向量, 如图 3 所示. 其中, $\lambda(\cdot)$ 和 $\delta(\cdot)$ 分别表示对不同类别云 API 的 QoS 数据生成方法, Null 表示不进行填充处理.

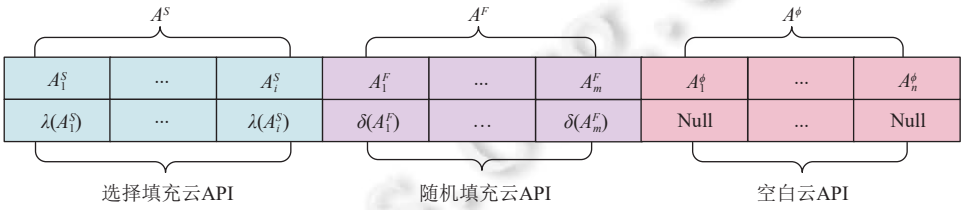


图 3 可信用户一般表示形式

2.2.2 可信用户生成

针对 QoS 感知云 API 推荐系统的数据投毒攻击, 持续防御核心在于注入高质量可信用户给推荐系统提供更丰富的交互数据, 而高质量可信用户生成关键在于可信用户生成算法. 一般地, 可信用户生成算法 $f(\cdot)$ 利用真实用户与云 API 交互数据集 $R(U)$ 生成可信用户数据 $R(\tilde{A})$ 的过程表示为:

$$f(R(U)) = R(\tilde{A}) \tag{3}$$

本文从以攻学防视角提出以下 3 类持续防御模型来设计可信用户生成算法.

(1) 启发式防御. 该类方法的基本思想是利用数据统计分析生成可信用户数据. 根据计算方法的不同, 可分为随机防御、均值防御、潮流防御.

(2) 深度防御. 该类方法的基本思想是利用深度学习模型, 如变分自编码器 (variational autoencoder, VAE)、生成对抗网络 (generative adversarial network, GAN), 建模用户和云 API 交互关系来生成可信用户数据.

(3) 扩散防御. 利用扩散模型对用户和云 API 交互信息进行建模, 模拟并学习复杂用户行为, 通过正向扩散和反向扩散两个过程以迭代去噪的方式生成可信用户.

利用启发式防御、深度防御和扩散防御模型生成可信用户选择填充云 API A^S、随机填充云 API A^F 和空白云 API A^ϕ 的部署方式如表 1 所示.

表 1 持续防御模型部署方式

防御类型	防御模型	A^S (选择填充云API)	A^F (随机填充云API)	A^ϕ (空白云API)
启发式防御	随机防御 (Rnd_D)	Null	$\delta(A^F) = r_{\text{random}}$	Null
	均值防御 (Avg_D)	Null	$\delta(A^F) = r_{\text{average}}$	Null
	潮流防御 (Bdg_D)	$\lambda(A^S) = r_{\max}$	$\delta(A^F) = r_{\text{average}}$	Null
深度防御	变分自编码防御 (Vae_D)	$\lambda(A^S) = r_{\text{vae}}$	$\delta(A^F) = r_{\text{vae}}$	Null
	生成对抗防御 (Gan_D)	$\lambda(A^S) = r_{\text{gan}}$	$\delta(A^F) = r_{\text{gan}}$	Null
扩散防御	扩散防御 (Diff_D)	$\lambda(A^S) = r_{\text{diffusion}}$	$\delta(A^F) = r_{\text{diffusion}}$	Null

(1) 随机防御根据正态分布 $r_{\text{random}} \sim N(\mu,\sigma)$ 为随机填充云 API 生成 QoS 反馈值, 其中 μ 和 σ 分别是数据集中所有 QoS 反馈值的均值和方差.

(2) 均值防御根据正态分布 $r_{\text{average}} \sim N(\mu_m,\sigma_m)$ 为随机填充云 API m 生成 QoS 反馈值, 其中 μ_m 和 σ_m 是云 API m QoS 反馈值的均值和方差.

(3) 潮流防御选择流行云 API 作为选择填充云 API, 以此来增强推荐的多样性. $r_{\max/\min}$ 表示数据集中的最高或者最低的 QoS 反馈值.

(4) 变分自编码防御采用变分自编码器为填充云 API 生成 QoS 反馈值.

(5) 生成对抗防御采用生成对抗网络为填充云 API 生成 QoS 反馈值.

(6) 扩散防御利用扩散模型为填充云 API 生成 QoS 反馈值.

2.2.3 持续防御

已有研究表明, 投毒攻击效果与恶意用户的数量呈正相关关系, 即数据集中恶意用户越多, 推荐系统学习到的特征表示越偏离正常表示[12]. 类似地, 本文通过注入可信用户, 为推荐系统提供更多可靠的用户与云 API 交互数据, 帮助推荐系统学习到更准确的真实用户和云 API 的特征表示, 从而提升推荐的性能. 在形式上, 基于可信数据增强的持续防御可以表示为:

$$R(U,V)\oplus R(\tilde{A}) \Rightarrow R(U,V,\tilde{A}) \tag{4}$$

其中, $\tilde{A}$ 表示可信用户集合, $R(\tilde{A})$ 表示可信用户与云 API 交互数据集. $R(U,V,\tilde{A})$ 表示注入可信用户后的数据集.

推荐系统 M 利用不同数据集 $R(\cdot)$ 训练后, 对目标用户 u 和目标云 API a 的服务质量预测表示如下:

$$M(u,a|R(\cdot)) = \hat{y}_{u,a} \tag{5}$$

根据预测值与实际值的偏差即可评估针对投毒攻击与防御方法的有效性. 此外, 还可以利用不同推荐模型、单一与混合攻击以及不同防御方法等多个维度评估攻击与防御的效果.

为了灵活应对不同方式的数据投毒攻击, 以及全面评估基于可信用户增强的持续防御方法, 引入防御强度和防御规模两种防御策略进一步提升防御效果.

定义 2. 防御强度. 防御强度 γ 是可信用户调用过的云 API 数量与云 API 总数的比率. 利用防御强度 γ 生成可信用户的过程如下:

$$f(R(U),\gamma) = R_{\gamma}(\tilde{A}) \tag{6}$$

防御强度越大, 意味着可信用户与更多的云 API 发生交互, 丰富的交互数据能够提供更全面的用户偏好和行为模式, 有利于推荐系统学到更好的特征表示, 进而提升推荐的准确性.

定义 3. 防御规模. 防御规模 η 是注入的可信用户的数量与用户总数的比率. 利用生成不同防御规模 η 可信用户过程如下:

$$\xi(f(R(U),\gamma),\eta) \Rightarrow R_{\gamma,\eta}(\tilde{A}) \tag{7}$$

其中, $\xi(\cdot)$ 控制注入的可信用户的数量. 防御规模越大意味着更多的可信用户被注入训练集中, 对 QoS 感知云 API 推荐系统产生更大的影响. 由于大多数云 API 调用都是付费的, 调用更多的云 API 意味着更多的成本. 因此, 在构建基于不同防御规模的可信用户数据集时, 需要平衡防御效果与防御成本之间的关系.

3 基于扩散模型的可信用户生成算法

基于可信数据增强的投毒攻击持续防御的核心是高质量的可信用户. 尽管基于启发式防御的方法利用数据统计特征生成可信用户数据易于理解, 但其生成的可信用户可能过于模式化, 无法充分反映真实用户的个性化偏好. 相比之下, 深度生成式方法在数据生成质量和自动化效率等方面展现出显著优势. 其中, 扩散模型凭借其卓越的生成能力、坚实的理论基础以及广泛的应用前景, 已成为该领域的重要研究方向[35,36]. 与变分自编码器和对抗生成网络等相比, 扩散模型在生成数据的真实性和多样性以及训练稳定性等方面表现更优[37]. 因此本文提出了基于扩

散模型的可信用户生成算法. 该算法通过迭代去噪的方式, 充分利用历史交互数据, 逐步恢复用户的个性化偏好, 从而生成高质量的可信用户. 图 4 展示了利用扩散模型生成可信用户的基本过程.

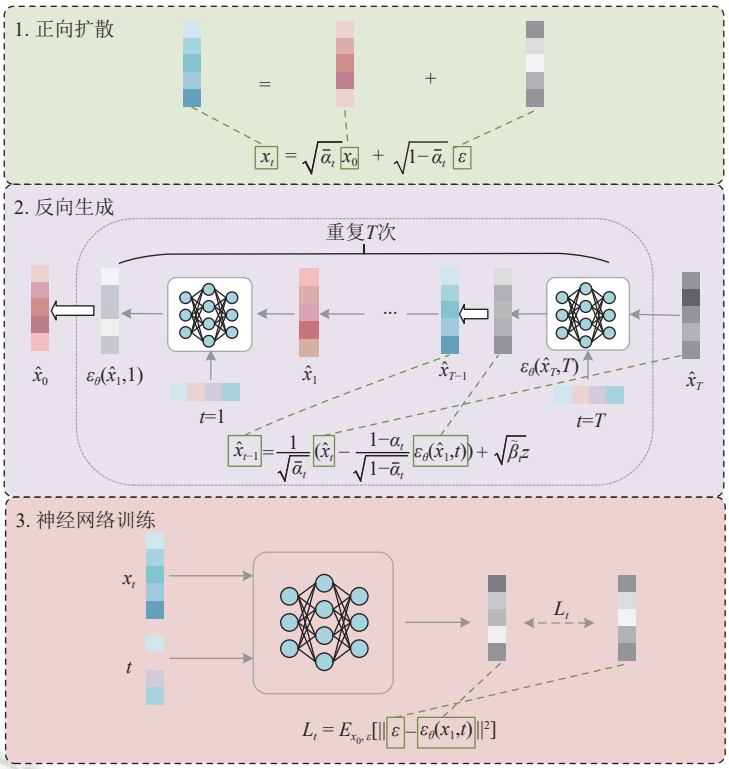


图 4 基于扩散模型的可信用户生成过程

(1) 正向扩散. 随机采样高斯噪声, 获得不同时间步长的加噪数据, 模拟真实世界中由于用户反馈不准确等原因造成的噪声交互, 并为后续神经网络训练提供数据支持.

(2) 反向生成. 通过迭代去噪恢复被噪声污染的交互信息, 推断用户未来的交互行为, 最后生成可信用户数据.

(3) 神经网络训练. 由于反向生成过程无法通过数学推理直接得出, 需要设计一个去噪神经网络来拟合并逼近反向生成过程.

3.1 正向扩散过程

给定真实用户向量 $x_0 \sim q(x_0)$ 作为初始数据, 正向扩散过程经过 T 步迭代将较小的噪声逐渐地加入初始数据中得到被噪声污染的数据 $\{x_1, x_2, \ldots, x_T\}$. 每一步加噪过程表示如下:

$$x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\varepsilon_t \tag{8}$$

其中, $t \in \{1, 2, \ldots, T\}$ 表示前向扩散过程中的步骤, $\varepsilon_t \in \mathcal{N}(0, \mathrm{I})$ 表示 t 时刻随机采样的高斯噪声向量, 加入每一步的噪声的大小是由超参数 $\{\beta_t \in (0, 1)\}_1^T$ 控制的. 正向扩散过程中当前时刻的数据 x_t 只与前一时刻数据 x_{t-1} 有关, 所以从 x_0 到 x_T 是一个马尔可夫链过程, 满足公式 (9):

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathrm{I}), q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}) \tag{9}$$

在每一步的噪声添加过程中, 噪声是从一个已知的分布中独立采样的. 这意味着噪声项与模型的参数无关, 因此在正向扩散过程中, 每一步都是可以根据公式 (8) 精确计算的. 利用重参数化技巧和两个独立高斯噪声的可加

性[38], 可以直接从 x_0 得到 x_t, 转换形式为公式 (10):

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\varepsilon \tag{10}$$

其中, $\alpha_t = 1-\beta_t$, $\bar{\alpha}_t = \prod_{i=1}^{t}\alpha_i$, $\varepsilon \in \mathcal{N}(0, \mathrm{I})$, I 表示单位矩阵. 因此, 正向扩散过程满足公式 (11):

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathrm{I}) \tag{11}$$

3.2 反向生成过程

在反向生成过程中, 不只是对 QoS 反馈值通过数学运算进行去噪, 而是要将用户、云 API 及其对应的 QoS 反馈值整合为一个具有实际意义的整体, 理解并重建用户复杂的交互行为以及云 API 之间的关联. 给定用户的历史交互, 在正向过程中添加噪声来逐渐破坏它们, 反向生成过程通过迭代去噪恢复原始交互. 如果能够从 $q(x_{t-1}|x_t)$ 中采样, 就可以从高斯噪声 x_T 推理得到 $x_0 \sim q(x_0)$. 由于 $q(x_{t-1}|x_t)$ 无法简单推断, 本文设计一个神经网络 $p_\theta(x_{t-1}|x_t)$ 来拟合反向扩散过程.

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}\left(x_{t-1}; \mu_\theta(x_t,t), \sum_\theta(x_t,t)\right) \tag{12}$$

其中, $\mu_\theta(x_t,t)$ 和 $\sum_\theta(x_t,t)$ 是由神经网络预测的高斯分布均值和协方差, 神经网络具有可学习参数 θ. 定义 $\sum_\theta(x_t,t) = \tilde{\beta}_t$. 给定 x_0 可计算得到条件概率 $q(x_{t-1}|x_t, x_0)$.

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}_t(x_t, x_0, t), \tilde{\beta}_t\mathrm{I}) \tag{13}$$

利用公式 (9) 和 (11) 以及贝叶斯公式计算条件概率 $q(x_{t-1}|x_t, x_0)$ 的均值 $\tilde{\mu}_t$ 和方差, 如公式 (14) 和 (15) 所示:

$$\tilde{\mu}_t = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon\right) \tag{14}$$

$$\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t \tag{15}$$

3.3 神经网络训练

神经网络的优化目标是使其预测的 $p_\theta(x_{t-1}|x_t)$ 分布更加接近准确的条件概率分布 $q(x_{t-1}|x_t, x_0)$. 故而在此, 本文使用 KL 散度计算两者之间的差异, 化简后得到损失函数 L_t.

$$L_t = E_{q(x_t|x_0)}[\|q(x_{t-1}|x_t, x_0) - p_\theta(x_{t-1}|x_t)\|^2]\tilde{\beta}_t \tag{16}$$

由于两个分布的协方差都是常数, 神经网络的优化目标可简化为使预测的均值 $\mu_\theta(x_t,t)$ 更接近真实的均值. $\mu_\theta(x_t,t)$ 可定义为:

$$\mu_\theta(x_t,t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon(x_t,t)\right) \tag{17}$$

由公式 (14) 和 (17) 可知均值只和超参数 x_t 和 ε 有关. 最后, 神经网络训练过程中的损失函数如公式 (18) 所示:

$$L_t = E_{x_0,\varepsilon}[\|\varepsilon - \varepsilon_\theta(x_t,t)\|^2] = E_{x_0,\varepsilon}\left[\left\|\varepsilon - \varepsilon_\theta\left(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\varepsilon, t\right)\right\|^2\right] \tag{18}$$

神经网络不需要直接预测条件概率的均值, 只需要预测出数据 x_t 被添加的噪声 $\varepsilon_\theta(\cdot)$. 噪声预测网络的输入为 x_t 和步长 t 的嵌入向量, 输出为预测的噪声向量. 利用训练好的神经网络通过公式 (19) 将一个高斯噪声数据通过迭代去噪生成高质量的可信用户数据, 其中 $z \in \mathcal{N}(0, \mathrm{I})$.

$$\hat{x}_{t-1} = \mu_\theta(\hat{x}_t,t) + \tilde{\beta}_t z = \frac{1}{\sqrt{\alpha_t}}\left(\hat{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon_\theta(\hat{x}_t,t)\right) + \sqrt{\tilde{\beta}_t}z \tag{19}$$

利用扩散模型生成可信用户时, 通过深入理解用户与云 API 之间的关系, 能够生成符合真实用户个性化偏好的高质量可信用户, 不局限于重现已有的用户交互信息. 这有助于推荐系统学习到真实用户与云 API 的特征表示. 算法 1 给出了基于扩散模型的可信用户生成算法并给出了计算复杂度的分析.

算法 1. 基于扩散模型的可信用户生成算法.

输入: 神经网络参数 θ, 真实数据集 $R(A_u)$;
输出: 可信用户向量 $\hat{x}_0$.

1. 随机抽取 $x \subset R(A_u)$ //O(1)
2. **for** all $x_0 \in x$ **do** //O(|x|)
3. 采样 $t \sim U(1, T)$, $\varepsilon \in \mathcal{N}(0, \mathrm{I})$ //O(1)
4. 通过公式 (8) 计算 x_t //O(D)
5. 优化神经网络参数 θ 通过 $\nabla_\theta \|\varepsilon - \varepsilon_\theta(x_t, t)\|^2$ //O(|θ|)
6. **end for**
7. **for** $t = T, \ldots, 1$ **do** //O(T)
8. $z \sim \mathcal{N}(0, \mathrm{I})$ if $t > 1$, else $z = 0$ //O(D)
9. 通过公式 (19) 计算 $\hat{x}_0$ //O(1)
10. **end for**
11. **return** $\hat{x}_0$

基于扩散模型的可信用户生成算法的计算复杂度主要源于两个部分: 正向扩散过程与神经网络训练 (第 2–6 行) 的复杂度为 O(|x|×(D+|θ|)), 以及反向生成过程 (第 7–10 行) 的复杂度为 O(T×D), 总体计算复杂度为 O(|x|×(D+|θ|)+T×D). 其中, |x|为训练样本数量, D 为数据大小, |θ|为网络参数数量, T 为扩散步长. 由于本文使用的数据集规模较小且网络参数量有限, 算法的复杂度主要由扩散步长 T 决定.

4 实　验

为了探究基于可信数据增强的持续防御方法对 QoS 感知云 API 推荐系统投毒攻击的防御效果, 进行了一系列的实验, 以回答以下研究问题.

RQ1: 可信数据增强的持续防御是否能够产生防御效果?

RQ2: 为什么可信用户数据可以产生防御作用?

RQ3: 持续防御在混合投毒攻击下是否有效?

RQ4: 防御强度如何影响持续防御效果?

RQ5: 防御规模如何影响持续防御效果?

4.1 实验设置

(1) 数据集. 实验使用真实响应时间 QoS 数据集 WS-DREAM. 该数据集记录了 339 个用户对分布在全球的 5 825 个云 API 服务的响应时间. 在开放的网络环境中, 每个用户通常只会调用少量的云 API, 因此用户与云 API 之间的交互矩阵非常稀疏. 为了模拟真实的云 API 应用场景, 从数据集中随机采样 5% 的数据作为训练集, 剩余的 95% 用作测试集. 实验使用的响应时间数据集统计特征如表 2 所示.

表 2　实验数据集的统计特征

统计特征	值
用户数量	339
云API数量	5 825
数据范围	(0, 20]
响应时间平均值	0.908 5

(2) 评价指标. 为了评估云 API 推荐系统的预测性能, 本文采用两个广泛使用的评估指标: 平均绝对误差 (mean absolute error, *MAE*) 和均方根误差 (root mean square error, *RMSE*) 衡量推荐系统的预测性能. *MAE* 和 *RMSE* 定义如下:

$$MAE = \frac{1}{N}\sum_{i=1}^{n}|\hat{y}_{u,a} - y_{u,a}| \tag{20}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{n}(\hat{y}_{u,a} - y_{u,a})^2} \tag{21}$$

其中, N 表示测试集的大小, $\hat{y}_{u,a}$ 和 $y_{u,a}$ 分别表示预测的 QoS 值和实际的 QoS 值. *MAE* 和 *RMSE* 反映了预测值与真实值之间的偏差. *MAE* 和 *RMSE* 的值越低表示预测结果与实际观测值之间的偏差越小, 即预测精度越高.

(3) 基线推荐模型. 选择 3 类 11 种代表性的服务质量感知云 API 推荐系统 (见表 3), 验证基于可信数据增强的持续防御方法的有效性.

表 3 基线推荐模型

类别	方法	关键思想
协同过滤	UPCC	选择用户侧相似邻域进行预测, 邻域大小为20, 相似模型为PCC
	APCC	选择API侧相似邻域进行预测, 邻域大小为20, 相似模型为PCC
	WSRec	WSRec是UPCC和APCC的组合, 其中融合系数为0.5
矩阵分解	FunkSVD	通过矩阵分解将交互矩阵分为用户相关矩阵和API相关矩阵进行预测
	BiasSVD	使用矩阵分解同时考虑用户和云API的偏好, 将交互矩阵分为用户相关矩阵和API相关矩阵进行预测
	SVD++	使用矩阵分解并结合用户的反馈, 将交互矩阵分为用户相关矩阵和API相关矩阵来进行预测
	NMF	使用矩阵分解并引入非负约束和局部性特性, 将交互矩阵分为用户相关矩阵和API相关矩阵来进行预测
深度学习	LR	使用经典的线性回归模型进行预测
	MLP	使用用户和API之间复杂的非线性关系进行预测
	AFM	使用二阶特征交互并引入注意力机制来建模用户和API之间的交互进行预测
	DeepFM	结合FM和MLP的优点, 增强模型对特征交互的建模能力和学习高阶特征之间的复杂关系

(4) 参数设置. 采用随机攻击、均值攻击、潮流攻击、生成对抗网络攻击这 4 种恶意用户生成算法进行数据投毒攻击. 默认恶意用户数量占整体用户比例为 10%. 利用启发式防御、深度防御和扩散防御模型生成可信用户, 防御强度默认为 5%, 防御规模默认为 10%. 在扩散防御中对协方差 β 的取值一般选用线性增加的方式, 即 $\beta_1 = 10^{-4}$ 且 $\beta_T = 0.002$, T=100.

4.2 持续防御的有效性

为了验证基于可信数据增强的持续防御方法是否能对投毒攻击产生防御效果, 使用表 1 中描述的 3 种启发式防御算法、2 种深度防御算法和扩散防御算法生成可信用户数据. 具体而言, 每种算法分别生成占数据集总量 10% 的可信用户数据, 并将其注入遭受攻击的数据集中. 表 4 和表 5 分别展示了 11 种服务质量感知云 API 推荐算法在攻击前后以及持续防御后的 *MAE* 和 *RMSE* 变化结果.

表 4 攻击前后、持续防御后推荐系统 *MAE* 的对比

推荐算法	攻击方式	无攻击	仅攻击	Avg_D	Bdg_D	Rnd_D	Vae_D	Gan_D	Diff_D
UPCC	Rnd_A	0.7110	0.7960	0.8037	0.8034	0.8548	0.8437	**0.7923**	0.8085
	Avg_A		0.7651	0.7700	0.7770	0.8166	0.8132	**0.7622**	0.7743
	Bdg_A		0.7639	0.7761	0.7755	0.8157	0.8120	**0.7613**	0.7732
	Gan_A		0.7644	0.7761	0.7763	0.8177	0.8131	**0.7619**	0.7750
APCC	Rnd_A	1.0784	1.0970	**0.9086**	1.0933	0.9852	1.1004	1.1606	0.9939
	Avg_A		1.0360	**0.8527**	1.0185	0.9937	1.0352	1.0585	0.9807
	Bdg_A		1.0095	0.9397	**0.8639**	0.9180	1.0109	0.9278	0.9313
	Gan_A		1.0742	1.0589	**1.0202**	1.0983	1.0969	1.0450	1.0793
WSRec	Rnd_A	0.8429	0.9082	**0.8123**	0.9059	0.8887	0.9344	0.9392	0.8694
	Avg_A		0.8470	**0.7541**	0.8440	0.8657	0.8714	0.8583	0.8362
	Bdg_A		0.8326	0.8022	**0.7596**	0.8240	0.8582	0.7892	0.8101
	Gan_A		0.8671	0.8659	**0.8455**	0.9206	0.9044	0.8520	0.8876

表 4 攻击前后、持续防御后推荐系统 *MAE* 的对比 (续)

推荐算法	攻击方式	无攻击	仅攻击	Avg_D	Bdg_D	Rnd_D	Vae_D	Gan_D	Diff_D
BiasSVD	Rnd_A	0.5743	0.5778	0.5806	0.5803	0.5832	0.5896	0.5785	**0.5773**
	Avg_A		0.5853	**0.5834**	0.5837	0.6003	0.5976	0.5886	0.5895
	Bdg_A		0.5852	**0.5824**	0.5831	0.5981	0.5967	0.5874	0.5878
	Gan_A		0.5948	0.5879	**0.5876**	0.6007	0.6028	0.5925	0.5921
FunkSVD	Rnd_A	0.5605	0.5852	0.5912	0.5910	0.5962	0.6005	0.5853	**0.5847**
	Avg_A		0.5811	0.5859	0.5867	0.5948	0.5974	0.5825	**0.5807**
	Bdg_A		0.5812	0.5863	0.5873	0.6047	0.5949	0.5827	**0.5808**
	Gan_A		0.5748	0.5814	0.5817	0.597	0.5895	0.5763	**0.5747**
NMF	Rnd_A	0.5648	0.5814	0.5851	0.5866	0.5910	0.5973	0.5817	**0.5810**
	Avg_A		0.5783	0.5819	0.5873	0.6021	0.5948	0.5797	**0.5779**
	Bdg_A		0.5781	0.5822	0.5830	0.6017	0.5950	0.5799	**0.5776**
	Gan_A		0.5742	0.5794	0.5793	0.5974	0.5760	0.5759	**0.5742**
SVD++	Rnd_A	0.5663	0.5961	**0.5924**	0.5944	0.6020	0.6002	**0.5924**	0.5952
	Avg_A		0.5882	0.5889	0.5899	0.5892	0.5904	**0.5814**	0.5836
	Bdg_A		0.5917	0.5922	0.5926	0.5909	0.5950	**0.5846**	0.5858
	Gan_A		0.5893	0.5871	0.5889	0.5889	0.5932	**0.5813**	0.5840
DeepFM	Rnd_A	0.5123	0.5988	0.5911	0.5941	0.5849	0.5913	0.5876	**0.5757**
	Avg_A		0.6053	**0.5839**	0.5911	0.5940	0.5947	0.5882	0.5890
	Bdg_A		0.5934	0.5913	0.5906	0.5951	**0.5785**	0.5926	0.5925
	Gan_A		0.5993	0.5887	0.5957	0.5934	0.5987	0.5949	**0.5869**
MLP	Rnd_A	0.5121	0.5956	0.5865	0.5882	0.5883	0.5918	0.5888	**0.5806**
	Avg_A		0.5973	0.5847	0.5848	**0.5818**	0.5886	0.5850	0.5851
	Bdg_A		0.5976	0.5840	0.5874	0.5974	0.5840	0.5887	**0.5749**
	Gan_A		0.5985	0.5765	0.5735	0.5763	0.5853	0.5749	**0.5733**
LR	Rnd_A	0.6627	0.6973	0.6889	0.6879	0.6822	0.6757	0.7110	**0.6625**
	Avg_A		0.7229	0.7079	0.7067	0.6928	0.6905	0.7146	**0.6652**
	Bdg_A		0.7232	0.7072	0.7056	0.6836	0.6898	0.7195	**0.6648**
	Gan_A		0.6843	0.6985	0.6973	0.6758	0.6825	0.6876	**0.6618**
AFM	Rnd_A	0.7861	0.8805	0.7970	0.7924	0.7987	0.7963	**0.7755**	0.7853
	Avg_A		0.8565	0.7905	0.7920	0.7916	0.8185	0.7767	**0.7609**
	Bdg_A		0.8830	0.7740	0.7960	0.8036	0.7868	0.7683	**0.7562**
	Gan_A		0.8547	0.7829	0.8092	0.7923	0.8046	0.7710	**0.7518**

根据表 4 和表 5 的实验结果, 可以得出以下结论.

(1) 推荐系统易受到投毒攻击的影响. 对比无攻击和攻击后推荐系统的指标可知, 攻击者注入 10% 的恶意用户数据就可以损害大多数云 API 推荐系统的结果, 说明当前云 API 推荐领域存在安全隐患.

(2) 基于可信数据增强的持续防御方法能够有效缓解投毒攻击对推荐性能的影响. 通过对比推荐系统在遭受攻击后和引入持续防御后的性能指标, 可以发现 6 种防御模型均能降低数据投毒攻击的影响, 提升推荐系统的鲁棒性. 生成的可信用户数据质量能够满足防御要求, 甚至在某些推荐系统中, 引入持续防御后的性能指标优于未受攻击时的表现. 例如, 对于 APCC 算法, 无攻击时 *MAE* 值为 1.0784, 扩散防御后 *MAE* 值为 0.9807. 因为可信用户的注入为数据稀疏云 API 增加了可利用的交互数据.

(3) 扩散防御有更好的防御效果. 通过对比不同持续防御方法下推荐系统的指标可知, 3 种启发式防御模型和变分自编码防御模型在基于矩阵分解的推荐算法上表现并不理想, 而扩散防御模型在大多数推荐系统上均展现出良好的防御效果, 防御效果更加稳定. 出现这种现象的原因是由于扩散模型以迭代去噪的方式生成可信用户向量, 使其能够更准确地学习到用户的个性化信息, 生成更高质量可信用户, 进而有助于推荐系统更准确地捕捉用户与云 API 的真实交互特征. 虽然生成对抗防御模型的防御效果也具有一定优势, 但其在模型训练过程中难以找到生成器和鉴别器之间的平衡点, 且容易出现模式崩溃等问题, 这在一定程度上限制了应用效果.

(4) 基于深度学习的推荐算法的鲁棒性较差一些. 基于深度学习的推荐算法易受到数据的影响, 无论是恶意用户注入还是可信用户注入都会给推荐结果带来较大的影响. 例如, 对于 DeepFM 算法, 无攻击时 *MAE* 值为 0.5123, 随机攻击后 *MAE* 值变为 0.5988, 扩散防御后 *MAE* 值为 0.5757. 对于 LR 算法, 无攻击时 *MAE* 值为 0.6627, 随机攻击后 *MAE* 值变为 0.6973, 扩散防御后 *MAE* 值为 0.6625.

表 5 攻击前后、持续防御后推荐系统 *RMSE* 的对比

推荐算法	攻击方式	无攻击	仅攻击	Avg_D	Bdg_D	Rnd_D	Vae_D	Gan_D	Diff_D
UPCC	Rnd_A	1.5739	1.7036	1.6951	**1.6939**	1.7070	1.7275	1.7076	1.6976
	Avg_A		1.7321	1.7259	1.7254	1.7274	1.7547	1.7351	**1.7188**
	Bdg_A		1.7310	1.7249	1.7229	1.7260	1.7534	1.7344	**1.7171**
	Gan_A		1.7262	1.718	1.717	1.7233	1.7490	1.7298	**1.7163**
APCC	Rnd_A	1.8998	1.8993	**1.8446**	1.8951	1.8495	1.8995	1.9236	1.8578
	Avg_A		1.8878	1.8649	1.8833	**1.8586**	1.8863	1.8924	1.8629
	Bdg_A		1.8811	1.8691	1.8618	**1.8449**	1.8804	1.8658	1.8478
	Gan_A		1.8979	1.8927	**1.8848**	1.8983	1.9052	1.8906	1.8949
WSRec	Rnd_A	1.5810	1.6388	**1.6120**	1.6333	1.6443	1.6529	1.6520	1.6466
	Avg_A		1.6213	**1.6126**	1.6198	1.6319	1.6327	1.6267	1.6353
	Bdg_A		1.6173	1.6140	**1.6081**	1.6214	1.6291	1.6144	1.6290
	Gan_A		1.6288	1.6251	**1.6204**	1.6525	1.6403	1.6243	1.6501
BiasSVD	Rnd_A	1.3549	1.3544	1.3568	1.3597	1.3591	1.3637	1.3553	1.3560
	Avg_A		1.4224	**1.4132**	1.4132	1.4344	1.4314	1.4268	1.4296
	Bdg_A		1.4151	**1.4042**	1.4055	1.4224	1 .4207	1.4167	1.4180
	Gan_A		1.3955	**1.3779**	1.3812	1.4005	1.4029	1.3933	1.3950
FunkSVD	Rnd_A	1.4226	1.4363	1.4371	1.4398	**1.4201**	1.4450	1.4355	1.4321
	Avg_A		1.4831	1.4780	1.4811	**1.4620**	1.4861	1.4820	1.4775
	Bdg_A		1.4778	1.4760	1.4778	**1.4567**	1.4788	1.4771	1.4723
	Gan_A		1.4558	1.4532	1.4545	**1.4367**	1.4618	1.4552	1.4507
NMF	Rnd_A	1.4261	1.4323	1.4276	1.4317	**1.4145**	1.4457	1.4315	1.4272
	Avg_A		1.4755	1.4676	1.4815	**1.4531**	1.4862	1.4745	1.4688
	Bdg_A		1.4730	1.4670	1.4678	**1.4505**	1.4859	1.4725	1.4665
	Gan_A		1.4550	1.4491	1.4482	**1.4346**	1.4546	1.4539	1.4489
SVD++	Rnd_A	1.3516	1.3946	**1.3847**	1.3889	1.3904	1.3895	1.3851	1.3903
	Avg_A		1.4079	1.4017	1.4040	1.4029	1.4054	**1.3975**	1.4017
	Bdg_A		1.4073	1.3999	1.4025	1.3993	1.4046	**1.3967**	1.3983
	Gan_A		1.3895	1.3811	1.3845	1.3831	1.3862	**1.3783**	1.3827
DeepFM	Rnd_A	1.3276	1.5106	1.4706	**1.4689**	1.4956	1.4991	1.5080	1.4795
	Avg_A		1.5376	1.5048	**1.4729**	1.5179	1.5047	1.5172	1.4936
	Bdg_A		1.5162	1.4833	**1.4803**	1.4995	1.5121	1.5138	1.4882
	Gan_A		1.5164	1.4871	**1.4869**	1.4989	1.4889	1.5014	1.4931
MLP	Rnd_A	1.3281	1.5056	1.4891	1.5063	1.5007	1.4915	1.4893	**1.4884**
	Avg_A		1.5085	1.5056	1.5071	1.5040	1.4976	1.5044	**1.4957**
	Bdg_A		1.5033	1.4917	1.5012	1.5034	1.5025	1.5056	**1.4901**
	Gan_A		1.5170	1.4967	1.4973	1.5086	1.5145	1.4974	**1.4942**
LR	Rnd_A	1.4080	1.5087	**1.4884**	1.4887	1.5177	1.5005	1.5097	1.5122
	Avg_A		1.5086	**1.4862**	1.4863	1.4918	1.4937	1.5084	1.5078
	Bdg_A		1.5073	1.4852	**1.4847**	1.4955	1.4928	1.5082	1.5072
	Gan_A		1.4892	1.4826	**1.4825**	1.4950	1.4913	1.4995	1.5091
AFM	Rnd_A	1.5551	1.6244	1.5661	1.5685	**1.5649**	1.5705	1.5781	1.5712
	Avg_A		1.6202	1.5660	1.5735	1.5740	1.5889	1.5701	**1.5643**
	Bdg_A		1.6349	1.5629	1.5749	1.5752	1.5783	1.5852	**1.5629**
	Gan_A		1.6279	1.5653	1.5832	1.5593	1.5837	1.5918	**1.5544**

4.3 可信用户数据产生防御效果的解释

第 4.2 节的实验结果显示, 基于可信数据增强的持续防御方法可以有效抵御投毒攻击. 为了进一步探究原因, 本文采用可视化方法来分析不同方法生成的可信用户数据. 本节实验使用 T-SNE (T-distributed stochastic neighbor embedding) 对可信用户进行可视化分析, 结果如图 5 所示.

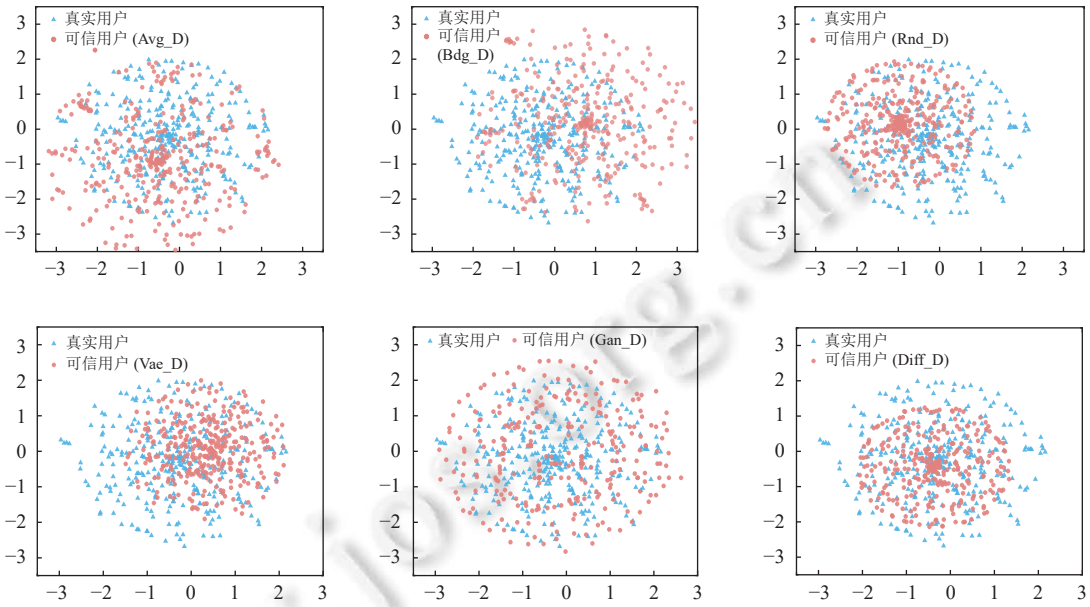


图 5 不同方法生成的可信用户可视分析

根据图 5 可以得出以下结论.

(1) 利用启发式防御、深度防御、扩散防御生成的可信用户数据, 其分布与真实用户数据分布整体十分相似, 证明生成的可信用户质量能够满足防御要求. 将可信用户数据注入推荐系统中可以带来更多的交互数据, 有助于推荐系统学习到更加准确的特征表示, 从而显著提升系统的鲁棒性. 这一结果从数据分布角度验证了可信用户数据的质量并从实际应用效果方面解释可信用户的防御有效性.

(2) 扩散模型可以生成更高质量的可信用户. 因为扩散模型更好地捕捉到真实用户的数据分布, 使得生成的可信用户与真实用户的数据分布紧密相连, 而且两者之间的界限变得更为模糊, 难以区分. 相比之下, 启发式算法和变分自编码器在数据稀疏区域时, 会有部分生成数据出现偏离真实数据分布的情况; 而生成对抗网络会出现模式崩溃问题, 即部分可信用户特征高度相似. 因此, 扩散模型在可信用户生成领域更具应用潜力.

4.4 混合投毒攻击对防御效果的影响

传统检测算法通常被设计用于有效地识别特定类型的恶意用户, 或是基于一种假设, 即投毒攻击的形式是单一的. 为了验证基于可信数据增强的持续防御方法是普遍有效的, 设计了混合投毒攻击实验. 利用 3 种数据投毒攻击方式各自生成占用户总数 5% 的恶意用户, 并将这些恶意用户两两混合后注入数据集, 以模拟混合攻击情况. 在混合攻击中, 恶意用户数量仍占用户总数的 10%, 确保与单一攻击的参数保持一致. 在 APCC、FunkSVD、MLP 这 3 种推荐算法上分别进行实验, 表 6 展示了面对混合攻击时推荐系统 *MAE* 和 *RMSE* 的变化.

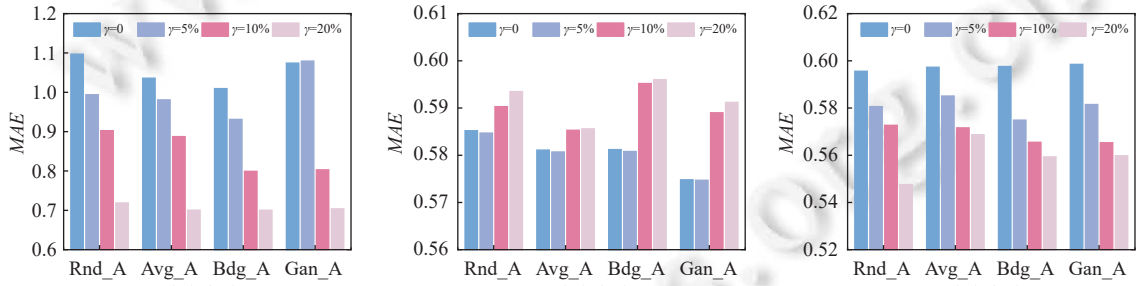
观察表 6 可知, 在应对混合投毒攻击时, 防御方法的表现与面对单一攻击类型时的实验结果 (如第 4.2 节所述) 相似. 在不同的混合攻击下, 3 种推荐算法利用有防御的数据集进行训练后性能都有所提高. 本文提出的基于可信数据增强的持续防御方法通过向数据集中注入可信用户数据来帮助推荐系统学习到更准确的特征表示, 减弱恶意用户数据对推荐系统性能的影响. 因此, 持续防御方法不再局限于保护特定推荐算法或抵御单一投毒攻击方式, 而是能够有效应对复杂多变的混合数据投毒攻击.

表 6 混合投毒攻击下的 *MAE* 和 *RMSE* 变化

指标	推荐算法	攻击方法	无攻击	仅攻击	Avg_D	Bdg_D	Rnd_D	Vae_D	Gan_D	Diff_D
MAE	APCC	Rnd_A+Avg_A	1.0784	1.0838	**0.8695**	1.0179	1.0501	1.0889	1.0876	0.8882
		Avg_A+Bdg_A		1.0837	**0.8702**	1.0179	1.0597	1.0889	1.0876	0.8868
		Bdg_A+Rnd_A		1.1309	0.9100	1.0626	1.0563	1.1339	1.1412	**0.8933**
	FunkSVD	Rnd_A+Avg_A	0.5605	0.5731	0.5778	0.5797	0.5945	0.5915	0.5741	**0.5726**
		Avg_A+Bdg_A		0.5727	0.5775	0.5789	0.5983	0.5915	0.5742	**0.5722**
		Bdg_A+Rnd_A		0.5799	0.5862	0.5871	0.5919	0.5965	0.5803	**0.5794**
	MLP	Rnd_A+Avg_A	0.5121	0.6012	0.5855	0.5878	0.5973	0.5901	0.5941	**0.5854**
		Avg_A+Bdg_A		0.6000	0.5968	**0.5876**	0.5976	0.5965	0.5898	0.5885
		Bdg_A+Rnd_A		0.6010	0.5950	0.5888	0.5917	0.5881	0.5901	**0.5742**
RMSE	APCC	Rnd_A+Avg_A	1.8998	1.9013	1.8689	1.8832	1.8790	1.9026	1.9003	**1.8365**
		Avg_A+Bdg_A		1.9010	1.8687	1.8827	1.8812	1.9026	1.9003	**1.8367**
		Bdg_A+Rnd_A		1.9140	1.8591	1.8893	1.8771	1.9148	1.9178	**1.8348**
	FunkSVD	Rnd_A+Avg_A	1.4226	1.4646	1.4596	1.4633	**1.4408**	1.4756	1.4606	1.4584
		Avg_A+Bdg_A		1.4617	1.4570	1.4596	**1.4409**	1.4756	1.4606	1.4554
		Bdg_A+Rnd_A		1.4388	1.4383	1.4412	**1.4177**	1.4493	1.4381	1.4339
	MLP	Rnd_A+Avg_A	1.3281	1.4932	1.4849	1.4837	1.4844	1.4924	1.4901	**1.4830**
		Avg_A+Bdg_A		1.4938	1.4894	1.4788	1.4812	1.5042	1.4876	**1.4694**
		Bdg_A+Rnd_A		1.5076	1.4795	1.4719	1.4814	1.4856	1.5167	**1.4684**

4.5 防御强度的影响

防御强度越大, 可信用户与越多的云 API 发生交互. 为了探究防御强度对持续防御效果的影响, 我们将防御强度设定为 5%、10%、20%, 其中防御强度为 5% 的可信用户数据集保持了与原始数据集相同的稀疏度. 在 APCC、FunkSVD、MLP 这 3 种推荐算法上, 针对 4 种投毒攻击方式利用扩散模型生成的可信用户数据进行实验. 图 6 和图 7 分别展示了随着防御强度的增加推荐系统的 *MAE* 和 *RMSE* 的变化.



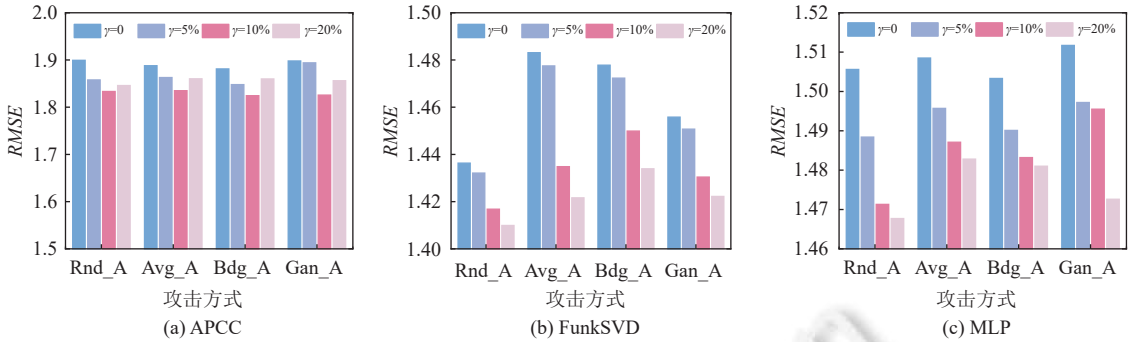
(a) APCC (b) FunkSVD (c) MLP

图 6 不同防御强度下的 *MAE* 变化

由图 6 和图 7 可知, 在不同的攻击方式下, 随着防御强度的提升, *MAE* 和 *RMSE* 呈下降趋势. 因为更大的防御强度意味着可信用户向量填充项数量的增加, 生成了更加稠密的可信用户数据集. 在用户数量不变的情况下, 向数据集中注入更多用户与云 API 交互数据, 有利于推荐系统对 QoS 数值的预测, 进而提高推荐的准确性.

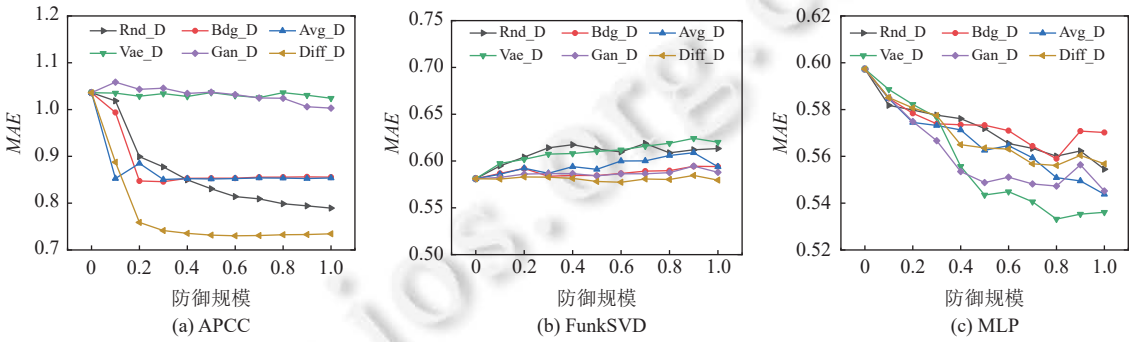
4.6 防御规模的影响

防御规模越大, 注入的可信用户数据就越多. 具体来说, 我们以 10% 的步长将防御规模从 0 增加到 100%, 以分析防御规模的影响. 防御强度固定为 5%. 攻击方式为均值攻击. 在 APCC、FunkSVD、MLP 这 3 种推荐算法上分别进行实验. 图 8 和图 9 分别展示了随着防御规模逐渐增大推荐系统的 *MAE* 和 *RMSE* 的变化趋势.



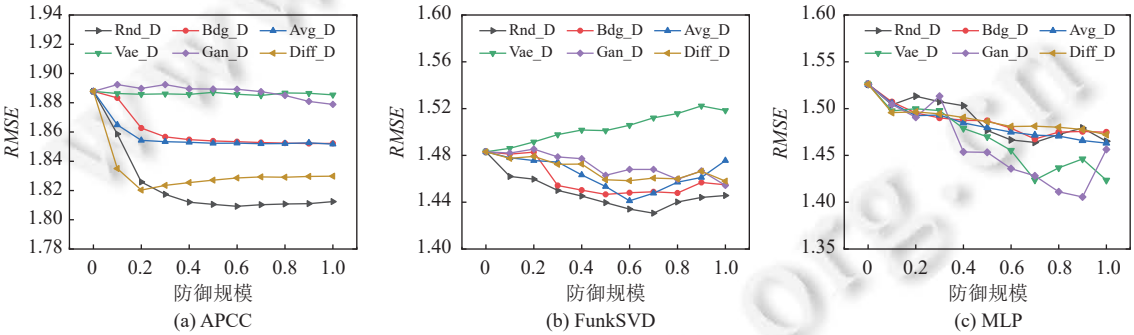
(a) APCC (b) FunkSVD (c) MLP

图 7 不同防御强度下的 *RMSE* 变化



(a) APCC (b) FunkSVD (c) MLP

图 8 不同防御规模下的 *MAE* 变化



(a) APCC (b) FunkSVD (c) MLP

图 9 不同防御规模下的 *RMSE* 变化

由图 8 和图 9 可知, *MAE* 和 *RMSE* 值均随着防御规模的增大而降低, 表明更多的可信用户注入使得推荐系统学得更准确的特征表示, 降低恶意用户的影响, 提高推荐系统的鲁棒性. 但是, 当防御规模达到某个阈值时, 这种下降趋势会明显放缓, 甚至出现上升情况. 这表明, 虽然增加防御规模可以在一定程度上提高防御的有效性, 但是可信用户的质量是非常重要的, 盲目增加防御规模可能难以得到理想的防御效果. 如果可信用户的质量难以保证, 防御规模过大可能不会产生良好的防御效果, 甚至会产生负面影响.

5 总 结

本文针对 QoS 感知的云 API 推荐系统投毒攻击问题, 从“以攻学防”视角提出一种基于可信数据增强的 QoS 感知云 API 推荐系统投毒攻击持续防御方法. 解决了现有检测防御方法难以避免的漏检和误检的不足. 首先, 构建基于可信数据增强的投毒攻击防御框架, 在投毒攻击已成功的情况下, 通过注入可信数据来降低攻击的影响和增

强推荐系统的鲁棒性. 其次, 设计基于扩散模型的可信用户生成算法, 通过迭代去噪的方式深入学习并模拟真实用户的行为特征和数据分布, 生成高质量的可信用户数据. 此外, 在真实的 WS-DREAM 响应时间数据集上对 11 种推荐算法进行实验, 并从混合攻击、防御规模和防御强度等多维度进行分析. 结果表明, 本文提出的基于可信数据增强的投毒攻击持续防御方法能够有效降低投毒攻击的影响. 未来工作将聚焦于优化可信用户生成算法, 为云 API 推荐系统提供更优质的训练数据, 进一步提升推荐算法的鲁棒性.

References

[1] Chen Z, Pan MS, He PF, Qi WC, Liu LL, Shen LM, You DL. Context and auto-interaction are all you need: Towards context embedding based QoS prediction via automatic feature interaction for high quality cloud API delivery. Future Generation Computer Systems, 2022, 128: 265–281. [doi: 10.1016/j.future.2021.10.014]

[2] Gong WW, Zhang XY, Chen YF, He Q, Beheshti A, Xu XL, Yan C, Qi LY. DAWAR: Diversity-aware Web APIs recommendation for mashup creation based on correlation graph. In: Proc. of the 45th Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. Madrid: ACM, 2022. 395–404. [doi: 10.1145/3477495.3531962]

[3] Gamez-Diaz A, Fernandez P, Ruiz-Cortés A. Governify for APIs: SLA-driven ecosystem for API governance. In: Proc. of the 27th ACM Joint Meeting on European Software Engineering Conf. and Symp. on the Foundations of Software Engineering. Tallinn: ACM, 2019. 1120–1123. [doi: 10.1145/3338906.3341176]

[4] Gartner. Gartner magic quadrant for full life cycle API management. 2021. https://www.gartner.com/en/documents/4006268

[5] He Q, Zhou R, Zhang XY, Wang YC, Ye DY, Chen FF, Chen SP, Grundy J, Yang Y. Efficient keyword search for building service-based systems based on dynamic programming. In: Proc. of the 15th Int'l Conf. on Service-oriented Computing. Malaga: Springer, 2017. 462–470. [doi: 10.1007/978-3-319-69035-3_33]

[6] Cao BQ, Liu JX, Wen YP, Li HT, Xiao QX, Chen JJ. QoS-aware service recommendation based on relational topic model and factorization machines for IoT mashup applications. Journal of Parallel and Distributed Computing, 2019, 132: 177–189. [doi: 10.1016/j.jpdc.2018.04.002]

[7] Bermbach D, Wittern E. Benchmarking Web API quality-revisited. Journal of Web Engineering, 2020, 19(5–6): 603–646. [doi: 10.13052/jwe1540-9589.19563]

[8] Zhang YW, Xiang T, Guo X, Jia ZH, He Q. Quality prediction for services based on SOM neural network. Ruan Jian Xue Bao/Journal of Software, 2018, 29(11): 3388–3399 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5474.htm [doi: 10.13328/j.cnki.jos.005474]

[9] Zheng ZB, Li XL, Tang MD, Xie FF, Lyu MR. Web service QoS prediction via collaborative filtering: A survey. IEEE Trans. on Services Computing, 2022, 15(4): 2455–2472. [doi: 10.1109/TSC.2020.2995571]

[10] Chen Z, Shen LM, Li F. Exploiting Web service geographical neighborhood for collaborative QoS prediction. Future Generation Computer Systems, 2017, 68: 248–259. [doi: 10.1016/j.future.2016.09.022]

[11] Hussain W, Merigó JM, Raza MR, Gao HH. A new QoS prediction model using hybrid IOWA-ANFIS with fuzzy C-means, subtractive clustering and grid partitioning. Information Sciences, 2022, 584: 280–300. [doi: 10.1016/j.ins.2021.10.054]

[12] Chen Z, Bao TY, Qi WC, You DL, Liu LL, Shen LM. Poisoning QoS-aware cloud API recommender system with generative adversarial network attack. Expert Systems with Applications, 2024, 238: 121630. [doi: 10.1016/j.eswa.2023.121630]

[13] Xing XY, Meng W, Doozan D, Snoeren AC, Feamster N, Lee W. Take this personally: Pollution attacks on personalized services. In: Proc. of the 22nd USENIX Conf. on Security. Washington: USENIX Association, 2013. 671–686.

[14] Yang F, Gao M, Yu JL, Song YQ, Wang XY. Detection of shilling attack based on Bayesian model and user embedding. In: Proc. of the 30th IEEE Int'l Conf. on Tools with Artificial Intelligence. Volos: IEEE, 2018. 639–646. [doi: 10.1109/ICTAI.2018.00102]

[15] Mehta B, Nejdl W. Unsupervised strategies for shilling detection and robust collaborative filtering. User Modeling and User-adapted Interaction, 2009, 19(1–2): 65–97. [doi: 10.1007/s11257-008-9050-4]

[16] Wu ZA, Wu JJ, Cao J, Tao DC. HySAD: A semi-supervised hybrid shilling attack detector for trustworthy product recommendation. In: Proc. of the 18th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. Beijing: ACM, 2012. 985–993. [doi: 10.1145/2339530.2339684]

[17] Lin C, Chen S, Li H, Xiao YH, Li LY, Yang Q. Attacking recommender systems with augmented user profiles. In: Proc. of the 29th ACM Int'l Conf. on Information & Knowledge Management. New York: ACM, 2020. 855–864. [doi: 10.1145/3340531.3411884]

[18] Shao LS, Zhang J, Wei Y, Zhao JF, Xie B, Mei H. Personalized QoS prediction for Web services via collaborative filtering. In: Proc. of the 2007 IEEE Int'l Conf. on Web Services. Salt Lake City: IEEE, 2007. 439–446. [doi: 10.1109/ICWS.2007.140]

[19] Zheng ZB, Ma H, Lyu MR, King I. WSRec: A collaborative filtering based Web service recommender system. In: Proc. of the 2009 IEEE Int'l Conf. on Web Services. Los Angeles: IEEE, 2009. 437–444. [doi: 10.1109/ICWS.2009.30]

[20] Chen YP, Zhang YQ, Xia H, Gao C, Wang ZM, Wang FW, Li G. A hybrid tensor factorization approach for QoS prediction in time-aware mobile edge computing. Applied Intelligence, 2022, 52(7): 8056–8072. [doi: 10.1007/s10489-021-02851-z]

[21] Nan ZH, Zhao F. Research on semi-supervised recommendation algorithm based on hybrid model. In: Proc. of the 2nd Int'l Conf. on Machine Learning, Big Data and Business Intelligence. Taiyuan: IEEE, 2020. 344–348. [doi: 10.1109/MLBDBI51377.2020.00073]

[22] Sun D, Nie T. A Web service recommendation algorithm based on BaisSVD. In: Proc. of the 5th IEEE Information Technology and Mechatronics Engineering Conf. Chongqing: IEEE, 2020. 29–32. [doi: 10.1109/ITOEC49072.2020.9141664]

[23] Anwar T, Uma V, Srivastava G. Rec-CFSVD++: Implementing recommendation system using collaborative filtering and singular value decomposition (SVD)++. Int'l Journal of Information Technology & Decision Making, 2021, 20(4): 1075–1093. [doi: 10.1142/S0219622021500310]

[24] Yang YT, Zheng ZB, Niu XD, Tang MD, Lu YT, Liao XK. A location-based factorization machine model for Web service QoS prediction. IEEE Trans. on Services Computing, 2021, 14(5): 1264–1277. [doi: 10.1109/TSC.2018.2876532]

[25] Zhang PY, Huang WJ, Chen YT, Zhou MC, Al-Turki Y. A novel deep-learning-based QoS prediction model for service recommendation utilizing multi-stage multi-scale feature fusion with individual evaluations. IEEE Trans. on Automation Science and Engineering, 2024, 21(2): 1740–1753. [doi: 10.1109/TASE.2023.3244184]

[26] OWASP. OWASP API Security Top Ten—2023. 2023. https://owasp.org/API-Security/editions/2023/en/0x00-header/.

[27] Gunes I, Kaleli C, Bilge A, Polat H. Shilling attacks against recommender systems: A comprehensive survey. Artificial Intelligence Review, 2014, 42(4): 767–799. [doi: 10.1007/s10462-012-9364-9]

[28] Chirita PA, Nejdl W, Zamfir C. Preventing shilling attacks in online recommender systems. In: Proc. of the 7th Annual ACM Int'l Workshop on Web Information and Data Management. Bremen: ACM, 2005. 67–74. [doi: 10.1145/1097047.1097061]

[29] Zhou QQ, Wu JX, Duan LL. Recommendation attack detection based on deep learning. Journal of Information Security and Applications, 2020, 52: 102493. [doi: 10.1016/j.jisa.2020.102493]

[30] Mehta B. Unsupervised shilling detection for collaborative filtering. In: Proc. of the 22nd National Conf. on Artificial Intelligence. Vancouver: AAAI Press, 2007. 1402–1407.

[31] Zhang F, Deng ZJ, He ZM, Lin XC, Sun LL. Detection of shilling attack in collaborative filtering recommender system by PCA and data complexity. In: Proc. of the 2018 Int'l Conf. on Machine Learning and Cybernetics. Chengdu: IEEE, 2018. 673–678. [doi: 10.1109/ICMLC.2018.8526965]

[32] Wu ZA, Cao J, Mao B, Wang YQ. Semi-SAD: Applying semi-supervised learning to shilling attack detection. In: Proc. of the 5th ACM Conf. on Recommender Systems. Chicago: ACM, 2011. 289–292. [doi: 10.1145/2043932.2043985]

[33] Chen J, Zhang XX, Zhang R, Wang C, Liu L. De-Pois: An attack-agnostic defense against data poisoning attacks. IEEE Trans. on Information Forensics and Security, 2021, 16: 3412–3425. [doi: 10.1109/TIFS.2021.3080522]

[34] Chen Z, Qi WC, Bao TY, Shen LM. Data poisoning attack detection approach for quality of service aware cloud API recommender system. Journal on Communications, 2023, 44(8): 155–167 (in Chinese with English abstract). [doi: 10.11959/j.issn.1000-436x.2023161]

[35] Toker A, Eisenberger M, Cremers D, Leal-Taixé L. SatSynth: Augmenting image-mask pairs through diffusion models for aerial semantic segmentation. In: Proc. of the 2024 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Seattle: IEEE, 2024. 27685–27695. [doi: 10.1109/CVPR52733.2024.02615]

[36] Cao HQ, Tan C, Gao ZY, Xu YL, Chen GY, Heng PA, Li SZ. A survey on generative diffusion models. IEEE Trans. on Knowledge and Data Engineering, 2024, 36(7): 2814–2830. [doi: 10.1109/TKDE.2024.3361474]

[37] Wang WJ, Xu YY, Feng FL, Lin XY, He XN, Chua TS. Diffusion recommender model. In: Proc. of the 46th Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. Taipei: ACM, 2023. 832–841. [doi: 10.1145/3539618.3591663]

[38] Ho J, Jain A, Abbeel P. Denoising diffusion probabilistic models. In: Proc. of the 34th Int'l Conf. on Neural Information Processing Systems. Vancouver: NeurIPS, 2020. 6840–6851.

附中文参考文献

[8] 张以文, 项涛, 郭星, 贾兆红, 何强. 基于 SOM 神经网络的服务质量预测. 软件学报, 2018, 29(11): 3388–3399. http://www.jos.org.cn/

1000-9825/5474.htm [doi: 10.13328/j.cnki.jos.005474]

[34] 陈真, 乞文超, 鲍泰宇, 申利民. 面向服务质量感知云 API 推荐系统的数据投毒攻击检测方法. 通信学报, 2023, 44(8): 155–167. [doi: 10.11959/j.issn.1000-436x.2023161]

作者简介

陈真, 博士, 副教授, 博士生导师, CCF 专业会员, 主要研究领域为服务计算, 推荐系统, 云 API 安全, 服务化软件开发.

范爽, 硕士生, 主要研究领域为推荐系统攻击与防御, 云 API 安全.

余建强, 硕士生, 主要研究领域为推荐系统攻击, 云 API 安全.

徐悦甡, 博士, 副教授, CCF 高级会员, 主要研究领域为边缘计算, 服务计算, 软件工程.

檀泽宇, 硕士生, 主要研究领域为推荐系统攻击与防御, 云 API 安全.

尤殿龙, 博士, 教授, 博士生导师, CCF 专业会员, 主要研究领域为数据挖掘, 特征选择, 推荐系统安全.

申利民, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为软件工程, 柔性软件, 信息安全.