基于代理生成对抗网络的服务质量感知云 API 推荐系统投毒攻击

陈真[1,2]，刘伟[1]，吕瑞民[1]，马佳洁[1]，冯佳音[3]，尤殿龙[1]
（1.燕山大学信息科学与工程学院，河北 秦皇岛 066004；
2.燕山大学河北省计算机虚拟技术与系统集成重点实验室，河北 秦皇岛 066004；
3.河北科技师范学院数学与信息科技学院，河北 秦皇岛 066004）

摘　要：针对现有投毒攻击方法生成的虚假用户攻击数据存在攻击效果差且易被检测的不足，提出一种基于代理生成对抗网络的投毒攻击方法。首先，在生成对抗网络中采用 K-means 算法将数据分类，并引入自注意力机制学习每个类中的全局特征，解决生成对抗网络在数据稀疏时难以有效捕捉真实用户复杂行为模式这一问题，提升虚假用户的隐蔽性。其次，引入代理模型评估生成对抗网络生成的虚假用户的攻击效果，将评估结果作为代理损失优化生成对抗网络，进而实现在兼顾虚假用户隐蔽性的同时增强攻击效果。云 API 服务质量数据集上的实验表明，所提方法在兼顾攻击的有效性和隐蔽性方面均优于现有方法。

关键词：推荐系统；云 API；投毒攻击；生成对抗网络；代理模型
中图分类号：TN92
文献标志码：A
DOI：10.11959/j.issn.1000–436x.2025056

Poisoning attack on quality of service aware cloud API recommender system via surrogate generative adversarial network

CHEN Zhen[1,2], LIU Wei[1], LYU Ruimin[1], MA Jiajie[1], FENG Jiayin[3], YOU Dianlong[1]
1. School of Information Science and Engineering, Yanshan University, Qinhuangdao 066004, China
2. Hebei Key Laboratory of Computer Virtual Technology and System Integration, Yanshan University, Qinhuangdao 066004, China
3. School of Mathematics and Information Technology, Hebei Normal University of Science and Technology, Qinhuangdao 066004, China

Abstract: To address the shortcomings of existing poisoning attack methods, where the generated fake user attack data suffers from poor attack effectiveness and high detectability, a poisoning attack method based on the surrogate generative adversarial network (S-GAN) was proposed. Firstly, K-means was used in the generative adversarial network to classify the data, and a self-attention mechanism was incorporated to learn the global features within each class, solving the problem of difficulty in capturing and mimicking key features of real users in sparse data, thereby enhancing the concealment of fake users. Secondly, a surrogate model was deployed to evaluate the attack effectiveness of the GAN-generated fake users, and the evaluation results were employed as a surrogate loss to optimize the GAN, thereby facilitating the attack effectiveness while considering the concealment of fake users. Experiments conducted on cloud API quality of service datasets demonstrate that the proposed method outperforms existing methods in balancing the effectiveness and concealment of attacks.

Keywords: recommender system, cloud API, poisoning attack, generative adversarial network, surrogate model

---

收稿日期：2024–12–31；**修回日期：**2025–03–12
基金项目：国家自然科学基金资助项目（No.62102348, No.62276226）；河北省自然科学基金资助项目（No.F2022203012）；河北省科技计划基金资助项目（No.236Z0103G, No.236Z7725G）；河北省创新能力提升计划基金资助项目（No.22567626H）
Foundation Items: The National Natural Science Foundation of China (No.62102348, No.62276226), The Natural Science Foundation of Hebei Province (No.F2022203012), The Science and Technology Program of Hebei Province (No.236Z0103G, No.236Z7725G), The Innovation Capability Improvement Plan Project of Hebei Province (No.22567626H)

0　引言

面向服务架构是一种网络环境下广泛采用的软件开发模式，它将应用程序分解为一系列独立、松耦合的服务。这些服务遵循标准接口和协议，以跨平台和跨语言方式进行交互，旨在提高软件系统的灵活性、可维护性和扩展性。云应用程序接口（API, application program interface）是一种预先定义的网络接口，通过网络为开发人员提供访问一组例程和数据的能力，以支持面向服务架构的软件开发以及不同软件和机器之间的通信。目前，云API凭借跨平台、易组合和轻量化等优势，成功实现了业务内部逻辑和开发者的解耦，成为当前面向服务架构软件开发与运行的主流使能技术[1]。

现如今，进入万物互联的云时代，云API是落地人工智能算法赋能和发挥数据要素乘数效应的最佳载体。例如，Open API通过云API使GPT赋能千行百业。医疗机构采用云API实现了检查检验结果数据标准统一和共享互认。随着企业和组织对开放自身竞品资源和快速接入云端服务的日渐重视，云API的种类和数量都在急剧增加[2]，由此极大地丰富了网络中可用的云API。在此背景下，如何从功能多样、持续增长的海量云API中快速有效地发现和选择满足用户业务需求的云API，成为一个迫切需要解决的现实问题。虽然已有平台对云API进行分类并附加标签，用户可以通过浏览特定类别或利用标签筛选和定位所需云API。然而数量众多的功能高度同质化云API阻碍了开发者个性化的云API选择[3]。因此，人们引入服务质量（QoS, quality of service）来评估和差异化云API之间的性能，以期解决云API功能同质化问题[4]。

QoS用于刻画云API非功能侧服务质量信息，如响应时间、吞吐量、可用性、丢包率等。优秀的QoS性能意味着用户在与云API交互时能够获得更优越的体验。因此，QoS感知的云API推荐系统在功能高度同质化的环境中能够全面地评估云API的质量，为用户提供满足个性化需求的高质量云API[5]。目前，研究人员采用协同过滤[6]、矩阵分解[7]和深度学习[8]等技术提出了众多QoS感知云API推荐方法，这些方法在构建基于云API的高质量软件系统开发中发挥了积极作用。然而，由于网络环境的开放性和云API的货币属性，QoS感知的推荐系统很容易受到来自攻击者的数据投毒攻击[9]。

数据投毒攻击是指攻击者将精心制作的虚假用户数据注入QoS感知云API推荐系统中，致使推荐结果遵照攻击者的意图进行改变。已有研究表明，推荐系统自身是脆弱的[10]，向推荐系统仅注入1%的虚假用户就足以对推荐结果造成严重破坏。现有投毒攻击方法大致可分为2种：基于启发式的投毒攻击方法和基于对抗性的投毒攻击方法。基于启发式的投毒攻击方法通过统计分析生成攻击数据。例如，均值攻击将最高评级分配给目标云API，并将平均评级分配选择云API和填充云API。由于这类方法生成的虚假用户与正常用户之间的特征存在显著差异，很容易被防御算法检测出来[11]。此外，这类方法仅对特定的云API推荐系统有效。例如，均值攻击对基于用户的协同过滤推荐系统更有效，而对基于云API的协同过滤推荐系统的攻击效果不佳[12]。因此，启发式投毒攻击方法的有效性值得怀疑，尤其是对于当前广泛使用的基于深度学习的推荐系统[13]。基于对抗性的投毒攻击方法采用对抗学习思想设计攻击模型实现对推荐系统的投毒攻击。文献[14]为解决传统投毒攻击生成的虚假用户隐蔽性不足问题，引入生成对抗网络（GAN, generative adversarial network）生成用于推荐系统投毒攻击的虚假用户。文献[15]借助WGAN的思想，将鉴别器的功能从识别样本的真实性转变为计算两组分布之间的距离，使生成器能够精确捕捉正常用户行为特征分布以取得更好的生成效果，从而提高生成的虚假用户的隐蔽性。进一步，文献[16]提出了一种基于潜在扩散模型的数据投毒攻击方法，利用扩散模型学习潜在空间中真实用户与云API交互数据的分布以提高虚假用户的隐蔽性。虽然利用对抗学习思想设计投毒攻击方法能够提高生成的虚假用户的隐蔽性，但生成的虚假用户对推荐系统的攻击效果不佳。

为了解决现有投毒攻击方法存在的隐蔽性不足以及攻击效果不佳的问题，推动构建更加稳健的推荐系统投毒攻击防御体系的研究，本文提出了一种基于代理生成对抗网络的投毒攻击模型S-GAN。具体而言，针对生成对抗网络在数据稀疏时难以学习并模仿真实用户复杂行为模式这一挑战，S-GAN对生成对抗网络进行优化，通过K-means聚类算法将数据分为多个类，并引入自注意力机制学习每个

类中的全局特征，使S-GAN可以更好地捕捉到真实用户的行为模式，从而生成更加逼真的虚假用户，提高虚假用户的隐蔽性。此外，S-GAN引入代理模型，通过对生成的虚假用户进行攻击效果评估，代理模型能够提供及时的反馈，从而使生成对抗网络可以进行针对性的优化。因此，S-GAN生成的虚假用户不仅在被防御检测算法识别的概率上低于传统启发式投毒攻击方法，而且在对QoS感知云API推荐系统的攻击效果上也表现出色。这一双重优势使S-GAN在实施投毒攻击时，既能有效影响推荐结果，又能保持较高的隐蔽性，从而提升攻击成功率。

综上，本文的贡献主要体现在以下3个方面。

1) 针对现有投毒攻击方法在模拟用户行为真实性方面的不足，提出基于代理生成对抗网络的投毒攻击模型S-GAN。S-GAN在生成对抗网络中引入K-means聚类分析真实用户群体的行为规律，然后利用自注意力学习用户与商品的交互特征，使生成的虚假用户既符合用户群体行为模式，又具备个性化特征，显著提升了攻击的隐蔽性。

2) 针对现有投毒攻击方法在攻击效果方面的不足，引入代理模型评估S-GAN生成的虚假用户的攻击效果，将检测结果转化为代理损失反馈给生成器，使其能持续提升虚假用户的攻击能力，从而提升虚假用户的攻击效果。

3) 在基于真实世界的服务质量数据集上，对3类6种代表性推荐算法进行了广泛的投毒攻击实验。实验结果表明，S-GAN在保证攻击有效性的同时，隐蔽性也优于现有方法。实验还探究了不同攻击规模和攻击强度对攻击效果的影响，阐明了攻击规模和攻击强度对防御难度的影响机制，为针对推荐系统投毒攻击的防护提供了新思路。

1 数据投毒攻击

数据投毒攻击是指攻击者通过注入有毒数据污染原始数据分布，造成数据分析出现偏差或错误，使推荐系统向攻击者期望的方向倾斜，从而达到攻击目的。总体而言，QoS感知云API推荐系统的投毒攻击过程主要包括虚假用户生成和虚假用户注入2个阶段。

1.1 虚假用户生成

虚假用户生成是投毒攻击的核心步骤。攻击者使用投毒攻击方法对不同云API生成QoS数据，并将生成的QoS数据组合起来构建虚假用户

$$\hat{r}_u = C\left(A^S, A^F, A^\Phi, A^T\right) \tag{1}$$

其中，$\hat{r}_u$为生成的虚假用户，C为不同的投毒攻击方法，A^S和A^F是为了更好地拟合真实数据特征和提高虚假用户的真实性，A^Φ是空白云API集合，即用户未调用过的云API集合，A^T是用来达到对推荐系统攻击目的的云API集合。常见投毒攻击方法有均值攻击、随机攻击、潮流攻击和GAN攻击等。

攻击强度η为虚假用户采样的目标云API的数量。攻击强度η越高意味着攻击者对更多的目标云API进行攻击，进而对推荐系统的干扰效果越显著。然而，攻击强度的增加也伴随着更高的被检测风险，因此攻击者通常需要在提高攻击强度与降低检测风险之间进行权衡。为了探究攻击强度对云API推荐系统攻击的影响，利用攻击强度η生成虚假用户过程定义为

$$C\left(A^S, A^F, A^T, \eta\right) = \hat{r}_u \tag{2}$$

1.2 虚假用户注入

攻击者将包含不同攻击目标的虚假用户注入真实云API推荐系统数据集中，使推荐系统无法区分真实用户与虚假用户，从而对云API推荐模型的训练产生干扰，生成遵循攻击者意愿的推荐结果。

攻击规模μ用来描述组合虚假用户的数量。攻击规模越大意味着攻击者伪造了更多的虚假用户来影响推荐系统。由于大多数云API仅提供付费服务，更多的用户访问意味着更多的攻击成本。利用攻击规模μ生成虚假用户过程为

$$\xi\left(\hat{r}_{u1}, \hat{r}_{u2}, \cdots, \hat{r}_{uk}, \mu\right) \Rightarrow \hat{R} \tag{3}$$

其中，$\hat{r}_{u1}, \hat{r}_{u2}, \cdots, \hat{r}_{uk}$为攻击者生成的$k$个虚假用户，$\hat{R}$为根据不同的攻击规模挑选虚假用户而组成的虚假用户集合。

将生成的虚假用户集合$\hat{R}$注入真实数据集中，完成对QoS感知云API推荐系统的攻击

$$R \oplus \hat{R} \Rightarrow \tilde{R} \tag{4}$$

其中，$\tilde{R}$为混合数据集，R为真实数据集，$\hat{R}$为虚假用户数据集合。

2　模型介绍

2.1　总体结构

S-GAN的总体结构如图1所示。S-GAN模型在生成对抗网络的基础上，结合了K-means聚类和自注意力机制，以及代理模型的优化策略，提升虚假用户的隐蔽性和攻击效果。S-GAN主要包含3个核心组件：生成器G、鉴别器D和代理模型S。生成器G旨在生成虚假用户，以实现2个目标：生成尽可能接近真实数据分布的虚假用户；确保生成的虚假用户能够有效操纵QoS感知云API推荐系统的推荐结果。鉴别器D则致力于区分生成的虚假用户和真实用户。代理模型S用于评估生成器生成虚假用户的有效性，并根据评估结果对生成器进行优化。利用上述3个组件，S-GAN生成虚假用户的训练过程可分为以下2个阶段。

1)　生成器和鉴别器对抗学习。首先利用生成器G生成一批虚假用户，并将其与真实用户一同输入至鉴别器D中进行评估。鉴别器D的任务是计算这批虚假用户被误认为是真实用户的概率，并将此概率反馈给生成器G。生成器G使用多层感知机对原始输入数据进行非线性变换，能够提取丰富的特征表示。通过K-means聚类和自注意力机制，生成器能够更精确地模拟真实用户的行为特征。同时，利用鉴别损失调整与优化生成器G，直至鉴别器D无法有效区分虚假用户与真实用户。通过这一过程，确保生成的虚假用户在特征上与真实用户高度相似，从而降低被投毒攻击检测算法识别的风险。

2) 代理模型优化虚假用户生成。首先将生成器生成的虚假用户与真实用户混合，形成一个混合数据集。在此基础上，代理模型S预测目标云API在面临混合虚假用户群体时的QoS值，然后通过评估虚假用户的攻击代理损失进一步优化生成器G，进而使生成的虚假用户既能有效影响推荐结果，又能保持较高的隐蔽性，提升攻击成功率。

通过上述2个过程以迭代的方式对生成器进行优化，能够在确保虚假用户隐蔽性的同时，提高对QoS感知云API推荐系统攻击的有效性。

2.2　生成器和鉴别器对抗学习

首先讨论虚假用户生成的训练过程，生成随机高斯噪声输入生成器G来生成虚假用户，并将其与真实用户一起放入鉴别器D中，鉴别器D区分用户是否真实，并将反馈信息发送给生成器G。这样，通过生成器G与鉴别器D的竞争，得到训练良好的生成器G来生成虚假用户。

2.2.1　生成器

为了生成尽可能接近真实数据分布的虚假用户，利用多层感知机（MLP, multi-layer perceptron）对原始输入数据进行非线性变换，初步编码并提取特征，为后续阶段提供丰富且具有表征力的中间表示。具体地，将随机高斯噪声Z输入MLP进行特征提取的过程为

$$X = \mathrm{MLP}(Z) = f_l(\cdots f_2(f_1(Z))\cdots) \tag{5}$$

其中，1,2,⋯,l表示隐藏层层数，第l层的激活函数可以表示为

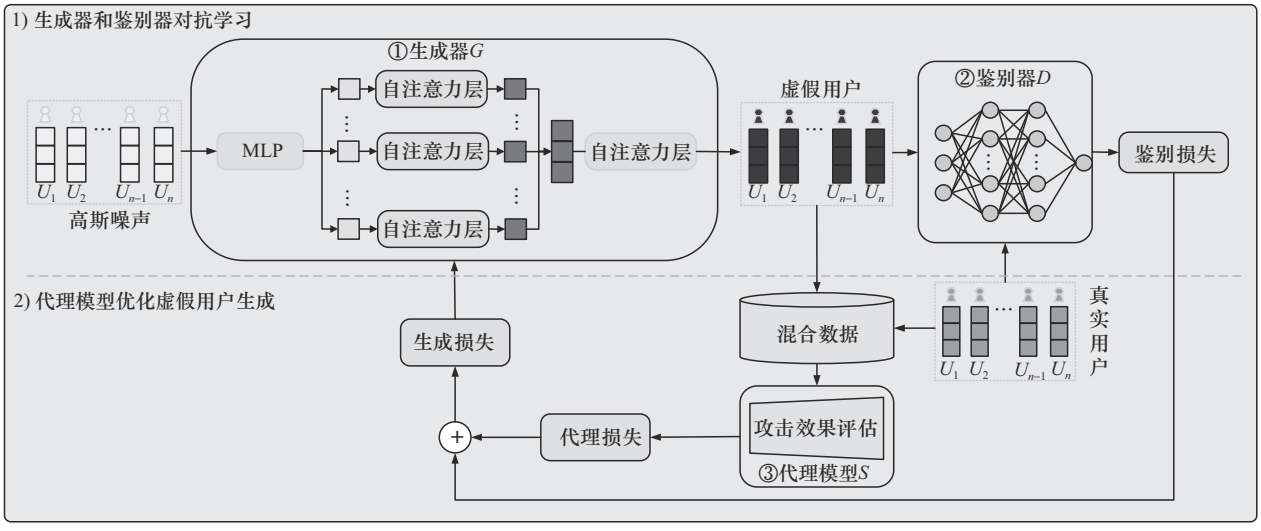


图1　S-GAN的总体结构

$$f_l(X_{l-1}) = \sigma(\boldsymbol{w}_l X_{l-1} + \boldsymbol{b}_l) \tag{6}$$

其中，$\boldsymbol{w}_l$和$\boldsymbol{b}_l$分别为学习到的权重矩阵和偏差向量，X_{l-1}为第l−1层的输出，σ是激活函数。

为了解决生成对抗网络在数据稀疏时难以有效捕捉真实用户复杂的行为模式这一问题，将云API按照K-means聚类分为m组，每组以相对集中的方式展现特定行为模式。这样，有利于生成器更精确地捕获这些行为模式，从而提高生成的虚假用户的隐蔽性。因此，在生成器内部，通过MLP得到的特征表示X，根据聚类结果划分为m组每组对应不同的真实用户行为模式

$$K(X) = \{X_1, X_2, \cdots, X_m\} \tag{7}$$

为了使模型能够更好地捕捉每组的全局特征，引入自注意力机制，对于每组X_i，计算查询$\boldsymbol{Q}_i$、键$\boldsymbol{K}_i$和值$\boldsymbol{V}_i$

$$\boldsymbol{Q}_i = X_i W_i^{\boldsymbol{Q}}, \boldsymbol{K}_i = X_i W_i^{\boldsymbol{K}}, \boldsymbol{V}_i = X_i W_i^{\boldsymbol{V}} \tag{8}$$

其中，$\boldsymbol{Q}_i$、$\boldsymbol{K}_i$、$\boldsymbol{V}_i$分别是学习到的权重矩阵。

然后应用点积注意力计算注意力分数，并使用softmax函数得到注意力权重W_i。最后根据注意力权重W_i对值$\boldsymbol{V}_i$进行加权求和，得到输出$\boldsymbol{O}_i$

$$\boldsymbol{O}_i = \text{softmax}\left(\frac{\boldsymbol{Q}_i \cdot \boldsymbol{K}_i^{\mathrm{T}}}{\sqrt{d_k}}\right) \cdot \boldsymbol{V}_i \tag{9}$$

完成分组训练后，遵循先前的聚类划分，合并各类输出$(\boldsymbol{O}_1, \boldsymbol{O}_2, \cdots, \boldsymbol{O}_m)$，形成一个综合性的特征表示$X_{\text{global}}$

$$X_{\text{global}} = \text{Context}(\boldsymbol{O}_1, \boldsymbol{O}_2, \cdots, \boldsymbol{O}_m) \tag{10}$$

最终，这个融合特征被送入一个全局自注意力层进行进一步提炼，该层负责在宏观层面整合局部特征，优化全局结构，最终生成高度逼真的虚假用户$\hat{r}_u$

$$\hat{r}_u = f_{\text{attention}}(X_{\text{global}}) = \text{softmax}\left(\frac{\boldsymbol{Q}_{\text{global}} \cdot \boldsymbol{K}_{\text{global}}^{\mathrm{T}}}{\sqrt{d_k}}\right) \cdot \boldsymbol{V}_{\text{global}} \tag{11}$$

其中，$\boldsymbol{Q}_{\text{global}}$、$\boldsymbol{K}_{\text{global}}$和$\boldsymbol{V}_{\text{global}}$分别是学习到的权重矩阵。

2.2.2 鉴别器

鉴别器D用于区分真实用户和由生成器G生成的虚假用户，并促使生成器G生成更真实的用户。鉴别器使用与生成器相同的MLP结构。

$$D(r_u) = \sigma(\boldsymbol{w}_i r_u + \boldsymbol{b}_i) \tag{12}$$

其中，$D()$表示输入是真实用户的概率，$\boldsymbol{w}_i$和$\boldsymbol{b}_i$分别表示学习到的权重矩阵和偏差向量。

生成器G和鉴别器D的目标函数如式(13)和式(14)所示。进一步，结合对抗学习的思想，将生成器和鉴别器的目标统一起来得到式(15)，来增强生成器生成无法检测到的虚假用户的能力。

$$\min \ \mathcal{L}_G = \min \sum_u \left(1 - D\left(\hat{r}_u\right)\right) \tag{13}$$

$$\max \ \mathcal{L}_D = \min \left(-\sum_u D\left(r_u\right) - \sum_u \left(1 - D\left(\hat{r}_u\right)\right)\right) \tag{14}$$

$$\min_G \max_D \ \mathcal{L}_{\text{GAN}} = \sum_u D(r_u) + \sum_u (1 - D(\hat{r}_u)) \tag{15}$$

其中，r_u为真实用户评级，$\hat{r}_u$为生成的虚假用户评级。

2.3 代理模型优化虚假用户生成

研究注意到，生成器G生成的虚假用户虽然能够模仿真实用户的行为特征，但不足以确保投毒攻击的有效性。为了克服这一挑战，引入代理模型S，用于评估生成的虚假用户在实施攻击时的攻击效果。具体而言，在式(13)的基础上，将生成器G的损失函数$\mathcal{L}_G$更新为式(16)。这样，生成器G不仅能够生成逼真的虚假用户，还能根据代理模型S提供的反馈来调整自身参数，提高攻击效果。

$$\min \ \mathcal{L}_G = \min\left(\sum_u \left(1 - D\left(\hat{r}_u\right)\right) + \mathcal{L}_{\text{attack}}\right) \tag{16}$$

其中，$\mathcal{L}_{\text{attack}}$为代理损失。

在设计损失函数时，考虑到攻击者的目标是通过优化目标云API的QoS预测值来实现攻击目的。为此，重新设计损失函数得到式(17)，使目标云API的预测值接近最大值，从而达到对推荐系统的攻击目的。通过式(17)中的损失函数，生成器G能够优化其生成的虚假用户，使得攻击后的QoS预测值可以显著高于攻击前。

$$\min \ \mathcal{L}_G = \min\left(\sum_u \left(1 - D\left(\hat{r}_u\right)\right) + \sum_{u \in U, i \in \text{target\_item}} (\max(\tilde{R}) - \tilde{P}_{u,i})\right) \tag{17}$$

其中，$\tilde{p}_{ui}$为攻击后代理模型S给出的用户u对目标云API的预测评级。

2.4　模型训练

S-GAN攻击模型的训练过程如算法1所示。

算法1　S-GAN攻击训练算法

输入　真实用户集合R、随机高斯噪声Z、总训练轮数T、鉴别器每轮训练步长K_D、生成器每轮训练步长K_G、代理模型训练步长K_S

输出　虚假用户配置文件中的API评级

1) 初始化生成器参数ϕ，鉴别器参数θ；

2) for 1 to T do

3)　for 1 to K_D do

4)　　采样真实用户，Sample $\{R_u\}_{u=1,\cdots,m}\sim R$

5)　　采样高斯噪声，生成虚假用户，

Sample $\{Z_u\}_{u=1,\cdots,m}\sim Z,\hat{R}=G(Z_u)$

6)　　更新鉴别器参数：

$$\theta\leftarrow\theta+\nabla_\theta\left(-\sum_u D\left(r_u\right)-\sum_u\left(1-D\left(\hat{r}_u\right)\right)\right)$$

7)　end for

8)　for 1 to K_G do

9)　　采样真实用户，Sample $\{R_u\}_{u=1,\cdots,m}\sim R$

10)　采样高斯噪声，生成虚假用户，

Sample $\{Z_u\}_{u=1,\cdots,m}\sim Z,\hat{R}=G(Z_u)$

11)　for 1 to K_S do

12)　混合数据，$\tilde{R}=(R,\hat{R})$

13)　　根据混合数据集预测目标项$\tilde{P}_{u,i}=S(\tilde{R})$

14)　end for

15)　更新生成器参数：

$$\phi\leftarrow\phi-\nabla_\phi\left(\sum_u\left(1-D\left(\hat{r}_u\right)\right)+\sum_{u\in U,i\in\text{target\_item}}\left(\max\left(\tilde{R}\right)-\tilde{P}_{u,i}\right)\right)$$

16)　end for

17) end for

在一个训练周期内，首先对鉴别器进行训练，目标是使其能够区分虚假用户与真实用户。鉴别器的训练过程如下。首先从真实用户集合中R随机抽取一组真实用户样本，同时生成一批基于高斯噪声Z的虚假用户，并将这2组样本一同输入鉴别器D。随后在固定生成器参数ϕ的条件下，利用式(14)优化鉴别器的参数θ。完成鉴别器的训练后，进入生成器的训练阶段。此时，再次从真实用户集合R中抽取一组真实用户样本，并生成一批基于高斯噪声Z的虚假用户，将这些虚假用户与真实用户混合，形成一个混合数据集。接下来，使用代理模型S预测该混合数据集中目标云API的QoS值。进一步，在固定鉴别器参数θ的前提下利用式(17)优化生成器的参数ϕ。通过反复迭代这个过程，就能够在保证生成的虚假用户隐蔽性的同时，提高攻击的有效性。

算法1的计算复杂度主要由生成器参数更新、鉴别器参数更新和代理模型预测评估3个部分组成。具体而言，生成器由MLP和自注意力网络组成，其参数更新一次的计算复杂度$C_G=O(I\times256+256\times512+512\times256+256\times I)+O(U^2\times\frac{I}{M})$；鉴别器MLP参数更新一次的计算复杂度$C_D=O(I\times512+512\times256+256\times128+128\times1)$；代理模型一次预测的计算复杂度$C_S=O(J\times(U+I))$，其中$J$为嵌入维度，$U$为用户数，$I$为云API数，$M$为聚类数。综上，结合总体训练轮数$T$、鉴别训练步长$K_D$、生成器训练步长$K_G$和代理训练步长$K_S$，算法1的总体计算复杂度为$O(T\times(K_D\times C_D+K_G\times(C_G+K_S\times C_S)))$。由于超参数$J$、$T$、$K_D$、$K_G$和$K_S$均是有限的固定值，因此算法1的复杂度可简化为$O(I+U^2\times\frac{I}{M}+U)$，主要依赖于用户数$U$和云API数$I$。

3　实验

3.1　实验准备

1) 实验数据集。本文采用真实世界云API的QoS数据集WS-DREAM[9]，评估投毒攻击的有效性及云API推荐系统的鲁棒性。WS-DREAM数据集包含了339名用户对全球范围内5 825个云API服务的响应时间服务质量数据。WS-DREAM QoS数据集的统计数据如表1所示。

表1　WS-DREAM QoS数据集的统计数据

参数	值
用户数/名	339
云API数量/个	5 825
QoS数据范围	(0, 20)
QoS平均值	0.908 5

2) 评价指标。为了衡量QoS感知云API推荐系统的性能，采用2种广泛应用于评估推荐系统准确

性的指标：平均绝对误差（MAE, mean absolute error）和均方根误差（RMSE, root mean square error）。MAE和RMSE定义式分别为

$$\mathrm{MAE}=\frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i-y_i| \tag{18}$$

$$\mathrm{RMSE}=\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\hat{y}_i-y_i\right)^2} \tag{19}$$

其中，N为测试集的大小，$\hat{y}_i$和y_i分别表示预测的QoS值和实际的QoS值。MAE和RMSE作为评估预测性能的关键指标，反映了预测结果的平均误差程度。显然，当MAE和RMSE的值越低时，说明预测结果与实际观测值之间的偏差越小，即预测精度越高。特别是RMSE，由于其对于较大误差的敏感性，使其成为识别预测误差显著性的有效指标。

此外，采用F1分数作为评估投毒攻击检测性能的指标。F1分数是一种将精确率和召回率相结合的综合性评价方法。精确率是真正例数除以真正例数和假正例数之和的比率，表示检测出的攻击样本中真实攻击样本的比例，高精确率意味着检测器误报少，对正常样本干扰低，如式(20)所示。召回率是真正例数除以真正例数和假负例数之和的比率，表示真实攻击样本中被正确检测的比例，高召回率意味着检测器漏报少，对攻击样本覆盖全面，如式(21)所示。

$$\mathrm{Precision}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \tag{20}$$

$$\mathrm{Recall}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{21}$$

F1分数作为精确率和召回率的调和平均数，其具体计算式如式(22)所示。F1分数综合考量了检测模型在识别攻击样本时的准确性和全面性。F1值越高，意味着模型能够更为有效地甄别正常样本与攻击样本：高精确率有助于减少误报情况，而高召回率则能降低漏报风险，这不仅代表检测方法的检测能力更为出色，同时也表明投毒攻击隐蔽性越低。

$$\mathrm{F1}=\frac{2\times\mathrm{Precision}\times\mathrm{Recall}}{\mathrm{Precision}+\mathrm{Recall}} \tag{22}$$

3) 实验环境。实验硬件环境为INTEL i7 13700和NVIDIA GeForce RTX 3060 Ti。软件环境为Pytorch 1.6.0，生成器中MLP的结构为256×512×256，鉴别器的结构为512×256×128×1。

4) 基线攻击模型。将均值攻击、随机攻击、潮流攻击、GAN攻击、扩散攻击作为基线攻击模型与S-GAN进行比较。

5) 推荐系统。选择3类6种代表性服务质量感知云API推荐系统（如表2所示），验证S-GAN攻击有效性。

6) 代理模型。使用加权正则化矩阵分解模型（WRMF, weighted regularized matrix factorization）作为代理模型。

3.2 投毒攻击效果分析

为了验证本文提出的S-GAN在QoS感知云API推荐系统中能否产生有效的攻击效果，向数据集中注入1%的虚假用户进行实验。表3和表4分别展示了在均值攻击、随机攻击、潮流攻击、GAN攻击、S-GAN W/O S攻击以及S-GAN攻击6种投毒攻击下，3类6种QoS感知云API推荐系统在攻击前和攻击后MAE和RMSE的变化情况。S-GAN W/O S攻击是S-GAN攻击去掉代理模型后的攻击方法。

表3和表4的实验结果显示，尽管只在数据集中注入了1%的虚假用户数据，这些投毒攻击依然能够显著影响推荐系统的准确性。具体表现为，各推荐系统的MAE和RMSE指标均出现了不同程度的上升。尤其值得关注的是，S-GAN在6种不同的QoS感知云API推荐系统中均展现出了优异的投毒

表2 基线推荐系统

类别	方法	核心算法
协同过滤	UserCF	选取用户侧相似邻居进行预测，其中邻域大小为20，相似度模型为PCC
	ItemCF	选取云API侧相似邻居进行预测，其中邻域大小为20，相似度模型为PCC
矩阵分解	Bias-SVD	通过矩阵分解为用户相关矩阵和云API相关矩阵，加上偏移量进行预测
	SVD++	通过添加用户反馈，将矩阵分解为用户相关矩阵和云API相关矩阵进行预测
深度学习	MLP	应用用户和云API之间交互的非线性变换
	DeepFM	DeepFM是FM和MLP方法的结合，FM应用二阶特征交互来模拟用户和云API交互

表3　6种推荐系统在不同投毒攻击下的MAE比较

推荐系统	无攻击	均值攻击	随机攻击	潮流攻击	GAN攻击	扩散攻击	S-GAN W/O S攻击	S-GAN攻击
UserCF	0.673 4	0.678 7	0.674 5	0.675 0	0.679 8	0.679 5	0.677 5	**0.681 2**
ItemCF	0.851 0	**0.853 8**	0.851 9	0.851 4	0.844 2	0.843 5	0.841 3	0.848 3
Bias-SVD	0.601 8	0.612 6	0.607 6	0.607 6	0.615 4	0.615 7	0.616 5	**0.620 3**
SVD++	0.592 4	0.596 9	0.598 1	0.597 2	0.599 2	0.602 3	0.601 8	**0.603 4**
MLP	0.510 0	0.510 3	0.514 6	0.518 0	0.523 8	0.525 3	0.521 6	**0.527 4**
DeepFM	0.507 9	0.512 3	0.524 5	0.526 8	0.530 7	0.528 6	0.528 8	**0.532 8**

表4　6种推荐系统在不同投毒攻击下的RMSE比较

推荐系统	无攻击	均值攻击	随机攻击	潮流攻击	GAN攻击	扩散攻击	S-GAN W/O S攻击	S-GAN攻击
UserCF	1.726 7	1.728 5	1.727 6	1.728 0	1.729 1	1.728 6	1.727 5	**1.730 5**
ItemCF	1.969 6	**1.978 8**	1.966 5	1.966 4	1.970 2	1.971 5	1.968 8	1.974 5
Bias-SVD	1.394 9	1.406 8	1.399 1	1.400 8	1.408 5	1.409 6	1.406 5	**1.411 2**
SVD++	1.394 5	1.396 5	1.399 8	1.402 3	1.418 9	1.418 6	1.419 2	**1.420 3**
MLP	1.363 5	1.367 6	1.375 4	1.378 9	1.385 7	1.386 8	1.382 4	**1.386 8**
DeepFM	1.352 4	1.359 8	1.361 7	1.360 8	1.369 8	1.368 5	1.371 1	**1.372 4**

攻击能力。S-GAN之所以能够取得如此显著的效果，关键在于其引入了代理模型。S-GAN通过代理模型获得代理损失，并利用代理损失不断优化生成器，从而生成更具攻击性的虚假用户。这些虚假用户能够有效污染数据，从而显著降低QoS感知云API推荐系统的准确性。

通过表3和表4还可以发现，基于深度学习的云API推荐系统鲁棒性较差，面对生成式对抗投毒时MAE变化比其他模型大，例如，MLP的MAE从0.510 0上升到0.527 4，增幅约3.4%；DeepFM的MAE从0.507 9到0.532 8，增幅约4.9%。这些数据清晰地反映出，在当前云API推荐领域占据主流地位的深度学习算法，正面临着严峻的数据投毒攻击威胁。鉴于此，深度学习在云API推荐中的应用必须高度重视数据安全防护工作。

3.3　投毒攻击规模影响分析

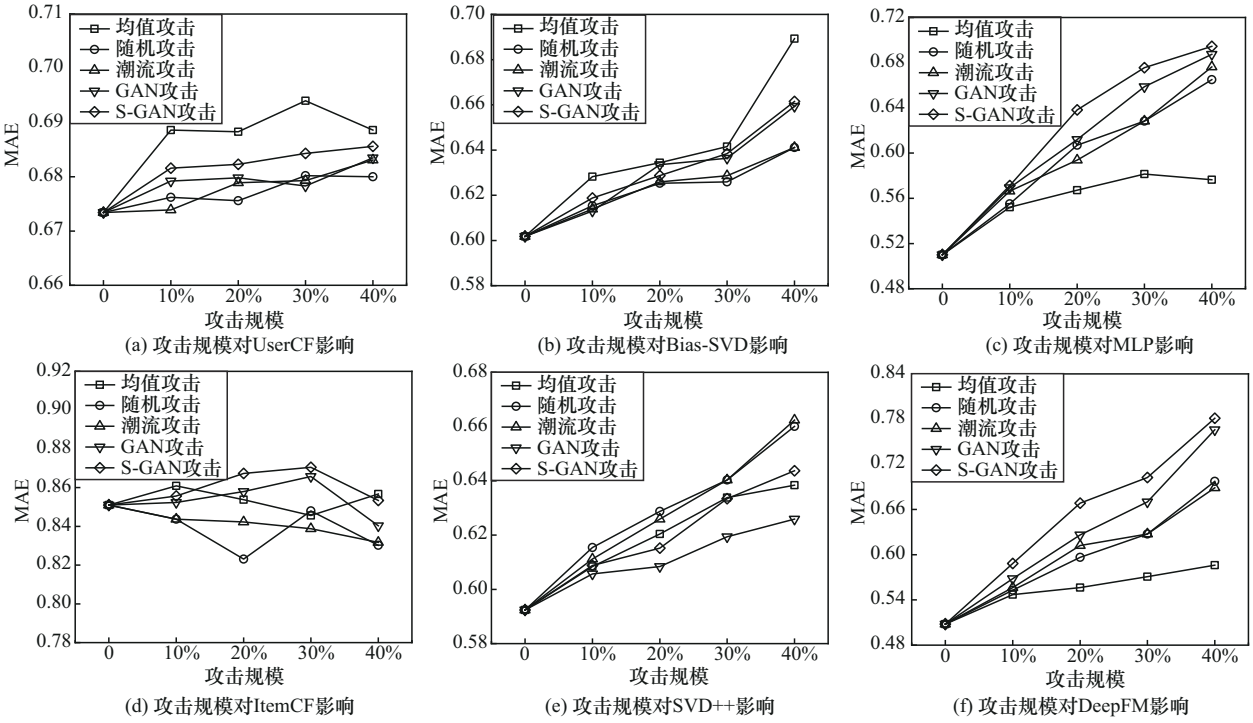
为了探究数据投毒攻击规模对服务质量感知云API推荐系统造成的影响，在固定攻击强度为20的条件下，逐步将投毒攻击规模从0增加到40%，每次以10%的步长递增，分析在不同攻击规模下，投毒攻击对QoS感知云API推荐系统性能的影响，如图2和图3所示。

从图2和图3可以看出，随着攻击规模的逐渐增大，对于大多数推荐算法，其MAE和RMSE指标都呈现出明显的上升趋势。这说明投毒攻击对云API推荐系统的影响存在明显的规模效应，攻击规模越大，对推荐系统性能的损害也越严重。此外，还有以下观察结果。

1)S-GAN对深度学习推荐方法的攻击表现明显优于其他攻击方法，例如，在MLP推荐系统中，当攻击规模为10%、攻击强度为20时，S-GAN攻击后的MAE值为0.571 3，均值攻击后的MAE值为0.552 1，S-GAN攻击比均值攻击提升了3.3%。

2)平均攻击对协同过滤和基于矩阵分解的推荐方法效果更明显，因为平均攻击产生的虚假用户包含了大部分真实用户对云API的平均反馈，而协同过滤和基于矩阵分解的方法是基于邻居相似性，因此虚假用户能够更好地融入真实用户中，但是平均攻击需要更高的攻击成本。

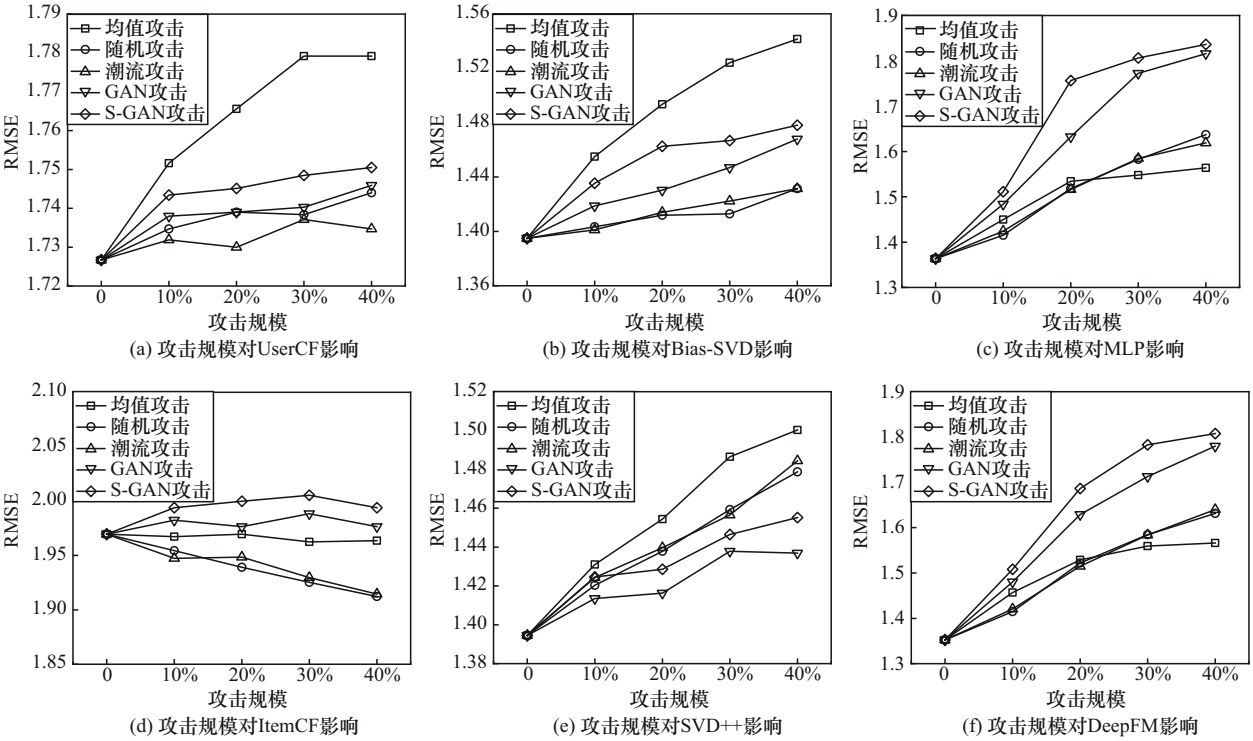
3)在对ItemCF进行投毒攻击时，部分投毒攻击方法随攻击规模的增加并没有提升MAE和RMSE。这是因为部分生成的虚假用户行为模式与真实用户差异显著，导致虚假用户被ItemCF的相似性计算机制过滤。



(a) 攻击规模对UserCF影响　(b) 攻击规模对Bias-SVD影响　(c) 攻击规模对MLP影响

(d) 攻击规模对ItemCF影响　(e) 攻击规模对SVD++影响　(f) 攻击规模对DeepFM影响

图2　攻击规模对不同推荐系统MAE的影响



(a) 攻击规模对UserCF影响　(b) 攻击规模对Bias-SVD影响　(c) 攻击规模对MLP影响

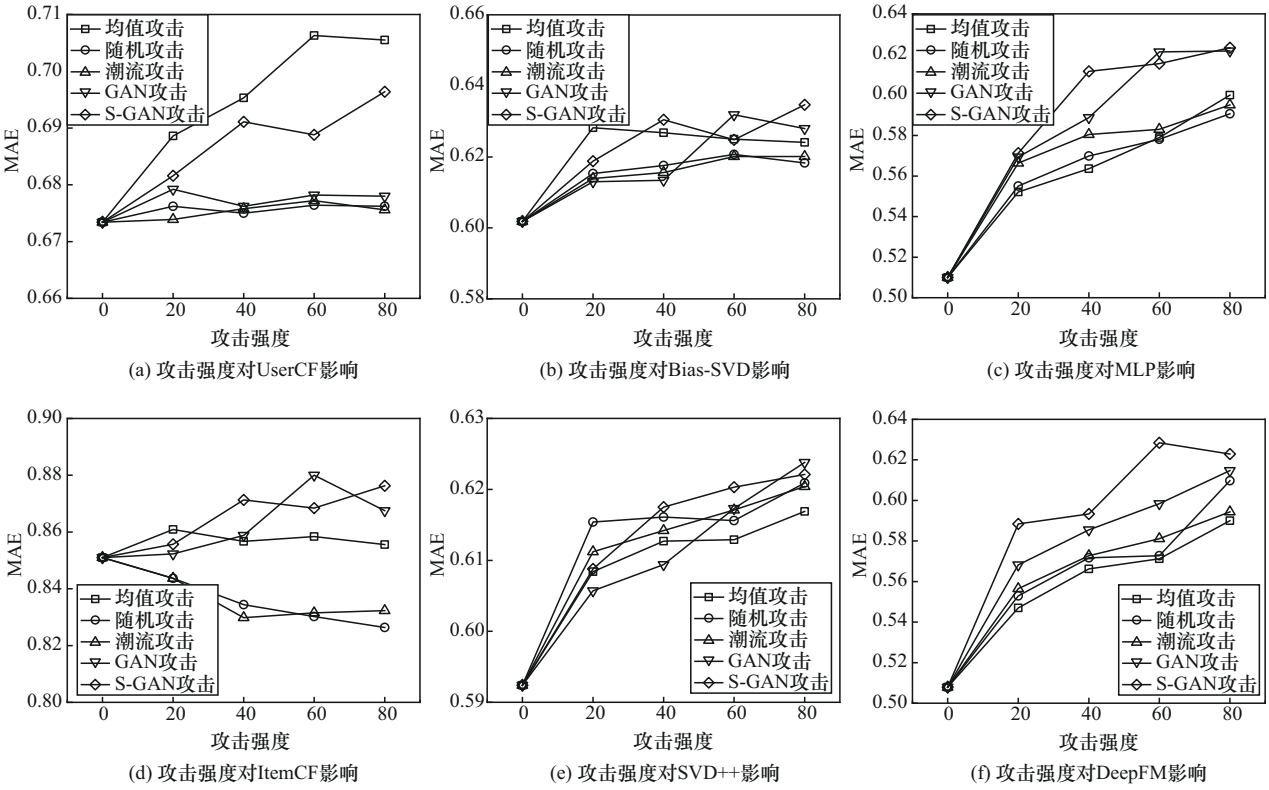
(d) 攻击规模对ItemCF影响　(e) 攻击规模对SVD++影响　(f) 攻击规模对DeepFM影响

图3　攻击规模对不同推荐系统RMSE的影响

3.4　投毒攻击强度影响分析

为了探究攻击强度对QoS感知云API推荐系统造成的影响，在固定攻击规模为10%的前提下，将攻击强度从0逐步增加至80，每次递增20。图4和图5显示了随着攻击强度的增加，云API推荐系统MAE和RMSE的变化趋势。

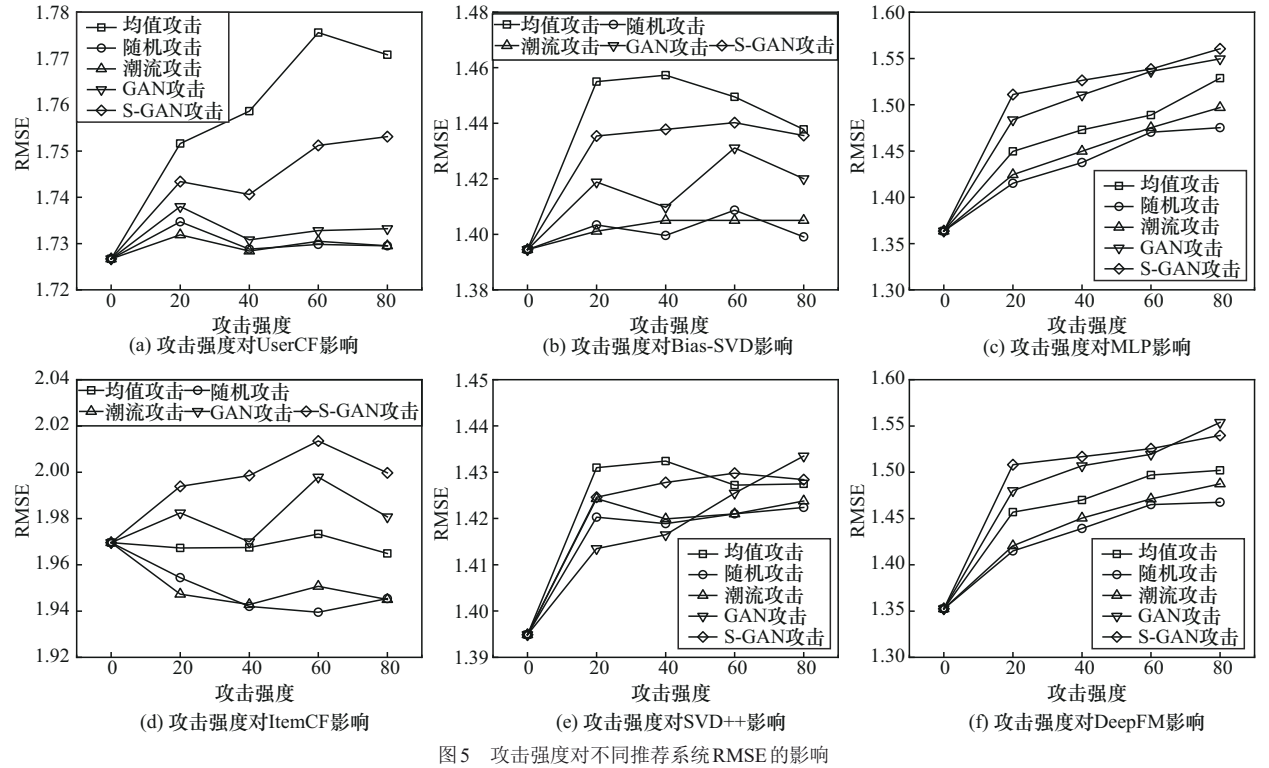
从图4和图5可以看出，随着攻击强度的增加，大多数推荐系统的MAE和RMSE指标呈现先大幅



(a) 攻击强度对UserCF影响　(b) 攻击强度对Bias-SVD影响　(c) 攻击强度对MLP影响

(d) 攻击强度对ItemCF影响　(e) 攻击强度对SVD++影响　(f) 攻击强度对DeepFM影响

图4　攻击强度对不同推荐系统MAE的影响



(a) 攻击强度对UserCF影响　(b) 攻击强度对Bias-SVD影响　(c) 攻击强度对MLP影响

(d) 攻击强度对ItemCF影响　(e) 攻击强度对SVD++影响　(f) 攻击强度对DeepFM影响

图5　攻击强度对不同推荐系统RMSE的影响

上升，再逐渐减缓的趋势。这一现象表明，当单个虚假用户中所包含的目标云API数量增多时，攻击的实际效果可能达不到预期效果，同时虚假用户被系统识别的风险也大幅提高。

注意到，S-GAN即使在攻击强度加大的情况下依然可以保持良好的攻击性能。这是由于S-

GAN 生成的虚假用户不是依据人工规则生成的，而是通过学习真实用户的行为模式生成的。

在现实世界中，对推荐系统实施攻击的成本较高。为了降低攻击成本，攻击者可能会为虚假用户设定多个目标 API。然而，增加攻击强度并不总能提升投毒攻击效果，有时甚至会降低攻击有效性。实验表明，对于响应时间数据集，虚假用户的最优目标 API 数量介于 20 至 40 之间。

3.5 投毒攻击隐蔽性分析

为了探究 S-GAN 的隐蔽性，在数据集中注入 10% 虚假用户进行实验，并采用 4 种具有代表性的检测算法，旨在精准区分真实用户与虚假用户。这些算法涵盖了无监督、半监督及监督学习方法，包括 PCA[17]、FAP[18]、SemiSAD[19]和 DegreeSAD[20]。在实验设计中，无监督方法采用 5 次独立实验的中位数作为性能指标，以确保结果的稳定性和可靠性。对于半监督与监督方法，则通过三重交叉验证来评估其表现，从而提供更加全面和精确的性能评估。表 5 给出了不同投毒攻击检测方法针对不同攻击的 F1 分数。其中，S-GAN W/O K&A 攻击表示 S-GAN 攻击去掉在生成器部分添加的 K-means 聚类和自注意力机制后的攻击方法。

从表 5 的实验结果来看，S-GAN 展现出较高隐蔽性，在 4 种不同检测算法中均难以被识别。例如，在 PCA 检测中，S-GAN 攻击的 F1 分数 0.793 3 比均值攻击的 F1 分数 0.961 0 降低了 21.2%，这是因为与传统投毒攻击方式相比，S-GAN 生成的虚假用户能够更加准确地模仿真实用户行为特征。传统投毒攻击在生成虚假用户时，往往会产生极端的投毒数据，这些数据更容易被检测算法识别和排除。S-GAN 则有效规避了这一缺陷，其生成的虚假用户具有高度的真实性，从而显著降低了被现有检测技术识别风险。

在检测性能方面，FAP 检测算法表现最优。FAP 对 S-GAN 检测的 F1 值为 0.936 2，较 PCA 检测算法提升 18.1%。其原因在于 FAP 通过标签数据驱动特征学习，能够有效捕捉生成数据与真实分布潜在统计偏差，从而在对抗性攻击检测中表现最优性能。因此，融合监督学习与深度生成对抗技术，可能是未来提升检测鲁棒性的有效路径。

为了进一步探究表 5 中 S-GAN 具有较好攻击隐蔽性的原因，利用 T-SNE 将虚假用户和真实用户投影到潜空间进行可视化处理。S-GAN 等不同攻击方法生成的虚假用户与真实用户分布的可视结果如图 6 所示。

从图 6 的结果可以看出，利用 S-GAN 攻击生成的虚假用户与真实用户的分布相似，这说明 S-GAN 能够有效学习到真实用户的行为模式，从而使其相对于其他攻击方式更难被检测到。这一突破揭示了生成对抗网络在数据投毒攻击中的潜在威胁，同时也为防御技术提出了新挑战。

4 结束语

数据投毒攻击通过混合虚假用户与真实用户反馈数据，操纵推荐系统按照攻击者期望的方向进行推荐，从而产生有偏推荐，破坏推荐系统的可信性。为了解决现有投毒攻击方法在攻击效果和隐蔽性方面的不足，本文提出了一种基于代理生成对抗网络的投毒攻击模型 S-GAN。首先在生成对抗网络中利用 K-means 聚类和自注意力机制帮助生成对抗网络有效捕捉真实用户复杂的行为模式，从而提升生成的虚假用户的隐蔽性。此外，引入代理模型对生成器进行优化，在保证攻击隐蔽性的同时提升了攻击有效性。真实世界服务质量数据集上的实验结果表明，S-GAN 不仅提升了攻击隐蔽性，同时也增强了攻击的有效性。但 S-GAN 需要同时训练生成器和判别器，在大型数据集上可能面临计算效率问题，限制了其在实际应用中的实用性。结合联邦学习场景，研究分布式环境下的投毒攻击方法是值得进一步探索的研究方向。

表 5　不同投毒攻击检测方法针对不同攻击的 F1 分数

检测方法	均值攻击	随机攻击	潮流攻击	GAN 攻击	扩散攻击	S-GAN W/O K&A 攻击	S-GAN 攻击
PCA	0.961 0	0.953 2	0.932 4	0.810 6	0.798 6	0.813 3	**0.793 3**
FAP	0.972 1	0.970 4	0.970 1	0.954 8	0.940 5	0.956 1	**0.936 2**
SemiSAD	0.856 7	0.882 4	0.868 9	0.743 1	0.735 9	0.746 8	**0.725 9**
DegreeSAD	0.915 4	0.882 1	0.883 4	0.813 7	0.803 5	0.815 4	**0.793 4**

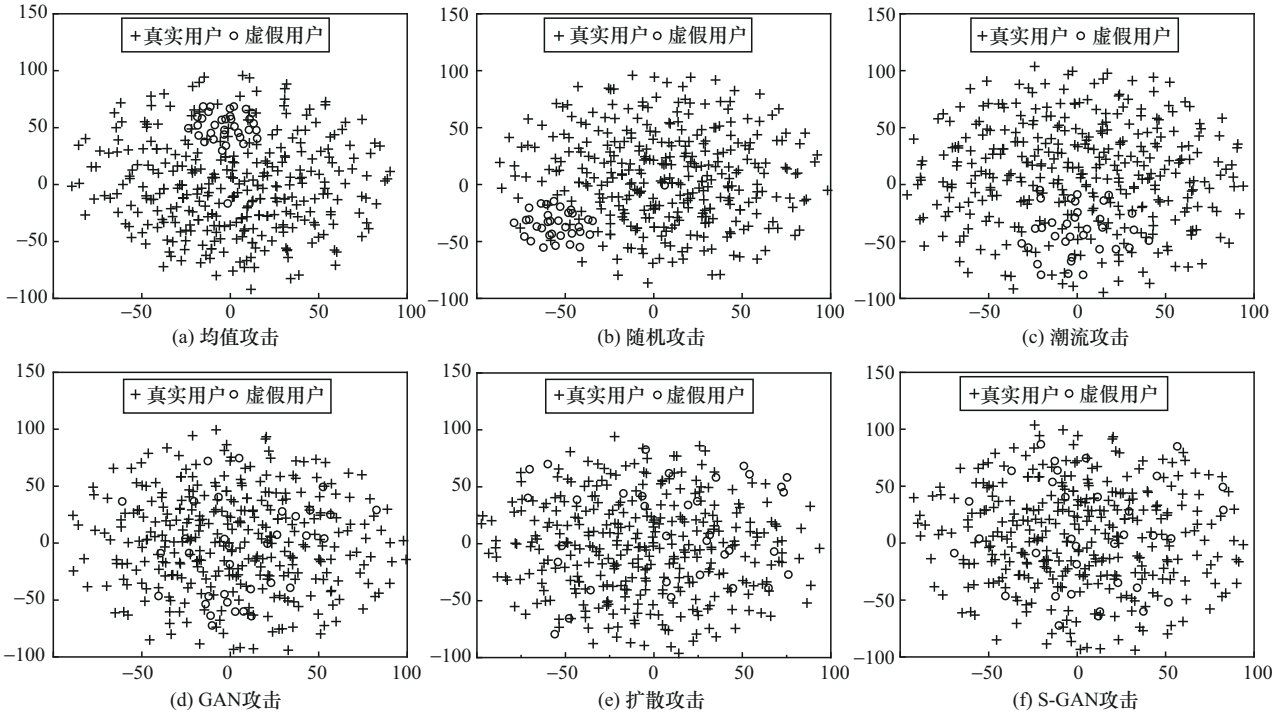


图6　不同攻击方法生成的虚假用户和真实用户在潜空间中的分布

参考文献：

[1] ZHANG M J, CAO J N, SAHNI Y, et al. EaaS: a service-oriented edge computing framework towards distributed intelligence[C]//Proceedings of the 2022 IEEE International Conference on Service-Oriented System Engineering (SOSE). Piscataway: IEEE Press, 2022: 165-175.

[2] WANG L, ZHANG Y Q, ZHU X H. Concept drift-aware temporal cloud service APIs recommendation for building composite cloud systems[J]. Journal of Systems and Software, 2021, 174: 110902.

[3] QI L Y, LIN W M, ZHANG X Y, et al. A correlation graph based approach for personalized and compatible web APIs recommendation in mobile APP development[J]. IEEE Transactions on Knowledge and Data Engineering, 2023, 35(6): 5444-5457.

[4] ANITHADEVI N, SUNDARAMBAL M. A design of intelligent QoS aware web service recommendation system[J]. Cluster Computing, 2019, 22(6): 14231-14240.

[5] EISA M, YOUNAS M, BASU K, et al. Modelling and simulation of QoS-aware service selection in cloud computing[J]. Simulation Modelling Practice and Theory, 2020, 103: 102108.

[6] CHEN Z, CHEN W H, LIU X W, et al. CCeACF: content and complementarity enhanced attentional collaborative filtering for cloud API recommendation[J]. The Journal of Supercomputing, 2024, 80(18): 26111-26139.

[7] LIU Z. Matrix factorization-based web service QoS prediction: methods and applications[J]. Academic Journal of Engineering and Technology Science, 2024, 7(3): 146-151.

[8] BOULAKBECH M, MESSAI N, SAM Y, et al. Deep learning model for personalized web service recommendations using attention mechanism[C]//International Conference on Service-Oriented Computing. Berlin: Springer, 2023: 19-33.

[9] ZHENG Z B, LI X L, TANG M D, et al. Web service QoS prediction via collaborative filtering: a survey[J]. IEEE Transactions on Services Computing, 2022, 15(4): 2455-2472.

[10] FANG M H, YANG G L, GONG N Z, et al. Poisoning attacks to graph-based recommender systems[C]//Proceedings of the 34th Annual Computer Security Applications Conference. New York: ACM Press, 2018: 381-392.

[11] ZHANG X X, CHEN J, ZHANG R, et al. Attacking recommender systems with plausible profile[J]. IEEE Transactions on Information Forensics and Security, 2021, 16: 4788-4800.

[12] NGUYEN T T, QUOC VIET HUNG N, NGUYEN T T, et al. Manipulating recommender systems: a survey of poisoning attacks and countermeasures[J]. ACM Computing Surveys, 2025, 57(1): 1-39.

[13] ZHANG S, YAO L N, SUN A X, et al. Deep learning based recommender system[J]. ACM Computing Surveys, 2020, 52(1): 1-38.

[14] CHRISTAKOPOULOU K, BANERJEE A. Adversarial attacks on an oblivious recommender[C]//Proceedings of the 13th ACM Conference on Recommender Systems. New York: ACM Press, 2019: 322-330.

[15] CHEN Z, BAO T Y, QI W C, et al. Poisoning QoS-aware cloud API recommender system with generative adversarial network attack[J]. Expert Systems with Applications, 2024, 238: 121630.

[16] CHEN Z, YU J Q, FAN S, et al. Latent diffusion model-based data poisoning attack against QoS-aware cloud API recommender system[J]. Computer Networks, 2025, 260: 111120.

[17] ZHANG F, CHAN P P K, HE Z M, et al. Unsupervised contaminated user profile identification against shilling attack in recommender system[J]. Intelligent Data Analysis, 28(6): 1411-1426.

[18] CHEN X, DENG X, HUANG C S, et al. Detection of trust shilling attacks in recommender systems[J]. IEICE Transactions on Information and Systems, 2022, E105.D(6): 1239-1242.

[19] ZHOU Q Q, DUAN L L. Semi-supervised recommendation attack detection based on Co-Forest[J]. Computers & Security, 2021, 109: 102390.

[20] SHENDE M K, VERMA V. Enhancing popSAD: a new approach to shilling attack detection in collaborative recommenders[C]//International Conference on Frontiers in Computing and Systems. Berlin: Springer, 2024: 51-62.

[作者简介]

陈真（1987– ），男，陕西宝鸡人，博士，燕山大学副教授、博士生导师，主要研究方向为服务计算、推荐系统、云API安全和服务化软件开发等。

刘伟（1997– ），男，河北衡水人，燕山大学硕士生，主要研究方向为云API安全。

吕瑞民（2001– ），男，山东德州人，燕山大学硕士生，主要研究方向为云API推荐系统攻击与防御。

马佳洁（2002– ），女，河北石家庄人，燕山大学硕士生，主要研究方向为云API安全。

冯佳音（1983– ），女，河北秦皇岛人，博士，河北科技师范学院副教授，主要研究方向为推荐系统与信息安全。

尤殿龙（1981– ），男，内蒙古赤峰人，博士，燕山大学教授，主要研究方向为数据挖掘、特征选择和推荐系统等。