本科生毕业论文（设计）

题目：面向联邦协同过滤的托攻击检测算法

姓　　名　姜洋帆

学　　号　17341068

院　　系　计算机学院

专　　业　计算机科学与技术

指导教师　吴迪 (教授)

2021 年 5 月 16 日

面向联邦协同过滤的

托攻击检测算法

On the Detection of Shilling Attacks in

Federated Collaborative Filtering

姓　　名	姜 洋 帆
学　　号	17341068
院　　系	计算机学院
专　　业	计算机科学与技术
指导教师	吴迪 (教授)
答辩委员	

2021 年 5 月 16 日

学术诚信声明

本人郑重声明：所呈交的毕业论文（设计），是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文（设计）不包含任何其他个人或集体已经发表或撰写过的作品成果。对本论文（设计）的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本论文（设计）的知识产权归属于培养单位。本人完全意识到本声明的法律结果由本人承担。

作者签名：

日　　期：　　年　　月　　日

【摘　要】

联邦学习 (Federated learning, FL) 是一种新兴的分布式机器学习范式，它可以让分布式用户仅交换梯度信息来协同训练模型，以达到保护用户数据隐私的目标。联邦协同过滤 (Federal Collaborative Filtering, Fed-CF) 是联邦学习在推荐问题中的一种典型应用。然而，Fed-CF 的去中心化特性使得它很容易遭受恶意用户发动的托攻击。在托攻击中，攻击者通过注入大量带有虚假评分数据的恶意用户，降低或提高目标项目在推荐系统中的评级，从而扭曲真实的推荐效果。传统的托攻击检测算法利用用户的评分数据，从中抽取特征进行聚类。但是在联邦学习场景下，用户的原始训练数据不会被公开，因此传统托攻击检测算法无法被应用于 Fed-CF。本文系统性地研究了联邦推荐系统场景下的托攻击问题，并设计了一种高效的联邦托攻击检测器 (Federated Shilling Attack Detector, FSAD) 来检测 Fed-CF 中的托攻击行为。我们首先展示了攻击者在 Fed-CF 中发动托攻击的可行性，然后针对托攻击检测问题设计了四个基于用户上传梯度的特征。利用这些基于梯度的特征，我们训练了一种半监督贝叶斯分类器来有效识别托攻击者生成的恶意用户。为了进一步保护隐私，用户可以给梯度增加噪声来达到差分隐私保证。我们分析了差分隐私噪声对 Fed-CF 算法和 FSAD 的性能影响。我们从理论上证明了只要差分隐私噪声不完全扭曲推荐效果，我们的检测算法仍然是有效的。最后，我们基于真实数据集进行了大量的实验来评估 FSAD 的性能。实验结果表明，FSAD 可以检测出未采用差分隐私机制的托攻击者，且准确率较高，在 Netflix 数据集上 $F1$ 值高达 0.90，接近利用原始用户评分信息的传统托攻击检测器的性能。而差分隐私机制下的 FSAD 性能取决于用户添加的噪声，检测效果与推荐精度呈正相关。

【关键词】 联邦协同过滤，托攻击，攻击检测，虚假评级，差分隐私

[ABSTRACT]

Federated learning (FL) is proposed to protect user privacy via training models across decentralized clients by exchanging gradients only. Federated Collaborative Filtering (Fed-CF) is a typical application of FL in recommender systems. However, the decentralized nature of Fed-CF makes it vulnerable to shilling attacks, which can be realized by inserting fake ratings of target items to distort recommendation results. Unfortunately, previous detection algorithms cannot work well in FL, as all original data samples are not disclosed at all. In this thesis, we are the first to systematically study the problem of shilling attacks in the context of FL, and propose an effective detection method called *Federated Shilling Attack Detector (FSAD)* to detect shilling attackers in Fed-CF. We first show the feasibility of shilling attacks in Fed-CF. Next, we dedicatedly design four novel features based on disclosed gradients. A semi-supervised Bayes classifier is trained to identify shilling attackers effectively. In addition, we analyze the performance of FSAD when the differential privacy (DP) is involved on clients to protect the gradient information. We prove that FSAD is still effective as long as the DP noise does not totally distort the recommendation accuracy. Finally, we conduct extensive experiments based on real-world datasets to evaluate the performance of FSAD. The results show that FSAD can detect shilling attackers without DP with high accuracy, with the F_1 value as high as 0.90 on the Netflix dataset, which approaches the performance of the optimal detector utilizing the complete information for detection. The detection performance with DP depends on the added noise, and the detection accuracy is positively correlated with the recommendation accuracy.

[Keywords] Federated collaborative filtering, shilling attack, attack detection, fake rating, differential privacy

目录

插图目录

表格目录

一、引言

1.1 研究背景与意义

近年来，联邦学习[1;2;3;4]作为一种可有效降低用户隐私泄露的新型去中心化机器学习框架，受到工业界和学术界的广泛关注。在联邦学习场景下，系统内的用户通过与参数服务器交换训练过程中的梯度信息来协同训练一个机器学习模型，而不必公开本地训练数据这一类敏感的隐私信息。联邦学习的出现为分布式机器学习系统中的用户隐私保护带来了新的可能性。

推荐系统[5]已经深入我们的日常生活[6]，并被广泛部署于电子商务、在线视频等各种应用中，但是却存在着较为严重的用户隐私泄露问题[7;8;9]。联邦学习的一种典型应用便是在推荐系统中保护用户隐私。联邦协同过滤[10](Federated Collaborative Filtering, Fed-CF) 是一种基于联邦学习的推荐算法，在 Fed-CF 中，终端用户被视为一个独立的联邦学习参与方（用户）。用户通过与参数服务器分享梯度信息的方式共同更新作为全局模型的项目隐向量 (item latent vectors)，而用户各自的评分数据与用户隐向量 (user latent vectors) 则作为隐私数据保存在本地。在经过多轮的训练迭代后，参数服务器聚合得到一个收敛的项目隐向量，而每个用户则在本地更新过程中得到收敛的用户隐向量。最终，参数服务器将最后一轮聚合得到的项目隐向量分发到所有用户，每个用户可以通过本地的用户隐向量与共同训练得到的项目隐向量进行模型推理，做出本地推荐决策。

虽然联邦学习可以在一定程度上保护用户隐私，但是托攻击者的攻击目的是扭曲推荐效果而非窃取用户隐私数据，因此托攻击在联邦学习场景下仍然有效[11;12;13]。托攻击者通过注入大量带有虚假评分数据的恶意用户对推荐系统发动攻击[14;15;16]。这种攻击的通常手段为，给目标项目赋予极高或极低的评分来提高或降低目标项目的推荐评级，从而达到特定的攻击目的。基于托攻击恶意用户的评分数据特点，目前已有多种检测算法[17;18;19;20;21]被开发出来以抵御托攻击，它们通过分析推荐系统中的用户评分数据来区分普通用户与托攻击恶意用户。然而，在 Fed-CF 系统中，用户只交换梯度信息而不会暴露评分数据，推荐系统平台（参数服务器）无法直接获得每个用户的评分数据，因此现有的检测算法在 Fed-CF 场景下不再适用。

此外，如果系统中的用户采用差分隐私机制对上传的梯度添加噪声来进一步保护数据隐私，托攻击检测将会变得更加困难。根据一些研究文献的报告，由于用

户上传的梯度一定程度上包含了本地训练数据的信息，联邦学习用户即使只分享梯度数据，也有可能造成本地隐私数据的泄露[22;23;24]。最新的研究结果表明，攻击者可以通过分析 Fed-CF 系统中用户上传的梯度信息来重构一些敏感数据（例如每个用户的评分数据）[25]。在联邦学习系统的用户端采用差分隐私机制来扰乱梯度数据，可以进一步模糊化梯度包含的隐私信息，从而有效提高隐私保护水平[26;27;28]。目前已有多种差分隐私机制被成功应用于联邦学习系统中[29;30;31]。在这些机制下，用户通常为梯度数据增加额外的噪声，然后再上传至参数服务器，以此模糊梯度信息，达到一定程度的差分隐私保证。由于噪声会使用户梯度信息失真，在用户采用差分隐私机制的场景下，托攻击检测任务将会变得更加困难与复杂。

1.2 论文主要研究内容

在本文中，为了抵御托攻击，我们提出了一种称为联邦托攻击检测器 (Federated Shilling Attacker Detector, FSAD) 的检测算法来有效识别 Fed-CF 中的托攻击恶意用户①。我们首先展示了托攻击可以通过注入大量带有虚假评分数据的恶意用户来显著地扭曲联邦推荐系统的推荐结果。为了逃避推荐系统平台的检测，托攻击恶意用户不仅会对目标项目进行极端评分以达到攻击目的，还会模仿正常的用户对其它项目进行评分，这使得托攻击恶意用户的评分数据与正常用户非常相似，增大检测难度。现有的文献中还没有出现对联邦学习场景下托攻击检测的研究。在联邦托攻击检测器 (FSAD) 的设计中，我们针对恶意用户的数据特点，基于梯度信息设计了新的特征。推荐系统平台（参数服务器）可以从用户上传的梯度中提取特征，训练半监督贝叶斯分类器来识别系统中的托攻击恶意用户。此外，我们还分析了当差分隐私机制作用于每个用户时，Fed-CF 的推荐性能和 FSAD 的有效性。我们从理论上证明了差分隐私噪声会以相同的幅度影响项目隐向量和特征的准确性。最后，我们基于真实数据集（Movielens 和 Netflix）进行大量的实验，证实了 FSAD 可以在用户没有采用差分隐私机制保护梯度的情况下有效检测出推荐系统中的托攻击恶意用户，并且检测精度接近利用完整用户隐私信息（完整的评分数据）进行托攻击检测所达到的水平。当用户采用了差分隐私机制后，检测效果主要取决于差分隐私噪声的大小。实验结果表明，只要梯度噪声不完全破坏推荐精度，我们的检测方法将仍然有效。

我们在本文的工作和贡献可以总结如下：

- 我们率先研究了联邦推荐系统中的托攻击检测问题。我们的研究表明了托攻

① 我们这项研究的基本方法和初步结果已经发表在一篇会议论文上[32]

击在联邦推荐系统场景下的可行性和有效性：托攻击者在联邦学习场景下，可以通过注入大量带有虚假评分数据的恶意用户来操纵推荐系统对目标项目的推荐结果。

- 我们提出了一种有效的检测方法，称为联邦托攻击检测器 (FSAD)，用以抵御联邦推荐系统中的托攻击行为。我们针对这个场景创新地设计了四个基于梯度的特征。通过利用这些新特征，我们训练了一种半监督贝叶斯分类器来有效地识别托攻击恶意用户。
- 我们从理论上推导出差分隐私噪声对项目隐藏向量和基于梯度的特征的估计精度的影响。理论分析结果表明，作用在梯度上的差分隐私噪声会以相同的幅度影响项目隐向量和特征的准确性。
- 我们基于 Movielens 和 Netflix 这两个真实世界数据集，通过大量的实验证实了 FSAD 的有效性。实验结果表明，我们设计的 FSAD 可以有效检测出托攻击恶意用户，从而提升推荐系统的性能。除此之外，实验还证实了 FSAD 的鲁棒性：即使用户采用了差分隐私机制对梯度添加噪声，FSAD 仍然能够达到较好的检测效果。

1.3 论文组织结构

本文的剩余章节内容安排如下：

第二章介绍问题背景和研究现状，章论述联邦推荐系统中托攻击的可行性与有效性。

第三章给出了基于梯度的四种特征的定义及有效性分析。对 FSAD 的设计细节进行具体阐述。

第四章理论分析差分隐私机制对 Fed-CF 和 FSAD 性能的影响。证明了差分隐私噪声会以相同的幅度影响项目隐向量和特征的准确性。

第五章展示与讨论实验结果。

最后，我们在第六章总结与展望了本文的工作。

二、相关工作及背景知识

2.1 推荐系统

推荐系统[5]已经在我们的日常生活中得到了广泛的应用。推荐系统提供的个性化推荐为终端用户和服务提供商都带来了巨大的利益[6;33;5]。经典的推荐模型（例如，协同过滤[34]）基于用户-项目评分矩阵，通过学习用户和项目（产品）的潜在向量来预测用户的个性化偏好。

这些经典推荐模型的一个缺点是，用户-项目评分信息必须被推荐服务平台收集与处理。为了训练推荐模型，服务提供商需要得到每个用户对项目的评分数据，这会为终端用户带来隐私泄露的风险。为了保护推荐系统中的用户隐私，多种机器学习隐私保护策略已经被提出并应用。终端用户可以采用差分隐私机制，将噪声添加于评分数据后再上传至服务器来保护隐私，然而这会损害推荐系统的性能[27;29]。分布式推荐系统可以避免集中化处理用户敏感数据，从而保护用户隐私，但是会带来额外的计算与通信开销[35;36]，降低推荐模型的实时性。

2.2 托攻击

托攻击[14]已在传统推荐模型下被广泛探讨与研究。托攻击者通过向推荐系统注入大量恶意用户来扭曲推荐结果。这些恶意用户会给目标项目赋予极高或极低的评分，让目标项目变得更加热门或冷门。相关研究表明，协同过滤算法很容易被拥有高偏置评分数据的托攻击恶意用户影响[14]，导致推荐结果被扭曲。同时由于推荐系统天然开放的特性，恶意用户很容易参与推荐模型的训练，因此推荐系统很容易遭受托攻击。目前已有大量检测算法[17;18;19]被设计并应用于传统推荐系统中。然而，这些算法都需要利用用户-项目评分矩阵来识别恶意用户。例如，一系列的无监督检测算法[20;21]根据从用户-项目评分矩阵中提取出的特征进行无监督聚类，将正常用户与托攻击恶意用户区分开来。

2.3 联邦学习

联邦学习[1;2;3;4]是一种新型分布式机器学习范式。联邦学习系统中的用户仅与参数服务器交换梯度信息，因此用户隐私可以在一定程度上得到保护。为了克服传统推荐算法泄露用户隐私的缺陷，Ammad-ud din 等人[10]提出了联邦协同过滤

(Fed-CF)，将基于奇异值分解 (SVD) 的推荐算法迁移到联邦学习框架中。Jalalirad 等人[37] 提出了一种联邦学习场景下基于神经网络的推荐算法。在[38] 中, Flanagan 等人提出一种称为联邦多视图矩阵分解的联邦推荐算法。这些算法的共同点是终端用户不需要将本地的原始数据上传至服务器，因此在很大程度上保护了用户隐私。

在联邦学习框架下，每个用户对项目的评分信息是私有的，这些信息无法被参数服务器获取。这意味着现有的托攻击检测算法在联邦学习场景下都不再适用。同时，由于分布式学习和推荐系统的开放性，托攻击者很容易将恶意用户注入系统[13;11]，从而造成对联邦推荐模型效果的破坏。因此，设计新的检测算法来抵御联邦推荐系统中的托攻击攻击是非常必要的。

2.4 差分隐私

差分隐私[26;27] 在统计和机器学习分析的背景下，为隐私保护提供了严格的数学定义。它的直观含义是，当数据集的一条记录产生变化时，该用户的输出与原数据集下执行相同算法的输出不会产生太大的差异。因此，若一个算法满足差分隐私，攻击者将难以通过用户的输出的变化来推断本地数据的变化，从而使本地隐私数据得到保护。实现差分隐私保证的一种常用方法是给输出添加一个服从特定分布（例如，高斯分布）的随机变量，我们将这个随机变量称作差分隐私噪声。最近的多项研究[29;31;30] 探索了如何将差分隐私机制部署到联邦学习系统中。它们的基本思想都是在梯度上添加差分隐私噪声来掩盖真实值，从而对梯度进行模糊化处理，防止攻击者从梯度信息反推出训练集信息，以此保护用户隐私。

2.5 联邦推荐系统中的托攻击

在推荐系统中，托攻击一般可分为三种形式：推攻击 (push attack)、核攻击 (nuke attack) 以及破坏攻击 (vandalism attack)。攻击者可能让目标项目变得更加热门（推攻击），或者变得更加冷门（核攻击），甚至只是想破坏推荐系统的推荐效果（破坏攻击[39]）。在大多数情况下，托攻击者的目的是为自己带来某种经济效益（例如，提高自己产品的评级，或者打压竞争对手的产品评级），因此破坏攻击在本文中将不被考虑。

为了进行有效的托攻击，攻击者需要在推荐系统中注入大量的恶意用户。恶意用户可以被抽象为人为生成的项目评分数据。从攻击者的角度来看，存在四种项目：填充项目 (filler items)（记为集合 $\mathcal{I}_F$），选择项目 (selected items)（记为集合

$\mathcal{I}_S$），未评分项目 (unrated items)（记为集合 $\mathcal{I}_\Phi$）以及目标项目 (target items)（记为集合 $\mathcal{I}_t$）。图2-1展示了托攻击恶意用户评分数据的一般形式。

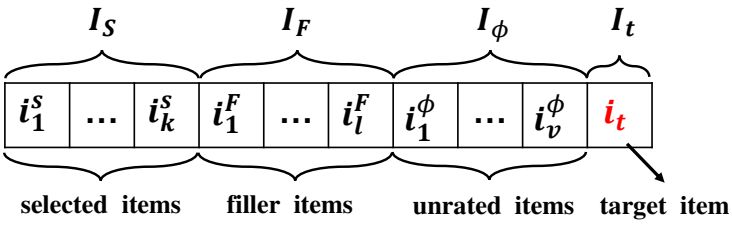


图 2-1 托攻击恶意用户的一般形式

攻击者使用不同的策略来为不同类型的项目赋予评分，归纳如下：

- $\mathcal{I}_F$（填充项目集合）：托攻击者模仿正常用户为集合 $\mathcal{I}_F$ 中的项目生成评分数据，以此将自己伪装成正常用户，使得推荐平台难以进行检测。根据攻击者得到的先验知识（例如，评分数据的分布），存在着多种为填充项目生成评分数据的策略。
- $\mathcal{I}_S$（选择项目集合）：此项目集合中的评分数据根据一个特殊的函数来生成。这些项目评分一般被用于特殊的攻击行为，比如被用于基于外部附加信息的群体攻击。这个项目集合在本文中被忽略，因为本文没有考虑带有外部附加信息的托攻击者。
- $\mathcal{I}_\Phi$（未评分项目集合）：此集合中的项目不进行评分。
- $\mathcal{I}_t$（目标项目集合）：恶意用户基于托攻击的目的，为这个集合中的目标项目赋予极端的评分。对于推攻击，恶意用户会给目标项目赋予非常高的评分。而对于核攻击，则会对目标项目赋予极低的评分。

根据上述讨论，如何为填充项目生成评分是托攻击最关键的一步，因为如果攻击者可以利用特殊的策略以及正常用户评分数据分布的知识来生成与正常用户评分相似的恶意用户，那么推荐平台将很难识别攻击者。例如，若采用随即攻击策略，托攻击者生成的恶意用户评分将服从以整个系统中的评分平均数为均值的正态分布。而采用均值攻击策略则会更加有效，因为在这种策略下，托攻击恶意用户的填充项目使用对应项目的平均评分作为正态分布的均值[14]，这将生成更加复杂且真实的评分数据分布。更全面的用户先验评分知识将有利于设计更加精细有效的托攻击策略，例如，完美知识攻击[40]，采样攻击[41]，热门项目攻击[42] 以及 Average-over-Popular (AoP) 攻击[43]。

接下来，我们讨论如何对现有的托攻击策略进行修改，让其可以在联邦学习场景下，对推荐系统发动有效托攻击。将现有的托攻击策略修改并移植到联邦学习系统中是相对容易的，因为联邦学习本身是一个相对开放的框架，用户作为训练

数据拥有方可以自由地加入到整个系统的训练过程中。因此，恶意用户很难被系统平台预先禁止加入训练，它们可以与正常用户一样直接加入到推荐系统中。托攻击者可以与针对传统推荐系统的策略一样，通过生成并注入带有特殊评分数据的恶意用户来影响联邦推荐系统的推荐结果。在联邦推荐系统中，所有用户都不会公开自己的评分数据，托攻击者难以获得关于正常用户的评分数据分布信息。因此，复杂的托攻击策略难以在联邦推荐系统场景下进行应用。然而，这实际上对攻击者和防御者都会产生影响。对于服务提供商（参数服务器）来说，获得相关的用户评分数据分布也是一样困难的，因此也难以利用真实用户的数据分布将正常用户与托攻击恶意用户区分开来。

在本文中，我们对评分数据分布的先验知识部分可获得以及先验知识不可获得这两种情况都进行了研究。一般来说，获取关于用户评分数据的分布信息是不容易的，因为在联邦学习场景下用户不会公开自己的隐私数据。但事实上，我们仍然有可能获得关于用户对于每一个项目的评分数据分布信息。例如，用户对视频的评分数据分布可以从 IMDB、豆瓣等平台获取，因为这些平台通常会公开电影评分的均值。即使我们无法通过这些公共平台获取每个项目的评分分布，我们仍然可以在不侵犯用户隐私的情况下，通过本地差分隐私来获取近似的数据分布特征[44]。同时，我们假设这类关于项目评分的先验知识可以同时被攻击者和推荐平台利用。对于攻击者而言，这部分信息可以用来生成更加“真实”的恶意用户，让推荐平台难以进行托攻击检测；对于推荐平台而言，可以利用真实用户产生的评分数据分布特征，更有效地区分恶意用户与真实用户。针对不同的先验知识，本文涵盖的攻击策略包括盲目攻击（对应于没有任何先验知识），随机攻击以及均值攻击（对应于攻击者可以获得部分先验知识）。

随机攻击和均值攻击已在上文进行详细阐述。盲目攻击指的是，在评分分布信息完全未知的场景下，恶意用户伪造的填充项目评分直接通过均匀分布产生。

2.6 本章小结

在这一章节中，我们简要介绍了本论文涉及课题的背景知识和研究现状。我们对推荐系统、托攻击、联邦学习以及差分隐私进行了简要介绍。同时我们论述了如何在联邦推荐系统中进行托攻击。

从下一章节开始，我们将介绍在 Fed-CF 用户在不考虑差分隐私机制的场景下的托攻击检测算法设计。在这个基础上，我们将进一步分析当用户采用差分隐私机制扰乱梯度真实值时，联邦托攻击检测器 (FSAD) 性能会受到的影响。

三、联邦托攻击检测算法的设计

在本章节中，我们将首先介绍 Fed-CF 算法以解释在联邦推荐系统场景下，哪些信息可用被参数服务器收集并用以设计特征来区分恶意用户。之后我们将详细对每一个特征的设计进行介绍。

表3.1列举了一系列本文的常用符号。

表 3.1 常用符号表

符号	定义
$\nabla q_i^{(u)}$	用户 u 的第 i 个梯度向量
∇q_i	所有用户的梯度 $\nabla q_i^{(u)}$ 的均值
$\mathcal{U}$	所有用户的集合
$\mathcal{U}_i$	对项目 i 进行评分的用户集合
m_u	用户 u 进行过的评分数量
n_i	所有用户提供的对项目 i 的评分数量
M	系统中用户的数量
N	系统中项目的数量, 即, $N = \|\mathcal{U}\|$
K	隐空间维度
$\mathcal{M}_u$	被用户 u 进行过评分的项目的集合
$\mathcal{M}_T$	潜在目标项目的集合
$\mathcal{M}_{u,T}$	用户 u 中的潜在目标项目集合
$\mathcal{M}_{u,F}$	用户 u 相关的剩余项目, 即, $\mathcal{M}_{u,F} = \mathcal{M}_u - \mathcal{M}_{u,T}$

3.1 Fed-CF 简介

目前已有大量的基于协同过滤的推荐算法被研究与应用。理论上，它们都能够被简单修改后应用在联邦学习场景下，因为联邦学习是一种通用的框架。在本文中，我们只考虑 Fed-CF 的应用场景，我们将基于 Fed-CF 算法的特点开展研究工作。事实上我们的研究结论能够被很容易的扩展到其他基于联邦学习的推荐系统上，因此我们的研究工作具有一定的普适性。

Fed-CF 是基于 SVD 的推荐系统的一种扩展版本。根据基于 SVD 的推荐系统的定义，系统最终给出的预测评分矩阵是两个低维矩阵的乘积。令 **P** 表示用户隐矩阵，**Q** 表示项目隐矩阵，则有

$$\mathbf{R} = \mathbf{P}^T\mathbf{Q}, \tag{3.1}$$

其中 $\mathbf{R} \in \mathbb{R}^{N \times M}$ 代表拥有 N 个用户和 M 个项目的用户-项目交互矩阵（例如，评分矩阵）。如果隐空间维度为 K，那么有 $\mathbf{P} \in \mathbb{R}^{K \times N}$ 和 $\mathbf{Q} \in \mathbb{R}^{K \times M}$。一般而言，$K \ll M, N$。对于某个用户 u 对项目 i 的预测评分而言，它会由 $\mathbf{P}$ 的第 u 列和 $\mathbf{Q}$ 的第 i 列的线性组合进行预测：

$$\hat{\mathbf{r}}_{ui} = \mathbf{p}_u^T \mathbf{q}_i = \sum_{k=1}^{K} p_{uk} q_{ik}. \tag{3.2}$$

$\mathbf{P}$ 和 $\mathbf{Q}$ 可以利用随机梯度下降 (Stochastic Gradient Descent, SGD) 算法，根据以下公式迭代计算出近似最优。

$$\epsilon_{ui} \leftarrow r_{ui} - \mathbf{q}_i^{\mathrm{T}} \cdot \mathbf{p}_u, \tag{3.3}$$

$$\mathbf{p}_u \leftarrow \mathbf{p}_u + \beta(\epsilon_{ui}\mathbf{q}_i - \rho\mathbf{p}_u), \tag{3.4}$$

$$\mathbf{q}_i \leftarrow \mathbf{q}_i + \beta(\epsilon_{ui}\mathbf{p}_u - \rho\mathbf{q}_i). \tag{3.5}$$

这里 r_{ui} 代表用户 u 对项目 i 的评分的真实值，β 代表本地迭代训练的学习率。

根据相关文献[10] 对联邦学习的定义，参数服务器会与终端用户进行多轮迭代通信。Fed-CF 扩展了由公式 (3.3)-(3.5) 给出的解决方案。用户只需要与参数服务器按照以下步骤来更新 $\mathbf{q}_i$：

- 若用户 u 被参数服务器选中，那么用户 u 在本地执行公式 (3.3)-(3.5) 中的计算任务。
- 用户 u 将 $\mathbf{p}_u$ 保持在本地进行更新，本地迭代结束后将更新后的 $\mathbf{q}_i$ 上传给参数服务器。
- 参数服务器根据特定规则对从选定用户中接收到的 $\mathbf{q}_i$ 进行聚合，然后将聚合后的模型分发给所有用户，进行下一轮全局迭代。

Fed-CF 的具体细节展示在算法3.1中。从该算法中我们可以看出，推荐系统平台无法获得用户评分数据，可能被用于托攻击检测算法特征设计的信息只有项目矩阵的梯度 $\nabla\mathbf{Q}^{(u)}$。

算法 3.1: 联邦协同过滤 (Fed-CF)

输入: $\mathcal{U}$: 用户的集合
输出: 每个用户 u 的推荐结果: $\mathbf{P}_u^T\mathbf{Q}$

Procedure `Server execution()`
  初始化 Q;
  对于每个 全局迭代轮次 *t = 1,2,...* **进行**
    对于每个 被选中的用户 $u \in \mathcal{U}$ 并行地 **进行**
      $\nabla\mathbf{Q}^{(u)}$ = `ClientUpdate(Q)`;
    $\mathbf{Q} = \mathbf{Q} - \frac{\alpha}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\nabla\mathbf{Q}^{(u)}$;
Function `ClientUpdate(Q)`
  对于 本地迭代轮次 $i = 1$ **转到** E **进行**
    更新用户隐向量 $\mathbf{P}_u$;
  计算项目隐向量的梯度: $\nabla\mathbf{Q}^{(u)}$;
  返回 $\nabla\mathbf{Q}^{(u)}$ 到参数服务器;

3.2 基于梯度的托攻击检测特征设计

推荐服务提供商（参数服务器）拥有了每个用户 u 的梯度信息（即，$\nabla\mathbf{Q}^{(u)}$）后，便可以开始设计基于这些梯度信息的特征来区分恶意用户与正常用户，以此进行托攻击检测。

直观上讲，我们设计这些特征的目的是将正常用户与恶意用户聚类到不同的类别中。在理想情况下，我们设计的特征能够帮助我们将具有相似攻击行为特征的恶意用户分类到一个聚类中，同时将具有相似兴趣偏好的正常用户聚集到其他不同的类别中。然后我们便可以通过训练一个分类器来区分系统中的恶意用户与正常用户。在联邦学习场景下，用户的数据分布特征为非独立同分布 (non-iid)，即每个用户的本地训练数据会存在一定偏好，与全局数据的分布情况差异较大。我们设计的特征从梯度信息中抽取得到，而梯度信息则是由参数服务器对系统中所有的用户进行收集得到。这意味着我们实际上是在利用全局信息进行托攻击恶意用户检测，而非仅根据某个特定用户的梯度信息进行分类。因此，本文提出的检测方法不会受到 Fed-CF 场景下用户数据非独立同分布特性的影响。

正如我们在上文对 Fed-CF 算法的描述，每一个参与训练的用户都需要在每一轮全局迭代中，将经过本地迭代更新得到的维度为 $M \times K$ 的梯度矩阵上传至参数服务器，然后由参数服务器聚合得到新的全局模型。我们将用户 u 在全局轮次 t 上传给参数服务器的梯度矩阵定义为 $\nabla\mathbf{Q}_t^{(u)}$。事实上，$\nabla\mathbf{Q}_t^{(u)}$ 是非常稀疏的矩阵，因为单个用户进行过评分的项目占平台总项目的比例是非常小的（例如，一个视频网站有上万部电影，而一个用户一般只会对其中几十部电影进行观看和评分）。

所有被用于联邦托攻击检测的特征都从每个用户所有轮次提交的梯度总和中抽取得到。每个用户的上传梯度总和可以形式化定义为：

$$\nabla \mathbf{Q}^{(u)} = \sum_{t=1}^{T} \nabla \mathbf{Q}_t^{(u)}, \tag{3.6}$$

其中 T 为全局迭代的轮次。令 $\nabla q_i^{(u)}$ 表示梯度矩阵 $\nabla \mathbf{Q}^{(u)}$ 的第 i 列，即，

$$\nabla \mathbf{Q}^{(u)} = \left[\nabla q_1^{(u)}, \nabla q_2^{(u)}, \nabla q_3^{(u)}, \ldots, \nabla q_M^{(u)}\right], \tag{3.7}$$

其中 M 为推荐系统中的项目总数。

我们选择从 $\nabla \mathbf{Q}^{(u)}$ 中提取梯度的原因可以从两个方面进行解释：首先，根据算法3.1中参数服务器端的模型更新规则可知，一个用户对 Fed-CF 的全局模型 $\mathbf{Q}$ 的贡献主要体现在该用户提交的所有梯度向量的矢量和。由于模型更新的随机性等原因，仅从一个全局迭代轮次上传的梯度信息中很难确定用户对模型的实际贡献，这个现象在一项关于联邦学习的研究中[11] 中有所体现。其次，攻击者的累计梯度向量的方向必须偏离正常用户的梯度，只有这样才能影响模型参数的更新，让模型参数最终无法收敛到真实的最优值附近，从而影响推荐模型的准确性。

在对特征设计的细节进行介绍之前，我们首先讨论与之相关的两个关键变量 m_u（用户 u 进行过的评分数量）和 n_i（所有用户提供的对项目 i 的评分数量）的计算方式。如果用户-项目评分矩阵可以获得（对应于传统的推荐系统场景），那么 m_u/n_i 能够直接通过计算第 u 行/第 i 列大于 0 的项目得到。在 Fed-CF 场景下，评分矩阵作为隐私数据不被公开，但是我们可以通过计算项目梯度中的非零元素来获得相应的数据。一个有趣的事实为，$\left\|\nabla q_i^{(u)}\right\|_1 > 0$ 表示用户 u 上传的梯度对项目矩阵 $\mathbf{Q}$ 第 i 列的贡献不为 0，因此可以推断出该用户在项目 i 上进行过评分。

变量 m_u 和 n_i 会出现在基于梯度的特征定义当中，因此在实际场景下，我们需要在计算特征之前计算出这两个变量值。

3.2.1 梯度偏离平均度 (GDMA)

我们设计梯度偏离平均度 (Gradient Deviation from Mean Agreement, GDMA) 来度量用户梯度与其平均值的偏差。我们利用 ℓ_2 范数来表示两个向量之间的距离，将 GDMA 定义为：

$$GDMA_u = \frac{\sum_{i \in \mathcal{M}_u} \frac{\left\|\nabla q_i^{(u)} - \nabla q_i\right\|_2}{n_i}}{m_u}. \tag{3.8}$$

其中 ∇q_i 为所有为项目 i 进行过评分的用户提供的梯度 $\nabla q_i^{(u)}$ 的平均值。因此，$\left\|\nabla q_i^{(u)} - \nabla q_i\right\|_2$ 度量了用户 u 对项目 i 的梯度与平均值的偏差。我们简要对 GDMA 的有效性进行分析：若一批恶意用户针对某个特定的项目 i 赋予极高或极低的评分进行攻击，那么这些攻击者的梯度与正常用户产生的梯度之间的距离将会非常远，因为根据公式 $\epsilon_{ui} = r_{ui} - \mathbf{q}_i^{\mathrm{T}} \cdot \mathbf{p}_u$，攻击者的预测误差 ϵ_{ui} 的绝对值将会显著高于正常的用户。例如，若攻击者通过对目标项目赋予极低的评分来发动核攻击，那么攻击者的 ϵ_{ui} 数值会比正常用户低很多。因此，只要攻击者的数量不能支配整个推荐系统，具有极高 GDMA 特征值的用户便可以被识别为潜在的攻击者。其余特征设计原理与 GDMA 相似，因此可以用类似的方式进行有效性分析。

一种与 GDMA 相似的，被称为评级偏离平均度 (Rating Deviation from Mean Agreement, RDMA) 的特征在已有的研究中[18] 被提出并证实可以有效用于托攻击检测。然而，评分数据在联邦学习场景下无法被参数服务器获得，因此我们设计 GDMA 以代替 RDMA。

3.2.2 加权梯度偏离度 (WGDA)

加权梯度偏离度 (Weighted Gradient Degree of Agreement, WGDA) 是 GDMA 的一种变体。WGDA 实际上是 GDMA 的分子部分，可以被形式化定义为：

$$WGDA_u = \sum_{i \in \mathcal{M}_u} \frac{\left\|\nabla q_i^{(u)} - \nabla q_i\right\|_2}{n_i}. \tag{3.9}$$

一种类似的特征在文献[17] 中被提出用以托攻击检测。WGDA 是该特征在联邦学习框架下的一个变体。

3.2.3 近邻相似度 (DegSim)

在梯度空间中，以攻击某一个项目为目的托攻击恶意用户会比其他正常用户更加相互靠近。在本文中，我们使用余弦相似度来度量两个用户梯度之间的相似程度。我们首先将梯度矩阵 $\nabla \mathbf{Q}^{(u)}$ 的每一个项目（每一列）取 ℓ_1 范数，即 $\left\|\nabla q_i^{(u)}\right\|_1$，从而将其梯度矩阵转换为一维向量。用户 u 经过转换得到的梯度向量可以表示为 $\mathbf{h}_u = \left[\left\|\nabla q_1^{(u)}\right\|_1, \left\|\nabla q_2^{(u)}\right\|_1, \ldots, \left\|\nabla q_M^{(u)}\right\|_1\right]^T$。在这基础上，我们给出近邻相似度 (Degree of Similarity with Top Neighbors, DegSim) 的定义：

$$DegSim_u = \frac{\sum\limits_{v \in Neigh(u,z)} cos(\mathbf{h}_u, \mathbf{h}_v)}{z}. \tag{3.10}$$

其中，$Neigh(u,z)$ 表示与用户 u 的余弦相似度前 z 大的用户集合。

近邻相似度在文献[18] 中出现并被用于度量两个评分向量的相似度。相关研究表明[11]，梯度的余弦相似度在联邦学习中可以被有效利用进行攻击检测。近邻相似度定义中的 z 是一个超参数，在我们的实验中，我们参考现有研究文献中的实验内容[18]，设置 $z=25$。

3.2.4 填充平均目标偏差 (FMTD)

假设我们可以确定一批很可能遭受了托攻击的目标项目，记为 $\mathcal{M}_T$。我们可以基于 $\mathcal{M}_T$ 设计填充平均目标偏差 (Filler Mean Target Difference, FMTD) 这个特征。我们将首先介绍如何根据 $\mathcal{M}_T$ 来定义 FMTD，然后再介绍如何利用两个基于项目的属性确定疑似遭受托攻击的项目集合 $\mathcal{M}_T$。

令 $\mathcal{M}_u$ 表示用户 u 进行过评分的项目的集合。若 $\mathcal{M}_T$ 已知，那么用户 u 的潜在被攻击项目集合可以表示为 $\mathcal{M}_{u,T}=\mathcal{M}_u\cap\mathcal{M}_T$，而用户 u 的潜在填充项目集合为 $\mathcal{M}_{u,F}=\mathcal{M}_u-\mathcal{M}_{u,T}$。FMTD 度量了目标项目对应的梯度向量和填充项目对应的梯度向量之间的差异。FMTD 形式化的定义如下：

$$FMTD_u=\left\|\frac{\sum_{\forall i\in\mathcal{M}_{u,T}}\nabla q_i^{(u)}}{|\mathcal{M}_{u,T}|}-\frac{\sum_{\forall i'\in\mathcal{M}_{u,F}}\nabla q_{i'}^{(u)}}{|\mathcal{M}_{u,F}|}\right\|_2. \tag{3.11}$$

对于托攻击者而言，为了达到可观的攻击效果，他们需要为目标项目赋予极端的数值，以此对目标项目的相关参数更新贡献更多的梯度以扰乱推荐模型。而为了躲避系统平台的检测，攻击者们会为填充项目提供与正常用户相近的分数使自己的评分与正常用户看起来相似，在这种情况下，填充项目对应的梯度数值会相对“温和”，不会像目标项目的梯度一样极端。FMTD 使用 ℓ_2 范数来度量这两类梯度之间的差异。用户的 FMTD 值越大，那么该用户就越有可能是恶意用户。一种类似的特征已被成功应用于托攻击检测[17]，但是需要获取用户评分信息。

接下来，我们详细介绍如何确定潜在的目标项目集合 $\mathcal{M}_T$。到目前为止，我们设计的特征都是从用户的角度出发的。同样的，我们也可以从推荐项目的角度出发来设计一些特征，通过这些特征来辨别疑似被攻击的项目。在这项研究中，我们采用了两种属性来确定潜在的目标项目集合 $\mathcal{M}_T$。

我们可以合理地假设，为了有效影响推荐模型的准确性以达到托攻击目的，攻击者会给目标项目赋予极高或极低的评分。因此，攻击者和正常用户贡献的梯度会倾向于分布在两个不同的方向，遭受攻击的项目对应的梯度很可能会形成两个不同的簇。为了验证这个观点，我们将协同过滤模型的隐维度设为 $K=2$，向拥

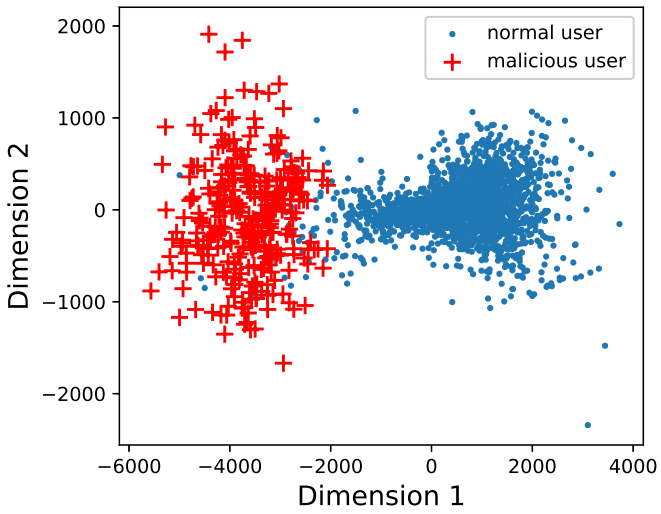


图 3-1 被攻击项目的梯度分布。梯度通过在有 6,040 个普通用户和 5 %（302 个）攻击者的 MovieLens 数据集上训练得到

有 6,040 个正常用户的 MoiveLens 数据集中注入 5% 的攻击者（302 个恶意用户）。图3-1展示了对目标项目进行过评分的用户的梯度分布。图中的红色“加号”表示攻击者，它们显著远离以蓝色“圆点”表示的正常用户梯度。

这个实验结果启发我们设计了称为梯度方差的度量：

$$Var_i = \frac{1}{n_i} \sum_{u \in \mathcal{U}_i} \left\| \nabla q_i^{(u)} - \nabla q_i \right\|_2^2, \tag{3.12}$$

其中 $\mathcal{U}_i$ 为对项目 i 进行评分的用户的集合，我们令 $n_i = |\mathcal{U}_i|$。直观地讲，攻击者倾向于为目标项目赋予极端的评分。因此，被攻击项目的梯度分布应该是非常分散的。项目 i 的梯度方差 Var_i 数值越大，那么它便越有可能遭受攻击。在本文的后续实验中，我们选取了梯度方差前 5 大的项目作为潜在目标项目。

第二个用于确定潜在目标项目的度量基于用户 u 对项目 i 的模型贡献程度设计。我们使用 ℓ_1 范数（即，$\left\| \nabla q_i^{(u)} \right\|_1$）来度量梯度的贡献程度。托攻击者的目标是通过更改目标项目的梯度方向来影响最终的推荐模型。因此，攻击者倾向于为目标项目对应的模型参数更新提供更极端并且数值更大的梯度数值，从而通过参数服务器梯度聚合步骤有效影响全局模型。相反，填充项梯度数值则会显著小于被攻击项目的梯度。我们将 $TopG(u, g)$ 定义为 $\left\| \nabla q_i^{(u)} \right\|_1$ 数值前 g 大的项目。

在拥有了 $TopG(u, g)$ 的相关数据后，我们便可以统计每一个项目在所有用户的 $TopG(u, g)$ 中出现的次数。我们给出以下定义：

$$\xi_i = \sum_{u=1}^{N} I\left(i \in TopG(u, g)\right). \tag{3.13}$$

其中 $I(\cdot)$ 是一个指示函数，若 i 出现在 $TopG(u,g)$ 中则返回 1，否则返回 0。我们选择 ξ_i 数值前 k 大的项目作为潜在目标项目。在后续的实验中，我们设置 $g=5$，并且选择前 5 大的 ξ 对应的项目。根据我们实验结果，托攻击检测效果对 g 和 k 的数值选择不是特别敏感。我们只需要避开一些极端的数值（例如，1 或 $\mathcal{O}(M)$）就能取得良好的检测效果。在实验中我们将 g 和 k 的值从 5 增长到 20，发现效果差别不大。最终我们选择将两个参数都设置为 5 进行实验效果作图展示。

最终由这两个度量确定下来的潜在目标项目集合 $\mathcal{M}_T$ 由上面给出的两个潜在项目集合的并集构成。综上所述，我们基于参数服务器可以获得的梯度信息，设计了 GDMA, WGDA, DegSim 和 FMTD 这四个特征值用来训练联邦托攻击检测器。

3.3 用于托攻击检测的半监督分类器

一种被称为 EM-λ 的半监督朴素贝叶斯分类器在文献[45] 被提出，并且被证实在托攻击检测任务中有良好的性能表现[19]。这个模型也能够用来处理联邦学习场景下的托攻击检测问题。

在本文定义的问题中，从参数服务器视角来看，我们有两类用户数据：无标签数据（由系统中的真实用户和托攻击者共同产生）和有标签数据（参数服务器模拟的真实用户和恶意用户）。所有用户的梯度特征数据和模拟用户的标签被共同用于训练 EM-λ 分类器。EM-λ 模型的训练思想是将带标签的数据和无标签数据共同耦合到同一个目标函数中，通过优化这个目标函数进行半监督式训练。我们将简要介绍如何将 EM-λ 进行调整以适用于联邦托攻击检测任务。

我们将所有用户的特征集合记为 $\mathcal{D}=\{\mathbf{x}_u\}_{u=1}^{N}$。其中 $\mathbf{x}_u$ 表示用户 u 的特征向量。推荐系统中所有用户都会被分类到 c 个类别的其中之一，我们将这些类别记为 $\mathcal{C}=\{C_1,C_2,\ldots,C_c\}$。在本文中，我们主要研究三种攻击策略（即，盲目攻击，随机攻击和均值攻击），具体选取何种攻击策略取决于攻击者可获得的关于评分分布的先验知识。利用不同程度的先验知识，可以使用不同类型的策略生成填充项评分，从而发动不同类别的托攻击。我们将这些攻击方式分别命名为均匀填充模型 (uniform filler model, UF)、随机填充模型 (random filler model, RF) 以及均值填充模型 (average filler model, AF)。因此，在我们研究的问题定义下，所有用户都将被分类到 $\mathcal{C}=\{N,AF,RF,UF\}$ 这四个类别之一，其中 "N" 代表的是正常用户类别。

假设我们有 Y 个特征值，各特征的类条件概率是独立且服从正态分布的，那

么有

$$\begin{aligned} p(\mathbf{x}_u|C_j;\boldsymbol{\theta}) &= \prod_{y=1}^{Y} p(x_{uy}|C_j;\boldsymbol{\theta}_j) \\ &= \prod_{y=1}^{Y} \frac{1}{\sqrt{2\pi}\sigma_{jy}} \exp\left(\frac{-(x_{uy}-\mu_{jy})^2}{2\sigma_{jy}^2}\right), \end{aligned} \tag{3.14}$$

其中 $\boldsymbol{\theta} = \{\boldsymbol{\theta}_j\}_{j=1}^{c}$ 为模型参数并且有 $\boldsymbol{\theta}_j = \{\mu_{jy}, \sigma_{jy}\}_{y=1}^{Y}$。

已知 C_j 的先验概率，即 $P(C_j)$，那么每一个用户出现的概率为

$$p(\mathbf{x_u}|\boldsymbol{\theta}) = \sum_{j=1}^{c} p(\mathbf{x_u}|C_j;\boldsymbol{\theta}_j)P(C_j). \tag{3.15}$$

令 $\mathcal{D}^U$ 表示 $\mathcal{D}$ 中的无标签数据，D^{L_j} 表示类别为 j 的有标签数据。那么，我们有 $\mathcal{D} = \mathcal{D}^U \cup \mathcal{D}^N \cup \mathcal{D}^{AF} \cup \mathcal{D}^{RF} \cup \mathcal{D}^{UF}$。我们的目标是执行最大似然估计来找到可以最大化 $\mathcal{D}$ 出现概率的最优参数 $\hat{\boldsymbol{\theta}}$ 。

为了减轻无标签数据的影响，我们将权重系数 $\lambda \in [0,1]$ 引入到分类模型当中。于是，数据 $\mathcal{D}$ 对于模型参数 $\boldsymbol{\theta}$ 的对数似然函数可以表示为

$$\begin{aligned} l_\lambda(\mathcal{D}|\boldsymbol{\theta}) &= \ln p(\mathcal{D}|\boldsymbol{\theta}) \\ &= \ln\left\{\left(\prod_{\mathbf{x_u}\in\mathcal{D}^U} p\left(\mathbf{x}_u|\boldsymbol{\theta}\right)\right)^{\lambda} \prod_{j=1}^{c}\left(\prod_{\mathbf{x_u}\in\mathcal{D}^{L_j}} p\left(\mathbf{x}_u|\boldsymbol{\theta}\right)\right)\right\} \\ &= \sum_{\mathbf{x_u}\in\mathcal{D}} \Lambda_u \ln p(\mathbf{x_u}|\boldsymbol{\theta}), \end{aligned} \tag{3.16}$$

其中

$$\Lambda_u = \begin{cases} \lambda, \text{若 } \mathbf{x}_u \in \mathcal{D}^U \\ 1, \text{其它}. \end{cases} \tag{3.17}$$

模型的目标函数定义如下：

$$\begin{aligned} \max_{\boldsymbol{\theta}} \quad & l_\lambda(\mathcal{D}|\boldsymbol{\theta}) \\ \text{s.t.} \quad & \sum_{j=1}^{c} P(C_j) = 1. \end{aligned} \tag{3.18}$$

公式 (3.18) 定义的优化问题可以通过 EM 算法求解

我们在下文列出了 EM 算法中需要用到的等式。根据公式 (3.14)，最优的参数

估计 $\hat{\mu}_{jy}$ 和 $\hat{\sigma}^2_{jy}$ 可以由以下公式得出:

$$\hat{\mu}_{jy} = \frac{\lambda \sum_{\mathbf{x_u} \in \mathcal{D}^U} P(C_j|\mathbf{x}_u; \hat{\boldsymbol{\theta}}) x_{uy} + \sum_{\mathbf{x}_u \in \mathcal{D}^{L_j}} x_{uy}}{\lambda \sum_{\mathbf{x}_u \in \mathcal{D}^U} P(C_j|\mathbf{x}_u; \hat{\boldsymbol{\theta}}) + |\mathcal{D}^{L_j}|}, \tag{3.19}$$

$$\hat{\sigma}^2_{jy} = \frac{\lambda \sum\limits_{\mathbf{x_u} \in \mathcal{D}^U} P(C_j|\mathbf{x}_u; \hat{\boldsymbol{\theta}})(x_{uy} - \hat{\mu}_{jy})^2 + \sum\limits_{\mathbf{x}_u \in \mathcal{D}^{L_j}} (x_{uy} - \hat{\mu}_{jy})^2}{\lambda \sum_{\mathbf{x}_u \in \mathcal{D}^U} P(C_j|\mathbf{x}_u; \hat{\boldsymbol{\theta}}) + |\mathcal{D}^{L_j}|}, \tag{3.20}$$

其中 $1 \le j \le c, 1 \le y \le Y$。任意用户属于类别 j 的概率为

$$\hat{P}(C_j) = \frac{\lambda \sum_{\mathbf{x}_u \in \mathcal{D}^U} P(C_j|\mathbf{x}_u; \hat{\boldsymbol{\theta}}) + |\mathcal{D}^{L_j}|}{\lambda |\mathcal{D}^U| + \sum_{j=1}^{c} |\mathcal{D}^{L_j}|}. \tag{3.21}$$

公式 (3.19) 和公式 (3.20) 中的概率 $P(C_j|\mathbf{x}_i; \boldsymbol{\theta})$ 可以通过贝叶斯公式进行计算:

$$P(C_j|\mathbf{x}_u; \hat{\boldsymbol{\theta}}) = \frac{p(\mathbf{x}_u|C_j; \hat{\boldsymbol{\theta}}) P(C_j)}{p(\mathbf{x}_u; \hat{\boldsymbol{\theta}})} \tag{3.22}$$

EM 算法通过以下步骤来迭代式地估计参数 $\boldsymbol{\theta}$:

- **参数初始化**: 对于带标签数据, $\hat{\mu}_{uy}$, $\hat{\sigma}^2_{uy}$ 以及 $\hat{P}(C_j)$, 分别由公式 (3.19), (3.20) 和 (3.21) 计算得到。对于无标签数据 $\mathbf{x}_u \in \mathcal{D}^U$, 使用公式 (3.14) 来计算 $p(\mathbf{x}_i|C_j; \boldsymbol{\theta})$。
- **E 步骤 (Expectation step)**: 对于无标签数据 $\mathbf{x}_u \in \mathcal{D}^U$, 使用公式 (3.22) 计算 $P(C_j|\mathbf{x}_u; \hat{\boldsymbol{\theta}})$。对于有标签数据 $\mathbf{x}_u \in \mathcal{D}^{L_j}$, 我们直接按照 $P(C_j|\mathbf{x}_u; \hat{\boldsymbol{\theta}}) = 1; P(C_{j'}|\mathbf{x}_u; \hat{\boldsymbol{\theta}}) = 0, \forall j' \ne j$ 进行赋值。
- **M 步骤 (Maximization step)**: 分别使用公式 (3.19), (3.20) 和 (3.21) 计算 $\hat{\mu}_{uy}$, $\hat{\sigma}^2_{uy}$ 和 $\hat{P}(C_j)$。对于五标签数据 $\mathbf{x}_u \in \mathcal{D}^U$, 使用公式 (3.14) 计算出 $p(\mathbf{x}_i|C_j; \boldsymbol{\theta})$。

E-M 两个步骤交替执行直至模型参数收敛。最终, 对于所有 $\mathbf{x}_u \in \mathcal{D}^U$, 我们可以计算出 $P(C_j|\mathbf{x}_u; \hat{\boldsymbol{\theta}})$。在我们的研究问题中, 用户可以被分为四个类别: N, RF, AF, UF。于是用户 u 为攻击者的总的概率为

$$P(A|\mathbf{x}_u; \hat{\boldsymbol{\theta}}) = P(RF|\mathbf{x}_u; \hat{\boldsymbol{\theta}}) + P(AF|\mathbf{x}_u; \hat{\boldsymbol{\theta}}) + P(UF|\mathbf{x}_u; \hat{\boldsymbol{\theta}}).$$

最后, 若用户 i 有 $P(N|\mathbf{x}_i; \hat{\boldsymbol{\theta}}) < P(A|\mathbf{x}_i; \hat{\boldsymbol{\theta}})$, 则该用户被归类为攻击者。

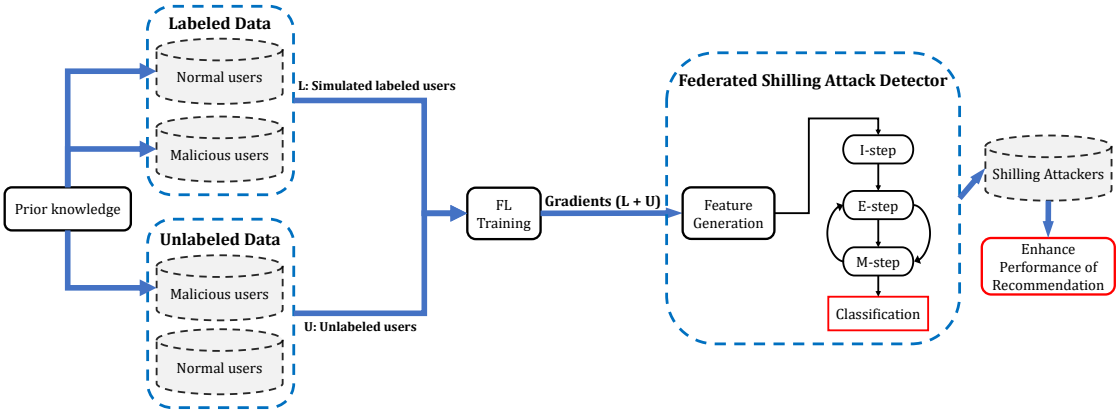


图 3-2 FSAD 框架

3.4 FSAD 的完整框架

图3-2展示了 FSAD 的工作流程，细节如下：

1) 托攻击发动者基于先验知识，生成针对目标项目的恶意用户。FSAD 以相似的方式生成带有不同标签的模拟用户，用于分类器的半监督训练。需要注意的是，真正的攻击者和普通用户的标签对检测器而言是未知的。
2) 所有用户通过联邦学习平台参与推荐模型训练。
3) 从参数服务器角度来看，可以获得的信息包括所有用户上传的梯度以及模拟用户的标签。
4) 基于梯度数据，抽取四个新的特征来捕获恶意用户和普通用户之间的差异。
5) 利用特征和标签信息训练分类器来识别托攻击者。

值得一提的是，第一步的模拟用户会人为地产生带标签的正常用户和恶意用户，这些模拟用户的标签对于分类器的训练至关重要。

3.5 本章小结

在这一章节中，我们主要介绍了 FSAD 的设计细节。我们首先对 Fed-CF 算法进行介绍，指出推荐体系平台在联邦学习场景下可获得的用户信息是受限的。然后我们对基于梯度的托攻击检测特征进行详细的描述，对四种特征给出形式化的定义，同时分析其有效性。最后我们给出了托攻击检测半监督分类器的具体定义，包括模型参数与关键的训练步骤。

四、差分隐私噪声下的检测算法性能分析

4.1 隐私增强的 Fed-CF

尽管联邦学习框架中用户只需要通过交换梯度（或模型参数）的方式进行协同训练来保护隐私，最近的研究表明[22;23;24]，本地敏感数据是可以通过梯度信息被重构的。在联邦学习场景下，解决隐私泄露问题的一种自然的方法是通过给公开数据（如梯度）添加噪声，使用户公布梯度数据的方式满足差分隐私。目前已有多项关于利用差分隐私机制来提升联邦学习系统隐私保护水平的研究[29;30;31]。在推荐系统场景下，Hua 等人[7] 提出了满足差分隐私的矩阵分解算法来保护用户的评分数据隐私，Shen 等人[8] 将差分隐私机制应用于扰乱用户评分过的项目集合这个敏感数据，Shin[9] 等人将本地化差分隐私应用到矩阵分解算法以保护评分数值和被评分项目集合这两种隐私数据。

在本文中，我们分析了 FSAD 在基于差分隐私机制下的 Fed-CF 系统中的性能。我们采用与论文[7] 类似的差分隐私机制。为了方便后续讨论，我们先简单介绍差分隐私以及基于差分隐私机制的 Fed-CF 算法。

定义 4.1　一个随机化机制 $\mathcal{M}:\mathcal{D}\to\mathcal{R}$ 满足 (ϵ,δ)-差分隐私，若对于任意两个相邻的输入 $d,d'\in\mathcal{D}$ 与任意输出集合 $S\in\mathcal{R}$ 都有

$$\Pr\left[\mathcal{M}\left(d\right)\in\mathcal{S}\right]\leq e^{\epsilon}\Pr\left[\mathcal{M}\left(d'\right)\in\mathcal{S}\right]+\delta.$$

实现 (ϵ,δ)-差分隐私的常见方法是在原始数据上添加人为噪声来模糊化其真实数值。例如，高斯机制[27] 的定义如下

$$\mathcal{M}\left(d\right)\triangleq f(d)+\mathcal{N}\left(0,\sigma^{2}\right),$$

其中 $\mathcal{N}\left(0,\sigma^{2}\right)$ 表示服从方差为 σ^{2} 的高斯分布的噪声。这种类型的随机化机制已经被成功地应用于多种机器学习算法[28;29;31;30]。

与以论文[31] 为代表的一系列已有工作类似，基于差分隐私的 Fed-CF 通过给梯度添加噪声方式实现隐私保护。换言之，每个用户 u 独立地在各自的梯度上添加高斯噪声来扰乱其真实值，然后再上传至参数服务器进行聚合。为了保证能够达到 (ϵ,δ)-差分隐私，我们需要设置 $\sigma\geq\frac{\sqrt{\ln(1.25/\delta)}\Delta_{2}f}{\epsilon}$，其中 $\Delta_{2}f$ 是函数 f 的 ℓ_{2} 敏感

度。ℓ_2 敏感度的定义为：$\Delta_2 f = \max_{d,d'} \|f(d) - f(d')\|_2$，其中 d 和 d' 是两个相邻的输入数据。在 Fed-CF 中，每个用户的梯度矩阵就是高斯机制的输入数据。与现有的相关工作[28;29;31] 类似，我们能够通过“裁剪”梯度的方式控制梯度的敏感度，将梯度的 ℓ_2 敏感度限制在某个范围 C 以内，即，$\nabla \mathbf{Q}^{*,(u)} = \nabla \mathbf{Q}^{(u)} / \max\left(1, \frac{\|\nabla \mathbf{Q}^{(u)}\|_2}{C}\right)$。随后我们在每一个经过裁剪的项目梯度上添加服从高斯分布 $\mathcal{N}\left(0, \sigma^2 \mathbb{I}^K\right)$ 的噪声。这里的 $\mathcal{N}\left(0, \sigma^2 \mathbb{I}^K\right)$ 表示 K 维的随机向量，其中每个元素都由方差为 σ^2 的高斯分布独立生成。基于差分隐私的 Fed-CF 的工作细节在算法 4.1中展示。

算法 4.1: 差分隐私联邦协同过滤

输入: $\mathcal{U}$: 用户的集合; C: 裁剪梯度 $\nabla \mathbf{Q}^{(u)}$ 的阈值
输出: 每个用户 u 的推荐结果: $\mathbf{P}_u^T \mathbf{Q}$

Procedure `Server execution()`
  初始化 Q;
  对于每个 全局迭代轮次 *t = 1,2,...* **进行**
    对于每个 被选中的用户 $u \in \mathcal{U}$ 并行地 **进行**
      $\nabla \mathbf{Q}^{*,(u)}$ = `ClientUpdate(Q)`;
    $\mathbf{Q} = \mathbf{Q} - \frac{\alpha}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \nabla \mathbf{Q}^{*,(u)}$;
Function `ClientUpdate(Q)`
  对于 本地迭代轮次 $i = 1$ **转到** E **进行**
    更新用户隐向量 $\mathbf{P}_u$;
  计算项目隐向量的梯度: $\nabla \mathbf{Q}^{(u)}$;
  裁剪本地梯度

$$\nabla \mathbf{Q}^{*,(u)} = \nabla \mathbf{Q}^{(u)} / \max\left(1, \frac{\left\|\nabla \mathbf{Q}^{(u)}\right\|_2}{C}\right);$$

  对于每个 项目梯度向量中的非零元素 **进行**
    $\nabla\left(\mathbf{Q}^{*,(u)}\right)_i = \nabla\left(\mathbf{Q}^{*,(u)}\right)_i + \mathcal{N}\left(0, \sigma^2 \mathbb{I}^K\right)$;
  返回 $\nabla \mathbf{Q}^{*,(u)}$ 到参数服务器;

4.2 理论分析

接下来，我们将理论分析 Fed-CF 的推荐准确性和 FSAD 精度被差分隐私噪声影响的程度。直接分析差分隐私噪声对精度的影响是非常困难的，因此我们分析了特征和项目隐矩阵 $\mathbf{Q}$ 的估计误差与高斯噪声的方差之间的关系。

定理 4.1　假设我们使用高斯机制来保证 (ϵ, δ) 差分隐私。用户 u 在全局迭代第 t 轮，高斯噪声 $v^* \sim \mathcal{N}(0, \sigma^2)$ 被添加到梯度矩阵 $\nabla \mathbf{Q}^{(u),t}$ 中每个非零元素上，然后将加噪后的梯度矩阵上传至参数服务器。令 α 表示参数服务器端的学习率，T 表示全局迭代次数，$\mathbf{Q}^{*,t} \in \mathbb{R}^{K \times M}$ 表示在引入差分隐私机制后的第 t 轮迭代后得到

的全局模型（项目隐矩阵）。令 $\mathbf{Q}^*$ 表示由算法4.1导出的全局模型。在至少 $1-\xi$ 的概率下，我们有

$$\|\mathbf{Q}^* - \mathbf{Q}\|_{\max} = O\left(\frac{\sqrt{MKT \cdot \mathbf{Var}\left(v^*\right)}}{\sqrt{N\xi}}\right). \tag{4.1}$$

证明 首先，观察到

$$\mathbf{Q}^{t+1} = \mathbf{Q}^t - \alpha \cdot \nabla \mathbf{Q}^t, \tag{4.2}$$

和

$$\mathbf{Q}^{*,t+1} = \mathbf{Q}^{*,t} - \alpha \cdot \nabla \mathbf{Q}^{*,t}. \tag{4.3}$$

由公式 (4.2) 和公式 (4.3) 我们有

$$\mathbf{Q}^* - \mathbf{Q} = -\sum_{t=1}^{T} \alpha \cdot \left(\nabla \mathbf{Q}^{*,t} - \nabla \mathbf{Q}^t\right). \tag{4.4}$$

然后，根据切比雪夫不等式 (Chebyshev's inequality)，我们有

$$\begin{aligned}
&\Pr\left[\|\mathbf{Q}^* - \mathbf{Q}\|_{\max} \geq \tau\right] \\
&= \Pr\left[\left\|\sum_{t=1}^{T} \alpha\left(\nabla \mathbf{Q}^{*,t} - \nabla \mathbf{Q}^t\right)\right\|_{\max} \geq \tau\right] \\
&\leq \sum_{j=1}^{M}\sum_{l=1}^{K} \Pr\left[\left|\sum_{t=1}^{T} \alpha\left(\left(\nabla \mathbf{Q}^{*,t}\right)_{j,l} - \left(\nabla \mathbf{Q}^t\right)_{j,l}\right)\right| \geq \tau\right] \\
&= MK \cdot \Pr\left[\left|\sum_{t=1}^{T} \alpha\left(\left(\nabla \mathbf{Q}^{*,t}\right)_{j,l} - \left(\nabla \mathbf{Q}^t\right)_{j,l}\right)\right| \geq \tau\right] \\
&\leq MK \cdot \frac{\mathbf{Var}\left(\sum_{t=1}^{T} \alpha\left(\nabla \mathbf{Q}^{*,t}\right)_{j,l}\right)}{\tau^2} \\
&= MK \cdot \frac{\sum_{t=1}^{T} \alpha^2 \cdot \mathbf{Var}\left(\nabla \mathbf{Q}^{*,t}\right)_{j,l}}{\tau^2}.
\end{aligned} \tag{4.5}$$

注意到在第 t 轮全局迭代中，$\nabla \mathbf{Q}^{*,t}$ 通过以下规则进行聚合：

$$\nabla \mathbf{Q}^{*,t} = \frac{1}{N}\sum_{i=1}^{N}\left(\nabla \mathbf{Q}^{(i),t} + V^*\right), \tag{4.6}$$

其中 $V^* \in \mathbb{R}^{K\times M}$ 是与梯度 $\nabla\mathbf{Q}$ 具有相同位置非零元素的噪声矩阵。因此，公式 4.5可以被重写为

$$\begin{aligned} & MK \cdot \frac{\sum_{t=1}^{T} \alpha^2 \cdot \mathbf{Var}\left(\nabla\mathbf{Q}^{*,t}\right)_{j,l}}{\tau^2} \\ &= MK \cdot \frac{\sum_{t=1}^{T} \alpha^2 \cdot \mathbf{Var}\left(\frac{1}{N}\sum_{i=1}^{N}\left(\nabla\mathbf{Q}^{(i),t} + V^*\right)_{j,l}\right)}{\tau^2} \\ &= \frac{MK}{N} \cdot \frac{\sum_{t=1}^{T} \alpha^2 \cdot \mathbf{Var}\left(V^*_{j,l}\right)}{\tau^2} \\ &= O\left(\frac{MKT \cdot \mathbf{Var}\left(V^*_{j,l}\right)}{N\tau^2}\right). \end{aligned} \tag{4.7}$$

取 $\tau = O\left(\frac{\sqrt{MKT\cdot\mathbf{Var}\left(V^*_{j,l}\right)}}{\sqrt{N\xi}}\right)$，我们有

$$\Pr\left[\|\mathbf{Q}^* - \mathbf{Q}\|_{\max} = O\left(\frac{\sqrt{MKT \cdot \mathbf{Var}\left(v^*\right)}}{\sqrt{N\xi}}\right)\right] \leq 1 - \xi.$$

□

定理4.1表明，若我们执行算法4.1来训练推荐模型，那么项目隐矩阵 $\mathbf{Q}$ 的估计误差上界可以在一个常数因子内被限制在 $O\left(\sqrt{T \cdot \mathbf{Var}\left(v^*\right)}\right)$。

接下来，我们将理论证明四个基于梯度的特征的估计误差。

定理 4.2 假设我们使用高斯机制来满足 (ϵ, δ)-差分隐私。我们令 $GDMA_u$ 和 $GDMA_u^*$ 分别表示从用户 u 中未添加噪声的梯度 $\nabla\mathbf{Q}^{(u)}$ 和被噪声扰乱后的梯度 $\nabla\mathbf{Q}^{*,(u)}$ 中提取出来的特征。令 T 表示全局迭代次数，$v^* \sim \mathcal{N}\left(0, \sigma^2\right)$ 表示添加在梯度非零元素上面的高斯噪声。在概率至少为 $1-\xi$ 的情况下，$GDMA_u$ 和 $GDMA_u^*$ 之间的差异可以被限制在

$$O\left(\frac{\Gamma\left((K+1)/2\right)}{\xi \cdot \Gamma\left(K/2\right)} \cdot \sqrt{2T \cdot \mathbf{Var}\left(v^*\right)}\right), \tag{4.8}$$

其中 $\Gamma(\cdot)$ 是一个标准的伽马函数 (Gamma function)，定义如下：

$$\Gamma\left(x\right) = \int_0^{+\infty} t^{x-1}e^{-t}\,dt.$$

证明 令 $v_i^{(t)} \sim \mathcal{N}\left(0, \sigma^2\mathbb{I}^K\right)$ 代表在 t 轮迭代中，添加到项目 i 对应的梯度上的噪声

向量，其中 $\mathcal{N}\left(0,\sigma^2\mathbb{I}^K\right)$ 是每个元素方差为 σ^2 的 K 维高斯噪声。由三角不等式，我们有

$$\begin{aligned}
&|GDMA_u^* - GDMA_u| \\
&= \left| \frac{\sum_{i\in\mathcal{M}_u} \frac{\left\|\nabla q_i^{(u)} - \nabla q_i\right\|_2}{n_i}}{m_u} - \frac{\sum_{i\in\mathcal{M}_u} \frac{\left\|\nabla q_i^{(u)} + \sum_{t=1}^{T} v_i^{(t)} - \nabla q_i\right\|_2}{n_i}}{m_u} \right| \\
&\leq \frac{\sum_{i\in\mathcal{M}_u} \frac{\left\|\sum_{t=1}^{T} v_i^{(t)}\right\|_2}{n_i}}{m_u} = O\left(\|\sum_{t=1}^{T} v_i^{(t)}\|_2\right).
\end{aligned} \tag{4.9}$$

令 $v_{i,j}^{(t)}$ 表示 $v_i^{(t)}$ 中的第 j 个元素，即，

$$v_i^{(t)} = \left[v_{i,1}^{(t)}, v_{i,2}^{(t)}, \ldots, v_{i,K}^{(t)}\right]. \tag{4.10}$$

于是，我们有

$$\begin{aligned}
\|\sum_{t=1}^{T} v_i^{(t)}\|_2 &= \left\| \left[\sum_{t=1}^{T} v_{i,1}^{(t)}, \sum_{t=1}^{T} v_{i,2}^{(t)}, \ldots, \sum_{t=1}^{T} v_{i,K}^{(t)} \right] \right\|_2 \\
&= \sqrt{\left(\sum_{t=1}^{T} v_{i,1}^{(t)}\right)^2 + \left(\sum_{t=1}^{T} v_{i,2}^{(t)}\right)^2 + \cdots + \left(\sum_{t=1}^{T} v_{i,K}^{(t)}\right)^2}.
\end{aligned} \tag{4.11}$$

令 $\sum_{t=1}^{T} v_{i,j}^{(t)} = X_j$，于是等式4.11可以被重写为

$$\|\sum_{t=1}^{T} v_i^{(t)}\|_2 = \sqrt{\sum_{j=1}^{K} X_j^2}. \tag{4.12}$$

又因为 $v_{i,j}^{(t)} \sim \mathcal{N}(0,\sigma^2)$，我们有

$$X_j \sim \mathcal{N}\left(0, T\sigma^2\right). \tag{4.13}$$

于是有

$$\frac{X_j}{\sqrt{T}\sigma} \sim \mathcal{N}(0,1). \tag{4.14}$$

因此，我们可以导出

$$\sqrt{\sum_{j=1}^{K}\frac{X_j^2}{T\sigma^2}}=\frac{1}{\sqrt{T}\sigma}\sqrt{\sum_{j=1}^{K}X_j^2}\sim\mathcal{X}_K, \tag{4.15}$$

其中 $\mathcal{X}_K$ 表示自由度为 K 的卡分布 (Chi distribution)。根据 $\mathcal{X}_K$ 的性质，我们有

$$\begin{aligned}\mathbf{E}\left[\frac{1}{\sqrt{T}\sigma}\sqrt{\sum_{j=1}^{K}X_j^2}\right]&=\frac{1}{\sqrt{T}\sigma}\mathbf{E}\left[\sqrt{\sum_{j=1}^{K}X_j^2}\right]\\&=\sqrt{2}\cdot\frac{\Gamma\left((K+1)/2\right)}{\Gamma\left(K/2\right)}.\end{aligned} \tag{4.16}$$

由公式 4.12 和公式. 4.16，我们可以得到

$$\mathbf{E}\left[\|\sum_{t=1}^{T}v_i^{(t)}\|_2\right]=\sqrt{2T}\sigma\cdot\frac{\Gamma\left((K+1)/2\right)}{\Gamma\left(K/2\right)}. \tag{4.17}$$

因为 $\|\sum_{t=1}^{T}v_i^{(t)}\|_2\geq 0$，根据马尔可夫不等式 (Markov's inequality)，我们有

$$\begin{aligned}\Pr\left(\|\sum_{t=1}^{T}v_i^{(t)}\|_2\geq\tau\right)&\leq\frac{\mathbf{E}\left[\|\sum_{t=1}^{T}v_i^{(t)}\|_2\right]}{\tau}\\&=\sqrt{2T}\sigma\cdot\frac{\Gamma\left((K+1)/2\right)}{\tau\cdot\Gamma\left(K/2\right)}\\&=\sqrt{2T\cdot\mathbf{Var}\left(v^*\right)}\cdot\frac{\Gamma\left((K+1)/2\right)}{\tau\cdot\Gamma\left(K/2\right)}.\end{aligned} \tag{4.18}$$

取 $\tau=\sqrt{2T\cdot\mathbf{Var}\cdot(v^*)}\frac{\Gamma((K+1)/2)}{\xi\cdot\Gamma(K/2)}$，我们有

$$\begin{aligned}&\Pr\left[\|\sum_{t=1}^{T}v_i^{(t)}\|_2<\frac{\Gamma\left((K+1)/2\right)}{\xi\cdot\Gamma\left(K/2\right)}\sqrt{2T\cdot\mathbf{Var}\left(v^*\right)}\right]\\&>1-\xi.\end{aligned} \tag{4.19}$$

证明到此结束。 □

定理 4.3　假设我们使用高斯机制来满足 (ϵ,δ)-差分隐私。我们令 $WGDA_u$ 和 $WGDA_u^*$ 分别表示从用户 u 中未添加噪声的梯度 $\nabla\mathbf{Q}^{(u)}$ 和被噪声扰乱后的梯度 $\nabla\mathbf{Q}^{*,(u)}$ 中提取出来的特征值。令 T 表示全局迭代次数，$v^*\sim\mathcal{N}\left(0,\sigma^2\right)$ 表示添加在

梯度非零元素上面的高斯噪声。在概率至少为 $1-\xi$ 的情况下，$WGDA_u$ 和 $WGDA_u^*$ 之间的差异可以被限制在

$$O\left(\frac{\Gamma\left((K+1)/2\right)}{\xi\cdot\Gamma\left(K/2\right)}\cdot\sqrt{2T\cdot\mathbf{Var}\left(v^*\right)}\right). \tag{4.20}$$

证明

$$\begin{aligned}
&|WGDA_u^* - WGDA_u| \\
&= \left| \sum_{i\in\mathcal{M}_u} \frac{\left\|\nabla q_i^{(u)} - \nabla q_i\right\|_2}{n_i} - \sum_{i\in\mathcal{M}_u} \frac{\left\|\nabla q_i^{(u)} + \sum_{t=1}^{T} v_i^{(t)} - \nabla q_i\right\|_2}{n_i} \right| \\
&\leq \sum_{i\in\mathcal{M}_u} \frac{\left\|\sum_{t=1}^{T} v_i^{(t)}\right\|_2}{n_i} = O\left(\|\sum_{t=1}^{T} v_i^{(t)}\|_2\right).
\end{aligned} \tag{4.21}$$

证明的剩余部分与定理4.2一致。 □

由于函数 $Neigh(u,z)$ 定义的复杂性，我们很难直接获取 DegSim 特征的估计误差。不过，我们观察到梯度向量 $\mathbf{h}_u$ 是直接影响该特征大小的关键因素。因此，我们不直接对 DegSim 进行分析，而是分析梯度向量 $\mathbf{h}_u$ 的估计误差。

定理 4.4　假设我们使用高斯机制来满足 (ϵ,δ)-差分隐私。我们令 $\mathbf{h}_u$ 表示定义在公式3.10中的用户 u 的梯度向量，同时令 $\mathbf{h}_u^*$ 表示用户 u 被加噪后的梯度向量。令 T 表示全局迭代次数，$v^* \sim \mathcal{N}(0,\sigma^2)$ 表示添加在梯度非零元素上面的高斯噪声。在概率至少为 $1-\xi$ 的情况下，我们有

$$\|\mathbf{h}_u^* - \mathbf{h}_u\|_{\max} = O\left(\frac{K\sqrt{2T\cdot\mathbf{Var}\left(v^*\right)}}{\xi\sqrt{\pi}}\right). \tag{4.22}$$

证明 注意到

$$\begin{aligned}
\mathbf{h}_u &= \left[\left\|\nabla q_1^{(u)}\right\|_1, \left\|\nabla q_2^{(u)}\right\|_1, \ldots, \left\|\nabla q_M^{(u)}\right\|_1\right]^T \\
&= \left[\left|\sum_{j=1}^{K}\nabla q_{1,j}^{(u)}\right|, \left|\sum_{j=2}^{K}\nabla q_{1,j}^{(u)}\right|, \ldots, \left|\sum_{j=3}^{K}\nabla q_{1,j}^{(u)}\right|\right]^T,
\end{aligned} \tag{4.23}$$

和

$$\mathbf{h}_u^* = \left[\left\|\nabla q_1^{*,(u)}\right\|_1, \left\|\nabla q_2^{*,(u)}\right\|_1, \ldots, \left\|\nabla q_M^{*,(u)}\right\|_1\right]^T, \tag{4.24}$$

其中

$$\nabla q_i^{*,(u)} = \sum_{j=1}^{K} \left| \nabla q_{i,j}^{(u)} + \sum_{t=1}^{T} v_{i,j}^{(t)} \right|.$$

由三角不等式，我们有

$$\begin{aligned} &\mathbf{h}_u^* - \mathbf{h}_u \leq \\ &\left[\sum_{j=1}^{K}\left|\sum_{t=1}^{T} v_{1,j}^{(t)}\right|, \sum_{j=1}^{K}\left|\sum_{t=1}^{T} v_{2,j}^{(t)}\right|, \ldots, \sum_{j=1}^{K}\left|\sum_{t=1}^{T} v_{M,j}^{(t)}\right|\right]^T. \end{aligned} \tag{4.25}$$

因此，我们有

$$\|\mathbf{h}_u^* - \mathbf{h}_u\|_{\max} = O\left(\sum_{j=1}^{K}\left|\sum_{t=1}^{T} v^*\right|\right). \tag{4.26}$$

现在，我们来约束 $\sum_{j=1}^{K}\left|\sum_{t=1}^{T} v^*\right|$。令 X 表示 $\sum_{t=1}^{T} v^*$。因为 v^* 是高斯噪声，我们有

$$\sum_{t=1}^{T} v^* \sim \mathcal{N}\left(0, T\sigma^2\right). \tag{4.27}$$

因此，X 服从折叠正态分布 (Folded normal distribution)。根据折叠正态分布的性质，我们有

$$\begin{aligned} \mathbf{E}[X] &= \frac{\sqrt{2T}}{\sqrt{\pi}} \cdot \sigma \\ &= \frac{\sqrt{2T \cdot \mathbf{Var}\left(v^*\right)}}{\sqrt{\pi}}. \end{aligned} \tag{4.28}$$

根据马尔可夫不等式，我们有

$$\begin{aligned} \Pr\left[\sum_{j=1}^{K}\left|\sum_{t=1}^{T} v^*\right| \geq \tau\right] &\leq \frac{\mathbf{E}\left[\sum_{j=1}^{K}\left|\sum_{t=1}^{T} v^*\right|\right]}{\tau} \\ &= \frac{K\sqrt{2T \cdot \mathbf{Var}\left(v^*\right)}}{\tau\sqrt{\pi}}. \end{aligned} \tag{4.29}$$

取 $\tau = \frac{K\sqrt{2T\cdot\mathbf{Var}(v^*)}}{\xi\sqrt{\pi}}$，则 $\|\mathbf{h}_u^* - \mathbf{h}_u\|_{\max}$ 达到定理所给出的上界。 □

定理 4.5 假设我们使用高斯机制来满足 (ϵ, δ)-差分隐私。我们令 $FMTD_u$ 和 $FMTD_u^*$ 分别表示从用户 u 中未添加噪声的梯度 $\nabla\mathbf{Q}^{(u)}$ 和被噪声扰乱后的梯度 $\nabla\mathbf{Q}^{*,(u)}$ 中提取出来的 FMTD 特征值。令 T 表示全局迭代次数，$v^* \sim \mathcal{N}(0, \sigma^2)$ 表示添加到梯度矩阵非零元素上面的高斯噪声。在概率至少为 $1-\xi$ 的情况下，$FMTD_u$ 和 $FMTD_u^*$ 之间的差异可以被限制在

$$O\left(\sqrt{2\left(\frac{1}{|\mathcal{M}_{u,T}|} - \frac{1}{|\mathcal{M}_{u,F}|}\right)T\cdot\mathbf{Var}\left(v^*\right)}\cdot\frac{\Gamma\left(\frac{K+1}{2}\right)}{\xi\cdot\Gamma\left(\frac{K}{2}\right)}\right). \tag{4.30}$$

证明 首先，根据三角不等式，我们有

$$\begin{aligned}
&|FMTD_u - FMTD_u^*| \\
&= \left|\left\|\frac{\sum_{i\in\mathcal{M}_{u,T}}\nabla q_i^{(u)}}{|\mathcal{M}_{u,T}|} - \frac{\sum_{i'\in\mathcal{M}_{u,F}}\nabla q_{i'}^{(u)}}{|\mathcal{M}_{u,F}|}\right\|_2\right. \\
&\quad \left. - \left\|\frac{\sum_{i\in\mathcal{M}_{u,T}}\left(\nabla q_i^{(u)} + \sum_{t=1}^{T} v_i^{(t)}\right)}{|\mathcal{M}_{u,T}|} - \frac{\sum_{i'\in\mathcal{M}_{u,F}}\left(\nabla q_{i'}^{(u)} + \sum_{t=1}^{T} v_{i'}^{(t)}\right)}{|\mathcal{M}_{u,F}|}\right\|_2\right| \qquad (4.31) \\
&\le \left\|\frac{\sum_{i\in\mathcal{M}_{u,T}}\left(\sum_{t=1}^{T} v_i^{(t)}\right)}{|\mathcal{M}_{u,T}|} - \frac{\sum_{i'\in\mathcal{M}_{u,F}}\left(\sum_{t=1}^{T} v_{i'}^{(t)}\right)}{|\mathcal{M}_{u,F}|}\right\|_2. \qquad (4.32)
\end{aligned}$$

我们知道噪声向量 $v_i^{(t)}$ 是服从高斯分布 $v_i^{(t)} \sim \mathcal{N}\left(0, \sigma^2\mathbb{I}^K\right)$ 且独立生成的。因此，我们有

$$\begin{aligned}
&\frac{\sum_{i\in\mathcal{M}_{u,T}}\left(\sum_{t=1}^{T} v_i^{(t)}\right)}{|\mathcal{M}_{u,T}|} - \frac{\sum_{i'\in\mathcal{M}_{u,F}}\left(\sum_{t=1}^{T} v_{i'}^{(t)}\right)}{|\mathcal{M}_{u,F}|} \\
&\quad \triangleq \mathcal{N}\left(0, \left(\frac{1}{|\mathcal{M}_{u,T}|} - \frac{1}{|\mathcal{M}_{u,F}|}\right)T\sigma^2\mathbb{I}^K\right). \qquad (4.33)
\end{aligned}$$

令 $X \sim \mathcal{N}\left(0, \left(\frac{1}{|\mathcal{M}_{u,T}|} - \frac{1}{|\mathcal{M}_{u,F}|}\right)T\sigma^2\mathbb{I}^K\right)$，等式4.33可以被重写为

$$|FMTD_u - FMTD_u^*| \le \|X\|_2.$$

于是，与定理4.2的证明类似，我们有

$$\begin{aligned}&\mathbf{E}\left[\|X\|_2\right]\\&=\sqrt{2\left(\frac{1}{|\mathcal{M}_{u,T}|}-\frac{1}{|\mathcal{M}_{u,F}|}\right)T\cdot\mathbf{Var}\left(v^*\right)}\cdot\frac{\Gamma\left(\frac{K+1}{2}\right)}{\Gamma\left(\frac{K}{2}\right)}.\end{aligned}\tag{4.34}$$

根据马尔可夫不等式，我们有

$$\begin{aligned}\Pr\left[\|X\|_2>\tau\right]&\leq\frac{\mathbf{E}\left[\|X\|_2\right]}{\tau}\\&=\sqrt{2\left(\frac{1}{|\mathcal{M}_{u,T}|}-\frac{1}{|\mathcal{M}_{u,F}|}\right)T\cdot\mathbf{Var}\left(v^*\right)}\cdot\frac{\Gamma\left(\frac{K+1}{2}\right)}{\tau\cdot\Gamma\left(\frac{K}{2}\right)}.\end{aligned}\tag{4.35}$$

取 $\tau=\sqrt{2\left(\frac{1}{|\mathcal{M}_{u,T}|}-\frac{1}{|\mathcal{M}_{u,F}|}\right)T\cdot\mathbf{Var}\left(v^*\right)}\cdot\frac{\Gamma\left(\frac{K+1}{2}\right)}{\xi\cdot\Gamma\left(\frac{K}{2}\right)}$，使得 $|FMTD_u-TMTD_u^*|$ 达到定理中描述的上界。 □

从上述定理中，我们观察到项目矩阵和特征的估计误差在忽略一些常数项后，都在同一个数量级 $O\left(\sqrt{T\cdot\mathbf{Var}(v^*)}\right)$ 下。因此，我们可以得出，只要项目矩阵的准确性（可以近似看作推荐模型的准确性）没有完全被差分隐私噪声破坏，那么 FSAD 就是有效的。同时，如果差分隐私噪声的方差太大以至于让检测器的准确性无法被接收，那么这也意味着推荐模型的性能也已经完全被噪声破坏了。在这种情况下，追求高准确性的托攻击检测器已经没有意义，因为差分隐私噪声对推荐系统造成的破坏性已经超过攻击者带来的破坏性。

4.3 本章小结

在这一章节，我们对联邦托攻击检测问题进行了扩展。我们首先将差分隐私机制引入 Fed-CF 中以增强用户隐私保护的程度。随后，我们理论分析了高斯机制下的差分隐私噪声对推荐系统模型性能以及托攻击检测特征的影响。理论结果表明，差分隐私噪声对两者的影响在同一个数量级上。

五、实验评估

在本章节中，我们进行了大量的实验来评估 FSAD 的有效性。我们首先评估了 FSAD 应用于算法3.1所描述的基础版 Fed-CF 的效果。然后我们使用算法4.1所描述的差分隐私 Fed-CF 算法，测试在采用不同隐私预算场景下的托攻击检测效果。

5.1 实验设置

数据集. 我们基于 MovieLens 和 Netflix 两个真实世界数据集来实验评估。

MovieLens 数据集[46] 已在教育、研究和工业等领域被广泛应用。在本文中，我们使用的 MovieLens 数据集由 3,952 部电影和 6,040 个用户构成的 1,000,209 条匿名评分组成。Netflix 数据集[47] 是网飞公司在 2006 年的公开竞赛中发布的数据集，随后这个数据集被广泛应用于各种推荐系统的研究。在本文的实验中，我们去除了 Netflix 数据集中的冷门视频与不活跃的用户。我们随机选取了 2,000 部被评分次数超过 200 的电影，以及 5,000 名至少为 50 部电影进行过评分的用户①。

超参数. 我们在两个数据集上分别运行了 Fed-CF 算法。推荐模型共有四个超参数：本地学习率 β，全局学习率 α，隐空间维度 K 以及正则化参数 η。对于两个数据集，我们对超参数的设置一致：$\beta = 10^{-4}$，$\alpha = 0.1$，$K = 4$，$\eta = 10^{-5}$。在联邦学习的每轮全局迭代中，用户执行五次本地更新。全局迭代的总数为 30。此外，用于托攻击检测的半监督朴素贝叶斯模型中的权重也是一个超参数。根据现有工作[19]，$\lambda = 0.5$ 是一个相对稳定的数值，因此在本实验中我们也设置 $\lambda = 0.5$。

攻击策略. 对于两个数据集，我们都注入了占真实用户数量 5% 的恶意用户来发动托攻击。在本文中，我们假设评分数据的分布信息可以作为先验知识被系统和攻击者获得。我们使用盲目攻击、随机攻击以及均值攻击这三种攻击模型来生成托攻击恶意用户。每一种攻击模型生成的恶意用户数量是相等的。对于攻击者而言，填充项目的大小会影响被检测的可能性。在本文实验中，我们采用了从占总项目数 5% 到 30% 的填充项目数以探索填充项目数对检测器性能的影响。

标签用户. 为了以半监督的方式运行 FSAD，我们需要手动生成带标签的模拟用户，包括正常用户与恶意用户，将这部分用户产生的梯度信息作为带标签的数据集。特别地，模拟的托攻击者也采用盲目攻击、随即攻击以及均值攻击这三种攻击模型来生成恶意用户。模拟攻击者所攻击的项目从前 10% 热门的项目中随机

① 这么做是合理的，因为攻击者通常不会去攻击不流行的项目（电影）。

挑选产生。填充项目的数量设置为 m_u 的均值，其中 m_u 为用户 u 进行过的评分数量。在实验中，生成带标签用户的过程通过脚本自动进行。我们使用半监督分类器，所以并不需要额外生成大量的带标签数据，因此生成标签数据的性能开销并不高。

基准算法. 我们将本文设计的 FSAD 的性能与 HySAD[19] 进行比较。HySAD 是一个非常有效的托攻击检测器，但是它需要获取用户的原始评分数据。我们在假设评分数据可知的条件下运行 HySAD，因此这并不是一个公平比较。我们将 HySAD 的检测效果作为本文提出的 FSAD 的性能上界。

评估指标. 在实验中，我们使用三个标准的评估指标：准确率 ($Precision$)，召回率 ($Recall$) 以及 F_1，它们的定义如下：

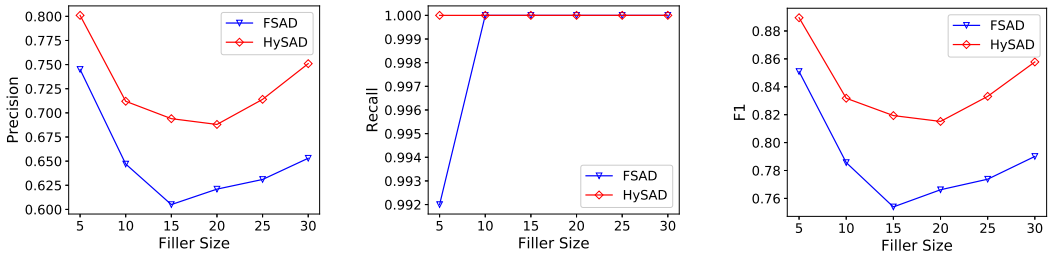
$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN}, \quad F_1 = \frac{2PR}{P + R}, \tag{5.1}$$

其中 TP 是真正识别出的攻击者数量，FP 是被错误识别为攻击者的正常用户数量，FN 是被错误识别为正常用户的攻击者数量。

5.2 实验结果

5.2.1 攻击检测性能

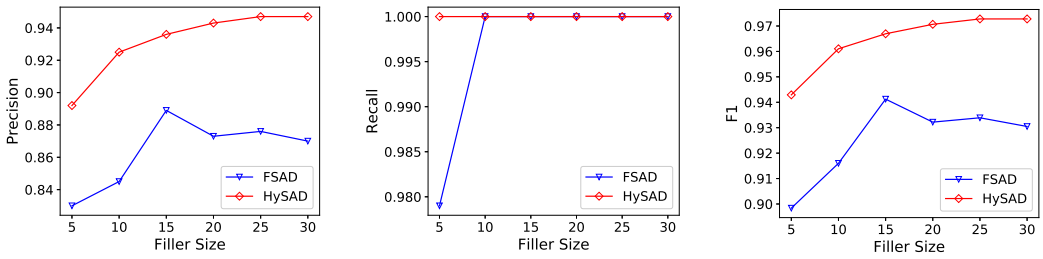
FSAD 和 HySAD 在 MovieLens 和 Netflix 数据集上的检测性能（使用准确率 P，召回率 R 以及 F_1 值进行评估）如图5-1和图5-2所示。对于每一个数据集实验，我们都模拟生成了 600 个带标签的用户，并且分别用 5% 的步长改变攻击者的填充数量大小，从 5% 到 30%。在 600 个模拟用户中，有 300 个正常用户，另外根据三种托攻击模型各自分别生成 100 个模拟恶意用户。



(a) 准确率 @MovieLens (b) 召回率 @MovieLens (c) F_1@MovieLens

图 5-1 基于 MovieLens 数据集的实验结果。两种检测器的准确率、召回率以及 F_1 分别展示在三个子图中。x 轴代表攻击者的填充数量（攻击者在系统中所评分的项目百分比）；y 轴代表评估指标

从图5-1和图5-2展示的实验结果中，我们可以观察到两种检测算法在两个数据



(a) 准确率 @Netflix (b) 召回率 @Netflix (c) F_1@Netflix

图 5-2 基于 Netflix 数据集的实验结果。两种检测器的准确率、召回率以及 F_1 分别展示在三个子图中。x 轴代表攻击者的填充数量（攻击者在系统中所评分的项目百分比）；y 轴代表评估指标

集上的召回率几乎相同，几乎都是 100%，除了当攻击者的填充大小只有 5% 时略小一些。这意味着 FSAD 和 HySAD 都可以捕获系统中的几乎所有托攻击者。然而，两种检测器的准确率都比召回率低。这意味着总有一些真实用户，他们有着与其他用户截然不同的特殊行为模式，导致他们的评分特征与托攻击恶意用户非常相似，我们很难将他们与真正的托攻击者区分开来。FSAD 的准确率略低于 HySAD，原因是 HySAD 可以获得更全面的信息（用户-项目评分矩阵）。总的来说，HySAD 的 F_1 比 FSAD 的 F_1 高约 5%。

在图5-3中，我们通过实验来评估标签用户数量对检测性能的影响。正如我们在章节三中讨论的，有标签用户对于 FSAD 分类器的半监督训练是必不可少的。我们将模拟标签用户的数量从 50 提高到 600，以观察检测性能的变化。如图5-3所示，随着两个数据集上标签用户数量的增加，托攻击检测性能得到了显著的提高。这表明我们需要生成足够的标签数据使 FSAD 能够达到相对理想的检测性能。

5.2.2 推荐性能提升效果

在本文实验中，我们通过在联邦推荐系统中部署 FSAD 来测试推荐性能的提升效果。我们采用均方误差 (Root Mean Squared Error, RMSE) 来评价推荐性能[5]，相关定义如下：

$$RMSE = \sqrt{\sum_{(u,i)\in\mathcal{D}_{test}} \frac{(r_{ui} - \hat{r}_{ui})^2}{|\mathcal{D}_{test}|}} \tag{5.2}$$

其中 $\mathcal{D}_{test}$ 为测试数据集，r_{ui} 是用户 u 对项目 i 真实的评分数据，$\hat{r}_{ui}$ 是联邦推荐模型给出的预测值。在移除潜在攻击者后，我们随机选取占目标项目 10% 的评分记录作为测试数据集。

我们使用 Netflix 数据集训练 Fed-CF 模型。全局迭代轮次为 30，测试数据集大小为 241，注入 250 个攻击者，攻击者填充大小从 5% 提高到到 30%（步长为

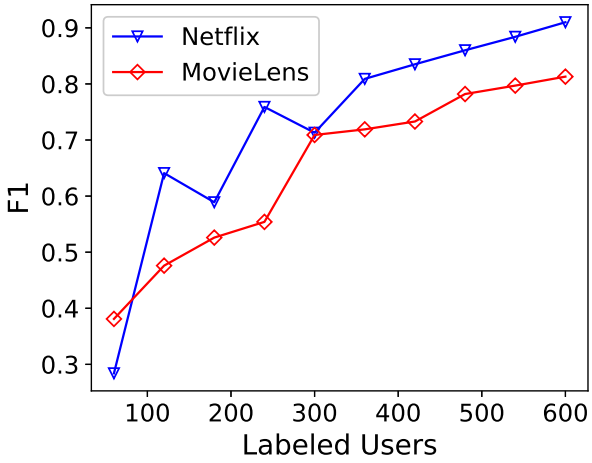


图 5-3 标签用户的数量从 50 改变到 600，比较不同数据集下 FSAD 的 F_1

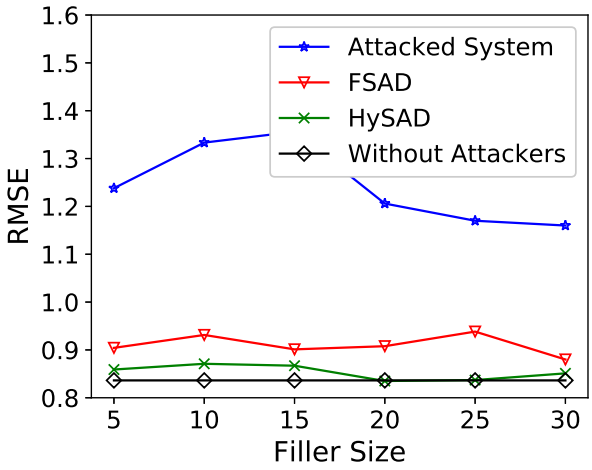
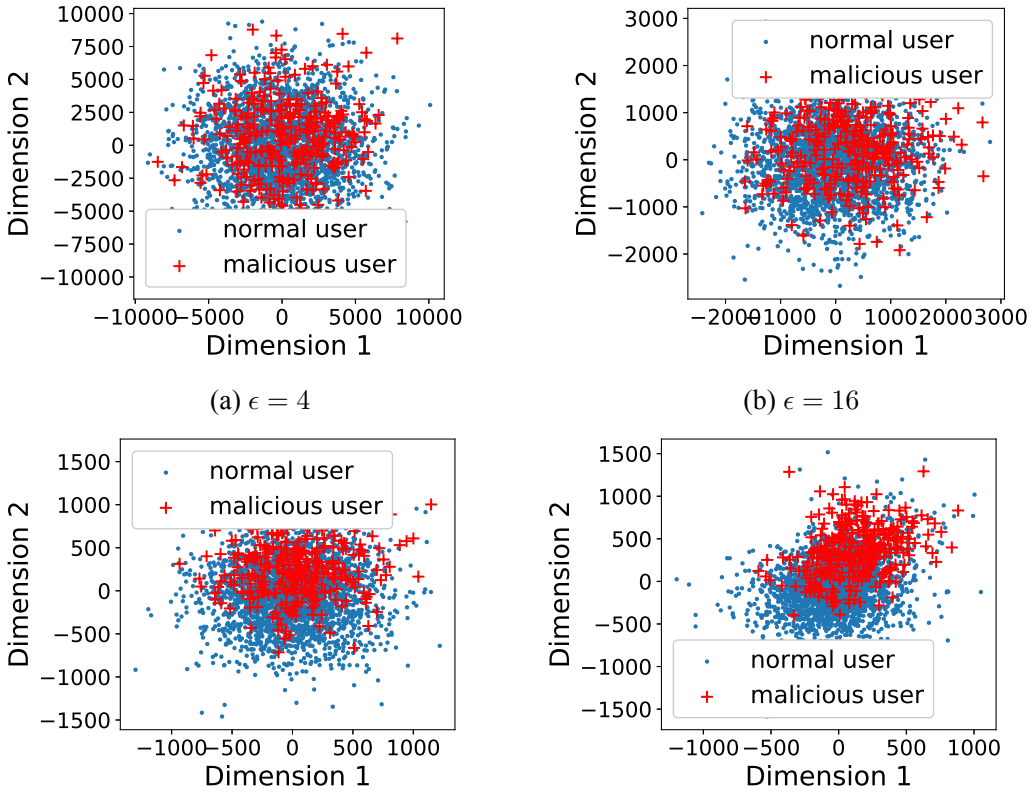


图 5-4 使用 Netflix 数据集比较部署 FSAD 前后的测试集 RMSE。攻击者生成的恶意用户数量为 250。填充项目数量从 5% 改变到 30%

5%)，来评估 FSAD 的鲁棒性。在经过 30 轮全局迭代后，联邦学习系统停止训练。然后分别使用 FSAD 和 HySAD 检测攻击者。检测器识别出的潜在攻击者将从原始数据集中删除，然后再次执行 Fed-CF 算法。在没有这部分潜在的攻击者的情况下，我们预计 Fed-CF 将取得更好的表现。为了证明 FSAD 通过移除托攻击者来提高推荐性能的效果，我们绘制了遭受攻击的 Fed-CF 的 RMSE 曲线，其中包含 5000 名真实用户和 250 名攻击者。此外，图中还分别绘制了有 FSAD 和 HySAD 时的 Fed-CF，以及没有攻击者时的 Fed-CF 各自对应的 RMSE 曲线。从图中我们可以明显地发现，部署了 FSAD 后推荐性能出现显著提升，RMSE 从没有检测器时的 1.3 降低到的 0.9。部署的 FSAD 后的 RMSE 与理想情况相比只相差大约 0.05，表明了 FSAD 的有效性。

5.3 差分隐私噪声的影响

为了研究差分隐私噪声对推荐系统和 FSAD 的影响，我们将 FSAD 应用于基于差分隐私的 Fed-CF 算法，并进行性能评估。我们选用的高斯噪声的标准差为 $\sigma = \frac{\sqrt{\ln(1.25/\delta)}\Delta_2 f}{\epsilon}$。注意到在高斯机制中，只有当 $\epsilon \in (0,1)$ 满足时，才可以保证达到 (ϵ,δ)-差分隐私。在本实验中，我们的目的是研究差分隐私噪声方差与 Fed-CF 和 FSAD 性能之间的关系。因此 ϵ 的值主要用来表示差分隐私噪声的大小，而不是隐私保护水平。由于梯度的裁剪和高斯噪声会影响 Fed-CF 的收敛速度，我们重置了一些超参数以达到理想的收敛速度。在差分隐私 Fed-CF 中，我们设置 $\alpha = 0.3, K = 5, \eta = 10^{-8}$。



(a) $\epsilon = 4$　(b) $\epsilon = 16$

(c) $\epsilon = 36$　(d) $\epsilon = 64$

图 5-5 不同高斯噪声下的梯度分布。梯度是通过在拥有 6,040 个正常用户和 302 个攻击者的 MovieLens 数据集上训练引入差分隐私机制的 FedCF 模型得到的

为了直观地显示差分隐私噪声对梯度分布的影响，我们将被攻击项目的梯度分布绘制在图5-5中。值得注意的是，当差分隐私噪声的方差非常大时，攻击者和正常用户的梯度分布都是相对均匀的。相反，当差分隐私噪声的方差较小时，它们会形成两个独立的簇。这个实验现象解释了为什么差分隐私噪声使得攻击者的检测更加困难。另外，如果差分隐私噪声太大，梯度信息将被完全扭曲。此时不仅 FSAD 的性能会被严重影响，推荐模型的准确性也会大幅度降低。

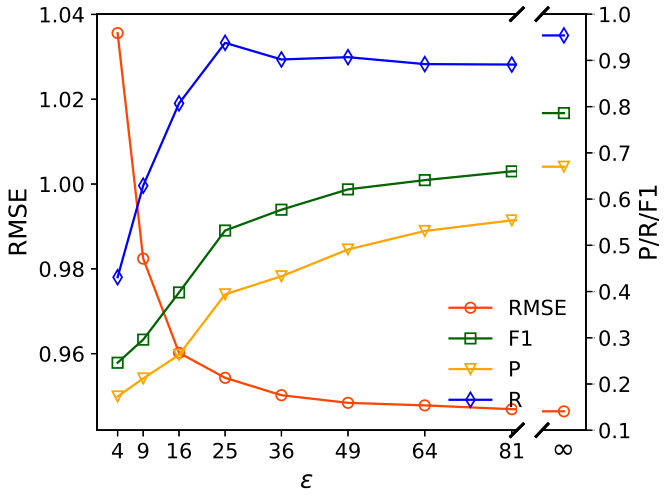


图 5-6 基于 MovieLens 数据集的差分隐私 Fed-CF 的 RMSE 和攻击检测性能。我们训练差分隐私 Fed-CF 模型 10 次，计算 RMSE 和攻击检测性能的平均值

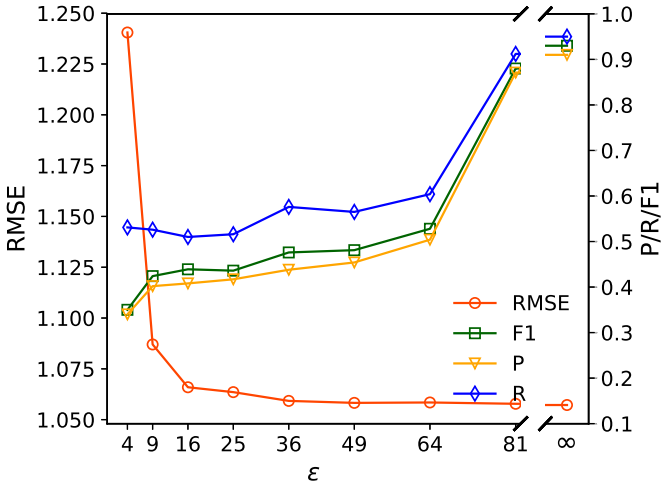


图 5-7 基于 Netflix 数据集的差分隐私 Fed-CF 的 RMSE 和攻击检测性能。我们训练差分隐私 Fed-CF 模型 10 次，计算 RMSE 和攻击检测性能的平均值

图5-6和图5-7展示了差分隐私 Fed-CF 在不同噪声下的推荐性能和检测性能。对于每个给定的 ϵ，我们训练差分隐私 Fed-CF 模型 10 次，并计算 RMSE 和攻击检测性能的平均值。图5-6和图5-7中的实验结果验证了我们的理论分析: 对于我们采用的两个数据集，随着差分隐私噪声方差的减小，推荐系统的均方误差在不断减小，同时攻击检测性能随之提高。图 5-7中 Netflix 数据集的实验曲线有较小的抖动，这可能是由于 Netflix 数据集比 MovieLens 数据集稀疏造成的。

5.4 本章小结

在本章节，我们基于 MovieLens 和 Netflix 两个真实世界数据集组织了大量的实验。我们首先考虑没有引入差分隐私机制的 Fed-CF 模型，通过注入恶意用户模

拟托攻击。实验表明，FSAD 在该场景下可以有效检测出系统中的托攻击恶意用户。在去除 FSAD 检测出的疑似恶意用户后，推荐系统的性能得到显著提升。随后我们在引入差分隐私机制后，在相同的数据集上训练了 Fed-CF 模型，提取梯度并使用 FSAD 进行攻击检测。实验结果与章节四中的理论分析结果相符合。

六、总结与展望

6.1 论文工作总结

在本文的工作中，我们证明了托攻击者可以很容易地攻击基于联邦学习的推荐系统。然而，现有的所有检测算法都依赖于原始训练样本这样的敏感信息，因此不能直接被应用于联邦推荐系统。在联邦学习场景下如何抵御托攻击仍然是一个开放式的问题。因此，我们设计 FSAD 来检测托攻击者。我们从参数服务器唯一可用的梯度信息中设计四个新的特征。然后基于这些特征值训练一个半监督贝叶斯分类器来区分正常用户与恶意用户。我们还证明了在使用差分隐私机制来保护用户梯度信息的场景下，FSAD 仍是有效的。最后，我们在 MovieLens 和 Netflix 数据集上进行了大量的实验。实验结果表明，FSAD 能够有效并且准确地检测托攻击者。FSAD 的托攻击检测性能仅略低于基于用户-物品评分矩阵信息的传统检测算法。此外，我们的实验表明，只要差分隐私噪声不完全破坏推荐系统性能，FSAD 仍是有效的。

6.2 未来研究工作设想

我们的实验结果表明，当差分隐私噪声相对较小时，对系统的推荐性能影响较小，但是对托攻击检测的影响仍然比较明显。这是由于全局模型由所有用户聚合而成，噪声方差在平均化过程中会大幅度减小；而特征从每一个用户的加噪梯度中直接提取获得，因此噪声水平相对较大。在差分隐私机制下，联邦托攻击检测器的准确性仍有较大的提升空间。设计更加高效的特征和开发更健壮的托攻击检测器是我们未来的工作方向。

参考文献

[1] KONEČNỲ J, MCMAHAN H B, RAMAGE D, et al. Federated optimization: Distributed machine learning for on-device intelligence[J]. arXiv preprint arXiv:1610.02527, 2016.

[2] MCMAHAN B, MOORE E, RAMAGE D, et al. Communication-Efficient Learning of Deep Networks from Decentralized Data[C] // Artificial Intelligence and Statistics. 2017 : 1273 – 1282.

[3] YANG Q, LIU Y, CHEN T, et al. Federated machine learning: Concept and applications[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2019, 10(2) : 1 – 19.

[4] KAIROUZ P, MCMAHAN H B, AVENT B, et al. Advances and open problems in federated learning[J]. arXiv preprint arXiv:1912.04977, 2019.

[5] RICCI F, ROKACH L, SHAPIRA B, et al. Recommender Systems Handbook[M]. 1st. Berlin, Heidelberg : Springer-Verlag, 2010.

[6] LU J, WU D, MAO M, et al. Recommender system application developments: a survey[J]. Decision Support Systems, 2015, 74 : 12 – 32.

[7] HUA J, XIA C, ZHONG S. Differentially Private Matrix Factorization[C] // IJCAI'15 : Proceedings of the 24th International Conference on Artificial Intelligence. [S.l.] : AAAI Press, 2015 : 1763–1770.

[8] SHEN Y, JIN H. Privacy-preserving personalized recommendation: An instance-based approach via differential privacy[C] // 2014 IEEE International Conference on Data Mining. 2014 : 540 – 549.

[9] SHIN H, KIM S, SHIN J, et al. Privacy enhanced matrix factorization for recommendation with local differential privacy[J]. IEEE Transactions on Knowledge and Data Engineering, 2018, 30(9) : 1770 – 1782.

[10] Ammad-ud DIN M, IVANNIKOVA E, KHAN S A, et al. Federated Collaborative Filtering for Privacy-Preserving Personalized Recommendation System[J]. arXiv preprint arXiv:1901.09888, 2019.

[11] FUNG C, YOON C J, BESCHASTNIKH I. Mitigating sybils in federated learning poisoning[J]. arXiv preprint arXiv:1808.04866, 2018.

[12] BIGGIO B, NELSON B, LASKOV P. Poisoning attacks against support vector machines[J]. arXiv preprint arXiv:1206.6389, 2012.

[13] BAGDASARYAN E, VEIT A, HUA Y, et al. How to backdoor federated learning[J]. arXiv preprint arXiv:1807.00459, 2018.

[14] LAM S K, RIEDL J. Shilling recommender systems for fun and profit[C] // Proceedings of the 13th international conference on World Wide Web. 2004 : 393 – 402.

[15] GUNES I, KALELI C, BILGE A, et al. Shilling attacks against recommender systems: a comprehensive survey[J]. Artificial Intelligence Review, 2014, 42(4) : 767 – 799.

[16] RAY S, MAHANTI A. Strategies for effective shilling attacks against recommender systems[C] // International Workshop on Privacy, Security, and Trust in KDD. 2008 : 111 – 125.

[17] BURKE R, MOBASHER B, WILLIAMS C, et al. Classification features for attack detection in collaborative recommender systems[C] // Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. 2006 : 542 – 547.

[18] CHIRITA P-A, NEJDL W, ZAMFIR C. Preventing shilling attacks in online recommender systems[C] // Proceedings of the 7th annual ACM international workshop on Web information and data management. 2005 : 67 – 74.

[19] WU Z, WU J, CAO J, et al. HySAD: A semi-supervised hybrid shilling attack detector for trustworthy product recommendation[C] // Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. 2012 : 985 – 993.

[20] MEHTA B. Unsupervised shilling detection for collaborative filtering[C] // AAAI. 2007 : 1402 – 1407.

[21] AKTUKMAK M, YILMAZ Y, UYSAL I. Quick and accurate attack detection in recommender systems through user attributes[C] // Proceedings of the 13th ACM Conference on Recommender Systems. 2019 : 348 – 352.

[22] ZHU L, LIU Z, HAN S. Deep leakage from gradients[C] // Advances in Neural Information Processing Systems. 2019 : 14774 – 14784.

[23] WANG Z, SONG M, ZHANG Z, et al. Beyond inferring class representatives: User-level privacy leakage from federated learning[C] // IEEE INFOCOM 2019-IEEE Conference on Computer Communications. 2019 : 2512 – 2520.

[24] GEIPING J, BAUERMEISTER H, DRÖGE H, et al. Inverting Gradients–How easy is it to break privacy in federated learning?[J]. arXiv preprint arXiv:2003.14053, 2020.

[25] CHAI D, WANG L, CHEN K, et al. Secure federated matrix factorization[J]. IEEE Intelligent Systems, 2020.

[26] DWORK C, MCSHERRY F, NISSIM K, et al. Calibrating noise to sensitivity in private data analysis[C] // Theory of cryptography conference. 2006 : 265 – 284.

[27] DWORK C, ROTH A, OTHERS. The algorithmic foundations of differential privacy[J]. Foundations and Trends® in Theoretical Computer Science, 2014, 9(3–4) : 211 – 407.

[28] ABADI M, CHU A, GOODFELLOW I, et al. Deep learning with differential privacy[C] // Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. 2016 : 308 – 318.

[29] GEYER R C, KLEIN T, NABI M. Differentially private federated learning: A client level perspective[J]. arXiv preprint arXiv:1712.07557, 2017.

[30] WU N, FAROKHI F, SMITH D, et al. The value of collaboration in convex machine learning with differential privacy[C] // 2020 IEEE Symposium on Security and Privacy (SP). 2020 : 304 – 317.

[31] WEI K, LI J, DING M, et al. Federated learning with differential privacy: Algorithms and performance analysis[J]. IEEE Transactions on Information Forensics and Security, 2020.

[32] Jiang Y, Zhou Y, Wu D, et al. On the Detection of Shilling Attacks in Federated Collaborative Filtering[C] // 2020 International Symposium on Reliable Distributed Systems (SRDS). 2020 : 185 – 194.

[33] GOMEZ-URIBE C A, HUNT N. The netflix recommender system: Algorithms, business value, and innovation[J]. ACM Transactions on Management Information Systems (TMIS), 2015, 6(4) : 1 – 19.

[34] SCHAFER J B, FRANKOWSKI D, HERLOCKER J, et al. Collaborative filtering recommender systems[G] // The adaptive web. [S.l.] : Springer, 2007 : 291 – 324.

[35] HAN P, XIE B, YANG F, et al. A scalable P2P recommender system based on distributed collaborative filtering[J]. Expert systems with applications, 2004, 27(2) : 203 – 210.

[36] ZHEN L, JIANG Z, SONG H. Distributed recommender for peer-to-peer knowledge sharing[J]. Information Sciences, 2010, 180(18) : 3546 – 3561.

[37] JALALIRAD A, SCAVUZZO M, CAPOTA C, et al. A Simple and Efficient Federated Recommender System[C] // Proceedings of the 6th IEEE/ACM International Conference on Big Data Computing, Applications and Technologies. 2019 : 53 – 58.

[38] FLANAGAN A, OYOMNO W, GRIGORIEVSKIY A, et al. Federated Multi-view Matrix Factorization for Personalized Recommendations[J]. arXiv preprint arXiv:2004.04256, 2020.

[39] O'MAHONY M, HURLEY N, KUSHMERICK N, et al. Collaborative recommendation: A robustness analysis[J]. ACM Transactions on Internet Technology (TOIT), 2004, 4(4) : 344 – 377.

[40] WILLIAMS C, MOBASHER B. Profile injection attack detection for securing collaborative recommender systems[J]. DePaul University CTI Technical Report, 2006 : 1 – 47.

[41] BURKE R, MOBASHER B, BHAUMIK R. Limited knowledge shilling attacks in collaborative filtering systems[C] // Proceedings of 3rd International Workshop on Intelligent Techniques for Web Personalization (ITWP 2005), 19th International Joint Conference on Artificial Intelligence (IJCAI 2005). 2005 : 17 – 24.

[42] BURKE R, MOBASHER B, ZABICKI R, et al. Identifying attack models for secure recommendation[J]. Beyond Personalization, 2005, 2005.

[43] HURLEY N, CHENG Z, ZHANG M. Statistical attack detection[C] // Proceedings of the third ACM conference on Recommender systems. 2009 : 149 – 156.

[44] ERLINGSSON Ú, PIHUR V, KOROLOVA A. Rappor: Randomized aggregatable privacy-preserving ordinal response[C] // Proceedings of the 2014 ACM SIGSAC conference on computer and communications security. 2014 : 1054 – 1067.

[45] NIGAM K, MCCALLUM A K, THRUN S, et al. Text classification from labeled and unlabeled documents using EM[J]. Machine learning, 2000, 39(2-3) : 103 – 134.

[46] HARPER F M, KONSTAN J A. The movielens datasets: History and context[J]. Acm transactions on interactive intelligent systems (tiis), 2015, 5(4) : 1 – 19.

[47] BENNETT J, LANNING S, OTHERS. The netflix prize[C] // Proceedings of KDD cup and workshop : Vol 2007. 2007 : 35.

致谢

首先我想对我的指导老师吴迪教授表示感谢。作为我的本科毕设导师以及逸仙班科研导师，吴老师这几年来在我的学业、学术以及个人发展的选择上给予了大量宝贵的建议与帮助。在吴老师的带领和指导下，我作为本科生参与了多项科研项目并发表学术论文，这坚定了我选择走学术道路的决心。在我申请博士的过程中，吴老师同样给予了大力的帮助与支持。吴老师是我在科研学术道路上的启蒙导师，同时也在我的职业规划、个人发展等方面提供了慷慨的指导，是人生路上的良师益友，在此向吴老师致以最诚挚的感谢。

在本科四年的学习中，我非常幸运能够与多位优秀的学者和志同道合的伙伴们进行合作。我要感谢澳洲麦考瑞大学的周义朋博士。周老师作为科研项目上的指导者与合作者，对我的科研工作提供了大量细节上的指导与建议。从周老师身上，我学会了如何对具体的研究问题进行把握和分析。我还要感谢符尧和钟志聪同学，作为同一课题组的本科生，我们合作参与了一些非常有意思的工作，这些合作项目令我受益匪浅。

最后我要感谢我的父母及家人，他们对我的支持与鼓励是我前进的最大动力。

姜洋帆

2021 年 5 月 16 日