计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
2025,34(9):46−56 [doi: 10.15888/j.cnki.csa.009926] [CSTR: 32024.14.csa.009926]
©中国科学院软件研究所版权所有.
E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

P-Tracing: 基于可控数据增强和多任务学习的可溯源模型指纹方法①

陈先意[1,2], 徐静雯[2], 张　欢[3], 张圣林[4], 刘庆程[1]

[1](南京信息工程大学 数字取证教育部工程研究中心, 南京 210044)
[2](南京信息工程大学 计算机学院、网络空间安全学院, 南京 210044)
[3](南京市专利行政执法支队, 南京 210008)
[4](南京信息工程大学雷丁学院, 南京 210044)
通信作者: 徐静雯, E-mail: 3466183218@qq.com

摘　要: 目前, 人工智能模型版权保护研究主要集中在水印技术的鲁棒性和模型精度提升, 而对于模型溯源追踪的研究仍然较少. 针对多用户模式下盗版难于追踪的问题, 提出了一个支持盗版溯源的模型指纹框架. 该框架为每个用户随机抽取唯一的可控数据增强组合, 基于此生成各用户的溯源指纹集. 接着结合多任务学习技术, 利用指纹集同时再训练源模型和接收源模型输出的溯源模型. 再训练完成后, 嵌入独特指纹集特征的源模型转化为用户专有模型, 溯源模型则能根据指纹集在对应专有模型上的预测结果推断出对应用户, 实现溯源有效性和专有模型性能之间的平衡. 在 CIFAR10 和 Fashion-MNIST 数据集上的实验结果表明, 所提出的方法在盗版检测和溯源追踪任务上均达到了 90% 以上的准确度, 验证了其在不同任务场景下的有效性.

关键词: 模型保护; 模型指纹; 多任务学习; 盗版溯源

引用格式: 陈先意,徐静雯,张欢,张圣林,刘庆程.P-Tracing: 基于可控数据增强和多任务学习的可溯源模型指纹方法.计算机系统应用,2025,34(9): 46–56. http://www.c-s-a.org.cn/1003-3254/9926.html

P-Tracing: Fingerprinting Method for Traceable Model Based on Controlled Data Augmentation and Multi-task Learning

CHEN Xian-Yi[1,2], XU Jing-Wen[2], ZHANG Huan[3], ZHANG Sheng-Lin[4], LIU Qing-Cheng[1]

[1](Engineering Research Center of Digital Forensics Ministry of Education, Nanjing University of Information Science & Technology, Nanjing 210044, China)
[2](School of Computer Science & School of Cyber Science and Engineering, Nanjing University of Information Science & Technology, Nanjing 210044, China)
[3](Nanjing Patent Administrative Enforcement Detachment, Nanjing 210008, China)
[4](NUIST Reading Academy, Nanjing 210044, China)

Abstract: Currently, research on artificial intelligence model copyright protection primarily focuses on the robustness of watermarking techniques and the enhancement of model accuracy, while model traceability remains relatively under explored. To address the challenge of tracking piracy in multi-user environments, a model fingerprinting framework enabling piracy traceability is proposed. In this framework, a unique and controllable data augmentation combination is randomly assigned to each user, based on which a traceable fingerprint set is generated. Multi-task learning techniques are then employed to simultaneously retrain the source model and the traceability model, which receives outputs from the source model, using the fingerprint sets. After retraining, the source model is transformed into a user-specific model with

① 基金项目: 国家重点研发计划 (2021YFB2700900); 国家自然科学基金 (62172232, 62172233); 江苏省杰出青年基金 (BK20200039)
收稿时间: 2025-01-10; 修改时间: 2025-01-26; 采用时间: 2025-02-24; csa 在线出版时间: 2025-07-14
CNKI 网络首发时间: 2025-07-15

© 中国科学院软件研究所 http://www.c-s-a.org.cn

embedded features from the unique fingerprint set. The traceability model is capable of inferring the corresponding user based on the prediction results of the fingerprint set from the respective proprietary model. This approach maintains a balance between traceability effectiveness and the performance of proprietary models. Experimental results on the CIFAR10 and Fashion-MNIST datasets demonstrate that the proposed method achieves over 90% accuracy in both piracy detection and traceability tasks, confirming its effectiveness across various task scenarios.

Key words: model protection; model fingerprinting; multi-task learning; piracy traceability

随着人工智能技术的快速发展, 深度神经网络模型 (DNN) 在图像识别[1−3]、自然语言处理[4,5]和自动驾驶[6,7]等领域得到了广泛的应用. 然而, 训练一个表现优异的模型往往需要大量的数据、复杂的架构设计和昂贵的计算资源. 因此, 研究针对 DNN 模型的攻击识别和防御, 防止其遭受侵害就变得极为重要.

常见的模型攻击主要包括对抗样本攻击[8,9]、数据中毒攻击[10]和模型窃取攻击等. 其中, 前两种攻击主要通过预先在模型或数据集中植入噪音或木马, 致使模型在特定条件下表现异常来实现. 模型窃取是指攻击者尝试通过非授权的模型访问实现源模型功能复现或盗用, 常见的模型窃取攻击可以分为黑盒和白盒两类, 在白盒攻击中, 攻击者可以访问模型的所有参数, 并通过微调[11]或剪枝[12]等方法对源模型进行修改, 从而实现非法使用. 在黑盒攻击中, 攻击者通过观察输入和输出的关系来推断模型的内部结构, 并对源模型进行提取攻击[13]从而训练一个替代模型来模仿源模型的行为.

近年来, 研究者已经提出了模型加密[14]、模型水印和模型指纹等多种技术来保护 DNN 模型的知识产权. 其中, 模型加密容易增加模型的计算开销, 导致推理速度变慢, 从而影响用户体验. 模型水印通过在训练期间在模型中嵌入数字标识实现版权验证, 这种方法虽然能有效地标记模型, 但存在被攻击者识别和移除的风险. 而模型指纹技术利用模型在特定数据集上的独特响应作为识别标志, 从而实现版权验证. 模型指纹不依赖于对模型训练过程的干预, 且标识难以被检测, 因而难以被篡改或移除, 具有较高的鲁棒性. 此外其为每个模型提供了一种可验证的身份标识, 能够有效地追踪模型的 IP, 从而更好地保护模型开发者的权益, 从而得到了广泛的关注.

尽管现有的 DNN 模型指纹方案能有效地声明模型版权, 但针对模型溯源追踪的研究仍然较少. 而随着云计算、分布式计算的逐步深入, 多用户应用变得越发重要, 而由于各用户的模型都是在相同的基础模型上根据自己的数据集和需求进行微调或再训练得到, 一旦发生版权纠纷, 所有者很难判断盗版模型的来源. 为此, 一些研究者提出了根据用户需求定制分发模型的方案, 但该类方案的开销太大, 不适于用户数量较多的情况. 还有部分研究将用户信息嵌入源模型, 再将这种带“标识”的源模型分发给用户. 这类方案虽然在一定程度上可以辅助判断盗版模型的来源, 但这种溯源特征是通过融合模型的训练过程嵌入的, 容易导致原始任务精度大幅下降. 此外, 这类添加标识的方案大多是基于后门攻击设计的模型水印方案, 难以抵御模型提取攻击.

针对上述问题, 本文提出了一个支持版权保护和盗版溯源的 DNN 模型指纹方案, 结合多任务学习技术, 能够在少量再训练轮次内将独一无二的溯源特征嵌入用户专有模型, 同时保证原始任务精度不会大幅下降. 本文贡献主要包括以下方面.

(1) 提出了一种创新的模型知识产权保护方案, 结合多任务学习技术, 设计了一个支持版权保护和盗版溯源的指纹方法, 从而提升了模型在多用户环境中的安全性.

(2) 开发了模型指纹溯源框架 P-Tracing, 该框架基于原始训练集的部分增强样本, 结合多任务学习技术, 为每个用户训练独特模型的同时, 更新溯源模型, 以实现对盗版行为泄露点的判断.

(3) 实验结果验证了所提方法在多种常见模型攻击场景中的有效性, 表明了其在版权保护的多用户环境中的广泛潜力.

1 相关工作

这一部分将主要介绍模型水印和模型指纹这两种模型知识产权 (IP) 保护方法的相关背景知识, 并对当前的研究现状进行归纳和分析.

近年来, 研究者针对模型版权保护提出了很多优

© 中国科学院软件研究所 http://www.c-s-a.org.cn

秀的方案, 常用的方案大致可分为模型水印和模型指纹两类. 模型水印利用了模型训练期间由于多个局部最优而导致的参数空间冗余现象[15], 将特定信息或模式嵌入到模型权重或输出中. Uchida 等人[16]首先提出了在模型中添加水印的方案, 模型所有者利用修改后的损失函数, 将水印成功地嵌入到源模型的权重中. 该方案虽然解决了微调和剪枝的鲁棒性问题, 但由于源模型权重分布的异常使得抗检测能力不高. 接着, Adi 等人[17]提出了一个基于后门攻击的 DNN 水印方案, 所有者通过精心构建的后门触发集对源模型进行微调来嵌入水印, 并在模型部署后向受信任的第三方公开验证集. 第三方查询到可疑模型, 根据结果与验证集的标签匹配度声明模型版权.

文献[7]的方案适用于所有者无法完全访问可疑模型的情形, 但该方案很难将抽象的触发集与所有者信息关联起来. Zhang 等人[18]还提出过另一个基于 DNN 后门攻击的水印方案, 他们将添加了有意义信息或无意义噪声的样本或者任务分布之外的样本作为水印集, 然后将水印集与原始训练集结合起来, 重新训练源模型从而嵌入后门. 该方案对于微调和剪枝攻击具有一定的防御力, 还能快速且准确地检验可疑模型. 但是, 该方案无法抵御模型提取攻击.

为了将附加信息嵌入源模型, 模型水印方法通常会重点设计模型的训练过程, 确保水印的鲁棒性, 但这种融合嵌入做法通常会降低源模型的准确率. 此外, 多数水印方案都无法检测新的攻击, 例如提取攻击. 而模型指纹不会将任何附加信息嵌入到源模型中, 而是利用源模型独特的行为或决策模式来构造指纹, 借此将源模型与模型所有者绑定. 这种技术通常不参与模型的训练过程, 保证了源模型准确率不受损.

Cao 等人[19]发现模型的决策边界可以唯一表征为 DNN 源模型, 为此, 他们提出了一个利用接近决策边界的对抗样本生成指纹集的方案 IPGuard. 所有者使用指纹验证集远程查询可疑模型, 通过对比返回的结果和验证集的标签判断可疑模型是否为盗版. IPGuard 解决了水印会降低源模型的准确率的问题, 但无法应对新的攻击手段.

针对上述问题, Lukas 等人[20]提出了一种利用可授予的对抗性例子表征模型决策边界的模型指纹的方法, 用以区分盗版模型和无辜模型. 可授予的对抗性例子是对抗性例子的子集, 可以通过提取攻击从源模型转移到提取模型中. 而之后提出的许多方法都依赖于使用对抗性例子进行指纹识别, 例如 Li 等人[21]提出的 ModelDiff, 该方案使用模型的决策距离向量表征模型的决策边界, 决策距离向量中的每个值都是两个对抗性例子在模型上的输出之间的距离. Li 等人[21]基于相同样本在具有相似决策边界的模型中会得到相似输出的理论, 计算了常见的和可转移的对抗性例子的决策距离向量来表征模型的决策边界. 验证时, 通过远程查询可疑模型的决策距离向量, 并对比其与源模型决策距离向量的余弦相似度判断是否盗版.

然而, 由于这些方法依赖对抗性示例, 攻击者可以通过对抗性防御 (例如对抗性训练或迁移学习) 轻松规避指纹验证. 此外, 这些方法通常需要防御者训练大量具有不同模型架构的盗版模型和无辜模型, 这造成很大的计算负担. DeepJudge[22]是一个统一的模型相似度比较框架, 该框架使用不同级别的指标检测白盒攻击和黑盒攻击中的可疑模型, 例如神经元输出到神经元级别的距离、层输出到层级别的距离等. DeepJudge 中的每个相似性度量都有基于统计分析的阈值, 当某个指标的相似度得分大于相应阈值时, DeepJudge 就会对该指标投赞成票. 验证者可以通过赞成和翻译票的占比来判断模型是否被盗. 虽然 DeepJudge 方案创建了一个可扩展的框架, 解决了依赖对抗性例子进行指纹识别方案的弊端, 但该方案仍旧只着眼于模型盗版检测, 忽视了追溯盗版模型具体的来源在模型 IP 保护中的重要性.

综上所述, 现有的模型版权保护方案仍存在以下局限性: 对于模型版权保护的重要环节——追踪溯源尚缺乏深入研究, 仍需探索能够在检测到盗版后进一步实现追踪盗版源头的方案. 为了解决这个问题, 需要分析在多用户环境中追踪溯源的难点: 首先, 鉴于用户模型是由源模型转化得到, 因此各用户模型在结构和功能上极为相似, 所有者很难判断盗版的源头. 其次, 即使选择在源模型中添加特征来标识各个用户模型, 特征的嵌入也可能会对用户模型的性能产生影响. 最后, 如何在多种攻击下仍能保证溯源方案的鲁棒性.

2 方法

P-Tracing 框架主要包含 3 个模块: 溯源指纹集生成模块、专有模型与溯源模型的再训练模块, 以及版权验证与溯源模块. 框架图如图 1 所示.

© 中国科学院软件研究所 http://www.c-s-a.org.cn

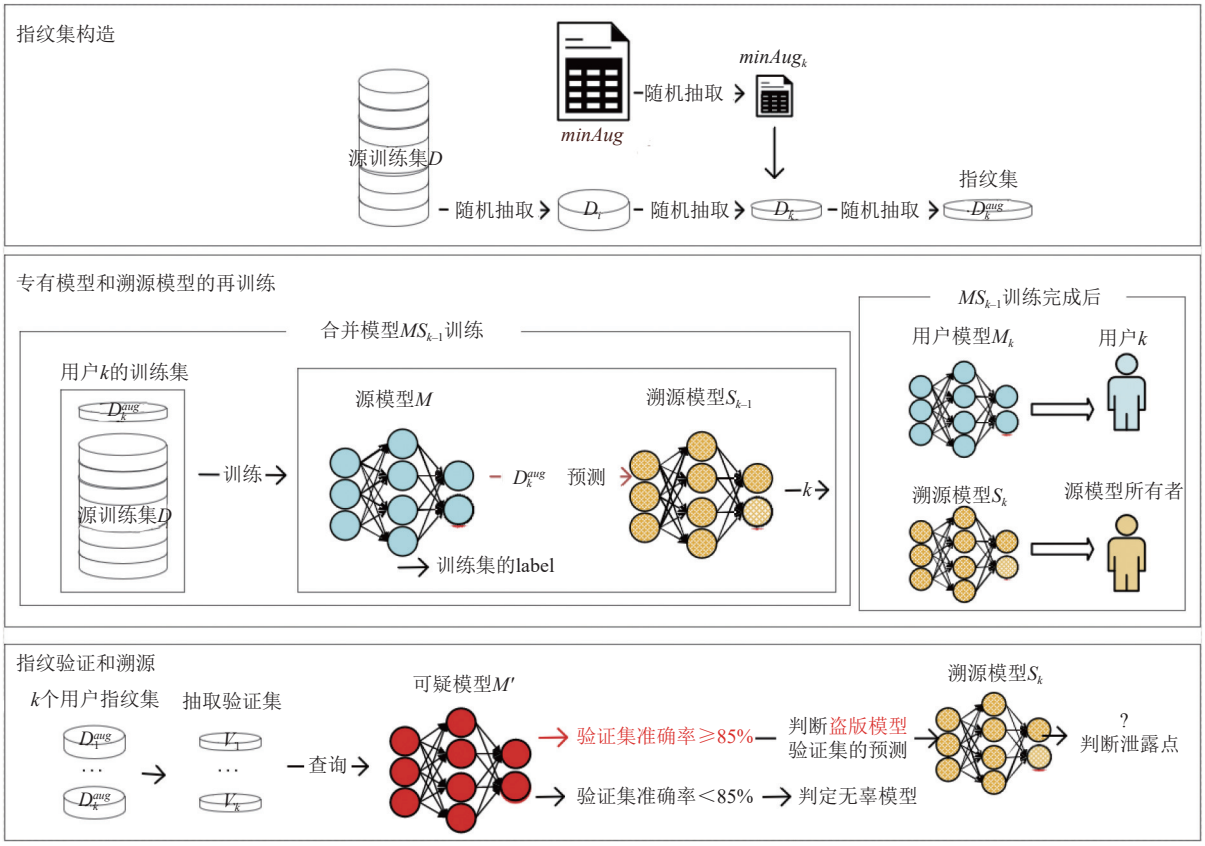


图 1 P-Tracing 框架的全局流程图

为了区分各用户模型, 就要在不显著影响原始任务精度的情况下, 将独特的指纹集嵌入用户模型. 在溯源指纹集生成模块中, 首先构建数据增强集 (涵盖几乎所有常见的数据增强方法, 包括颜色调整、几何变换、噪声添加、模糊、弹性变换、擦除、图像特效等), 接着再为每个用户随机不重复抽取唯一的数据增强组合, 并将其应用于原始训练集的部分样本上, 并为其分配对应用户的标签, 从而生成该用户独有的指纹集. 接着, 在专有模型与溯源模型的再训练模块中, 使用溯源指纹集和原始训练集少轮次地再训练源模型, 进而将指纹集特征嵌入到源模型中, 生成与指纹集对应的用户专有模型. 同时, 结合多任务学习技术, 获取指纹集在每轮再训练的源模型中的输出结果, 并根据用户标签更新溯源模型. 训练完成后, 用户专有模型将被分发给用户, 而溯源模型则保留下来, 用于版权检测和溯源. 最后, 在版权验证与溯源模块中, 将指纹集的子集作为验证集输入可疑模型, 并将结果传入溯源模型, 根据溯源模型的输出判断是否发生盗版行为, 以及盗版模型的来源.

2.1 问题定义

本框架中, 我们假设数据增强操作表为$minAug=\{minAug_1,minAug_2,\cdots,minAug_n\}$, n为数据增强操作的数量, 从$minAug$中为用户k (k为用户编号) 挑选唯一的数据增强组合$minAug_k$. 设源模型为M, 溯源模型为一个简易的多分类模型S, 输出为用户标签. 对于用户k, 定义源模型与溯源模型的串联为组合模型MS_{k-1}. 本框架的研究是将$minAug_k$作用于源模型训练集的少量样本上生成溯源指纹集, 接着利用溯源指纹集和源模型训练集训练MS_{k-1}, 将之转化为M_kS_k. 训练完成后, 拆分组合模型, 将M_k分发给用户, 将S_k保留下来, 用于下一次更新和版权保护.

2.2 溯源指纹集生成模块

为了让用户专有模型是唯一的, 用于训练它们的溯源指纹集就得是唯一的. 此外, 指纹集的嵌入不能显著损害原始任务的精度. 因此, 指纹集以原始训练数据为基础, 使用可控的、独特的数据增强组合制作而成. 每个用户的数据增强组合都是从数据增强集中随机不重复抽取的, 保证了数据增强组合的独特性. 为了控制

© 中国科学院软件研究所 http://www.c-s-a.org.cn

数据增强组合对结果的影响, 就要控制数据增强集中变换的幅度. 针对增强集中常见的图像变换, 如颜色调整、仿射变换、模糊、噪声添加、弹性变换等, 以及一些自定义的变换 (如模拟雾霾、盐和胡椒噪声), 以 CIFAR10 为例, 表 1 分析了原始数据和轻 (例如旋转角度±10°/颜色抖动: brightness=0.05, contrast=0.05, saturation=0.05, hue=0.02)、中 (例如旋转角度±30°/颜色抖动: brightness=0.1, contrast=0.1, saturation=0.1, hue=0.05)、强 (例如旋转角度±45°/颜色抖动: brightness=0.2, contrast=0.2, saturation=0.2, hue=0.1) 这 3 种增强数据之间的差异.

表 1 不同幅度的数据增强和原始数据的差异

数据增强幅度	原始数据	轻度增强	中度增强	强度增强
平均SSIM	0.98	0.97	0.95	0.89
平均PSNR (dB)	32.5	31.8	30.2	27.5
平均MSE	0.002	0.003	0.005	0.008

根据表 1 的结果, 轻度增强的平均 SSIM 值、PSNR 值和 MSE 值与原始数据最为接近, 说明轻度增强数据与原始数据的相似性更高、图片质量更好、差异性也更低. 而指纹集是原始训练数据集的增强数据, 和原始训练数据一起再训练用户模型时, 指纹集既要具有特殊性, 能让用户模型将其与原始数据集区分开; 又不能严重扭曲原始数据, 致使用户模型的精度大幅度下降. 因此, 选用轻度数据增强组合对原始数据做变换, 这样既不会改变原始数据的全局特征分布和类别信息, 又能在局部区域造成与原始数据不一样的特征分布, 生成符合条件的溯源指纹集, 具体构造方式如下.

(1) 对于每个用户, 抽取原始训练集D的随机类D_i的部分样本D_k.

$$\begin{cases} D_k = \{(x_1,y_1),(x_2,y_2),\cdots,(x_n,y_n)\} \\ D_k \in D_i \in D,\ k\text{为用户编号} \end{cases}$$

(2) 从所有者构造的数据增强集$minAug$中随机抽取若干变换$minAug_k$, 除了第 1 位用户, 后续的每位用户k抽取的数据增强组合都要与前面$k-1$个用户使用的做对比, 若有重复, 则重新抽取. 将抽取的$minAug_k$记录下来, 将其作用于样本集D_k, 为其添加额外标签, 得到指纹集D_k^{aug}:

$$D_k^{aug} = \left\{(x_1^{minAug_k},y_1,k),(x_2^{minAug_k},y_2,k),\cdots,(x_n^{minAug_k},y_n,k)\right\}$$

算法 1 描述了构造溯源指纹集的方法. D_k^{aug}的加入不仅丰富了训练集的多样性, 提高了专有模型的泛化能力, 还不会显著影响专有模型原始任务的精度. 此外, D_k^{aug}的特殊性作为用户的指纹被嵌入专有模型, 其在专有模型上的输出结果被用于训练溯源模型. 这样训练得到的专有模型是唯一的, 专有模型和对应的D_k^{aug}之间的联系也是唯一的, 有助于溯源模型在验证阶段更准确地判断模型窃取的泄露点. 最后, D_k^{aug}参与了专有模型的参数调整, 其针对潜在的模型窃取攻击的鲁棒性也随之增强.

算法 1. 可溯源指纹集生成

输入: 原始训练集D, 用户 ID k, 轻度数据增强集$minAug$.
输出: 用户指纹集D_k^{aug}.

1. $D_i \in D$; //随机选择一个来自D的D_i类
2. $D_k=\{(x_1,y_1),(x_2,y_2),\cdots,(x_n,y_n)\}$, $D_k \in D_i$;
3. $minAug_k \in minAug$
   //从数据增强组合集合$minAug$中随机无放回地选取一个组合$minAug_k$
4. $D_k^{aug}=\{\}$;
5. for each sample (x,y) in D_k:
   $x \xrightarrow{minAug_k} x^{minAug_k}$;
   //将$minAug_k$作用于样本x上
   $(x^{minAug_k},y) \rightarrow (x^{minAug_k},y,k)$;
   将(x^{minAug_k},y,k)加入D_k^{aug};

2.3 专有模型与溯源模型的再训练模块

在本模块中, P-Tracing 框架借助多任务学习技术同时再训练专有模型和溯源模型. 其中, 指纹集被用于调整专有模型参数, 将溯源特征嵌入到专有模型中. 在这个过程中, 为了保证原始任务的精度不会随着指纹嵌入而大幅下降, 原始训练集被添加到训练集中, 成为专有模型参数调整的重要标准. 而为了溯源的有效性, 指纹集在每轮调整后的专有模型中的输出结果被用于调整溯源模型的参数.

P-Tracing 框架本模块接收 4 个输入: 原始训练集D、指纹集D_k^{aug}、预训练的源模型M和上一轮的溯源模型S_{k-1}.

输出: 用户的专有模型M_k和更新后的溯源模型S_k. 再训练过程如下.

(1) 源模型所有者根据用户生成指纹集D_k^{aug}后, 若当前用户是第 1 个用户, 则将源模型M与预训练的多分类模型S组合起来得到MS; 否则, 就将源模型M与上一个用户训练完成后保存下来的溯源模型S_{k-1}串联得到组合模型MS_{k-1}.

(2) 为了同时再训练专有模型和溯源模型, 平衡指纹的嵌入效率和专有模型的性能, 就要结合多任务学

© 中国科学院软件研究所 http://www.c-s-a.org.cn

习技术, 利用专有模型在指纹集和原始训练集上针对原始任务的损失以及溯源模型针对溯源任务的损失对组合模型MS_{k-1}进行微调.

(3) 组合模型MS_{k-1}微调时, 接收原始训练集D和指纹集D_k^{aug}, 将D在M中的预测与D的标签进行对比, 得到$Loss1$, D_k^{aug}在M中的预测与D_k^{aug}的原始标签进行对比, 得到$Loss2$. 将D_k^{aug}在MS_{k-1}中的预测与附加标签k进行对比, 得到$Loss3$.

(4) 计算加权组合损失函数L:

$$\begin{cases} L=\alpha Loss1+\beta Loss2+(1-\alpha-\beta)Loss3 \\ \quad =\alpha Loss(M(x),y)+\beta Loss(M(x^{minAug_k}),y) \\ \quad\quad +(1-\alpha-\beta)Loss(S_{k-1}(M(x^{minAug_k}),k)) \\ (x,y)\in D,\ (x^{minAug_k},y,k)\in D_k^{aug} \\ 0<\alpha,\beta<1,\ \alpha,\beta\text{为超参数} \end{cases}$$

(5) 根据L调整MS_{k-1}的参数, 训练完成之后, 将MS_{k-1}拆分为M_k (分发给用户) 和S_k (由所有者保存).

算法 2 描述了专有模型和溯源模型的再训练过程.

算法 2. 专有模型与溯源模型的再训练

输入: 原始训练集D, 指纹集D_k^{aug}, 预训练源模型M, 上一轮的溯源模型S_{k-1}, 总训练轮数 TE, 加权损失参数 α, β, 经验回放 ERD.
输出: 用户专有模型M_k, 更新后的溯源模型S_k.

1. if k==1: //若当前为第 1 位用户
   $combine(M,S)\rightarrow MS$; //将源模型与预训练溯源模型组合起来
   else:
   $combine(M,S_{k-1})\rightarrow MS_{k-1}$; //否则就组合源模型与上一轮的溯源模型
2. L_f=$crossEntropy()$
3. for each epoch in TE:
   //使用加权损失函数再训练组合模型
   $L=\alpha Loss1+\beta Loss2+(1-\alpha-\beta)Loss3$
   $=\alpha Loss(M(x),y)+\beta Loss(M(x^{minAug_k}),y)$
   $+(1-\alpha-\beta)Loss(S_{k-1}(M(x^{minAug_k}),k))$;
   $optimize(L)$;
   if epoch is the last one:
   //保存最后一轮用户模型的结果和用户编号
   $ERD_k=Save(M(x^{minAug_k}),k)$;
   $L_S=Loss(ERD)$;
   $ERD=\{ERD_1,ERD_2,\cdots,ERD_{k-1}\}$;
   $MS_{k-1}\rightarrow M_kS_k$;
4. $split(M_kS_k)\rightarrow M_k,S_k$;
   M_k to $User_k$, $Save(S_k)$;

由于模型所有者只保留最后一个用户的溯源模型S_k, 所以每增加一个用户, 溯源模型就相当于更新了一次, 这样的更新会导致一个问题: 每更新一次, 溯源模型S_{k-1}都接收了一个新用户的数据, S_{k-1}的参数会逐渐调整以适应这个新用户的数据, 但这样的调整会破坏S_{k-1}在之前$k-1$个用户数据上学到的信息. 为了处理这种“灾难性遗忘”现象, 在组合模型MS_{k-1}训练到最后一轮时, 将$(M(x^{minAug_k}),k)$保存下来, 作为组合模型训练时, 对溯源模型进行“经验回放”使用的数据集, 记为ERD_k. 而在MS_{k-1}的每一轮训练完成后, 都要结合之前保存下来的$\{ERD_1,ERD_2,\cdots,ERD_{k-1}\}$来保证溯源模型对之前的指纹集的记忆.

训练完成后, 将组合模型拆分为用户专有模型M_k和更新后的溯源模型S_k. M_k分发给编号为k的用户, S_k被保留在模型所有者这里, 用于盗版行为检测和溯源, 以及下一次用户专有模型和溯源模型的再训练.

2.4 指纹验证和溯源模块

验证阶段分两步: 先根据指纹集在可疑模型上的预测结果判断是否发生盗版行为, 再根据该预测结果在溯源模型上的表现推断盗版的源头. 验证阶段的流程如下.

(1) 当出现可疑模型M'时, 源模型所有者从每个用户的指纹集中抽取少量的样本, 得到查询集$V=\{V_1,V_2,\cdots,V_k\}$, $V_k\in D_k^{aug}$.

(2) 所有者远程查询可疑模型, 并将结果输入溯源模型. 如果某个用户的查询集V_i, $i\in k$在M'中以高精度被正确分类, 就将该查询集在可疑模型中的预测结果$M'(V_i)$传递给溯源模型S_k, 根据S_k的输出标签k验证所有者的所有权, 判断模型盗版行为的源头.

3 实验

3.1 实验设置

3.1.1 原始训练集、源模型和溯源模型设置

本文使用 CIFAR10 和 Fashion-MNIST 数据集这两个图像分类数据集. CIFAR10 有 10 个类别, 图像尺寸为 32×32, 通道数量为 3 (RGB). Fashion-MNIST 数据集有 10 个类别, 图像尺寸为 28×28, 通道数量为 1 (灰度).

表 2 总结了针对不同的原始训练集, 对应的源模型在原始任务上的精度.

表 2 源模型在原始任务上的精度 (%)

原始训练集	源模型结构	原始任务精度
CIFAR10	ResNet18	91.85
Fashion-MNIST	VGG16	90.680
	SimpleCNN	88.88

溯源模型: 溯源模型是一个深度前馈神经网络, 该网络包含 7 个线性 (全连接) 层, 每个层都有指定的输入和输出神经元数.

© 中国科学院软件研究所 http://www.c-s-a.org.cn

3.1.2 训练设置

在本实验中, 源模型使用交叉熵损失函数和随机梯度下降 (SGD) 优化器进行了 250 轮的训练. 学习率设置 0.001, 动量 0.9, 权重衰减 0.000 5, 每次处理批大小为 128.

为了成功将指纹嵌入到专有模型的同时训练溯源模型, 本实验使用原始训练集和指纹集对组合模型进行了少量轮次的微调, 我们称其为组合模型微调过程.

原始训练集为 CIFAR10 时, 组合模型微调 40 轮, 对于加权组合损失函数L, 为了保持专有模型在 CIFAR10 上的精度, 超参数α设置为 0.7, 为了将指纹集特征嵌入专有模型上, 超参数β设置为 0.2, 最后, 为了训练溯源模型正确识别专有模型的输出, 超参数$(1-\alpha-\beta)$设置为 0.1; 原始训练集为 Fashion-MNIST 时, 组合模型微调 30 轮, 超参数α设置为 0.6, 超参数β设置为 0.35, 超参数$(1-\alpha-\beta)$设置为 0.05.

3.1.3 攻击设置

本文针对以下 3 类窃取攻击评估了 P-Tracing 框架的性能: 微调、剪枝、模型提取.

(1) 微调: 本文假设攻击者使用攻击数据集通过 SGD 优化器对专有模型进行 30 轮的微调.

(2) 剪枝: 本文假设攻击者使用精细剪枝, 根据神经元的激活情况对用户专有模型进行剪枝 20 轮剪枝, 剪枝率初始值为 0.05, 步长为 0.05.

(3) 模型提取: 本文假设攻击者使用基于标签的模型提取攻击, 利用攻击者数据集在专有模型上的预测标签对攻击者模型进行 150 轮的调整, 窃取专有模型的知识. 原始训练集为 CIFAR10 时, 提取攻击的模型结构为 ResNet34 和 VGG13; 原始训练集为 Fashion-MNIST 时, 提取攻击的模型结构为自构建的 CNN 和 LeNet.

3.2 实验结果

对于不同数据集和结构训练的源模型, 实验模拟了将其分发给 50 个用户的情况, 为了分析本框架在保真度、可靠性和鲁棒性上的结果, 实验在每个源模型的用户专有模型池中任意抽取了 5 个用户的专有模型, 并分析这些专有模型在未受攻击和受攻击两种状态下, 原始任务和指纹任务的表现.

实验结果部分将会使用M_i表示第 i 个被抽取用户的专有模型, 用V_i表示第 i 个被抽取用户的验证集.

3.2.1 保真度

保真度实验主要证明, 用户专有模型在原始任务上的性能不会随着指纹的嵌入而大幅度下降. 测试原始任务测试集在被抽取的 15 个专有模型上的精度, 结果如表 3 所示. 原始测试集 T 在专有模型和源模型上的精度几乎一致. 主要是因为在训练组合模型时, 损失函数被设置为加权组合损失, 它同时最小化了原始任务、指纹任务和溯源任务的损失, 保证了在少量轮次内将指纹嵌入源模型的同时, 原始任务的精度不会显著下降.

表 3 专有模型和源模型在原始测试集 T 上的精度 (%)

原始训练集	源模型结构	源模型精度	M_1	M_2	M_3	M_4	M_5
CIFAR10	ResNet18	91.85	90.33	91.00	91.69	89.40	89.71
	VGG16	90.68	90.57	90.62	90.65	90.73	90.59
Fashion-MNIST	SimpleCNN	88.88	87.20	86.86	86.37	86.68	86.15

3.2.2 可靠性

实验主要验证两个任务的可靠性: 指纹任务和溯源任务. 指纹任务的可靠性要求用户的指纹集特征被成功嵌入对应的用户专有模型中. 表 4 展示了查询集 V 在对应用户模型 M 上的精度, 可以看到, 查询集在专有模型上的精度都在 90% 以上, 指纹被成功嵌入. 而溯源任务的可靠性则要求, 将查询集在对应的专有模型中的预测结果输入溯源模型后, 溯源模型应该以高准确率输出对应的用户编号, 而不匹配的查询集和专有模型的预测结果输入溯源模型后, 溯源模型根本无法正确分类. 如图 2 所示, 只有对应的查询集和专有模型的预测结果才能同时在指纹任务和溯源任务上被准确识别.

表 4 专有模型在指纹任务上的精度 (%)

原始训练集	源模型结构	M_1 (V_1)	M_2 (V_2)	M_3 (V_3)	M_4 (V_4)	M_5 (V_5)
CIFAR10	ResNet18	99.75	100.00	99.75	96.5	99.88
	VGG16	99.62	100.00	100.00	100.00	99.75
Fashion-MNIST	SimpleCNN	99.50	91.88	100.00	98.38	98.83

为了进一步验证可靠性, 还增加了更复杂多用户场景下的实验结果, 即用户群体规模不同、数据分布不同时, 指纹集的嵌入和用户模型的溯源是否仍能成功.

鉴于不同数据分布存在以下两种情况: 不同数据集和同数据集的不同分类, 而不同数据集的结果已在表 4 和图 2 中显示, 此处仅针对不同规模用户群体和同数据集不同类别的情况做分析. 对于每个源数据集和模型结构的组合, 实验重新再训练了 20 个用户模型. 结果如表 5–表 7 所示 (第 1、3 列为用户编号/指纹集的数据集类别; 第 2、4 列为指纹集在对应用户模型上的精度/用户模型和对应指纹集在溯源模型上的精度).

© 中国科学院软件研究所 http://www.c-s-a.org.cn

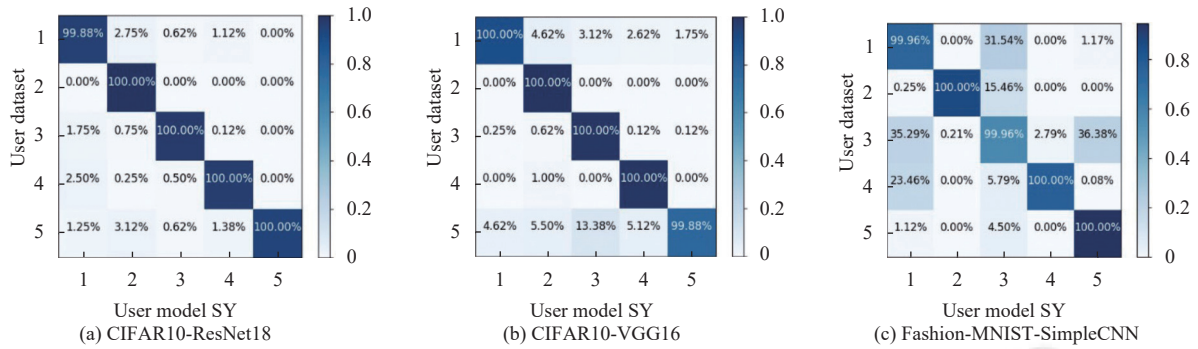


图 2 查询集和专有模型在溯源任务上的匹配度

表 5 复杂场景下 CIFAR10-ResNet18 的可靠性 (%)

用户号/分类号	指纹/溯源	用户号/分类号	指纹/溯源
1/5	99.62/100.00	11/9	98.69/100.00
2/8	99.75/100.00	12/3	99.57/100.00
3/2	96.55/99.90	13/1	99.50/100.00
4/7	99.88/99.88	14/7	99.73/99.80
5/3	97.34/100.00	15/8	98.57/100.00
6/9	99.80/99.70	16/2	96.55/100.00
7/1	98.75/100.00	17/5	96.50/99.90
8/4	99.76/100.00	18/10	97.11/100.00
9/10	95.77/100.00	19/4	96.55/100.00
10/6	100.00/99.95	20/6	98.65/100.00

表 6 复杂场景下 CIFAR10-VGG16 的可靠性 (%)

用户号/分类号	指纹/溯源	用户号/分类号	指纹/溯源
1/3	99.81/100.00	11/5	99.55/100.00
2/9	100.00/99.88	12/2	100.00/99.86
3/2	99.92/100.00	13/4	99.99/100.00
4/4	100.00/100.00	14/1	99.88/99.90
5/1	100.00/99.80	15/10	100.00/100.00
6/6	100.00/100.00	16/8	100.00/100.00
7/1	99.75/100.00	17/9	100.00/100.00
8/8	100.00/99.93	18/7	99.95/100.00
9/7	100.00/100.00	19/5	100.00/100.00
10/3	99.63/100.00	20/10	100.00/100.00

如表 5–表 7 所示, 当用户群体规模变小 (对比 50 个用户模型)、数据分布不同 (用户随机到的数据集分类并不相同) 时, 指纹任务和溯源任务仍然成功. 指纹集在用户模型上的精度都在 90% 以上, 指纹被嵌入. 对应的指纹集和用户模型能够让溯源模型输出正确的用户编号, 而非对应的指纹集和用户模型在溯源模型上无法被正确分类, 实验证明, 即使在更复杂的多用户场景下, 实验结果仍然可靠.

3.2.3 鲁棒性

实验评估了 P-Tracing 框架针对 3 种常见攻击: 微调 (finetune)、剪枝 (pruning) 和模型提取 (extract-L) 的鲁棒性: 针对被抽取的 15 个专有模型, 将 3 类攻击分别作用于其上, 生成了若干专有模型的攻击副本, 用于模拟盗版模型. 实验通过对比验证集在专有模型上的结果和在对应副本上的结果, 验证了 P-Tracing 框架的鲁棒性.

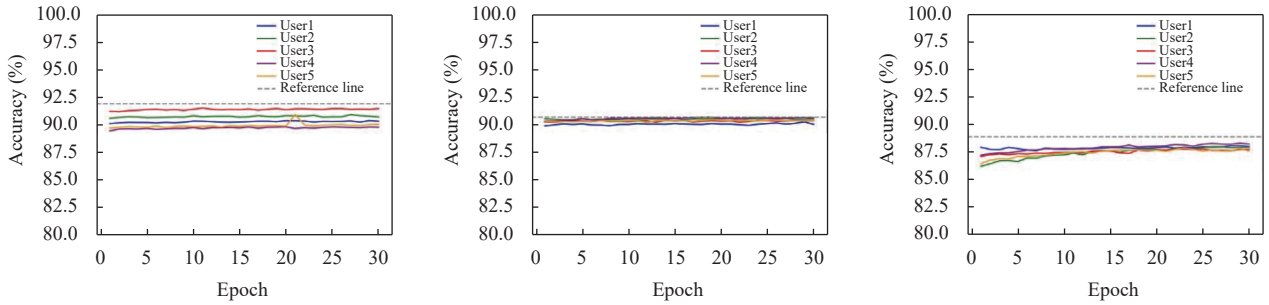
表 7 复杂场景下 Fashion-MNIST-SimpleCNN 的可靠性 (%)

用户号/分类号	指纹/溯源	用户号/分类号	指纹/溯源
1/5	98.52/99.84	11/3	92.71/99.73
2/3	92.10/99.92	12/1	99.48/99.82
3/9	96.55/100.00	13/4	92.50/100.00
4/1	100.00/100.00	14/8	97.56/100.00
5/7	97.25/99.90	15/6	96.93/99.94
6/7	97.25/99.79	16/2	97.30/100.00
7/2	99.83/99.68	17/10	97.25/99.96
8/4	91.78/100.00	18/7	98.11/99.96
9/9	98.65/100.00	19/4	97.25/100.00
10/5	94.67/100.00	20/5	97.25/100.00

图 3–图 5 首先展示了 3 种类型的攻击对专有模型原始任务精度的影响. 可以看出, 微调对于原始任务的精度影响不大, 这符合微调的特性. 模型提取会导致原始任务精度有小幅度的下降, 这是因为攻击者有很大可能使用与原训练集不一致的数据集训练提取模型. 而在剪枝攻击中, 随着剪枝率的上升, 模型参数的减少会导致模型原始任务的精度不断下降, 直至降到 10% 以下.

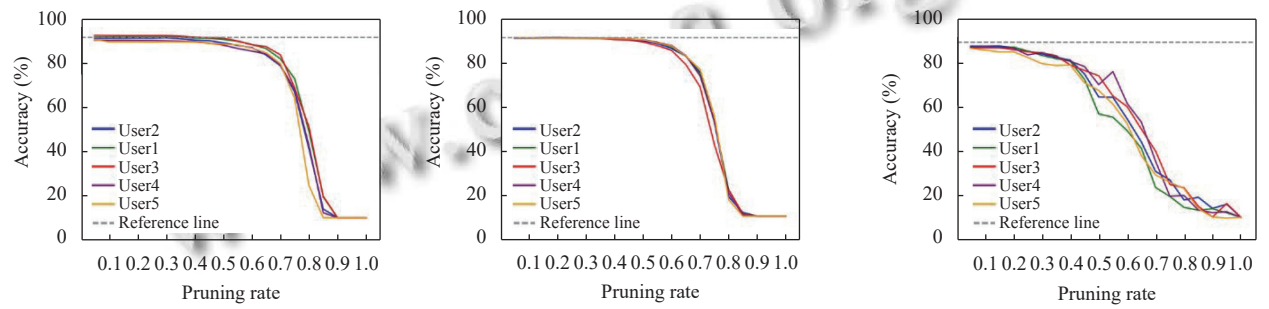
图 6–图 8 展示了 3 种类型的攻击对溯源任务的影响. 可以看出, 即使专有模型经过攻击, 查询集在攻击后的专有模型副本上的结果输入溯源模型后, 仍能得到与未攻击时几乎一致的结果. 溯源任务的准确度没有下降, 这是因为, 指纹集对源模型进行再训练时, 指纹集提高了训练集的多样性. 这种多样性有助于提高专有模型的泛化能力, 使其在查询集上的表现更加的鲁棒和稳定.

© 中国科学院软件研究所 http://www.c-s-a.org.cn



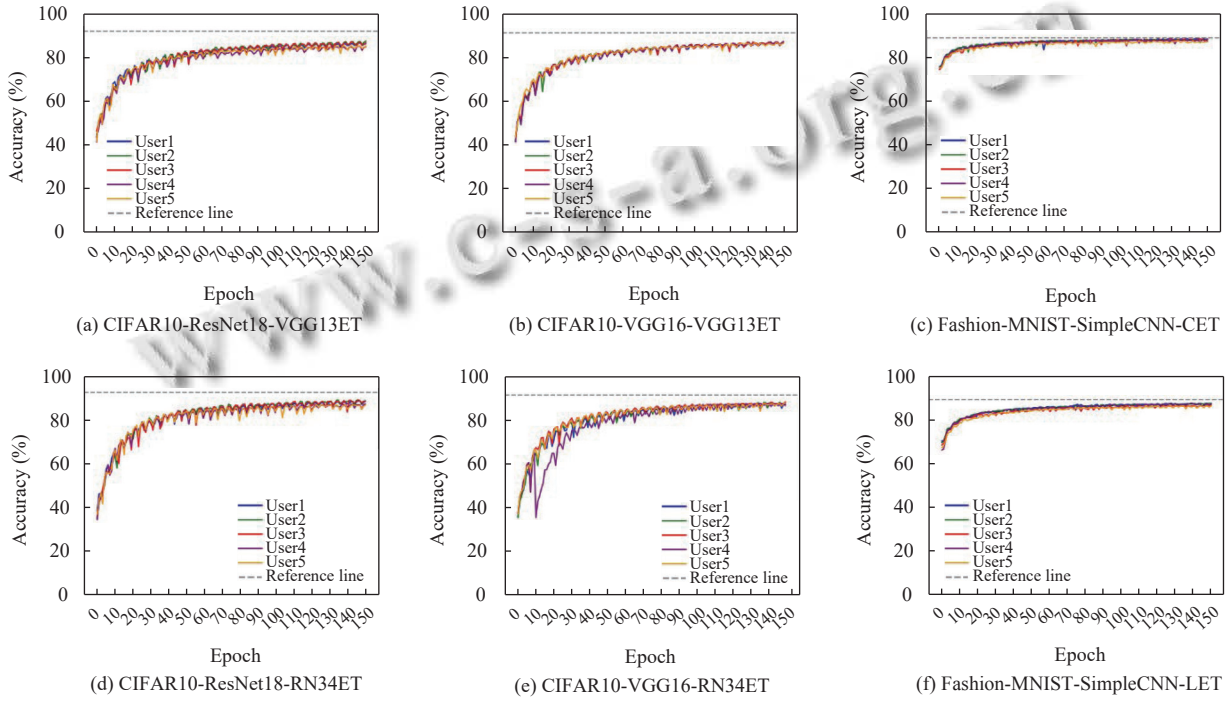
(a) CIFAR10-ResNet18 (b) CIFAR10-VGG16 (c) Fashion-MNIST-SimpleCNN

图 3 Finetune 对专有模型原始任务的影响



(a) CIFAR10-ResNet18 (b) CIFAR10-VGG16 (c) Fashion-MNIST-SimpleCNN

图 4 Pruning 对专有模型原始任务的影响



(a) CIFAR10-ResNet18-VGG13ET (b) CIFAR10-VGG16-VGG13ET (c) Fashion-MNIST-SimpleCNN-CET

(d) CIFAR10-ResNet18-RN34ET (e) CIFAR10-VGG16-RN34ET (f) Fashion-MNIST-SimpleCNN-LET

图 5 Extract-L 对专有模型原始任务的影响

© 中国科学院软件研究所 http://www.c-s-a.org.cn

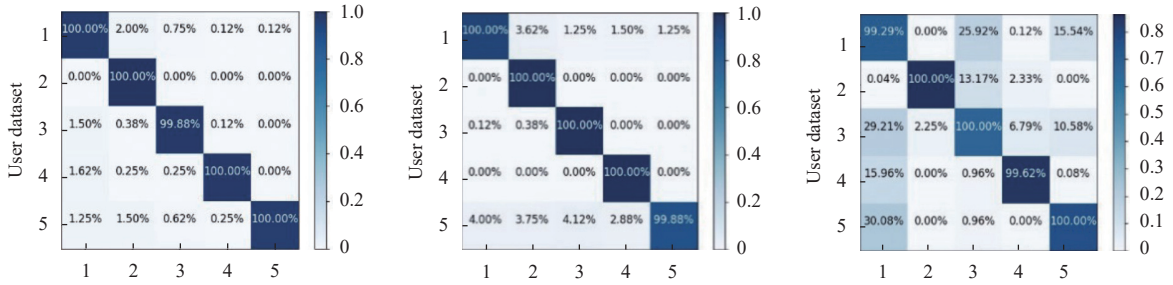


图 6 查询集和专有模型 (finetune) 在溯源任务上的匹配度

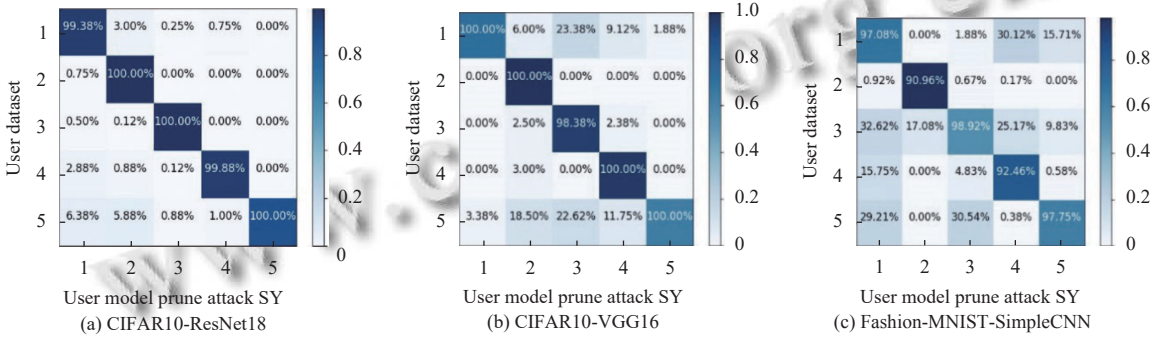


图 7 查询集和专有模型 (Pruning) 在溯源任务上的匹配度

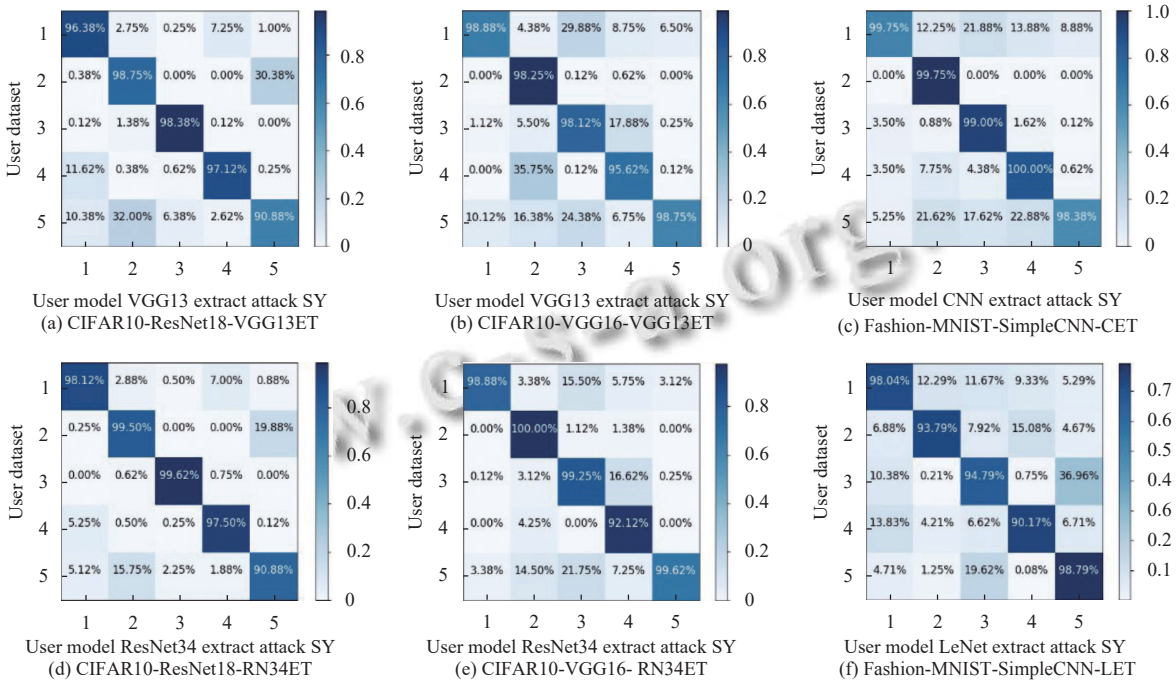


图 8 查询集和专有模型 (Extract-L) 在溯源任务上的匹配度

4 结束语

本研究通过对现有的模型指纹方案进行深入分析, 得出了以下结论. 现有的模型指纹方案虽然能够验证模型所有者身份, 但在多用户环境下的盗版溯源方面仍存在不足. 针对该问题, 本研究首先提出了溯源问题的难点. 接着, 开发了一个模型版权验证和溯源框架用于处理上述难点. 该框架基于用户唯一的指纹集, 结合多任务学习技术, 同时再训练个性化用户模型和所有

© 中国科学院软件研究所 http://www.c-s-a.org.cn

者的溯源模型. 这样得到的用户模型与溯源模型不仅能够验证所有者身份, 还能追踪盗版行为, 为模型指纹研究提供了新的视角与见解.

然而, 本研究也存在一些局限性, 例如, 单个溯源模型的输出毕竟是有限的、被提前定义的, 随着用户群体的进一步扩大, 溯源的效果可能会受限于溯源模型的输出. 这些局限性可能在一定程度上影响研究结果的可靠性. 基于上述局限性, 未来研究可以尝试使用更复杂的框架, 例如将单个溯源模型拆分为多个模型, 结合多个模型的输出判断盗版行为的泄漏点.

参考文献

1 Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Communications of the ACM, 2017, 60(6): 80–94.

2 Johnson J, Alahi A, Li FF. Perceptual losses for real-time style transfer and super-resolution. Proceedings of the 14th European Conference on Computer Vision. Amsterdam: Springer, 2016. 694–711.

3 邓志军, 田秋红. 改进 Inception-v3 网络的手势图像识别. 计算机系统应用, 2022, 31(11): 157–166. [doi: 10.15888/j.cnki.csa.008793]

4 Devlin J, Chang MW, Lee K, *et al*. BERT: Pre-training of deep bidirectional Transformers for language understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Minneapolis: ACL, 2019. 4171–4186.

5 Brown TB, Mann B, Ryder N, *et al*. Language models are few-shot learners. Proceedings of the 34th International Conference on Neural Information Processing Systems. NeurIPS, 2020.

6 Bojarski M, Del Testa DW, Dworakowski D, *et al*. End to end learning for self-driving cars. arXiv:1604.07316, 2016.

7 Teichmann M, Weber M, Zöllner JM, *et al*. MultiNet: Real-time joint semantic reasoning for autonomous driving. Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV). Changshu: IEEE, 2016. 1013–1020.

8 Goodfellow IJ, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. arXiv:1412.6572, 2014.

9 王正来, 关胜晓. 应用引导积分梯度的对抗样本生成. 计算机系统应用, 2023, 32(7): 171–178. [doi: 10.15888/j.cnki.csa.009177]

10 Shafahi A, Huang WR, Najibi M, *et al*. Poison frogs! Targeted clean-label poisoning attacks on neural networks. Proceedings of the 32nd International Conference on Neural Information Processing Systems. Montréal: Curran Associates Inc., 2018. 6106–6116.

11 Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. Science, 2006, 313(5786): 504–507. [doi: 10.1126/science.1127647]

12 Han S, Pool J, Tran J, *et al*. Learning both weights and connections for efficient neural network. Proceedings of the 29th International Conference on Neural Information Processing Systems. Montreal: MIT Press, 2015. 1135–1143.

13 Tramèr F, Zhang F, Juels A, *et al*. Stealing machine learning models via prediction APIs. Proceedings of the 25th USENIX Conference on Security Symposium. Austin: USENIX Association, 2016. 601–618.

14 Acar A, Aksu H, Uluagac AS, *et al*. A survey on homomorphic encryption schemes. ACM Computing Surveys (CSUR), 2019, 51(1): 79.

15 Li Y, Wang Hx, Barni M. A survey of deep neural network watermarking techniques. Neurocomputing, 2021, 461: 171–193. [doi: 10.1016/j.neucom.2021.07.051]

16 Uchida Y, Nagai Y, Sakazawa S, *et al*. Embedding watermarks into deep neural networks. Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval. Bucharest: ACM, 2017. 269–277.

17 Adi Y, Baum C, Cissé M, *et al*. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. Proceedings of the 27th USENIX Conference on Security Symposium. Baltimore: USENIX Association, 2018. 1615–1631.

18 Zhang JL, Gu ZS, Jang JY, *et al*. Protecting intellectual property of deep neural networks with watermarking. Proceedings of the 2018 on Asia Conference on Computer and Communications Security. Incheon: ACM, 2018. 159–172.

19 Cao XY, Jia JY, Gong NZ. IPGuard: Protecting intellectual property of deep neural networks via fingerprinting the classification boundary. Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security. Hong Kong: ACM, 2021. 14–25.

20 Lukas N, Zhang YX, Kerschbaum F. Deep neural network fingerprinting by conferrable adversarial examples. Proceedings of the 9th International Conference on Learning Representations. OpenReview.net, 2021. 1–18.

21 Li YC, Zhang ZQ, Liu BY, *et al*. ModelDiff: Testing-based DNN similarity comparison for model reuse detection. Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis. ACM, 2021. 139–151.

22 Chen JL, Wang JY, Peng TL, *et al*. Copy, right? A testing framework for copyright protection of deep learning models. Proceedings of the 2022 IEEE Symposium on Security and Privacy (SP). San Francisco: IEEE, 2021. 824–884.

(校对责编: 张重毅)

© 中国科学院软件研究所 http://www.c-s-a.org.cn