基于模型遗忘的深度神经网络鲁棒性水印方法

任纪星, 许 葳, 汪 润, 李勃衡, 张钰洋, 王丽娜

武汉大学国家网络安全学院, 空天信息安全与可信计算教育部重点实验室 武汉 中国 430072

摘要 近些年来, 深度神经网络(Deep Neural Networks, DNN)在许多前沿领域取得了巨大成功, 比如图像、语音、自然语言处理。这些DNN模型为它们的开发公司团队带来了巨大的经济收益, 同时, DNN模型的训练需要大量的数据资源和计算资源, 其成本会随着模型参数量的增加而成倍增长。因此, 一个训练良好的DNN模型对其所有者具有很高的价值。但不幸的是, 高价值、训练良好的DNN模型正受到各种模型窃取攻击、滥用和非法分发等安全威胁。深度神经网络水印是一种保护模型版权的重要手段, 根据水印是否嵌入到模型参数中, 深度神经网络水印可以分为静态水印和动态水印。静态水印由于在验证过程中需要白盒权限难以在实际应用中使用, 而向深度神经网络模型中添加验证样本和标签对映射的动态水印技术范式, 面临着难以抵御水印移除攻击的威胁。因此现有水印方法存在鲁棒性不足的问题, 导致其在实际部署应用中存在着大量风险与安全隐患。本文提出了一种基于模型遗忘的深度神经网络鲁棒性水印方法。与现有方法不同, 该方法通过使用模型遗忘技术消除原有的样本映射嵌入水印, 代替传统的添加样本标签映射方式, 规避水印移除攻击的消除, 从而极大地提高了水印的鲁棒性。具体来说, 该方法使用基于样本相似度的样本选择方法筛选需要遗忘的样本, 再通过梯度上升策略有针对性地遗忘部分训练样本的映射关系, 以提高水印的鲁棒性。该方法有效应对多种水印移除攻击, 并在CIFAR-10、CIFAR-100和TinyImageNet三个数据集上的实验中表现出优异的鲁棒性, 面对多种水印移除攻击, 本文方法水印提取有效性平均超过98%。

关键词 深度神经网络模型; 版权保护; 深度神经网络水印; 模型遗忘

中图法分类号 TP391 DOI 号 10.19363/J.cnki.cn10-1380/tn.2026.03.01

A Robust Watermarking Scheme for Deep Neural Networks based on Machine Unlearning

REN Jixing, XU Wei, WANG Run, LI Boheng, ZHANG Yuyang, WANG Lina

Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China

Abstract In recent years, Deep Neural Networks (DNNs) have achieved significant success in many cutting-edge fields, such as image processing, speech recognition, and natural language processing. These DNN models have brought substantial economic benefits to their developers and teams. However, training DNN models requires extensive data and computational resources, with costs that multiply as model parameters increase. Consequently, a well-trained DNN model holds high value for its owner. Unfortunately, high-value, well-trained DNN models face various security threats, including model theft, misuse, and unauthorized distribution. DNN watermarking has become an essential means of protecting model copyrights. Based on whether the watermark is embedded into the model parameters, DNN watermarking can be classified into static and dynamic watermarks. Due to the need for white-box access during verification, static watermarking is challenging to use in practical applications. Dynamic watermarking, which involves adding validation samples and label mappings to DNN models, is vulnerable to watermark removal attacks. Consequently, existing watermarking methods often lack robustness, posing significant risks and security concerns in real-world deployments. This paper proposes a robust watermarking method for DNNs based on machine unlearning. Unlike existing methods, this approach leverages machine unlearning techniques to eliminate original sample mapping-embedded watermarks, replacing traditional sample-label mapping methods to prevent watermark removal attacks, thereby greatly enhancing watermark robustness. Specifically, this method uses a sample selection technique based on sample similarity to identify samples that need to be forgotten. It then selectively forgets the mapping relationships of certain training samples using a gradient ascent strategy to improve watermark robustness. This method effectively counters multiple watermark removal attacks and demonstrates excellent robustness in experiments conducted on CIFAR-10, CIFAR-100, and TinyImageNet datasets, achieving an average watermark extraction accuracy exceeding 98% in the face of various watermark removal attacks.

Key words deep neural networks; copyright protection; DNN watermarking; machine unlearning

---

通讯作者: 汪润, 副教授, 博士生导师, Email: wangrun@whu.edu.cn。

本课题得到国家重点研发计划, 国家自然科学基金(No. 62576255), 中央高校基本科研业务费专项资金(No. 2025AFB455) , 湖北省自然科学基金(No. 2025AFB455)资助。

收稿日期: 2024-08-10; 修改日期: 2024-10-25; 定稿日期: 2026-01-26

1 引言

近些年来, 深度神经网络(Deep Neural Networks, DNNs)在许多前沿领域取得了巨大成功, 如图像识别、语音识别、自然语言处理等[1]。这些深度神经网络模型不仅推动着相关领域的发展进步, 同时也为它们的开发公司团队带来了巨大的收益。然而, 深度神经网络模型的训练需要大量的数据资源和计算资源, 其成本会随着模型参数量的增加而成倍增长。据一份报告称, OpenAI 花费超过 1200 万美元来训练 GPT-3[2], 对于一些模型参数量更大的大型语言模型, 他们的训练成本也会增至 200 万美元至 1200 万美元之间。因此, 一个训练良好的 DNN 模型对其所有者具有很高的价值。不幸的是, 高价值的、训练良好的深度神经网络模型正受到各种模型窃取攻击、滥用和非法分发的威胁[3-4]。因此, DNN 模型的版权应该得到保护并为此制定有效的对策。现有的 DNN 模型版权保护策略可以归纳为两大类。第一类是通过模型加密等技术限制模型的使用权, 从而预防未授权访问和模型窃取。这种方法确保只有获得授权的用户能够使用 DNN 模型进行推理, 此外这种思路难以抵御基于知识蒸馏的模型窃取。第二类则是采用 DNN 版权验证技术以检测模型是否遭到窃取, 从而对模型窃取者形成威慑。

现有主流的版权验证技术根据是否向深度神经网络模型中添加辅助的验证信息可以分为深度神经网络指纹技术和深度神经网络水印技术。深度神经网络指纹技术[5-9]是一种用于识别和区分不同深度神经网络模型的方法。这种技术无需向模型中添加任何辅助验证信息, 其通过分析和提取 DNN 模型内部结构和行为的特征, 生成一种独特的“指纹”, 用于唯一标识和识别该模型。通过对 DNN 模型进行指纹提取和比对, 可以用于验证模型来源的合法性, 防止模型被恶意篡改或窃取。然而, DNN 指纹技术也存在着通用性差的问题, 面对不同架构的 DNN 模型需要生成不同的指纹, 这一过程费时费力难以应用到现实场景中。与 DNN 指纹技术不同, 深度神经网络水印技术[10-13]将设计好的辅助验证信息嵌入到训练好的深度神经网络模型中, 在验证过程中再从 DNN 模型中提取验证信息进行验证。最近, DNN 水印技术被广泛应用于 DNN 模型的知识产权保护。DNN 数字水印技术的原始理念源自数字多媒体保护, 即在不引入明显视觉质量降低的情况下将识别信号嵌入到多媒体中。现有的 DNN 水印技术可以分为静态水印技术[14-15]和动态水印技术[16]两大类。静态水印将辅助验证信息嵌入到 DNN 模型的参数或者结构中, 在验证的过程再从中提取出来, 因此需要获得对可疑模型白盒的访问权限, 通过研究模型中的参数, 确认 DNN 模型的版权, 因而在实际环境中存在部署困难的问题。而动态水印技术则是通过在设计良好的输入样本(也称为验证样本)和输出结果之间创建特殊映射, 动态地将验证消息嵌入到模型的功能中。动态水印技术允许在黑盒设置上进行版权的验证, 并通过仅查询可疑模型提取嵌入的水印以进行所有权验证。具体来说, 模型所有者通过检查验证样本的期望输出标签与其实际输出结果的一致性来确定所有权。

不幸的是, 现有广泛采用的动态水印技术在实践中存在鲁棒性不足的问题, 这种技术通过向深度神经网络模型中添加验证样本-标签对映射, 难以抵御水印移除攻击。水印移除攻击技术旨在删除嵌入的水印而不显著降低源模型的性能, 它可以分为三类: 输入预处理攻击、模型修改攻击和模型提取攻击。有研究[17]表明当前水印方法均无法同时有效应对以上三种常见的水印去除技术。现有的动态水印范式通过在输入样本(也称为验证样本)和输出结果之间创建特殊映射将水印嵌入到模型的功能中。具体而言, 在训练过程中, 模型所有者强制目标模型记住一组预设的验证样本—标签对。在验证过程中, 模型所有者在黑盒设置下使用验证样本对可疑模型进行查询, 并根据返回的推理结果来验证水印[17]。基于添加验证样本映射的水印范式很难应对有意针对水印映射进行削弱、删除或净化的水印移除攻击, 因此现有技术存在鲁棒性不足的问题难以应对多种水印移除攻击带来的安全威胁。

为了解决现有水印方法面对水印去除攻击鲁棒性不足的问题, 本文提出了一种基于移除目标模型原有映射的 DNN 水印范式。与现有通过强制记忆样本标签对向目标模型添加映射以嵌入水印范式不同, 本文提出的水印范式利用了水印移除攻击的特性: 这些攻击通常通过抹除与目标模型不符的映射, 但在无法访问目标模型的数据集和资源时, 难以引入新的映射。基于这一点, 本文采用了移除目标模型原有映射的方式来嵌入水印, 从而有效规避水印移除攻击。具体来说, 该范式首先使用样本选择算法从训练数据集的一个类别中选择一定数量的训练样本作为遗忘样本和恢复样本, 这些样本也被指定为验证样本。接着在水印嵌入阶段, 应用梯度上升策略等方法来消除目标模型中遗忘样本与其标签对的映射, 并重新稳固恢复样本与其标签对的映射。最后在水

印验证阶段, 将遗忘样本和恢复样本同时输入到可疑模型进行评估。通过比较两种样本的映射在可疑模型之中准确率之差来验证模型的版权。图 1 展示了本文提出的水印范式与先前水印范式的对比。图 1 的上半部分描述了之前水印范式水印嵌入和验证过程的框架图, 并揭示了该水印范式面对水印移除攻击的脆弱性。下半部分则描述了本文提出的基于消除目标模型中映射的水印范式水印嵌入和验证过程的框架图, 并说明了文中所提出的水印范式在抵抗各种水印去除攻击方面表现出的强大鲁棒性。本文提出的水印范式具有强鲁棒性的原因源于这样一个事实, 即这些攻击通常会通过一定手段抹除与目标模型不符的映射(现有水印方法添加的映射大多属于此类), 但在无法访问目标模型数据集和其他资源的情况下很难引入新的映射。因此, 水印去除攻击不能影响遗忘样本(被移除样本)的验证结果。此外, 恢复样本是训练数据集的子集。水印去除攻击并不会移除它们, 因为它们倾向于保持目标模型的性能。为了实现本文提出的水印范式, 需要在不影响目标模型功能的情况下消除目标模型中的部分映射, 因此本文将利用模型遗忘技术完成水印的嵌入。模型遗忘技术受数据隐私规定的启发, 旨在从目标模型中有策略地遗忘一些指定的样本[18-19]。具体而言, 本文首先定义了样本相似度, 并利用基于样本相似度的样本选择算法筛选出需要遗忘的样本标记为遗忘样本, 将同类别中与遗忘样本相差较大的样本标记为恢复样本。接着使用基于梯度上升的模型遗忘方法将遗忘样本映射从目标模型中抹除。最后将通过识别遗忘样本和恢复样本输出进行模型版权验证。本文的方法在三个常见数据集、三种典型模型架构上对目前所有的水印移除攻击进行评估均取得极高的鲁棒性。

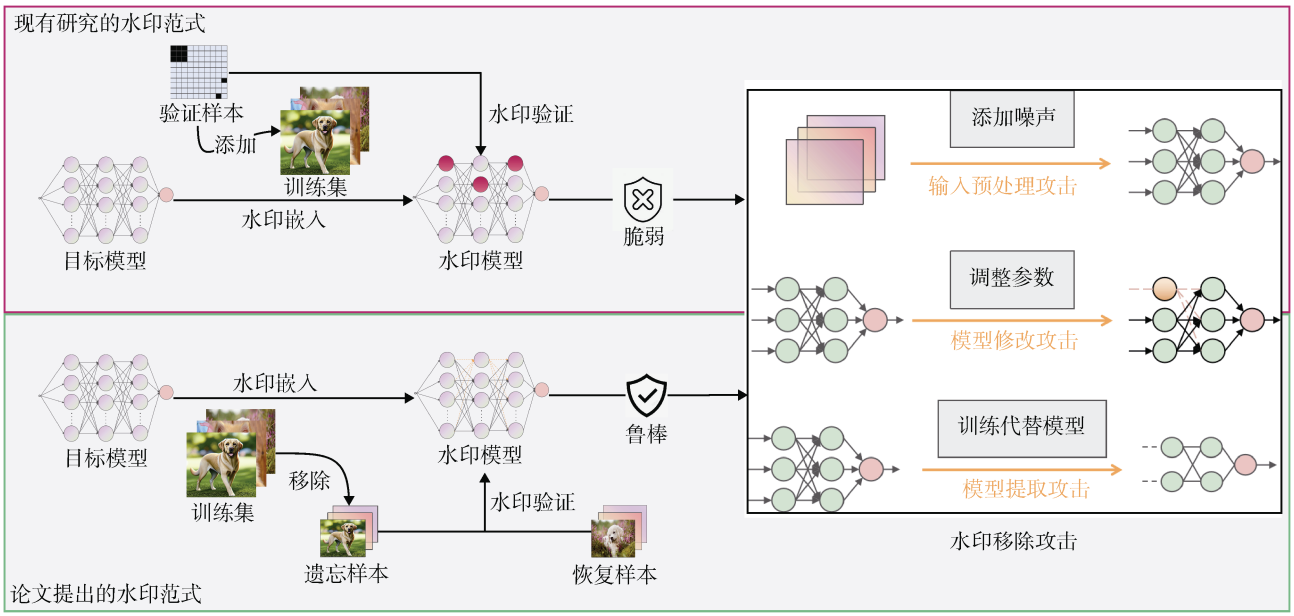


图 1　基于模型遗忘的水印方法的框架图

Figure 1　Framework Diagram of the Machine Unlearning-based Watermarking Method

本文的核心贡献归纳如下:

(1) 首先揭示了现有水印方法对各种水印移除攻击的鲁棒性不足的现象。具体来说, 现有水印方法通过将验证样本映射添加到目标模型中嵌入水印, 使得其在抵御有意针对水印映射进行削弱、删除或净化的水印移除攻击时表现非常脆弱。

(2) 本文提出了一种面向深度神经网络的鲁棒性水印方法。与现有方法不同, 该方法利用了水印去除攻击能抹除与目标模型不符的映射但很难引入新映射的特点, 通过使用基于梯度上升的模型遗忘技术消除原有的样本映射嵌入水印, 代替传统的样本标签映射方式, 规避水印移除攻击的消除, 从而极大提高了水印的鲁棒性。

(3) 实验评估中, 在三个数据集、三种模型架构上对目前所有的主流水印移除攻击进行测试均取得极高的鲁棒性, 面对三类水印移除攻击的平均水印有效性超过 **98.2%**。

本文的其他部分安排如下: 在第 2 节, 介绍了水印方法的相关工作; 在第 3 节, 介绍了水印方法所需要的预备知识与相关技术。在第 4 节, 从验证样本生成、水印嵌入、水印验证三个方面介绍了水印方法的工作流程; 第 5 节是方法的性能评估; 第 6 节讨论

了本文局限性, 第 7 节则给出了全文总结。

2 相关工作

2.1 深度神经网络水印

深度神经网络水印根据将水印嵌入到目标模型的是参数结构还是推断功能, 可以分为静态水印技术和动态水印技术[20]。

(1) 深度神经网络静态水印技术

静态 DNN 水印技术[14,21-23]通常通过在目标模型训练或微调阶段修改优化损失函数来嵌入水印, 因此存在影响模型正常性能的问题。它将水印验证信息嵌入到模型的参数或结构中, 在验证阶段再提取进行验证。但由于静态 DNN 水印提取需要获取目标可疑模型的白盒访问权限, 因此在对手以机器学习即服务(MLaaS)模式(即模型所有者通过远程 API 向用户提供推理服务模式)下部署窃取的模型时, 静态水印技术不适用。

Uchida 等人[21]首次提出将水印技术应用于深度神经网络知识产权保护[24]。模型所有者通过使用修改的损失函数训练目标模型来将水印嵌入到模型权重中。这种水印可以通过强制嵌入层的权重具有特定偏差分布来嵌入到目标模型的任何隐藏层中, 而通过投影操作可以提取出这种水印。

有研究指出 Uchida 的方法修改了模型的权重分布, 使得水印容易被检测到并可能被移除[25]。Chen 等人[22]声称, Uchida 的方法与用户身份的强关联性不强, 使其容易受到水印伪造攻击的影响。Uchida 方法生成的水印在碰撞攻击方面也存在不足, 即几个对手可以合作移除嵌入的水印。相反, 他们提出了 DeepMarks, 这是一种基于反碰撞码书的静态水印技术, 用于防御碰撞攻击。基于反碰撞码书理论, DeepMarks 为每个目标模型副本生成一种水印, 这种水印不仅与所有者唯一绑定, 而且只与一个特定用户绑定。DeepMarks 通过修改损失函数将水印嵌入到目标模型副本的一个隐藏层中, 与 Uchida 的想法相同。然而, 由于 DeepMarks 为每个目标模型副本生成唯一的水印, 因此嵌入是在目标模型的微调过程中进行的, 而不是在其训练过程中。

为了提高水印的鲁棒性和不可感知性, Wang 等人[14]提出了静态水印技术 RIGA。它通过一个对抗训练网络嵌入水印, 其中水印训练部分是生成器, 检测部分是鉴别器。这种设计强制水印模型的权重分布几乎与无辜模型相同, 提高了嵌入水印的不可感知性。此外, RIGA 训练另一个 DNN 模型来从嵌入的权重中提取水印, 而不是使用投影矩阵。这种设计增加了在模型训练过程中嵌入水印的容量。

有研究提出将水印嵌入到从目标模型权重中贪婪构造出的残差信息中[23]。他们首先基于模型所有者身份信息生成了一个私钥和公钥对, 并使用私钥加密消息以获得所有权水印。然后, 他们通过贪婪选择将水印嵌入到从模型权重构造出的残差中。这将鼓励水印嵌入到更少但更重要的权重中, 从而增强了其鲁棒性。从可疑模型中提取水印需要在白盒设置中进行。要验证提取出的水印, 可以使用来自所有者的公钥进行解密, 从而揭示模型的所有权。

由于静态 DNN 水印提取需要在白盒设置下访问可疑模型, 因此在攻击者以 MLaaS 模式(即模型所有者通过远程 API 向用户提供推理服务模式)下部署窃取的模型时, 静态水印技术不适用。

(2) 深度神经网络动态水印技术

动态水印技术[16,26-34]则是通过在设计良好的输入样本(也称为验证样本)和输出结果之间创建特殊映射, 动态地将验证消息嵌入到模型的功能中。通过调整目标模型的参数, 使得带水印的模型在面对水印数据集中的样本(也称为验证样本集)输入时, 给出模型所有者设计的特定推理结果。动态水印的提取只需要访问水印验证样本集的推理结果, 即可以在黑盒验证部署下完成水印验证。因此动态水印技术能够在对手以 MLaaS 模式部署窃取的模型时进行防御。

2017 年, Le Merrer 等人首次提出了动态水印技术, 指出静态水印在对手以 MLaaS 模式部署被盗模型时并不适用[28]。因此, 他们提出的水印方法利用 DNN 模型的对抗样本将水印嵌入到模型的功能中。具体来说, 生成了一个包含对抗样本和其他正确分类样本的水印数据集, 这些样本位于决策边界周围。通过使用这个水印数据集对目标模型进行微调来嵌入水印, 使其倾向于产生与水印数据集相同的推断结果。之后可以通过远程 API 查询可疑模型中的水印数据集中样本的预测结果来提取嵌入的水印。模型所有者可以比较预测结果与标签的相似性并验证置信度, 由此模型所有者可以宣称可疑模型是否为被盗模型。

然而, Adi 等人[16]认为基于对抗样本的动态水印方法的有效性在很大程度上取决于对抗样本在不同架构 DNN 模型之间的可传递性。此外, Le Merrer 等人[28]提出的水印具有无限的参数空间, 这也使得其方法容易受到水印伪造攻击的影响。水印伪造攻击是一种针对 DNN 模型版权保护手段的攻击方式, 其目的是在不影响模型原有功能的前提下, 修改或添

加伪造的水印, 从而挑战模型所有权的验证过程。因此, 在 2018 年, Adi 等人[16]提出了一种基于 DNN 后门攻击的动态水印方法。与对抗样本相比, 后门样本的可传递性取决于它们的构造而不是模型的决策边界。他们将 DNN 后门视为要嵌入的水印, 并使用精心构造的后门触发数据集对模型进行微调来嵌入它。他们将触发集封装在具有密钥的验证样本中, 使得水印与模型所有者的身份绑定。在部署了带有水印的模型后, 模型所有者可以向公众公开验证密钥。水印提取过程涉及一个受信任的第三方, 该第三方从所有者那里获取私有水印密钥, 并查询可疑模型以获取推断结果。如果结果与触发数据集中的标签匹配, 则第三方可以宣布可疑模型为被盗模型。

同年, Zhang 等人[31]还提出了一种基于 DNN 后门攻击的动态水印方法。他们在水印数据集中为后门样本添加信息作为模式。具体来说, 他们根据触发器的不同开发了三种水印模式。

1) 基于内容触发器的水印方法: 模型所有者将有意义的信息(通常与所有者的身份有关)添加到水印数据集中的验证样本中, 并为它们分配原始任务中的特定标签。

2) 基于分布外样本触发器的水印方法: 模型所有者使用超出目标模型任务分布范围的图像样本作为验证样本, 并为其分配原始任务中的标签。

3) 基于噪声触发器的水印方法: 模型所有者向验证样本中添加无意义的噪声, 例如高斯噪声, 并为其分配原始任务中的特定标签。

然后, 所有者可以将水印数据集与目标模型训练数据集结合起来, 并从头开始对目标模型进行训练以嵌入后门。提取和验证过程遵循与上述动态方法相同的范例。

2018 年, 考虑在应用场景中使用, Guo 等人[27]提出了一种用于嵌入系统的基于后门的动态水印方法。它采用了与水印方法中的基于内容触发器的水印方法类似的思想[33], 首先通过哈希所有者身份信息获得的水印密钥生成一个对抗性触发器, 再利用伪随机生成器为水印样本分配特定标签。

2019 年, Rouhani 等人[33]提出了 DeepSigns, 该方法首先生成一个二进制比特串作为水印, 并使用一个小的触发器数据集进行后续提取。然后, 它通过在目标模型微调期间修改损失函数, 将水印嵌入到隐藏激活值中, 这些激活值是激活后隐藏层的输出值。他们假设激活值分布遵循高斯混合模型(GMM)分布, 并将水印嵌入到 GMM 分布的均值中。要从可疑模型中提取水印, 验证器需要将触发器数据集中的样本输入到模型中并获得隐藏的激活值。然后可以使用线性投影操作从激活值中提取水印。

同年, Li 等人[29]将水印移除攻击和水印伪造攻击视为一种威胁。为了防御这两种攻击, 他们提出了一种基于 DNN 后门的方法, 将不可感知的水印嵌入到目标模型中。它利用编码器和鉴别器生成与原始任务相同分布的水印样本。编码器从任务分布中构造水印样本, 并使验证样本的触发器包含所有者身份信息, 鉴别器则确保水印样本在分布上不可检测。

2021 年, 在研究[26]中, 作者认为另一个对水印产生严重威胁的是模型提取攻击。当目标模型以 MLaaS 模式部署时, 模型提取攻击会严重威胁到水印的鲁棒性。他们观察到, 在水印提取过程中激活的神经元与正常任务推断中的激活不同, 这意味着带有水印的模型在处理原始任务和水印任务时激活的神经元不同。这个观察解释了为什么水印可以通过对目标模型进行几轮微调轻松嵌入到目标模型中, 也可以通过模型修改攻击被移除。基于这一观察, 他们提出了 EWE, 一种混合水印嵌入方法, 以抵御模型提取攻击。EWE 首先生成一个包含后门样本的水印数据集, 并将其与原始训练数据集相结合。然后它在训练损失函数中为水印嵌入添加了一项损失约束。这个损失约束了表征空间中的水印数据流形, 使其不与任务数据混合在一起, 确保在两个任务中激活相同的神经元。

同年, Szyller 等人[34]为了应对模型提取攻击提出了 DAWN。DAWN 首先生成二进制比特串的水印密钥, 并通过随机置换函数查询样本分配修改后的标签。然后, DAWN 为模型所有者在推断 API 输出后提供一个额外的组件。这个组件根据使用水印作为随机性来源的加密哈希函数随机决定是否修改查询输出。由于对手使用与任务提取数据集中的预测标签相结合的查询样本作为模型提取攻击中的训练数据集, 因此水印在对手的训练过程中得以保留, 从而可以验证被盗模型的版权。

不幸的是, 现有广泛采用的通过向深度神经网络模型中添加验证样本和标签对映射的动态水印技术范式在实践中存在对水印移除攻击鲁棒性不足的问题。有研究[17]表明当前水印方法均无法同时有效应对以上三种常见的水印去除技术。现有的动态水印范式通过在输入样本(也称为验证样本)和输出结果之间创建特殊映射将水印嵌入到模型的功能中。具体而言, 在训练过程中, 模型所有者强制目标模型记住一组预设的验证样本—标签对。在验证过程中, 模型所有者在黑盒设置下使用验证样本对可

疑模型进行查询, 并根据返回的推理结果来验证水印[17]。基于添加验证样本映射的水印范式很难应对有意针对水印映射进行削弱、删除或净化的水印移除攻击, 因此现有技术存在鲁棒性不足的问题难以应对多种水印移除攻击带来的安全威胁。

2.2 水印移除攻击

水印移除攻击技术旨在删除嵌入的水印而不显著降低源模型的性能。现有的水印移除攻击根据攻击的方式不同可以分为三类[17]: 输入预处理攻击、模型修改攻击和模型提取攻击。

(1) 输入预处理攻击在将输入样本通过目标模型之前, 通过添加噪声等手段影响它们的推理过程。攻击者必须具有对目标模型的白盒访问权限。常见的技术包括输入重构[35]、JPEG 压缩[36]、输入量化[37]、输入平滑[38]、输入噪声[39]、输入翻转、特征压缩[38]。输入重构[35]使用自动编码器[40]压缩和重建图像, 达到去除输入数据触发器的目的。输入量化将输入数据从高精度表示转换为低精度表示, 从而导致水印信号的部分或完全丢失。通过对输入数据进行量化, 攻击者可以模糊或扭曲水印信号, 使其无法被有效地提取或识别, 从而达到移除水印的目的。一项最新的工作[41]引入了自然感知重照明扰动来掩盖嵌入的水印触发器, 该方法在破坏验证样本方面取得了 SOTA 性能。由于输入预处理技术通常不依赖于水印方法、与模型无关且对训练数据不敏感, 它对 DNN 水印构成了极大的威胁。

(2) 模型修改攻击通过调整目标模型的参数、结构来干扰模型的推理过程。攻击者必须具有对目标模型的白盒访问权限。例如模型微调, 通过让目标模型在少量原数据集上降低学习率进行训练来去除模型中的水印。剪枝则是通过对模型中结构参数的修剪达到去除水印的目的。其余常见的模型调整攻击还有对抗性训练[42]、权重量化[43]、标签平滑[44]、微调-剪枝[45]、特征排列、权重修剪[46]、神经元清理[47]、正则化[48]。

(3) 模型提取攻击通过将目标模型的知识迁移到替代模型中来训练替代模型。现有的模型提取攻击中, 除了知识蒸馏[49]需要白盒访问权限, 其余攻击只需黑盒访问目标模型即可完成。常见的模型提取攻击技术包括迁移学习、重训练[50]、知识蒸馏、Knockoff Nets[51]等。Knockoff Nets 旨在通过目标模型的输入输出生成一个替代模型(即 Knockoff 模型), 该模型在功能上类似于目标模型, 但不包含水印。知识蒸馏技术则是攻击者使用目标模型的输出软标签分布来训练一个新的模型, 该模型被设计为在不包含水印的情况下模拟目标模型的行为。通过知识蒸馏, 攻击者可以在不直接访问原始模型的情况下复制其功能。在迁移学习技术中, 攻击者将目标模型的知识迁移到一个新的模型中, 该模型在训练过程中可能会删除水印。通过在新任务上微调原始模型, 攻击者可以使新模型在不包含水印的情况下执行与目标模型相似的任务。重训练会对目标模型进行微调或重新训练, 以删除或破坏水印。通过在水印模型上进行重新训练, 攻击者可以改变模型的参数和行为, 从而减弱或破坏水印。

2.3 模型遗忘

模型遗忘是一种有策略地从目标模型中忘记一些指定的样本的技术[19,52]。一般来说, 模型遗忘可以根据是否允许在遗忘模型和重训练模型之间存在一定程度的近似性, 分类为精确遗忘和近似遗忘。

(1) 精确遗忘

精确遗忘要求通过遗忘算法产生的所有模型的分布与从头重新训练的所有模型的分布之间的距离为零[53-54]。文献[19]提出了 SISA 算法, 这是一种实用的遗忘方法, 借鉴了分布式训练和集成学习的元素。然而, 精确遗忘技术仅证明了它们在传统的机器学习模型上的有效性, 比如决策树和随机森林。另外, 一些精确遗忘算法是为特定类型的模型所设计的, 比如 DaRE[53], 它专门针对基于决策树和随机森林的模型。

(2) 近似遗忘

近似遗忘放宽了精确遗忘的要求, 但限制了遗忘机制的近似误差[55]。 研究[56]首次提出了近似遗忘的概念, 并通过在模型参数上应用牛顿步骤, 为使用可微凸损失函数训练的模型开发了一种移除机制。研究[57]通过将标准深度网络替换为适当的线性逼近, 对非凸模型(如 DNN 模型)进行了近似遗忘的一些工作。

现有的对抗遗忘算法要么需要大量的计算和存储资源, 要么是为特定模型设计的, 因而难以应用于本文中的方法。为此, 本文提出了一种更高效、与模型无关的对抗遗忘方法, 即梯度上升算法, 以有效地抵御多种水印去除攻击。

3 背景知识

3.1 深度神经网络模型

深度神经网络模型[58]是一种数学函数 $F: X \to Y$, 它用于将输入分布 X 映射到输出分布 Y 上。F 通常由一系列的层 $l_i(\cdot), i \in \{1,2,\cdots,N\}$ 组成, 其中每一层 l_i 都有多个神经元排列, 并由一个线性函

数和一个非线性函数(激活函数)构成。深度神经网络模型的输入层和输出层之间的层称为隐藏层, 隐藏层中的每一层都由参数权重 w 和偏差 b 这两类参数共同计算本层的激活值。深度神经网络模型是由多个隐藏层构成的神经网络。对于分类任务的深度神经网络而言, Softmax 层 softmax(·) 用于将输出层 $l_N(\cdot)$ 的每个类从预测的可能性转换为概率。

深度神经网络模型的训练需要数据集 $D=\{x_i,y_i\}_{i=1}^N$ 和可微的损失函数 $\mathcal{L}(\cdot)$。x_i 是数据样本, y_i 则是对应的标签。通过梯度下降的方式对模型中可训练的参数权重和偏差进行优化, 使得损失函数 $\mathcal{L}(\cdot)$ 最小来完成深度神经网络的训练。

现有的深度神经网络模型部署方式[59]有两种: 本地分发模式和MLaaS模式。在本地分发模式下, 模型所有者将训练好的模型分发给用户进行使用, 即为白盒部署, 每个用户单独使用一个模型。随着近些年深度神经网络任务的复杂化, DNN 模型的参数和所需的计算资源增加, 普通用户难以负担这个成本, 因此本地分发模式较难满足如今的模型使用需求。MLaaS 模式则是多个用户共同使用一个模型, 模型部署在云服务器上, 用户仅可通过模型开放的 API 访问使用模型。这种模式下, 模型所有者可以限制未授权者对模型的再分配, 也使得用户的模型使用环境不再苛刻, 因而 MLaaS 模式更受欢迎。由于用户难以访问模型内部的信息, 这种部署方式也被称为黑盒部署。

不幸的是, 以上两种部署方式都面临着未经授权的使用、分发和篡改的威胁。在本地分发模式下, 模型直接部署在用户手中, 用户可以很轻易地修改、分发模型, 因而严重威胁了 DNN 模型的版权安全。在 MLaaS 模式下, 模型部署在云端, 但通过模型窃取攻击, 攻击者仅需通过调用 API 进行查询就能窃取整个网络模型, 从而对模型所有者造成极大的经济损失。因此, 在这两种部署模式下, 都需要使用版权保护策略对 DNN 模型进行保护。

3.2 深度神经网络水印方法

本文主要考虑两个角色: 模型所有者和模型窃取者。模型所有者拥有完整的训练集, 并使用训练集训练了目标模型。在白盒访问权限下, 模型拥有者通过运行水印嵌入算法将水印嵌入到目标模型中。一个 DNN 水印方法是一组概率多项式时间算法 $(\mathrm{VSGen},\mathrm{Embed},\mathrm{Verify})$, 其中:

$\mathrm{VSGen}(D)$: 给定数据集 D, 生成水印的验证样本集 $V=\{(x_{v_i},y_{v_i})\}_{i=1}^N$, 其中包含 N 个带有信息的验证样本 $X_v=\{x_{v_i}\}_{i=1}^N$ 和相应的验证信息 $Y_v=\{y_{v_i}\}_{i=1}^N$。算法旨在生成精心设计的用于水印验证的验证样本集。

$\mathrm{Embed}(V,f)$: 算法旨在将精心设计的验证样本集作为水印嵌入到训练好的模型中。算法通过输入验证样本集 V 和训练好的模型 f, 输出一个包含验证样本集 V 验证信息的水印模型 $\hat{f}$。

$\mathrm{Verify}(\hat{f},V)$: 算法旨在应用验证样本集对可疑模型的版权进行验证。通过比较 $\hat{f}(x_{v_i})$ 和 y_{v_i} 的差异, 计算验证集的准确率。如果准确率小于预定义的阈值 σ, $\mathrm{Verify}(\hat{f},V)$ 输出 0 表示验证失败。否则, 输出 1, 表示验证成功。

模型所有者希望能够保护其知识产权, 确保其拥有的模型不被未经授权的用户使用。他们希望能够通过嵌入水印的方式对模型进行标记, 以便在需要时验证模型的所有权。模型所有者希望在面对可疑模型时, 能提取出完整的验证信息, 证明可疑模型为被盗模型或无辜模型。在验证可疑模型时, 模型所有者仅拥有可疑模型的黑盒访问权限。

模型窃取者希望未经授权地使用目标模型, 并且获得与目标模型功能一致且不含水印的模型。模型窃取者仅拥有有限的训练集知识, 他们拥有三类数据集: 一小部分不超过三分之一的训练集 $D_1=\{x_i,y_i\}_{i=1}^M, M<\frac{N}{3}$、与训练集同分布的无标签数据集 $D_2=\{x_i\}_{i=1}^N$、与训练集不同分布的带标签数据集。模型窃取者拥有目标模型的白盒权限, 但 $D_3=\{m_i,n_i\}_{i=1}^T$ 没有水印技术的相关知识。因此, 模型窃取者可以通过输入预处理攻击、模型修改攻击和模型提取攻击等手段故意回避嵌入的水印验证。

输入预处理攻击。模型窃取者希望通过修改模型的输入, 影响它们的推理过程, 使得验证样本集中的输入失效。通过攻击算法 $\mathrm{Attack}_I(\cdot)$, 修改模型输入 $\mathrm{Attack}_I(x_{v_i})$, 使得 $\hat{f}(\mathrm{Attack}_I(x_{v_i}))\neq y_{v_i}$。

模型修改攻击。模型窃取者希望通过调整目标模型的参数、结构来干扰模型的推理过程, 使得模型中的水印失效。使用攻击算法 $\mathrm{Attack}_M(\cdot)$, 修改模型 $\hat{f}$, 使得 $\hat{f}_M=\mathrm{Attack}_M(\hat{f})$, 并且 $\hat{f}_M(x_{v_i})\neq y_{v_i}$。

模型提取攻击。模型窃取者通过将目标模型的知识迁移到替代模型中来训练替代模型。通过攻击算法 $\mathrm{Attack}_E(\cdot)$, 使用与训练集同分布的无标签数据集 D_2 训练代替模型 $\hat{f}_E=\mathrm{Attack}_E(x_i,\hat{f}(x_i)_{i=1}^N)$, 使得 $\hat{f}_E(x_{v_i})\neq y_{v_i}$。

3.3 深度神经网络水印的评估指标

深度神经网络水印有以下四个可以量化的评估指标: 保真度、水印有效性、鲁棒性、嵌入效率。

(1) 保真度

深度神经网络水印的保真度[17]用于衡量嵌入水印之后, 模型原有功能受到的影响, 通常用嵌入损失 EL(Embedding Loss)来衡量。本文首先定义辅助验证函数 $\Pr(\cdot)$。

$$\Pr(y,\hat{y})=\begin{cases}1, y=\hat{y}\\ 0, y\neq\hat{y}\end{cases} \tag{1}$$

接着, 本文定义了一个模型 f 在数据集 $D=\{x_i,y_i\}_{i=1}^{N}$ 上的准确率, $\mathrm{Acc}(f,D)$。

$$\mathrm{Acc}(f,D)=\frac{\sum_{i=1}^{N}\Pr(y_i,f(x_i))}{N} \tag{2}$$

那么一个水印方法的嵌入损失 EL 可以表示为

$$\mathrm{EL}=\mathrm{Acc}\left(f,D_{\mathrm{val}}\right)-\mathrm{Acc}\left(\hat{f},D_{\mathrm{val}}\right) \tag{3}$$

D_{val} 表示目标模型的验证集。

(2) 水印有效性

深度神经网络模型水印有效性[17]反映目标模型在嵌入水印后, 模型所有者从中提取出水印的能力, 通常使用验证集在水印模型上的准确率来代替。水印有效性 wmacc 定义如下:

$$\mathrm{wmacc}=\mathrm{Acc}\left(\hat{f},V\right) \tag{4}$$

(3) 鲁棒性

深度神经网络水印的鲁棒性体现了水印方法在面对特定水印移除攻击时的有效性。面对水印移除攻击 R 的攻击算法 $\mathrm{Attack}_R(\cdot)$, 本文可以定义水印方法在面对攻击 R 时的准确率:

$$\mathrm{wmacc}_R=\mathrm{Acc}\left(\mathrm{Attack}_R\left(\hat{f}\right),\mathrm{Attack}_R\left(V\right)\right) \tag{5}$$

如果 wmacc_R 大于阈值 σ, 那么就称该水印方法面对攻击 R 具有鲁棒性。

4 基于模型遗忘的深度神经网络水印方法

本文提出了一种基于模型遗忘的深度神经网络水印方法。该方法利用了水印移除攻击难以引入新的映射的特性, 采用了移除目标模型原有映射的方式来嵌入水印, 从而有效规避水印移除攻击。图 2 所示为本文所提方法的框架, 主要包括验证样本生成、水印嵌入和水印验证。首先, 在验证样本生成阶段, 需要选取用于版权验证的验证样本。本文使用的验证样本分为两个部分即遗忘样本和恢复样本。遗忘样本和恢复样本都是目标模型训练集中的正常样本, 区别在于在水印嵌入阶段遗忘样本的映射需要从目标模型中抹除, 而恢复样本的映射需要加强和巩固。本文利用一个基于样本相似度的样本选择算法从训练集中挑选出这两类验证样本, 确保它们适用于后续的水印验证任务。接下来, 在水印嵌入阶段, 需要将水印嵌入到目标模型中, 即抹除目标模型中遗忘样本的映射, 并加强恢复样本的映射。本文将利用基于梯度上升的模型遗忘算法抹除目标模型中遗忘样本的映射将水印嵌入到目标模型中。最后, 在水印验证阶段, 本文将使用验证样本完成对可疑模型的版权所有权验证。本文将利用验证样本中遗忘样本和恢复样本在可疑模型中的正确率之差完成水印验证。

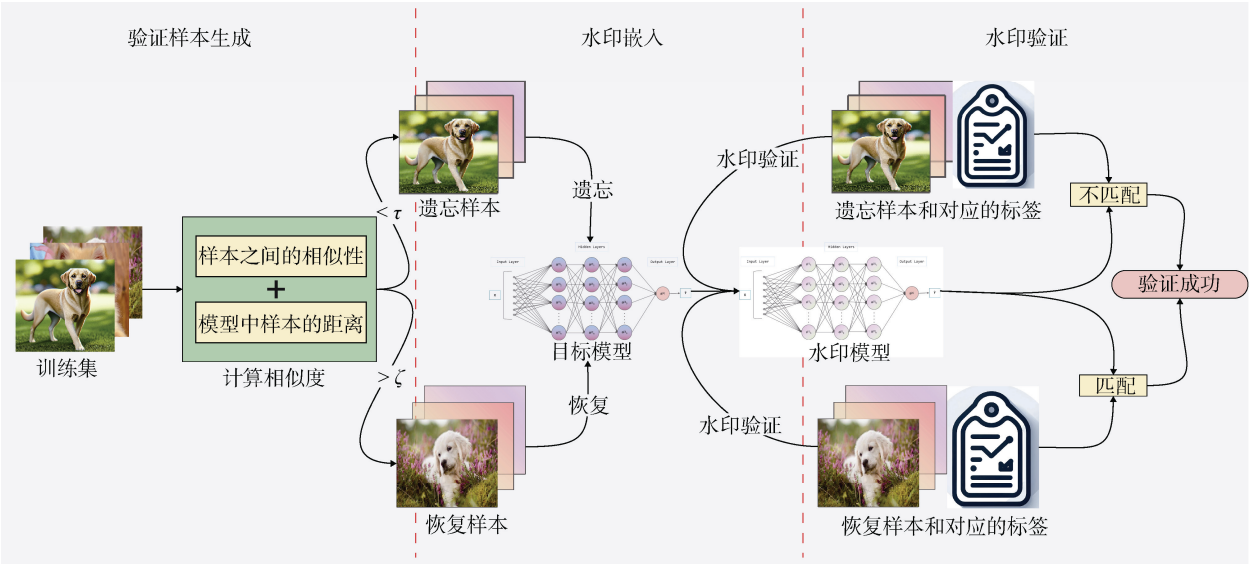


图 2 基于模型遗忘的水印方法示意图

Figure 2 Schematic Diagram of the Machine Unlearning-based Watermarking Method

4.1 验证样本生成

在验证样本生成阶段, 需要挑选用于版权验证的验证样本。本文使用的验证样本分为两个部分即遗忘样本和恢复样本。水印方法在水印验证过程中需要同时验证遗忘样本和恢复样本的正确率。遗忘样本和恢复样本都是目标模型训练集中的正常样本, 区别在于在水印嵌入阶段遗忘样本的映射需要从目标模型中抹除, 而恢复样本的映射需要加强和巩固。所有遗忘样本和恢复样本都具有相同的类标签, 并属于训练数据集。遗忘样本和恢复样本存在一定程度的类内距离, 但这个距离通常小于训练集类与类之间的距离。由于深度神经网络模型拥有泛化能力, 距离较近的样本更有可能被模型分类为相同的类别, 即使它们以前从未被模型学习过。因此, 该方法通过遗忘样本和恢复样本之间预测准确率的差异性来完成水印验证, 确定模型的版权所有权。具体来说, 本文采用模型遗忘技术来实现提出的水印范式。模型遗忘旨在定向抹除目标模型中的样本, 使得遗忘后的模型与从一开始就从未学习过被抹除样本的模型一致, 这意味着处理后的模型在遇到遗忘样本时结果是不确定的。在训练集中, 属于同一类别的样本通常在数据分布上表现出高度相似性。这可能导致这样一种情况: 由于深度神经网络模型的泛化性, 即使模型成功地忘记了遗忘样本, 同一类别中的其他相似样本仍然可能使模型学习到遗忘样本, 从而导致水印验证失败。因此, 本方法需要选择其相似的同类样本加入遗忘数据集, 或者将它们从训练集中剔除。

为了衡量样本与样本之间的相似性, 本文提出了一种样本相似性定义方法, 通过计算模型目标模型中样本之间的距离和样本之间的相似性来衡量两个样本之间的真实距离。

目标模型中样本之间的距离: 表示样本在目标模型推断阶段的相似性, 它直接反映了目标模型的样本分布和决策边界。如果两个样本在目标模型中距离接近, 则学习一个样本会使得模型有很大概率能够学习距离它接近的样本。由于在直接测量推断期间两个样本之间的距离存在困难, 本文选择使用模型推断两样本输出结果的余弦相似性分数。具体来说, 给定两个样本 x_i 和 x_j, 对应输出 $\widehat{y_i}$ 和 $\widehat{y_j}$ 的余弦相似性分数 $s_{i,j}$ 在模型 f 上的计算如下:

$$\widehat{y_i} = f(x_i) \tag{6}$$

$$s_{i,j} = \frac{\widehat{y_i}\widehat{y_j}^{\mathrm{T}}}{\|\widehat{y_i}\|\|\widehat{y_j}\|} = \frac{f(x_i)f(x_j)^{\mathrm{T}}}{\|f(x_i)\|\|f(x_j)\|} \tag{7}$$

同样, 从模型角度定义的样本 x_i 和 x_j 之间的相似性分数 S_1 在目标模型中如下定义。

$$S_1 = s_{i,i} - s_{i,j} \tag{8}$$

样本之间的相似性: 直观地展示了两样本本身的相似性。为了估计样本本身之间的距离, 需要提取样本的轮廓信息进行比对, 本文利用离散小波变换(DWT)处理样本。离散小波变换可以提取图像的低频信息, 低频图像变化缓慢, 描述了原始图像的局部平均值, 存储了原始图像的轮廓信息。给定一个低通滤波器 L, 原始图像通过 DWT 算法进行滤波, 产生一个低频图像和三个高频图像。样本 x_{ll} 的相似性可以通过低频图像的 L_2 距离表示如下。

$$x_{ll} = LxL^{\mathrm{T}} \tag{8}$$

然后, 可以定义样本 x_i 和 x_j 之间本身的相似性分数 S_2 定义如下:

$$S_2 = \| Lx_iL^{\mathrm{T}} - Lx_jL^{\mathrm{T}} \| \tag{9}$$

最终的相似性分数 S 由上述两种相似性测量方法确定。随着 S 值的降低, 样本之间的相似性增加。可以描述如下。

$$S = S_1 + \alpha S_2 \tag{10}$$

其中, α 是一个超参数, 用于调整两种不同距离测量方法的权重。在实际应用中, 本文经验性地将 α 的值设定为 10^{-4}。本文首先从训练集中随机选择一小部分初始样本, 计算其他所有样本与初始样本的相似性分数, 并进行排序。对于其中相似性分数低于阈值 τ 的样本称为高相关性样本, 其中相似性分数大于阈值 ζ 的样本称为低相关性样本。本文将高相关性样本作为遗忘样本集 $D_u = \{x_{u_i}, y_{u_i}\}_{i=1}^{M}$ 并使用模型遗忘算法进行处理, 将低相关性样本作为恢复样本集 $D_r = \{x_{r_i}, y_{r_i}\}_{i=1}^{N}$ 进行重新训练, 以实现功能保留。相似性距离大于 τ 且小于 ζ 的样本则称为废弃样本, 在水印嵌入阶段不进行任何操作。最后, 本文将同时利用遗忘样本集 D_u 和恢复样本集 D_r 进行水印验证过程。

4.2 水印嵌入

在水印嵌入阶段, 需要将水印嵌入到目标模型中, 即抹除目标模型中遗忘样本的映射, 并加强恢复样本的映射。因为水印移除攻击通常会通过一定手段抹除与目标模型不符的映射, 但在无法访问目标模型数据集和其他资源的情况下很难引入新的映射, 因此通过抹除目标模型中的映射作为水印能在面对水印移除攻击时具有强大的鲁棒性。在确定了

遗忘样本和恢复样本之后, 需要删除遗忘样本的映射并重新学习恢复样本的映射, 以将水印嵌入到目标模型中。为实现本文的水印范式, 本文将采用模型遗忘技术。然而, 现有模型遗忘算法所需的计算和存储资源在实际应用中是无法承受的, 并且在本文的水印方法中不需要严格限制遗忘数据产生的近似信息泄露。因此, 为了将水印嵌入目标模型, 本文利用梯度上升算法对遗忘数据集进行擦除。然而, 模型遗忘过程会导致模型功能的指数级下降, 因此应该同时重新训练恢复样本以修复模型。为了生成水印模型 $\hat{f}$, 样本可以分为用于模型遗忘的遗忘样本 x_u 和用于学习以恢复目标模型性能 f 的恢复样本 x_r, 描述如下:

$$\hat{f}=\underbrace{\arg\max}_{f}\sum_{i=1}^{M}\mathcal{L}(\widehat{y_{u_i}},y_{u_i})+\underbrace{\arg\min}_{f}\sum_{i=1}^{N}\mathcal{L}(\widehat{y_{r_i}},y_{r_i}) \quad (11)$$

其中, $\mathcal{L}$ 表示损失函数, y_u 和 $\widehat{y_u}$ 分别是 x_u 的真实标签和推断结果, y_r 和 $\widehat{y_r}$ 分别是 x_r 的真实标签和推断结果。为了保持一致的优化方向, 可以重写公式(11):

$$\hat{f}=\underbrace{\arg\min}_{f}\left(\sum_{i=1}^{M}\mathcal{L}(\widehat{y_{u_i}},1-y_{u_i})+\sum_{i=1}^{N}\mathcal{L}(\widehat{y_{r_i}},y_{r_i})\right) \quad (12)$$

本文将交叉熵损失函数记为 $\mathcal{L}_{\mathrm{CE}}$, 整个过程的损失可以写成如下形式。

$$\mathcal{L}_{\mathrm{CE}}=-\left(\beta\sum_{i=1}^{M}(1-y_{u_i})\log f(x_{u_i})+\sum_{i=1}^{N}y_{r_i}\log f(x_{r_i})\right) \quad (13)$$

其中, M 和 N 分别是遗忘样本和恢复样本的数量。β 是一个超参数, 用于控制遗忘算法的程度。

4.3 水印验证

在水印验证阶段, 本文将使用验证样本完成对可疑模型的版权所有权验证。为验证可疑模型的版权, 本文将利用遗忘样本和恢复样本在模型中准确率的差异进行水印验证。正如本文之前提到的, 即使样本从未被学习过, 距离较近的样本更有可能被分类到特定类别中。遗忘样本和恢复样本存在差异, 但这个差异要小于类与类之间的差异。本文将利用这个差异完成模型版权验证。换句话说, 如果给定的可疑模型能够识别恢复样本, 但无法识别遗忘样本, 那么这个模型就有很大可能被怀疑为被盗模型。具体的验证流程如下所示:

给定水印验证样本 (x_u,x_r), 本文可以通过以下方式验证可疑模型 $\hat{f}$, 首先将验证样本输入到可疑模型中:

$$\widehat{y_u}=\hat{f}(x_u,\hat{\theta}) \quad (14)$$

$$\widehat{y_r}=\hat{f}(x_r,\hat{\theta}) \quad (15)$$

它将输出验证消息 $\widehat{y_u}$ 和 $\widehat{y_r}$。然后, 本文使用第 2 节提到的辅助函数计算遗忘样本和恢复样本的准确率。遗忘样本的准确率 Acc_u 和恢复样本的准确率 Acc_r 定义如下:

$$\mathrm{Acc}_u=\mathrm{Acc}(\hat{f},D_u)=\frac{\sum_{i=1}^{M}\Pr(y_{u_i},f(x_{u_i}))}{M} \quad (16)$$

$$\mathrm{Acc}_r=\mathrm{Acc}(\hat{f},D_r)=\frac{\sum_{i=1}^{N}\Pr(y_{r_i},f(x_{r_i}))}{N} \quad (17)$$

如果遗忘样本的准确率 Acc_u 和恢复样本的准确率 Acc_r 之间的差异小于预定义的阈值 σ, $\mathrm{Verify}(\hat{f},D_u\cup D_r)$ 输出 0, 表示验证失败。否则, 它输出 1。然而, 如果水印验证对象不受限制, 这将增大对无辜模型造成误报的概率。由于被盗模型通常在功能上类似于目标模型, 本文仅对声称与目标模型 f 具有类似功能的可疑模型 $\hat{f}$ 使用水印验证。

为了更好地确定两个样本的平均准确率是否显著不同, 本文将使用 T 检验来计算阈值 σ。T 检验是一种比较从同一组或不同类别中收集的两个样本的均值或比例的统计方法。它旨在通过假设检验的方法检验数据均值和一个假设值之间是否存在差异。

$$t=\frac{|\overline{\mathrm{Acc}_u}-\overline{\mathrm{Acc}_r}|-\sigma}{\sqrt{\dfrac{(n_1-1)S_1^2+(n_2-1)S_2^2}{n_1+n_2-2}\left(\dfrac{1}{n_1}+\dfrac{1}{n_2}\right)}} \quad (18)$$

其中, n_1 和 n_2 是模型 $\hat{f}$ 的验证实验次数。S_1^2、S_2^2 是它们的样本方差。σ 是两个样本准确率之间差异的阈值。本文根据统计学领域中显著差异的定义和先前工作的设置, 将其设为 0.05。查询 T 分布表, 如果 $t>t_{(n_1+n_2-2)\sim 0.01}$, 则成功水印验证的概率将达到 99.9%。

5 实验验证及分析

在本节中, 本文首先介绍实验设置, 然后评估本文的方法在水印验证方面的保真度和水印有效性, 接着评估抵抗常见水印移除攻击(包括输入预处理、模型修改和模型提取)的鲁棒性。最后, 本文通过消融性实验评估了超参数以及水印各步骤的重要性。

5.1 实验设置

数据集和 DNN 模型: 本文的实验在三个流行的数据集(例如, CIFAR-10、CIFAR-100、TinyImageNet)

上，使用了三个流行的 DNN 模型架构，包括 ResNet-18、DenseNet-121 和 WideResNet-34 进行评估。表 1 显示了嵌入水印的目标模型在其不同任务上的原始性能。

表 1　三种目标模型在三个数据集上的原始性能

Table 1　Original Performance of Three Target Models on Three Datasets

数据集	目标模型架构		
	ResNet-18	DenseNet-121	WideResNet-34
CIFAR-10	82.00%	82.02%	91.20%
CIFAR-100	53.30%	55.70%	67.79%
TinyImageNet	44.60%	48.10%	54.00%

水印去除攻击：在鲁棒性评估中，本文评估了提出的方法对三种水印去除攻击的鲁棒性，其实现如下。(1) 输入预处理攻击包括 JPEG 压缩[36]和高斯模糊。(2) 模型修改攻击包括模型微调[21]和微调-剪枝[45]。(3) 模型提取攻击包括跨架构重训练[17]和迁移学习[60]。

基准方法：为了进行全面评估，本文使用了四个基准方法。第一个基准方法是基于后门的动态水印方法，通过后门攻击方式将触发器加入到正常数据集作为水印嵌入[16]。一项最近的研究表明，基于后门的动态水印方法在功能保留的有效性和对各种水印去除攻击的鲁棒性方面实现了最佳性能[17]。第二个基准方法来自文献[31]，该基准方法提出了三种动态的 DNN 适用水印生成算法，通过向训练集注入额外验证样本来实现。第三个基准方法是一种使用参数正则化器将水印嵌入模型参数的静态水印方法[21]。第四个基准方法[61]是一种通过嵌入专有模型 PTYNet 将水印注入到目标模型的动态水印方法。

5.2　有效性

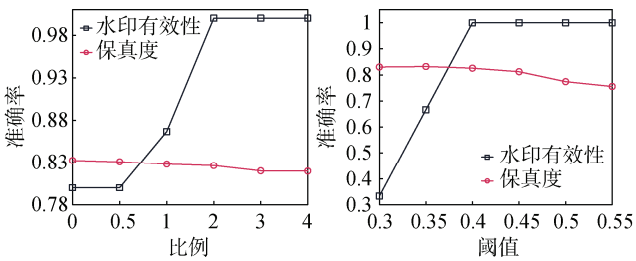
为了验证水印方法的有效性，本节通过保真度和水印有效性两个指标评估了方法在三种数据集和三个 DNN 模型架构上的性能，并与基准方法进行比较。本文首先探讨了提出的水印方法是否可以用于在黑盒设置中验证目标 DNN 模型的版权所有权，并评估了模型水印嵌入带来的保真度损失。随后，比较了水印方法与基准方法的保真度和水印有效性。

表 2 展示了在水印嵌入中确定出适当的遗忘样本和废弃样本比例的实验结果，表中的废弃样本和遗忘样本的比例表示遗忘样本和学习样本之间的比例，水印有效性表示水印验证的准确性，保真度表示模型嵌入水印后识别良性样本的性能，嵌入损失表示嵌入水印导致的对良性样本预测性能的降低，遗忘样本比例表示遗忘样本占训练集同类样本的比例。本文在固定遗忘样本的比例下通过一系列实验研究了遗忘样本和废弃样本之间的比例关系对水印结果的影响。实验结果表明，遗忘样本的数量越多，水印有效性就会提高。当废弃样本和遗忘样本的比例达到 2 且嵌入损失小于 1.9%时，本文的方法在水印验证中的准确率为 100%。图 3(a)显示，当比例大于 2 时，水印有效性和保真度保持稳定。

表 2　遗忘样本和废弃样本之间比例对水印性能的影响

Table 2　Impact of the Proportion between Unlearning Samples and Discarded Samples on Watermarking Performance

废弃样本和遗忘样本的比例	水印有效性	保真度	嵌入损失	遗忘样本比例
0	0.80	0.832	0.013	10%
0.5	0.80	0.831	0.015	10%
1	0.867	0.828	0.018	10%
2	1.0	0.826	0.019	10%
3	1.0	0.820	0.025	10%
4	1.0	0.820	0.026	10%



(a) 样本比例与水印性能的变化趋势　(b) 阈值τ与水印性能的变化趋势

图 3　采用不同比率和阈值 τ 值对水印有效性评估的变化趋势

Figure 3　Trend of Watermarking Effectiveness with Different Proportions and Threshold Values

在水印嵌入中，本文需要确定高相关样本的阈值 τ，以便在水印嵌入中进行模型遗忘的操作。表 3 展示了对水印嵌入中不同 τ 值的调查实验结果。实验结果表明，当 τ 值大于 0.4 时，遗忘样本在水印嵌入中的比例小于 11%，嵌入损失小于 2%时，水印有效性达到 100%。然而，当大量样本进行模型遗忘时(参见图 3(b))，良性样本的预测功能会受到影响。

在效果评估中，本文首先确定了遗忘样本和恢复样本的比例，以实现水印有效性和保真度保持的平衡，并确定了阈值 τ 的值，使得方法得以在保持水

表 3 阈值 τ 对水印性能的影响

Table 3 Impact of Threshold τ on Watermarking Performance

τ	水印有效性	保真度	嵌入损失	遗忘样本比例
0.30	0.333	0.830	0.015	1.42%
0.35	0.667	0.832	0.013	4.72%
0.40	1.0	0.826	0.020	10.82%
0.45	1.0	0.812	0.033	20.54%
0.50	1.0	0.774	0.072	31.86%
0.55	1.0	0.756	0.090	44.20%

印有效性时不会降低模型对良性样本的预测性能。接着, 本节设定废弃样本和遗忘样本的比率为 2, 阈值 τ 的值被设置为 0.4, 并通过该比例和阈值 τ 在不同数据集的模型中的有效性, 验证了该比例和阈值的可拓展性。表 4 展示了本文提出的水印方法在水印有效性和保真度保持方面的实验结果。水印有效性表示水印验证的准确性, 嵌入损失是嵌入水印过程中对模型预测性能的降低值, t 值表示 t 检验的结果。实验进行了三次, 并呈现了标准偏差。表 4 中的实验结果表明, 本文的方法在所有权验证方面平均准确率为 99%, 平均嵌入损失小于 1.11%。我们观察到, 本文方法在 WideResNet-34 架构上表现出更高的水印有效性, 这可能是因为该架构对模型遗忘操作的容忍度更高。此外, 在 TinyImageNet 上, 水印嵌入损失略高于 CIFAR-10, 可能是由于 TinyImageNet 的每一类的样本较少, 模型遗忘操作影响较大。

表 4 本文提出的水印方法在三个数据集三种模型架构上的有效性

Table 4 Effectiveness of the Proposed Watermarking Method on Three Models Across Three Datasets

数据集	模型架构	嵌入损失	t 值	水印有效性
CIFAR-10	ResNet-18	1.68(±0.15)%	18.10	99.90%
	DenseNet-121	0.63(±0.11)%	9.45	99.90%
	WideResNet-34	0.62(±0.22)%	7.14	99.75%
CIFAR-100	ResNet-18	0.36(±0.72)%	2.97	97.50%
	DenseNet-121	1.00(±1.12)%	4.01	99.00%
	WideResNet-34	1.30(±0.13)%	8.65	99.90%
TinyImageNet	ResNet-18	1.25(±0.40)%	4.93	99.50%
	DenseNet-121	1.54(±0.67)%	2.56	95.00%
	WideResNet-34	1.61(±0.40)%	6.39	99.75%

表 5 展示了水印方法[16,21,31]在四个不同数据集上使用三种流行的 DNN 模型进行的有效性评估性能。实验结果表明, 与基准方法相比, 本文的方法在有效性评估方面略有优势。

5.3 鲁棒性

为了验证水印方法面对水印移除攻击的鲁棒性, 本节对模型修改攻击、模型提取攻击和输入预处理攻击这三种常见类型的水印移除攻击[17]进行实验评

表 5 四种基准方法在三个数据集三种模型架构上的有效性

Table 5 Effectiveness of Four Benchmark Methods on Three Models Across Three Datasets

水印方法	数据集	模型架构	嵌入损失	水印有效性
Adi[16]	CIFAR-10	ResNet-18	0.46%	99%
		DenseNet-121	0.30%	100%
		WideResNet-34	0.22%	100%
	CIFAR-100	ResNet-18	1.16%	92%
		DenseNet-121	0.85%	95%
		WideResNet-34	1.08%	98%
	TinyImageNet	ResNet-18	0.33%	82%
		DenseNet-121	0.30%	43%
		WideResNet-34	0.10%	26%
Uchida[21]	CIFAR-10	ResNet-18	1.06%	99%
		DenseNet-121	0.99%	100%
		WideResNet-34	0.55%	100%
	CIFAR-100	ResNet-18	0.79%	100%
		DenseNet-121	1.40%	100%
		WideResNet-34	1.37%	100%
	TinyImageNet	ResNet-18	1.24%	96%
		DenseNet-121	0.90%	100%
		WideResNet-34	4.05%	100%

续表

水印方法	数据集	模型架构	嵌入损失	水印有效性
基于内容触发器的水印方法[31]	CIFAR-10	ResNet-18	1.58%	100%
		DenseNet-121	0.42%	100%
		WideResNet-34	0.25%	100%
	CIFAR-100	ResNet-18	4.42%	100%
		DenseNet-121	6.17%	|100%
		WideResNet-34	0.17%	100%
	TinyImageNet	ResNet-18	3.30%	100%
		DenseNet-121	2.42%	100%
		WideResNet-34	0.66%	100%
基于噪声触发器的水印方法[31]	CIFAR-10	ResNet-18	1.35%	100%
		DenseNet-121	0.52%	100%
		WideResNet-34	0.23%	100%
	CIFAR-100	ResNet-18	5.21%	100%
		DenseNet-121	6.51%	100%
		WideResNet-34	0.35%	100%
	TinyImageNet	ResNet-18	2.44%	100%
		DenseNet-121	2.79%	100%
		WideResNet-34	4.03%	100%
基于分布外样本触发器的水印方法[31]	CIFAR-10	ResNet-18	0.41%	100%
		DenseNet-121	0.87%	100%
		WideResNet-34	0.12%	100%
	CIFAR-100	ResNet-18	0.55%	100%
		DenseNet-121	0.43%	100%
		WideResNet-34	0.33%	100%
	TinyImageNet	ResNet-18	0.41%	75%
		DenseNet-121	0.67%	96%
		WideResNet-34	0.40%	100%
PTY[61]	CIFAR-10	ResNet-18	0.99%	100%
		DenseNet-121	0.46%	100%
		WideResNet-34	0.64%	100%
	CIFAR-100	ResNet-18	1.12%	100%
		DenseNet-121	1.19%	100%
		WideResNet-34	1.68%	100%
	TinyImageNet	ResNet-18	0.03%	100%
		DenseNet-121	0.09%	100%
		WideResNet-34	0.13%	100%

估, 通过测定水印移除攻击后的模型水印有效性, 并将其与基准方法进行比较, 说明了水印方法对该水印移除攻击卓越的鲁棒性。本文在 CIFAR-10 和 CIFAR-100 数据集上使用 ResNet-18、WideResNet-34 和 DenseNet-121 模型进行了实验。

(1) 输入预处理攻击通过在将数据样本输入目标模型之前修改数据样本的方式去除水印。为了验证水印方法对输入预处理攻击的鲁棒性, 本文评估了提出的水印方法抵御输入压缩和高斯模糊攻击后的水印有效性, 并将其与基准方法进行比较。图 4 表明在面对输入压缩攻击时, 所提水印方法在七种不同输入压缩率中的水印有效性平均达到99.9%。然而, 其余动态水印方法基准方法达到最佳性能的 Unrelated 只有接近 75.6%, 鲁棒性远不如本文中的水印验证方法。图 5 显示本章水印方法能够抵御高斯模糊攻击, 水印有效性的平均性能下降不到 0.1%, 并在五种不同的高斯模糊核大小下均优于动态水印方法的基准方法。

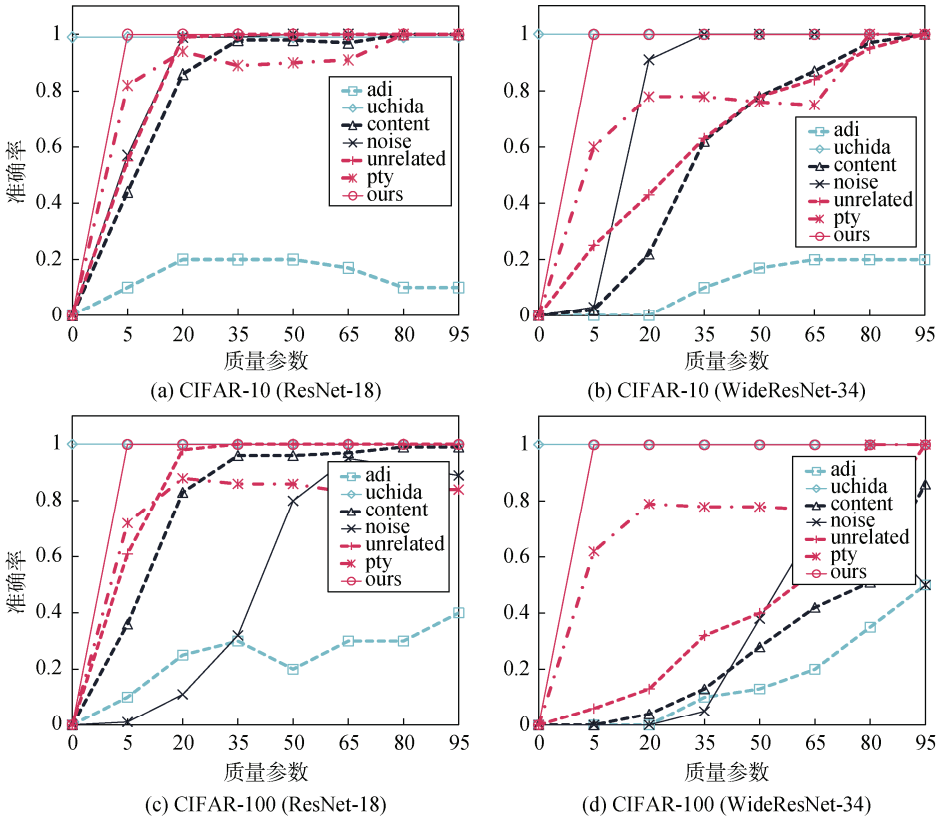


图 4　对抗 JPEG 压缩的鲁棒性评估

Figure 4　Robustness Evaluation Against JPEG Compression

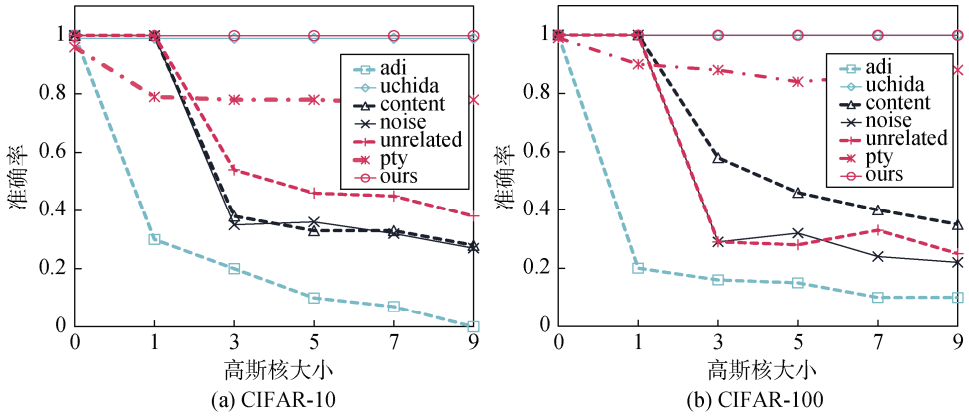


图 5　对抗高斯模糊的鲁棒性评估

Figure 5　The Robustness Evaluation Against Gaussian Blur

(2) 模型修改攻击通过修改模型参数去除水印。为了验证水印方法对模型修改攻击的鲁棒性, 本文评估了水印方法在抵御模型微调和微调-剪枝之后的水印有效性, 并与基准方法进行比较。在模型微调中, 本文采用了一种流行的微调策略, 即全层微调 (Fine-Tuning All Layer, FTAL), 其中所有层的权重都将参与微调训练。图 6(a)(b) 中的实验结果表明, 本文的方法在模型微调中的性能下降最少, 平均水印有效性下降不到 1.87%, 超过了动态水印基准方法。

微调-剪枝结合了模型剪枝和微调, 通过修剪休眠神经元去除水印映射, 然后对模型进行微调以保留功能。如图 7 所示, 本文的方法在面对微调-剪枝时表现良好, 水印有效性下降不到 0.1%, 而其他部分方法则经历了显著的性能下降。

(3) 模型提取攻击

模型提取攻击旨在训练一个替代模型, 将目标模型的知识转移到代替模型中, 达到去除水印的目的。为了验证水印方法对模型提取攻击的鲁棒性, 本

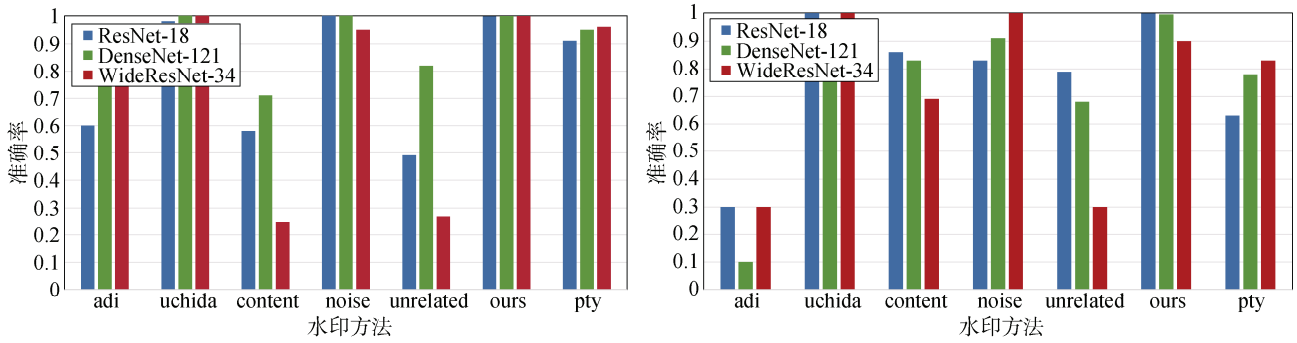


图 6 对抗微调攻击的鲁棒性评估

Figure 6 Robustness Evaluation Against Fine-tuning Attack

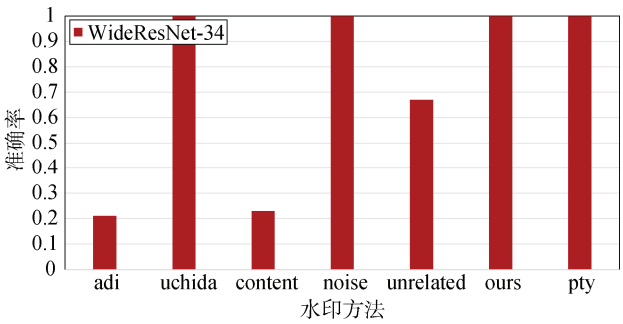


图 7 对抗微调-剪枝的鲁棒性评估

Figure 7 Robustness Evaluation Against Fine-tuning and Pruning Attack

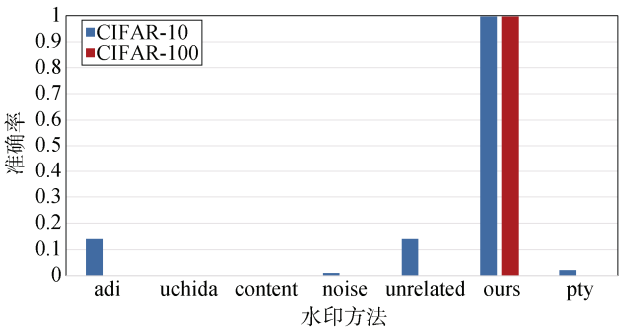


图 8 对抗跨架构重训练的鲁棒性评估

Figure 8 Robustness Evaluation Against Cross-architecture Retraining Attack

文评估了水印方法在抵御跨架构重新训练和迁移学习攻击之后的水印有效性，并与基准方法进行比较。

跨架构重训练通过使用与源模型不同的DNN模型架构，通过 API 访问源模型来训练替代模型。图 8 展示了在DenseNet上重新训练替代模型的实验结果，其中目标水印模型的架构是 WideResNet-34，数据集是 CIFAR-10 和 CIFAR-100。本文提出的方法在获取的训练数据集的各个部分中进行水印验证时没有错误。然而，跨架构重新训练误导了所有基准方法，最佳基准方法的性能在面对此攻击时也不足 7%，几乎使得水印完全无效，而本文提出的方法却仍保持着99.75%的水印有效性远超所有的基准方法。

迁移学习类似于对水印模型使用分布外数据集进行微调，并通过不同数据集上生成替代模型去除水印。在实验中，本文将在 CIFAR-100 上使用 ResNet-18、WideResNet-34 和 DenseNet-121 训练的目标水印模型迁移到 CIFAR-10。图 9 中的实验结果显示，本文的方法在水印验证方面优于基准方法。在模型迁移学习中，本文的方法在水印验证方面的性能下降最少，平均水印有效性仍有 91.6%，远远超过了最佳基准方法的 47.33%。

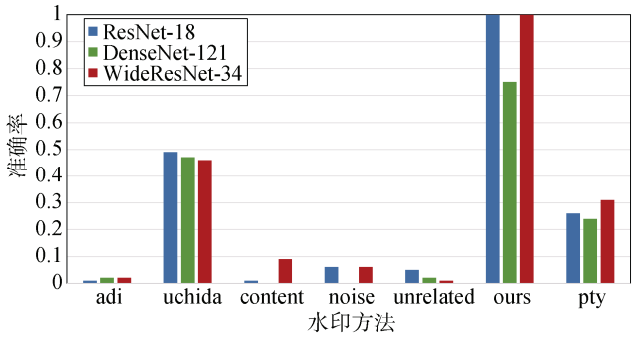


图 9 对抗迁移学习的鲁棒性评估

Figure 9 Robustness Evaluation Against Transfer Learning Attack

总的来说，实验结果表明本文的方法在抵御现有的三种常见水印去除攻击(如输入预处理攻击、模型修改攻击和模型提取攻击)方面的鲁棒性优于基准方法。实验结果还显示，现有的水印方法并没有准备好如何应对水印去除攻击的威胁。

5.4 消融实验

为了研究方法中超参数和各步骤对水印方法的影响，本节评估了调整超参数和消除每个步骤对整体水印有效性的影响。在消融研究中，本文对以下四

个研究内容进行了广泛的实验: (1) 调整超参数 β 的值。(2) 采用遗忘样本选择策略的有效性。(3) 采用抹除遗忘样本策略的有效性。(4) 采用重新学习恢复样本策略的有效性。

(1) 超参数 β 的有效性

为了研究超参数 β 的有效性, 本文评估了不同超参数 β 下, 水印方法的有效性。表 6 中的实验结果表明, 较大的 β 在水印验证中提供了较高的准确性, 并且在 β 大于 1.0 时趋于稳定。随着超参数 β 的增加, 水印的有效性会增加, 而模型的保真度会降低。因此, 本文在有效性和保真度之间取得平衡, 并最终选择 1.0。

表 6 超参数 β 对水印性能的影响

Table 6 Impact of Hyperparameters on Watermarking Performance

β	水印有效性	保真度
0.5	0.9333	0.836
1.0	1.0	0.834
1.5	1.0	0.829
2.0	1.0	0.826

(2) 采用遗忘样本选择策略的有效性

为了研究采用遗忘样本选择策略的有效性, 本文评估了所提水印方法移除样本选择步骤后的有效性。移除样本选择步骤后, 本文使用随机样本选择策略进行替代, 选择相同数量的随机样本作为遗忘样本。然后, 在整个数据集上进行水印嵌入过程, 并将结果与完整方法进行比较, 以了解本文方法中这一初始步骤的重要性。表 7 显示, 当从训练数据集中随机选择未学习样本时, 本文提出的方法在水印验证中的平均准确性低于 20%[62]。这证明了本文方法提出的样本选择策略的有效性。这主要是因为在数据集的类中, 样本具有很高的相似性。简单地去遗忘一个样本不能改变决策边界, 其他高相关样本也具有遗忘样本相关的知识。因此, 本文的样本选择方法是必不可少的。

表 7 遗忘样本选择策略对水印性能的影响

Table 7 Impact of Unlearning Sample Selection Strategy on Watermarking Performance

训练轮次	水印有效性	保真度
10	0.166	0.825
20	0.222	0.827
30	0.185	0.834
40	0.196	0.831

(3) 采用抹除遗忘样本策略的有效性

为了研究所采用遗忘样本选择策略的有效性, 本文评估了水印方法移除抹除遗忘样本策略后的有效性。当移除该步骤时, 先前训练的 DNN 模型保留其原始权重, 没有经历遗忘过程。在完成样本选择后, 仅执行重新学习恢复样本操作, 而忽略抹除遗忘样本操作。表 8 显示, 随着训练轮次的增加, 模型的保真度将略高于 80%, 并且保持稳定。而水印有效性将在初始训练轮次内缓慢增加, 并在之后在 20%和 40%之间波动。这表明, 仅保留低相关样本不能使模型完全忘记遗忘样本, 并展示了抹除遗忘样本策略的有效性。

表 8 抹除遗忘样本策略对水印性能的影响

Table 8 Impact of Unlearning Sample Erasure Strategy on Watermarking Performance

训练轮次	水印有效性	保真度
0	0	0.8148
10	0.1	0.8253
20	0.2	0.827
30	0.1	0.8368
40	0.1	0.8303

(4) 采用重新学习恢复样本策略的有效性

为了研究采用重新学习恢复样本策略的有效性, 本文评估了本章水印方法移除重新学习恢复样本策略后的有效性。在这个实验中, 本文探讨了在水印嵌入后不重新学习恢复样本的影响。在完成样本选择后, 仅执行抹除遗忘样本操作, 而忽略重新学习恢复样本操作。表 9 显示, 随着训练轮次的增加, 水印有效性将迅速增加到 100%, 而模型的保真度将迅速下降。这主要是因为灾难性遗忘, 这意味着虽然遗忘样本被抹除, 但模型功能的退化可能是指数级的。这表明, 仅抹除遗忘样本将导致模型功能的指数级退化, 并显示了重新学习恢复样本策略的有效性。

表 9 重新学习恢复样本对水印性能的影响

Table 9 Impact of Re-learning Recovery Samples on Watermarking Performance

训练轮次	水印有效性	保真度
0	0	0.8146
1	0	0.7362
2	0.9	0.6192
3	1.0	0.5038
4	1.0	0.3381

本节对水印方法进行了消融性实验研究, 以评

估调整超参数和消除每个步骤对水印性能的影响。研究包括调整超参数 β、采用遗忘样本选择策略的有效性、采用抹除遗忘样本策略的有效性、采用重新学习恢复样本策略的有效性。实验结果表明, 超参数 β 在 1.0 左右时能够在水印验证中取得较高准确性并保持稳定。采用遗忘样本选择策略可以有效提高水印验证的准确性, 而不采用抹除遗忘样本策略会导致模型功能的指数级退化。采用重新学习恢复样本策略可以提高水印验证的效果, 并在一定程度上保持模型的功能性。整体实验结果表明, 各个步骤在水印性能中发挥着重要作用, 对于保证水印的有效性和模型的功能性具有重要意义。

6 分析与讨论

尽管本文提出的基于模型遗忘的深度神经网络水印方法在鲁棒性和水印移除攻击抵抗能力上表现出色, 但仍存在一些局限性值得进一步讨论。

首先是水印的容量问题, 本文水印方法嵌入的水印容量有限, 由于消除的原有数据集的映射, 因此水印容量与原有数据集大小和遗忘样本与恢复样本之间的类内距离差异相关。根据表 3 实验显示水印容量一般为数据集一类样本的 10%。因此在较小的数据集中, 由于容量的限制, 水印的嵌入信息相对较少, 可能无法承载复杂的验证信息或多次嵌入不同的水印。

其次则是多次验证的安全性, 在实际应用中, 模型所有者可能需要进行多次水印验证, 特别是在长期使用模型的过程中。尽管本文使用的水印样本均为原有数据集的样本, 与寻常输入无异。然而, 频繁地对模型进行验证可能导致水印泄露风险。攻击者可能通过反复查询验证样本, 可能推测出水印嵌入的机制或特征。因此, 如何确保多次验证过程的安全性, 避免因过度验证而导致水印被推测或移除, 是一个值得进一步研究的方向。

这些局限性提示了本文方法在实际应用中的挑战, 未来的工作可以在提升水印容量、多次验证的安全性上进行进一步研究。

7 结论

水印移除攻击通常会通过一定手段抹除与目标模型不符的映射, 基于添加验证样本映射的水印范式很难应对有意针对水印映射进行削弱、删除或净化的水印移除攻击, 因此现有技术存在鲁棒性不足的问题难以应对多种水印移除攻击带来的安全威胁, 这也导致在实际应用中存在着大量风险与安全隐患的问题。针对现有水印方法鲁棒性不足, 本文提出了一种基于模型遗忘的深度神经网络鲁棒性水印方法。与传统的基于添加水印映射的水印范式相比, 本文通过抹除目标模型中的映射来嵌入水印。水印移除攻击一般通过移除添加的水印映射抹除水印, 但在无法访问目标模型数据集和其他资源的情况下很难引入新的映射, 因此通过抹除目标模型中的映射作为水印能在面对水印移除攻击时具有强大的鲁棒性。本文首先定义了样本相似度, 并利用基于样本相似度的样本选择算法筛选出需要遗忘的映射。然后通过基于梯度上升的模型遗忘方法抹除目标模型中的映射, 最后利用验证样本中遗忘样本和恢复样本在可疑模型中的正确率之差完成水印验证, 以此实现一个对现有水印移除攻击均有鲁棒性的水印方法。本文的方法在三个数据集、三种模型架构上对现有的主流水印移除攻击进行评估, 均取得极高的鲁棒性。

参考文献

[1] Dong S, Wang P, Abbas K. A Survey on Deep Learning and Its Applications[J]. *Computer Science Review*, 2021, 40: 100379.

[2] Brown T B, Mann B, Ryder N, et al. Language Models Are Few-Shot Learners[EB/OL]. 2020: arXiv: 2005.14165. https:// arxiv.org/abs/2005.14165.

[3] Krishna K, Tomar G S, Parikh A P, et al. Thieves on Sesame Street! Model Extraction of BERT-Based APIs[EB/OL]. 2019: arXiv: 1910.12366. https://arxiv.org/abs/1910.12366.

[4] Oliynyk D, Mayer R, Rauber A. I Know What You Trained last Summer: A Survey on Stealing Machine Learning Models and Defences[EB/OL]. 2022: arXiv: 2206.08451. https://arxiv.org/abs/2206.08451.

[5] Peng Z R, Li S F, Chen G X, et al. Fingerprinting Deep Neural Networks Globally *via* Universal Adversarial Perturbations[C]. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022: 13420-13429.

[6] Zhao J J, Hu Q Y, Liu G Y, et al. AFA: Adversarial Fingerprinting Authentication for Deep Neural Networks[J]. *Computer Communications*, 2020, 150: 488-497.

[7] Zheng Y, Wang S, Chang C H. A DNN Fingerprint for Non-Repudiable Model Ownership Identification and Piracy Detection[J]. *IEEE Transactions on Information Forensics and Security*, 2022, 17: 2977-2989.

[8] Yang K, Wang R, Wang L N. MetaFinger: Fingerprinting the Deep Neural Networks with Meta-Training[C]. *The Thirty-First International Joint Conference on Artificial Intelligence*, 2022: 776-782.

[9] Chen J L, Wang J Y, Peng T L, et al. Copy, Right? A Testing Framework for Copyright Protection of Deep Learning Models[C]. *2022 IEEE Symposium on Security and Privacy*, 2022: 824-841.

[10] Li X, Deng T P, Xiong J B, et al. Federated Learning Watermark

Based on Model Backdoor[J]. *Journal of Software*, 2024, 35(7): 3454-3468.
(李璇, 邓天鹏, 熊金波, 等. 基于模型后门的联邦学习水印[J]. *软件学报*, 2024, 35(7): 3454-3468.)

[11] Guo S, Zhang T, Qiu H, et al. Fine-tuning is not Enough: A Simple Yet Effective Watermark Removal Attack for Dnn Models[M]. [C]. Proceedings of the 30th International Joint Conference on Artificial Intelligence, 2021: 3640-3646. 2021.

[12] Lao Y J, Zhao W J, Yang P, et al. DeepAuth: A DNN Authentication Framework by Model-Unique and Fragile Signature Embedding[J]. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022, 36(9): 9595-9603.

[13] Li W, Zhang X Y, Lin S, et al. Chameleon DNN Watermarking: Dynamically Public Model Ownership Verification[M]. Information Security Applications. Cham: Springer Nature Switzerland, 2023: 344-356.

[14] Wang T H, Kerschbaum F. RIGA: Covert and Robust White-Box Watermarking of Deep Neural Networks[C]. *The Web Conference 2021*, 2021: 993-1004.

[15] Yasui T, Tanaka T, Malik A, et al. Coded DNN Watermark: Robustness Against Pruning Models Using Constant Weight Code[J]. *Journal of Imaging*, 2022, 8(6): 152.

[16] Adi Y, Baum C, Cisse M, et al. 2018. Turning Your Weakness into a Strength: Watermarking Deep Neural Networks by Backdooring[C]. 27th USENIX Security Symposium (USENIXSecurity 18). 1615-1631.

[17] Lukas N, Jiang E, Li X D, et al. SoK: How Robust Is Image Classification Deep Neural Network Watermarking? [C]. *2022 IEEE Symposium on Security and Privacy*, 2022: 787-804.

[18] Bevan P J, Atapour-Abarghouei A. Skin Deep Unlearning: Artefact and Instrument Debiasing in the Context of Melanoma Classification[EB/OL]. 2021: arXiv: 2109.09818. https://arxiv.org/abs/2109.09818.

[19] Bourtoule L, Chandrasekaran V, Choquette-Choo C A, et al. Machine Unlearning[C]. *2021 IEEE Symposium on Security and Privacy*, 2021: 141-159.

[20] Li Y, Wang H X, Barni M. A Survey of Deep Neural Network Watermarking Techniques[J]. *Neurocomputing*, 2021, 461: 171-193.

[21] Uchida Y, Nagai Y, Sakazawa S, et al. Embedding Watermarks into Deep Neural Networks[C]. *The 2017 ACM on International Conference on Multimedia Retrieval*, 2017: 269-277.

[22] Chen H L, Rohani B D, Koushanfar F. DeepMarks: A Digital Fingerprinting Framework for Deep Neural Networks[EB/OL]. 2018: arXiv: 1804.03648. https://arxiv.org/abs/1804.03648.

[23] Liu H, Weng Z, Zhu Y. Watermarking Deep Neural Networks with Greedy Residuals[C]. *ICML*, 2021: 6978-6988.

[24] Nagai Y, Uchida Y, Sakazawa S, et al. Digital Watermarking for Deep Neural Networks[J]. *International Journal of Multimedia Information Retrieval*, 2018, 7(1): 3-16.

[25] Wang T H, Kerschbaum F. Attacks on Digital Watermarks for Deep Neural Networks[C]. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019: 2622-2626.

[26] Jia H R, Yaghini M, Choquette-Choo C A, et al. Proof-of-Learning: Definitions and Practice[C]. *2021 IEEE Symposium on Security and Privacy*, 2021: 1039-1056.

[27] Guo J, Potkonjak M. Watermarking Deep Neural Networks for Embedded Systems[C]. *The International Conference on Computer-Aided Design*, 2018: 1-8.

[28] Le Merrer E, Pérez P, Trédan G. Adversarial Frontier Stitching for Remote Neural Network Watermarking[J]. *Neural Computing and Applications*, 2020, 32(13): 9233-9244.

[29] Li Z, Hu C Y, Zhang Y, et al. How to Prove Your Model Belongs to You: A Blind-Watermark Based Framework to Protect Intellectual Property of DNN[C]. *The 35th Annual Computer Security Applications Conference*, 2019: 126-137.

[30] Yang P, Lao Y J, Li P. Robust Watermarking for Deep Neural Networks *via* Bi-Level Optimization[C]. *2021 IEEE/CVF International Conference on Computer Vision*, 2021: 14821-14830.

[31] Zhang J L, Gu Z S, Jang J, et al. Protecting Intellectual Property of Deep Neural Networks with Watermarking[C]. *The 2018 on Asia Conference on Computer and Communications Security*, 2018: 159-172.

[32] Namba R, Sakuma J. Robust Watermarking of Neural Network with Exponential Weighting[C]. *The 2019 ACM Asia Conference on Computer and Communications Security*, 2019: 228-240.

[33] Darvish Rouhani B, Chen H L, Koushanfar F. DeepSigns: An End-to-End Watermarking Framework for Ownership Protection of Deep Neural Networks[C]. *The Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, 2019: 485-497.

[34] Szyller S, Atli B G, Marchal S, et al. DAWN: Dynamic Adversarial Watermarking of Neural Networks[C]. *The 29th ACM International Conference on Multimedia*, 2021: 4417-4425.

[35] Lin W A, Balaji Y, Samangouei P, et al. Invert and Defend: Model-Based Approximate Inversion of Generative Adversarial Networks for Secure Inference[EB/OL]. 2019: arXiv: 1911.10291. https://arxiv.org/abs/1911.10291.

[36] Dziugaite G K, Ghahramani Z, Roy D M. A Study of the Effect of JPG Compression on Adversarial Images[EB/OL]. 2016: arXiv: 1608.00853. https://arxiv.org/abs/1608.00853.

[37] Lin J, Gan C, Han S. Defensive Quantization: When Efficiency Meets Robustness[EB/OL]. 2019: arXiv: 1904.08444. https:// arxiv.org/abs/1904.08444.

[38] Xu W L, Evans D, Qi Y J. Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks[EB/OL]. 2017: arXiv: 1704.01155. https://arxiv.org/abs/1704.01155.

[39] Zantedeschi V, Nicolae M I, Rawat A. Efficient Defenses Against Adversarial Attacks[C]. *The 10th ACM Workshop on Artificial Intelligence and Security*, 2017: 39-49.

[40] NG A, et al. Sparse Autoencoder[J]. *CS294A Lecture notes*, 2011: 72(2011): 1-19.

[41] Wang R, Li H X, Mu L Z, et al. Rethinking the Vulnerability of DNN Watermarking: Are Watermarks Robust Against Natural-

ness-Aware Perturbations? [C]. *The 30th ACM International Conference on Multimedia*, 2022: 1808-1818.

[42] Madry A, Makelov A, Schmidt L, et al. Towards Deep Learning Models Resistant to Adversarial Attacks[EB/OL]. 2017: arXiv: 1706.06083. https://arxiv.org/abs/1706.06083.

[43] Hubara I, Courbariaux M, Soudry D, et al. Quantized Neural Networks: Training Neural Networks with Low Precision Weights and Activations[C]. *New York: ACM*, 2017: 6869-6898.

[44] Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the Inception Architecture for Computer Vision[C]. *2016 IEEE Conference on Computer Vision and Pattern Recognition*, 2016: 2818-2826.

[45] Liu K, Dolan-Gavitt B, Garg S. Fine-Pruning: Defending Against Backdooring Attacks on Deep Neural Networks[C]. *Research in Attacks, Intrusions, and Defenses*, 2018: 273-294.

[46] Zhu M, Gupta S. To Prune, or Not to Prune: Exploring the Efficacy of Pruning for Model Compression[EB/OL]. 2017: arXiv: 1710.01878. https://arxiv.org/abs/1710.01878.

[47] Wang B L, Yao Y S, Shan S, et al. Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks[C]. *2019 IEEE Symposium on Security and Privacy*, 2019: 707-723.

[48] Shafieinejad M, Lukas N, Wang J Q, et al. On the Robustness of Backdoor-Based Watermarking in Deep Neural Networks[C]. *The 2021 ACM Workshop on Information Hiding and Multimedia Security*, 2021: 177-188.

[49] Hinton G, Vinyals O, Dean J. Distilling the Knowledge in a Neural Network[EB/OL]. 2015: arXiv: 1503.02531. https://arxiv.org/abs/1503.02531.

[50] Tramèr F, Zhang F, Juels A, et al. Stealing Machine Learning Models via Prediction {APIs}[C]. *25th USENIX security symposium*, 2016: 601-618.

[51] Orekondy T, Schiele B, Fritz M. Knockoff Nets: Stealing Functionality of Black-Box Models[C]. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019: 4949-4958.

[52] Nguyen T T, Huynh T T, Ren Z, et al. A Survey of Machine Unlearning[EB/OL]. 2022: arXiv: 2209.02299. https://arxiv.org/abs/2209.02299.

[53] Brophy J, Lowd D. Machine Unlearning for Random Forests[C]. *International Conference on Machine Learning*, 2021: 1092-1104.

[54] Thudi A, Deza G, Chandrasekaran V, et al. Unrolling SGD: Understanding Factors Influencing Machine Unlearning[C]. *2022 IEEE 7th European Symposium on Security and Privacy*, 2022: 303-319.

[55] Graves L, Nagisetty V, Ganesh V. Amnesiac Machine Learning[J]. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, 35(13): 11516-11524.

[56] Guo C, Goldstein T, Hannun A, et al. Certified Data Removal from Machine Learning Models[EB/OL]. 2019: arXiv: 1911.03030. https://arxiv.org/abs/1911.03030.

[57] Golatkar A, Achille A, Ravichandran A, et al. Mixed-Privacy Forgetting in Deep Networks[C]. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021: 792-801.

[58] LeCun Y, Bengio Y, Hinton G. Deep Learning[J]. *Nature*, 2015, 521(7553): 436-444.

[59] Peng S, Chen Y F, Xu J, et al. Intellectual Property Protection of DNN Models[J]. *World Wide Web*, 2023, 26(4): 1877-1911.

[60] Torrey L, Shavlik J. 2010. Transfer Learning[M]. Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques. IGI global: 242-264.

[61] Wang R, Ren J X, Li B H, et al. Free Fine-Tuning: A Plug-and-Play Watermarking Scheme for Deep Neural Networks[C]. *The 31st ACM International Conference on Multimedia*, 2023: 8463-8474.

[62] Cao X Y, Jia J Y, Gong N Z. IPGuard: Protecting Intellectual Property of Deep Neural Networks *via* Fingerprinting the Classification Boundary[C]. *The 2021 ACM Asia Conference on Computer and Communications Security*, 2021: 14-25.

任纪星 于 2021 年在武汉大学信息安全专业获得工学学士学位。现在武汉大学网络空间安全专业攻读硕士学位。研究领域为人工智能安全。研究兴趣包括: 深度神经网络版权保护。Email: 2017301500283@whu.edu.cn

李勃衡 现在武汉大学信息安全专业攻读学士学位。研究领域为可信机器学习、人工智能安全。Email: boheng.li@whu.edu.cn

汪润 于 2018 年在武汉大学信息安全专业获得博士学位。现任武汉大学国家网络安全学院副教授, 博士生导师。研究兴趣包括: 人工智能安全、软件与系统安全。Email: wangrun@whu.edu.cn

许葳 于 2023 年在上海交通大学工业工程专业获得学士学位。现在武汉大学网络空间安全专业攻读硕士学位。研究领域为人工智能安全。Email: xuwei233@sjtu.edu.cn

张钰洋 于 2023 年在武汉大学信息安全专业获得学士学位。现在武汉大学网络空间安全专业攻读硕士学位。研究领域为人工智能安全。Email: yuwanz@whu.edu.cn

王丽娜 于 2001 年在东北大学获得博士学位。现任武汉大学二级教授, 博士生导师, 空天信息安全与可信计算教育部重点实验室主任。研究领域为软件与系统安全、隐写分析和人工智能安全。Email: lnwang@whu.edu.cn