

中图分类号:

学校代码: 10055

UDC:

密级: 公开

南开大学
硕士 学位 论文

一种基于近边界数据的模型所有权推断方法研究

Research on Model Ownership Inference Based on

Near-boundary Data

论文作者 杨宗稳 指导教师 蒲凌君副教授

申请学位 工学硕士 培养单位 南开大学

学科专业 计算机科学与技术 研究方向 模型的知识产权保护

答辩委员会主席 评 阅 人

南开大学研究生院

二〇二三年四月

南开大学学位论文使用授权书

本人完全了解《南开大学关于研究生学位论文收藏和利用管理办法》关于南开大学(简称“学校”)研究生学位论文收藏和利用的管理规定，同意向南开大学提交本人的学位论文电子版及相应的纸质本。

本人了解南开大学拥有在《中华人民共和国著作权法》规定范围内的学位论文使用权，同意在以下几方面向学校授权。即：

1. 学校将学位论文编入《南开大学博硕士学位论文全文数据库》，并作为资料在学校图书馆等场所提供阅览，在校园网上提供论文目录检索、文摘及前16页的浏览等信息服务；
2. 学校可以采用影印、缩印或其他复制手段保存学位论文；学校根据规定向教育部指定的收藏和存档单位提交学位论文；
3. 非公开学位论文在解密后的使用权同公开论文。

本人承诺：本人的学位论文是在南开大学学习期间创作完成的作品，并已通过论文答辩；提交的学位论文电子版与纸质本论文的内容一致，如因不同造成不良后果由本人自负。

本人签署本授权书一份（此授权书为论文中一页），交图书馆留存。

学位论文作者暨授权人(亲笔)签字：_____

20 年 月 日

南开大学研究生学位论文作者信息

论 文 题 目	一种基于近边界数据的模型所有权推断方法研究				
姓 名	杨宗稳	学号	2120200439	答 辩 期 间	
论 文 类 别	博士 <input type="checkbox"/> 学历硕士 <input checked="" type="checkbox"/> 专业学位硕士 <input type="checkbox"/> 同等学力硕士 <input type="checkbox"/> 划 <input checked="" type="checkbox"/> 选择				
学院(单位)	计算机学院		学科/专业(专业学位)名称	计算机科学与技术	
联 系 电 话	13102257615		电子邮箱	yzw@mail.nankai.edu.cn	
通讯地址(邮编): 300000					
非公开论文编号			备注		

注：本授权书适用我校授予的所有博士、硕士的学位论文。如已批准为非公开学位论文，须向图书馆提供批准通过的《南开大学研究生申请非公开学位论文审批表》复印件和“非公开学位论文标注说明”页原件。

南开大学学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师指导下进行研究工作所取得的研究成果。除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人创作的、已公开发表或者没有公开发表的作品的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。本学位论文原创性声明的法律责任由本人承担。

学位论文作者签名：_____ 年 月 日

非公开学位论文标注说明

(本页表中填写内容须打印)

根据南开大学有关规定，非公开学位论文须经指导教师同意、作者本人申请和相关部门批准方能标注。未经批准的均为公开学位论文，公开学位论文本说明为空白。

论文题目			
申请密级	<input type="checkbox"/> 限制 (≤2 年) <input type="checkbox"/> 秘密 (≤10 年) <input type="checkbox"/> 机密 (≤20 年)		
保密期限	20	年	月
审批表编号		批准日期	20

南开大学学位评定委员会办公室盖章(有效)

注：限制 ★2 年 (可少于 2 年); 秘密 ★10 年 (可少于 10 年); 机密 ★20 年 (可少于 20 年)

摘要

深度神经网络 (Deep Neural Network, DNN) 因其优越的性能已经应用于科学的研究和日常生活的方方面面，给人们的生活带来了极大地便利。一个高性能 DNN 模型离不开复杂的结构设计，覆盖全面的数据集和昂贵的计算资源以及漫长的训练时间，所以 DNN 模型具有很高的商业价值和知识产权价值。

近年来，模型盗窃行为时常出现，不法分子对 DNN 模型非法复制，派生和发布的行为都严重侵犯了模型所有者的知识产权。这引起人们逐渐重视模型的知识产权保护，许多研究者受到传统数字媒体水印的启发，从而设计模型水印和指纹用于验证模型所有权。然而，歧义性声明等攻击手段被用于破解模型水印和指纹，这对模型所有权验证工作造成了挑战。因此，本文提出了一种近边界数据作为获得模型所有权的证据，并创新性地提出了推断模型所有权而不是验证模型所有权。本文提出采用对抗性样本生成算法和生成对抗网络构造私有化的近边界数据，我们主要的观察结果是近边界数据在源模型和其衍生的盗窃模型中均表现出靠近分类边界的结果，模型所有者可以使用近边界数据作为 DNN 模型知识产权保护的技术支持。本文的主要工作如下：

- 1) 揭示了当前 DNN 模型所有权验证方案的脆弱性并确认了数据驱动推断模型所有权的有效性。
- 2) 提出了利用对抗性样本构造近边界数据以抵御模型窃取攻击。
- 3) 设计了基于生成对抗网络的近边界数据生成器和提出了一种损失函数用以微调源模型的目标分类边界，增加推断模型所有权的置信度。
- 4) 基于 ResNet18 和三个公开数据集进行了广泛的实验，实验结果证明了本文提出的近边界数据在推断模型所有权上的显著效果。

关键词： 知识产权保护；所有权推断；近边界数据；深度神经网络；生成对抗网络

Abstract

This is the abstract.

Key Words: Intellectual property protection; Ownership inference; Near-boundary data; Deep neural network; Generative adversarial network

目录

摘要	I
Abstract	II
第一章 绪论	1
第一节 研究背景与意义	1
第二节 相关研究现状	2
第三节 本文主要工作	5
第四节 本文组织架构	6
第二章 技术背景	8
第一节 深度神经网络及相关术语	8
第二节 对抗性攻击	9
2.2.1 对抗性样本	9
2.2.2 对抗性攻击的类别	10
第三节 生成对抗网络	11
第四节 深度神经网络的模型窃取攻击	12
2.4.1 模型修改攻击	12
2.4.2 删减攻击	13
2.4.3 主动攻击	13
第五节 深度神经网络模型的知识产权保护	13
2.5.1 模型水印	14
2.5.2 模型指纹	14
第六节 本章小结	15
第三章 基于生成对抗网络特征提取的近边界数据研究	17
第一节 近边界对抗性样本	17
第二节 生成近边界对抗性样本	18
第三节 近边界数据私有化	22
第四节 本章小结	25

目录

第四章 基于近边界数据的模型所有权推断方法研究	26
第一节 理论驱动	26
4.1.1 所有权验证局限性	26
4.1.2 利用数据推断模型所有权	27
第二节 近边界数据推断模型所有权	28
4.2.1 设计目标	28
4.2.2 方法概述	30
4.2.3 假设检验	30
第三节 本章小结	31
第五章 基于近边界数据的模型所有权推断方法分析	33
第一节 实验设置	33
5.1.1 数据集	33
5.1.2 目标模型	33
5.1.3 实验环境和参数设置	34
第二节 生成初始近边界数据的算法选择	35
第三节 数据近边界特性的评估	36
第四节 微调目标分类边界的影响	38
第五节 推断模型所有权的有效性	40
第六节 不同规模近边界数据的可伸缩性扩展	42
第七节 本章小结	43
第六章 总结与展望	45
第一节 工作总结	45
第二节 工作展望	45
参考文献	47
图索引	51
表索引	52
致谢	53
个人简历	54

第一章 绪论

第一节 研究背景与意义

近年来，科技迅速发展，计算资源日益丰富，计算能力得到显著提升，我们正在进入人工智能 (Artificial Intelligence, AI)^[1] 的时代。随着互联网的快速发展，产生了海量的数据，得益于深度神经网络 (Deep Neural Network, DNN)^[2] 强大的数据处理能力，DNN 已经成为应用最为广泛的人工智能方法之一。自 DNN 在计算机视觉^[3-5]，语音识别^[6]，自然语言处理^[7-9] 等领域取得突破性应用以来，DNN 的应用数量呈爆炸式增长。这些 DNN 应用被广泛应用于自动驾驶^[10]，癌症检测^[11]，复杂游戏^[12] 等众场景下。并且在许多领域中，DNN 已经能够超越人类的准确性，取得了惊人的成就。

DNN 在许多领域取得巨大成功，为人类社会生活带来极大便利的同时，也引发了非常严重的侵犯知识产权 (Intellectual Property, IP) 问题。训练一个大型的高性能的 DNN 模型都离不开该领域专家的专业知识，规模巨大的数据集以及大量的训练时间和强大的计算资源，具体体现在以下三个方面：

- 1) 人力资源，对于不同场景不同目的的 DNN 模型，需要不同领域的知识，包含对模型结构的设计分析、模型参数的调试校验等；
- 2) 大量的训练数据，模型所有者要在特定领域训练出一个高性能的模型，通常需要该领域大量的数据，并且需要覆盖到应用场景中的各种情况，这些数据的获取和整理本身就需要昂贵的价格，有的领域的数据还涉及到隐私性问题；
- 3) 昂贵的计算资源和大量的训练时间，DNN 模型的规模越来越大，层数越来越多，需要的训练时间也越多，并且训练过程中也需要越来越多的计算资源支持，才能对网络权重等进行精确的调整，这些都是巨额的经济成本。如 GPT-3^[13]，包含了 1750 亿参数，仅训练成本需花费 460 万美元以上。

所以高性能 DNN 模型是模型所有者智慧的结晶，同时需要高额的经济开销，模型所有者享有 DNN 模型的知识产权^[14, 15]。

模型所有者出于学术目的将 DNN 模型放到开源社区上。或者，使用机器学习即服务 (Machine Learning as a Service, MLaaS)^[16] 的商业模式，即 MLaaS 平台通过训练好的 DNN 模型来向用户提供应用程序接口 (Application Programming Interface, API)^[17]，用户可以通过支付一定的费用来使用 API。或者，训练好的 DNN 模型将成为像我们日常商品一样的消费品，它们由不同的公司或个人进行训练，由不同的供应商分发，最终由用户消费。如图1.1所示，这样的方式极大的方便了科研工作者和一般的消费者，但是不法分子却可以以比模型所有者低很多的成本复制一个替代模型，用于自己盈利。

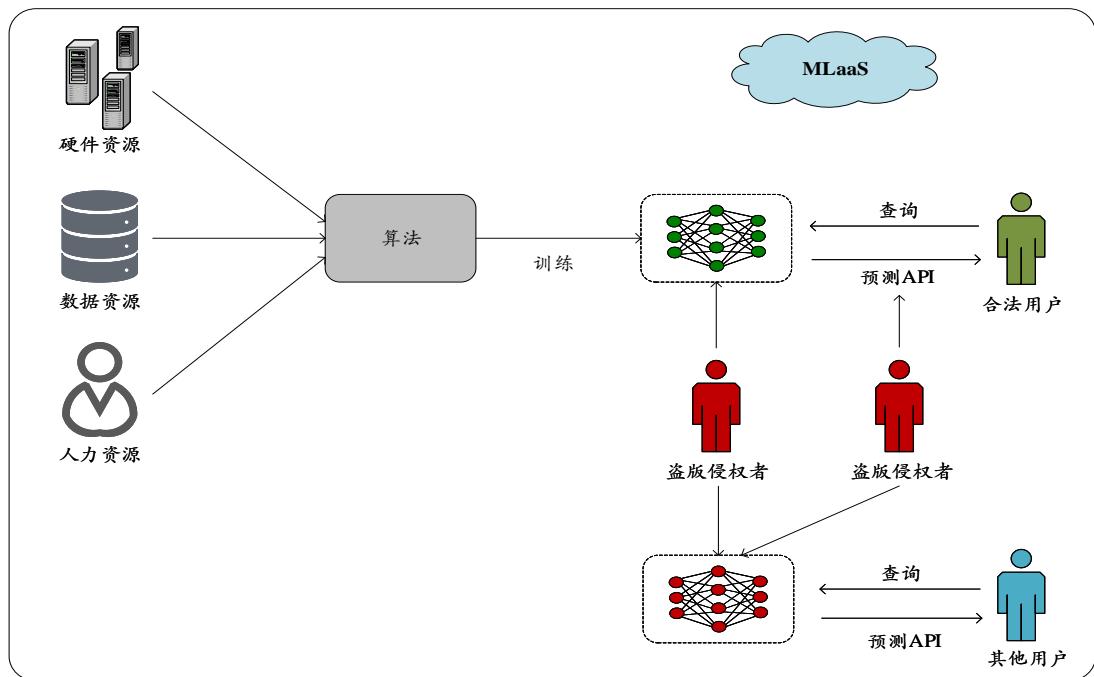


图 1.1 DNN 模型服务和盗窃示意图

所以如何在训练和部署时保护 DNN 模型所有者的知识产权是 AI 领域亟待解决的问题。

第二节 相关研究现状

作为一种数字产品，DNN 模型不仅凝结了设计者的智慧，还需要消耗大量的训练数据和昂贵的计算资源。近年来，拥有先进的的模型带来的工业优势已经被人们广泛认可，这开始激发一些不法分子窃取这些模型的攻击^[18, 19]。现在可以明确的是，DNN 模型将在未来的 IT 发展中发挥核心作用，因此保护这些

模型的必要性显得更加突出。1994 年，Van Schyndel 等人^[20]第一次提出数字水印的概念，将标记隐蔽的嵌入到如音频、视频等数字内容中，来识别其所有权，具体来说，版权所有者通过显示此类标记的存在可以证明其对内容的所有权。DNN 模型也是一种数字产品，所以，许多研究者从数字媒体水印得到启发，从而设计模型水印和模型指纹用于解决 DNN 模型的所有权问题。

模型水印^[21]是解决 DNN 模型知识产权问题的主要方式之一，Uchida 等人^[22]在 2017 年首次提出了在 DNN 模型中嵌入水印的通用框架。该方法是一种白盒的模式，通过训练时使用正则化器，并且这种正则化在参数中引入了所需要的统计偏差来作为嵌入的水印。模型所有者清楚模型内部的细节，并且可以提取嵌入的水印，以此来作为模型所有权的依据^[23]。Fan 等人^[24]提出了一种在 DNN 模型中嵌入数字护照的方案，嵌入数字护照的要点是设计和训练 DNN 模型，使得在伪造护照的情况下 DNN 的推断性能显著下降，而真正的护照可以通过查找预定签名来验证。Chen 等人^[25]提出了一种新颖的端到端框架，该框架同时依赖于用户和模型，它需要为每一个用户分配一个代码向量，并将该信息嵌入到可训练权重的的概率密度函数中，同时保持模型的准确性。不同于白盒的模式，另一种黑盒的模式，可以在不访问模型内部的情况下，通过特定的输入输出来验证模型的所有权。Le 等人^[26]提出了一种零比特水印算法，该算法标记模型的操作本身，稍微调整它的决策边界，来使特定的查询得到特定的输出。在减少模型性能损失的同时，该算法可以远程操作 DNN 或 API 服务，通过少量的查询提取水印。Zhang 等人^[27]提出了一种水印植入方法，将水印注入 DNN 模型。通过扩展 DNN 的内在泛化和记忆能力，使得模型能够在训练时学习特意制作的水印，然后在推断时激活预先指定的预测。Adi 等人^[28]提出了利用模型的后门机制当作 DNN 模型水印。后门通常是 DNN 将输入预测为错误的标签，虽然在大多数情况下这是不可取的，但是却可以将为 DNN 模型制作水印的任务转化为设计后门的任务。这些黑盒的方法利用对抗性样本作为触发集，或者使用一组特定的训练样本，然后根据特殊样本的输出来提取水印。因此黑盒的方法在所有权验证中不需要访问模型的权重参数。Rouhani 等人^[29]提出了一种端到端的 IP 保护框架 DeepSigns，可以在 DNN 模型中插入连贯的数字水印。DeepSigns 引入了一种通用水印方法，不同于直接将水印信息嵌入到模型的权重中，DeepSigns 将任意 N 位字符串嵌入到各层激活集的概率密度函数中，这意味着水印信息嵌入在 DNN 的动态内容中，并且只能通过特定的输入数据来触

发，并且对权重矩阵等静态属性没有影响。但是 DNN 模型水印的嵌入步骤总是会对原始进行修改。具体来说，白盒水印修改模型内部，比如模型权重，激活函数等，而黑盒水印通过特殊的训练调整模型来指定特定的输出。这些修改将会影响 DNN 模型在原始任务上的性能。

模型指纹是解决 DNN 模型知识产权问题的又一主流方法。不同与模型水印，模型指纹不需要对模型本身进行修改，而是利用模型本身来寻找和提取一些独特的特征作为模型指纹，一般来说，模型指纹不会影响模型的性能。Zhao 等人^[30] 提出了一种新的 DNN 模型指纹技术，该技术旨在提取模型本身的固有特征，而不是嵌入额外的水印。具体来说，该方法选择一组专门设计的对抗性样本作为模型指纹特征，称为对抗性标记，相比于其他不相关的模型，它可以更好的转移到从原始模型派生出的模型上。与 Zhao 等人^[30] 的方法类似，Lukas 等人^[31] 提出了一种用于 DNN 分类器的指纹识别方法，该方法从源模型中提取一组输入，以便只有源模型的派生模型在此类输入的分类上与源模型一致。这些输入是可转移对抗性样本的一个子类，它们的目标标签会从源模型转移到其派生模型上。Cao 等人^[32] 针对 DNN 分类器提出了一种名叫 IPGuard 的指纹方法，该方法的关键是 DNN 分类器可以由其分类边界唯一的表示。基于这一原理，IPGuard 在模型所有者的 DNN 分类器分类边界上提取了一些数据点，并使用它们对分类器进行指纹识别，如果 DNN 分类器对大多数指纹数据点预测相同的标签，那么该模型被认为是模型所有者分类器的盗版。Li 等人^[33] 提出了一种适用于生成对抗网络 (Generative Adversarial Network, GAN)^[34] 知识产权保护的指纹识别方案。该方案从目标 GAN 和分类器构建了一个复合深度学习模型，然后从该复合模型中生成隐蔽的指纹样本，并将其注册到分类器中进行有效的所有权验证。Dong 等人^[35] 针对 DNN 水印和指纹容易受到最抗性训练攻击，不适用于多出口 DNN 模型的 IP 验证的问题，提出了一种根据推理时间而不是推理预测的结果来为多出口模型建立指纹的新方法。

一般的模型窃取攻击涉及到模型的修改，主要包括模型微调，模型剪枝，模型压缩等。模型微调通常用于迁移学习，可以重新调整模型以更改模型参数，同时保持模型的性能。通过微调现有的模型，可以派生出许多功能相似的模型。模型剪枝是部署 DNN 模型的常见方法，通过参数修剪来减少 DNN 的内存需求和计算开销，而盗窃者可能会使用修剪来删除水印或指纹。模型压缩中常见的知识蒸馏，通过将大模型中的知识蒸馏到小模型中，可以显著降低模型的训

练成本，内存需求和计算开销，同时达到与大模型接近的性能。研究^[36]表明甚至不需要原始训练数据就可以直接利用 API 蒸馏模型，因此蒸馏常被用来派生模型。

虽然模型水印和模型指纹在保护模型知识产权方面已经取得了很大的进展，但是无论是水印还是指纹都容易受到歧义攻击^[24, 37]，歧义攻击是指通过为 DNN 模型伪造其他水印或指纹来对所有权验证产生干扰。直觉上，如果模型盗窃者可以在水印模型上嵌入第二个水印或者提取第二个指纹，那么该模型的知识产权归属存在巨大的歧义。

第三节 本文主要工作

为了解决 DNN 模型的知识产权问题，本文提出了近边界数据，一种分布在分类边界附近的特殊样本。模型指纹^[32] 使用对抗性样本抽象地反映模型分类边界，同一组对抗性样本的输入，其引起的决策模式的变化可以用于比较模型知识的相似性，但这种方法是脆弱的，对模型的任意操作都有可能破坏这种特性。因此，我们不直接比较决策模式的变化，它是不可信任的，而是比较对抗性样本与决策边界的距离。有意思的是大多数对抗性样本都是位于决策边界附近的，也就是说，它们与决策边界的距离很近。对抗性样本的这种性质被我们所利用并构造近边界数据，经过测试我们发现绝大多数的模型窃取方法都无法改变这种结果，即使样本分类被影响，其仍然位于分类边界附近。近边界数据背后的意义是如果被用于所有权验证如果两个模型的决策模式相似，参与训练的近边界数据一定可以反映出来。受这个特性的启发，将近边界数据作为水印验证所有权是传统的思路，即使不会对模型的精度造成影响，这样的水印也是脆弱的，很难抵御歧义攻击，因此我们提出由近边界数据驱动的所有权推断方法，其思想是构造私有的近边界数据，当验证一个模型的所有权时，模型所有者和盗窃者分别提供各自的私有近边界数据，距离分类边界最近的被推断获得所有权。

本文的主要贡献如下：

- 1) 揭示了当前 DNN 模型所有权验证方案的脆弱性并确认了数据驱动推断模型所有权的有效性。当前模型水印和模型指纹的方法通常是在模型内部嵌入特定的水印或者通过特定的触发集来验证模型的所有权，这种方式在面对歧义攻击，删除攻击等强攻击时，并不具备很强的鲁棒性。DNN 模型是通过数据集训练的，所以不管是源模型还是派生出来的模

型，总会包含数据中的知识，本文确认了这种数据驱动的推断模型所有权方法的有效性。

- 2) 提出了利用对抗性样本构造近边界数据以抵御模型窃取攻击。DNN 分类器可以由其分类边界唯一的表示，而对抗性样本一般位于模型分类边界上，可以很好的反应分类边界，并且相较于其他不相关的模型，对抗性样本可以更好的转移到从原始模型派生出的模型上。因此，本文利用对抗性样本构造了近边界数据来推断模型所有权，抵御模型窃取攻击。
- 3) 设计了基于 DCGAN 的近边界数据生成器和提出了一种损失函数用以微调源模型的目标分类边界，增加推断模型所有权的置信度。DCGAN 是深度卷积生成对抗网络，本文利用 DCGAN 对图像特征的强大处理能力，通过对近边界对抗性样本的特征提取，使用 DCGAN 的生成器生成我们私有的近边界数据。在此基础之上，为了使生成的私有数据更加接近分类边界，重新设计了模型损失函数，并且和原始数据交替训练源模型。在保持 DNN 模型性能的情况下，微调源模型分类边界，以 95% 以上的置信度推断模型所有权。
- 4) 基于 ResNet18^[38] 和三个公开数据集进行了广泛的实验，实验结果证明了近边界数据在推断模型所有权上的显著效果。本文在三个公开数据集上分别训练了 ResNet18 作为源模型，并且使用模型微调，不同比例模型剪枝，知识蒸馏几种方式派生出替代模型，使用 VGG11^[5] 作为无关对照模型。在生成初始近边界数据的方法选择，数据近边界特性评估，微调目标分类边界的影响，推断模型所有权的有效性和不同规模近边界数据的可伸缩性扩展几个方面对本文提出的方法进行了详细的实验和分析，实验结果证明了基于近边界数据推断模型所有权方法的有效性和鲁棒性。

第四节 本文组织架构

本文对模型的近边界数据进行了研究，并提出了生成私有近边界数据的方法以及基于近边界数据推断模型所有权的方法。全文共分为六个章节，每个章节的主要内容如下：

第一章：绪论。本章首先介绍了 DNN 模型在当今时代的广泛应用和研发的昂贵成本，引出了保护 DNN 模型知识产权的必要性和重大意义，然后介绍了模

型水印和模型指纹两种保护方法的研究现状，并针对相关研究存在的问题提出了本文的研究内容，最后简要说明了各个章节的内容安排。

第二章：技术背景。本章主要介绍了深度神经网络的结构和相关概念，对抗性攻击和生成对抗网络的基本原理，常见的模型窃取攻击方式和模型水印，模型指纹两种模型知识产权保护方法。

第三章：基于生成对抗网络特征提取的近边界数据研究。本章首先给出了分类边界和近边界数据的定义，研究了近边界数据在源模型和派生模型上的特性。然后对比了常见的生成对抗性样本的方法并对 CW- L_2 方法进行改进，使之生成近边界对抗性样本。最后利用生成对抗网络的特征提取功能，使用生成器生成我们的私有化近边界数据，并设计了新的损失函数来微调模型分类边界，增加所有权推断置信度。

第四章：基于近边界数据的模型所有权推断方法研究。本章首先阐述了本文方法的理论驱动，然后提出了所有权验证和数据集推断的局限性。针对之前的不足，提出了基于近边界数据的模型所有权推断方法，并说明了该方法的设计目标和详细的工作流程，最后提出使用假设检验的方法来统计对比结果。

第五章：基于近边界数据的模型所有权推断方法分析。本章在 ResNet18 和三个公开数据集上，对生成初始近边界数据的方法选择，数据近边界特性评估，微调目标分类边界的影响，推断模型所有权的有效性和不同规模近边界数据的可伸缩性扩展几个方面进行了详细的实验，证明了本文提出方法在推断模型所有权时的有效性和鲁棒性。

第六章：总结与展望。本章总结了全文的工作，分析了本文提出方法的优势和不足，并针对不足之处提出对未来工作的展望。

第二章 技术背景

本文的私有近边界数据主要是在近边界对抗性样本的基础上通过生成对抗网络生成的。本章首先介绍了深度神经网络的基本结构和知识产权保护领域相关术语，然后着重介绍了对抗性攻击和生成对抗网络的原理，为第三章的近边界数据的生成提供理论基础。最后说明了DNN的模型窃取攻击的种类以及模型水印和指纹两种知识产权保护方法。

第一节 深度神经网络及相关术语

人工神经网络是一种类似于人类大脑生物神经系统的信息处理模型，它由许多相互连接的神经元（网络中的节点）组成，这些神经元都可以向其他神经元发送信号。一般的神经网络由输入层，隐藏层和输出层组成，如图2.1所示，如果一个神经网络有多个隐藏层，那么这个神经网络就被称为深度神经网络。DNN的隐藏层一般由卷积层，池化层，全连接层，Dropout层和Softmax层构成，数据输入输入层后，会经过每一层，每层提取的抽象特征会作为下一层的输入，最终由输出层输出。

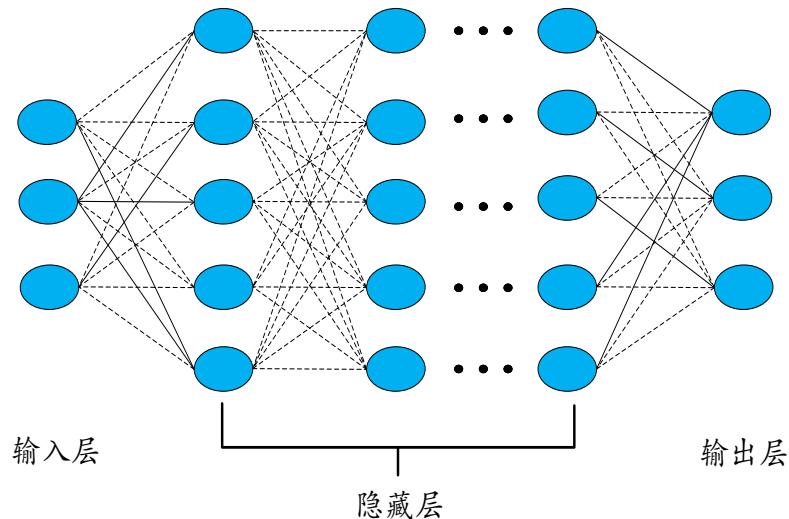


图 2.1 深度神经网络结构图

DNN 可以看作是将一组输入变量转化为一组输出变量的非线性数学函数。每个神经元都有对应的权重和偏置参数，控制着输入的精确转化，这些参数在反向传播的过程中，通过损失函数和梯度下降算法来更新。确定这些参数的过程称为 DNN 模型的学习或者训练，并且需要大量的计算资源，然而权重一旦确定，DNN 模型就可以快速的处理相似类型的新数据，识别并提取海量数据中的复杂特征。

以下是本文中涉及到 DNN 知识产权保护领域中的相关术语：

- 1) 源模型。源模型也称作目标模型，是指模型所有者在私有或公共数据集上，消耗大量计算资源和人力资源训练出的高性能 DNN 模型，可能因学术研究放置在开源社区，或者作为商用给用户提供远程 API。
- 2) 可疑模型。可疑模型也称作替代模型，是指该模型可能是通过模型窃取攻击方法从源模型派生出来的模型，判断一个可疑模型是否是从源模型派生是模型知识产权保护领域的主要目标。
- 3) 白盒模式。白盒模式是指能够获得 DNN 模型的所有知识，包括训练集，训练方式，模型参数，模型结构等。
- 4) 黑盒模式。黑盒模式指不清楚模型内部参数和结构等，但可以通过模型提供的 API 获得指定输入的输出。

第二节 对抗性攻击

2.2.1 对抗性样本

对抗性样本的概念是 Szegedy 等人^[39] 提出的。这篇文章中指出，通常情况下，一个良好性能的 DNN 模型具备很好的泛化能力，对输入的随机微小扰动具有鲁棒性，因此小扰动不应该改变图像的预测类别。然而，对图像添加特定的非随机扰动，使得损失函数的值增大，可以任意改变 DNN 模型的预测结果。这种人类肉眼上难以察觉但可以使模型输出错误类别的样本称为对抗性样本。

用 $f: R^m \rightarrow 1, 2, \dots, n$ 表示将一张图片映射为 n 个标签的 DNN 分类器，对一个正常样本 $x \in R^m$ 以及一个错误标签 l ，目标是找到一个最小的扰动 δ ，使得分类器将样本 x 错误分类为 l ，如式2.1所示：

$$\begin{aligned} & \min \| \delta \|_2, \\ & \text{s.t. } f(x + \delta) = l, \quad x + \delta \in [0, 1]^m \end{aligned} \tag{2.1}$$

其中叠加了扰动的 $x + \delta$ 即为一个对抗性样本。2.1这种方式通常用在黑盒的场景下，仅根据 DNN 分类器的输出进行扰动 δ 的调整。

在白盒场景下，由于知道模型的所有知识，可以根据这些信息来寻找对抗性样本，通常利用 DNN 分类器的损失函数来寻找对抗性样本。

用 $f: R^m \rightarrow 1, 2, \dots, n$ 表示将一张图片映射为 n 个标签的 DNN 分类器，对一个正常样本 $x \in R^m$ 以及它对应的正确标签 y ，目标是找到一个足够小的扰动 $\delta: \delta \leq \gamma$ ，使得加上扰动后的样本输入 DNN 模型后，损失函数 L 达到最大值，如式2.2所示：

$$\delta = \arg \max_{\delta \leq \gamma} L(f(\theta, x + \delta), y) \quad (2.2)$$

其中 θ 是分类器 f 的参数， $x + \delta$ 是一个扰动后的对抗性样本。

2.2.2 对抗性攻击的类别

对抗性攻击技术是指生成对抗性样本的方法，不同的方法生成对抗性样本的效率，质量也不相同。根据方式的不同，可以分为以下几类：

- 1) 白盒攻击与黑盒攻击。白盒攻击指敌手知道 DNN 模型的参数和内部结构等信息，利用这些信息发起的攻击。黑盒攻击指敌手仅根据模型的输入输出发起攻击。
- 2) 有目标攻击和无目标攻击。有目标攻击指对抗性样本的预测类别为敌手指定的类别，例如将一张牛的图片识别为羊，而不能是其他类别，常采取的方式是向各个方向搜索扰动来最大化 DNN 模型预测特定类上的可能性。无目标攻击指添加扰动来改变原始预测类别，对具体分类类别不做要求。通常来说有两种攻击方式，一种是最小化 DNN 模型预测正确类的可能性，一种是进行多次不同类别的有目标攻击，然后在多个对抗性样本中选取扰动最小的。
- 3) 单步攻击和迭代攻击。单步攻击指通过一次添加扰动生成对抗性样本，迭代攻击指通过多次迭代添加微小扰动来生成对抗性样本。通常来说迭代攻击的成功率较高，但是相应的算法复杂度更高，效率较低。
- 4) 个体攻击和普适性攻击。个体攻击指针对每个样本都需要重新生成扰动，普适性攻击指找到一个通用的扰动，对数据集中的一类数据都叠加该扰动，普适性攻击效率较高，但是寻找通用扰动的难度较大。

第三节 生成对抗网络

Goodfellow 等人^[40]第一次提出了生成对抗网络 (Generative Adversarial Network, GAN)，是一种通过生成模型实现无监督学习的特殊方法。GAN 由一个生成器和一个判别器构成，训练是一个相互博弈的过程，如图2.2所示，首先随机噪声作为生成器的输入，生成器生成和真实图片维度一致的图像，使用原始图片和生成图片分别输入判定器，训练判定器区分它们的能力，再训练生成器，使之尽可能接近真实图片，通过迭代的交替训练，最终生成器生成的图片和原始图片在空间分布上基本一致，判定器判定生成图片和原始图片为真的概率均为 $1/2$ ，也就是无法区分生成图片和原始图片。

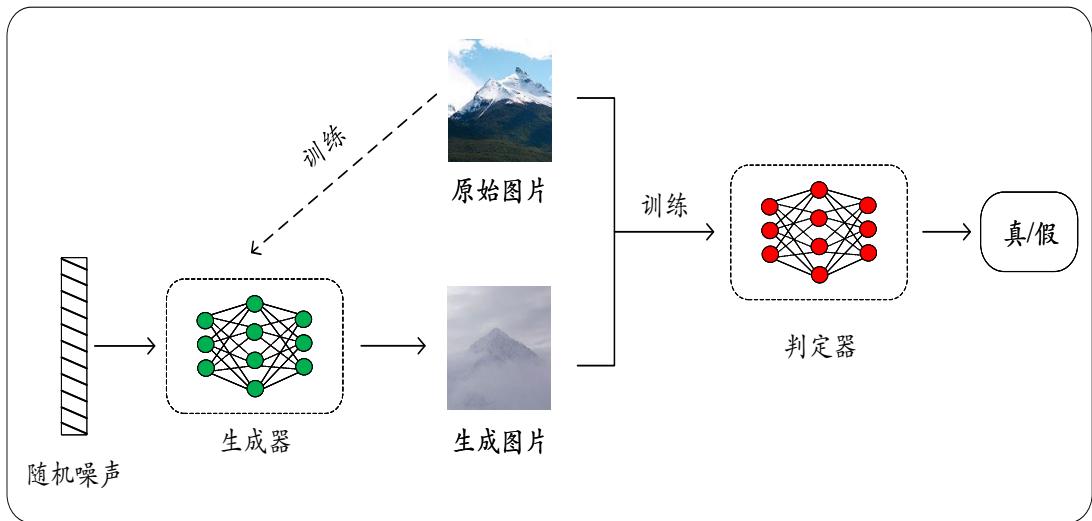


图 2.2 生成对抗网络结构图

具体而言，生成器 G 和判别器 D 可视为博弈中的双方，当训练 GAN 模型时，生成器 G 和判别器 D 通过更新各自的参数使损失达到最小，经过不断迭代优化，最后 G 和 D 达到纳什均衡。GAN 的目标函数如式2.3所示，对于原始图片 x ，判别器希望 $D(x)$ 变大，对应于式中的 $\max D$ ，对于生成图片 $G(T)$ ，生成器希望 $D(G(T))$ 变大，即 $\log(1 - D(G(T)))$ 变小，对应于式中的 $\min G$ ，所以 GAN 的目标函数由两个目标构成。

$$\begin{aligned} \min_G \max_D V(D, G) = & \min_G \max_D E_{x \sim P_{data}(x)} [\log D(x)] \\ & + E_{T \sim P_T(T)} [\log(1 - D(G(T)))] \end{aligned} \quad (2.3)$$

其中 x 表示原始图片， T 表示用于生成样本的随机噪声，GAN 对噪声 T 的分布没有特别要求，但是常用的有高斯分布，均匀分布， E 表示数学期望。

第四节 深度神经网络的模型窃取攻击

自 DNN 在各个领域取得巨大成功以来，针对 DNN 模型的攻击就层出不穷，按照攻击方式的不同，可以分为以下三类^[41]：(1) 模型修改攻击。指常见的模型修改，主要包括包括模型微调，模型剪枝，模型压缩，模型再训练等方式。(2) 删除攻击。指攻击者试图逃避水印或指纹的检测，主要包括删除攻击，篡改攻击，逆向工程攻击等方式。(3) 主动攻击。指攻击者主动攻击和强攻击，主要包括歧义攻击，水印和指纹覆盖攻击，查询修改攻击等方式。

2.4.1 模型修改攻击

模型窃取者在盗窃 DNN 模型后，通常会对 DNN 模型进行修改或者压缩，然后部署模型作为 MLaaS 来非法盈利。模型修改主要包括：

- 1) 模型微调。微调通常用于迁移学习中，包括在源模型的基础上，根据自己定制的任务，继续训练模型，使得 DNN 模型在保持性能的同时修改内部的参数。模型微调可以从源模型派生出非常多的模型。由于内部参数发生改变，水印等可能也会随之变化，因此这对水印的鲁棒性是一个考验。
- 2) 模型剪枝。由于 DNN 模型通常内存占用多，计算开销大，因此模型剪枝是在小型设备上部署 DNN 模型的常用方法。但是模型窃取者可能会利用剪枝来删除水印，因此有效的水印技术应该能够抵御由模型剪枝引起的参数变化。
- 3) 模型压缩。模型压缩可以显著降低 DNN 模型的内存需求和计算开销，常用的方法是知识蒸馏，通过将大型模型包含的知识转移到小模型上来达到模型压缩的目的。
- 4) 模型再训练^[42]。模型再训练是一种很直接的方法，这样能尽可能的去除或者减少原有水印的影响，相应的，这种攻击方式成本也比较高。

2.4.2 删 除 攻 击

目前大部分 DNN 模型的知识产权保护工作专注于水印对 DNN 模型被修改时的鲁棒性，而很少考虑水印或指纹本身受到的攻击。删除攻击主要包括：

- 1) 删 除 攻 击^[43]。攻击者试图修改模型以删除原有的水印。
- 2) 篡 改 攻 击。攻击者知道 DNN 模型中存在水印，试图篡改模型来删除原有的水印和指纹特征。
- 3) 逆 向 工 程 攻 击^[24]。如果攻击者知道并可以获得原始训练数据，可能会直接对内部参数进行逆向工程。

Shafieinejad 等人^[43]研究了 DNN 中基于后门的水印方法的移除攻击，表明攻击者可以仅依靠公共数据集删除水印，而不用访问训练集和模型参数。还提出了一种检验水印的方法，表明基于后门的水印不够安全，无法保持水印的隐藏。

2.4.3 主 动 攻 击

除了被动的攻击方式，攻击者还可能对 DNN 模型发动更强的主动攻击。主动攻击主要包括：

- 1) 歧 义 攻 击。歧 义 攻 击 指 在 DNN 模型上伪造额外的水印来混淆所有者的验证。研究表明，除非采取不可逆的水印方案，否者即使是鲁棒性的水印，也不一定能验证模型的所有权^[24]。
- 2) 水 印 覆 盖 攻 击^[15, 44, 45]。即使攻击者不知道具体的私有水印信息，但他知道模型水印嵌入的方法，就可能通过在 DNN 模型中嵌入新的水印来覆盖原有的水印，从未破坏原有的水印使其不可读。
- 3) 查 询 修改 攻 击。攻 击 者 修改查 询 结 果 来 使 得 水 印 验 证 过 程 无 效。一个典 型 的 方 式 是 攻 击 者 获 得 DNN 模 型 并 部 署 为 MLaaS 后，会 主 动 检 测 一 个查 询 是 否 为 水 印 验 证 查 询，从 而 修改 或 者 屏 蔽 该 查 询，使 水 印 验 证 无 效。

第五节 深 度 神 经 网 络 模 型 的 知 识 产 权 保 护

训练一个高性能 DNN 模型需要该领域专家的先验知识来设计模型结构，大量的训练数据和昂贵的计算资源和漫长的训练时间，因此，训练后的 DNN 模型属于模型所有者的知识产权。得益于 DNN 模型在各个领域的高效应用，许多不法分子开始偷盗，复制和修改这些模型来提供服务盈利。为了保护 DNN 模型的知识产权^[46, 47]，许多学者受多媒体数字水印的启发，使用模型水印和模型指纹

来验证 DNN 模型知识产权。

2.5.1 模型水印

模型水印是第一种被提出的保护 DNN 模型知识产权的方法，根据水印嵌入方式和提取方式的不同，主要分为白盒水印和黑盒水印。

在白盒场景下，模型所有者可以利用模型的全部知识构造水印，这些知识包括训练数据集，训练方法，模型内部权重参数和结构。Kuribayashi 等人^[48] 提出一种基于全连接层权重的可量化水印嵌入方法，通过在训练中改变参数，可以量化水印的影响，从而保证嵌入水印引起的变化较小。不同于基于权重的方法，基于内部结构的水印方法抵抗模型修改的鲁棒性更强。可以在 DNN 模型中添加一个额外的护照^[24]，比如在模型卷积层后面添加一个额外的护照层，以此来作为数字签名，这种方式还可以解决模型受到歧义攻击的问题。

在黑盒场景下，模型所有者不知道可疑模型的内部结构和权重参数等，只能通过 API 进行访问。一般而言，是通过构造特殊的触发集来实现的，主要有以下几种方式：

- 1) 通过更改样本标签构造触发集，将原始样本标签更改为模型所有者指定的与原始内容不符合的标签，这样仅修改标签不做任何其他修改的水印方法称为零位水印。
- 2) 通过在原始样本中嵌入额外水印信息和更改标签构造触发集，这样可以在模型输出中嵌入模型所有者的版权信息。
- 3) 通过添加新的样本构造触发集，这样的方式对模型的精度影响较大，一般通过模型微调最大限度的减少新样本对模型决策的影响。

2.5.2 模型指纹

模型指纹一般是利用模型本身来寻找和提取一些固有的特征来作为指纹。相较于模型水印的方法，模型指纹一般不对模型进行修改，因此不会影响模型的精度。一般来说，可以选择靠近决策边界的对抗性样本作为模型的指纹特征，来验证模型所有权。模型指纹分为指纹生成和指纹验证两个阶段。

(1) 模型指纹生成

模型指纹与生物学上的指纹类似，具有唯一标识性，用以标记 DNN 模型的所有权。根据之前对指纹的相关研究^[30-32]，可以把靠近决策边界的对抗性样本作为模型的指纹。具体来说，通过将此类样本输入 DNN 模型，将其对应的输出

标签作为模型的指纹。

对于一个正常样本 x , 添加一个微小的扰动 δ , 使得第 i 个输出满足:

$$\arg \max_i g_i(x) = y \wedge \arg \max_i g_i(x + \delta) = y' \quad (2.4)$$

其中 y 表示输入样本 x 对真实标签, 叠加扰动后输出 $y' \neq y$, δ 是对抗扰动, $x + \delta$ 即为对抗样本。

所以一组对抗性样本作为指纹样本, 对应的输出作为指纹标签, 共同构成模型指纹。即 $X' = \{x'_1, x'_2, \dots, x'_n\}$ 作为模型指纹样本, 对应的预测结果 $Y' = \{y'_1, y'_2, \dots, y'_n\}$ 为指纹标签。

(2) 模型指纹验证

在模型指纹验证阶段, 通常的做法是使用指纹样本查询可疑模型 API, 比较 API 返回的预测标签和指纹标签的匹配程度。设定一个阈值 τ , 当阈值标签和指纹标签的匹配率超过阈值 τ 时, 视为匹配成功。

设 $X' = \{x'_1, x'_2, \dots, x'_n\}$ 为 n 个指纹样本, $Y' = \{y'_1, y'_2, \dots, y'_n\}$ 为对应的指纹标签, 将 n 个指纹样本输入可疑模型 API 后, 预测标签为 $\tilde{Y}' = \{\tilde{y}'_1, \tilde{y}'_2, \dots, \tilde{y}'_n\}$ 。其中预测标签与指纹标签相同的数量为 q , 即 $y'_i = \tilde{y}'_i (i = 1, 2, \dots, q)$ 。

定义验证函数如下:

$$Verify(Y', \tilde{Y}') = \begin{cases} 1, & \frac{q}{n} \geq \tau \\ 0, & \text{其他} \end{cases} \quad (2.5)$$

式2.5中, q/n 表示匹配成功率, τ 是阈值, 结果为 1 代表判定可疑模型为盗版模型。

第六节 本章小结

本章主要介绍了深度神经网络, 对抗性攻击, 生成对抗网络, 深度神经网络的模型窃取攻击和知识产权保护这五个方面的相关概念和理论基础。DNN 复杂的结构使其训练昂贵耗时, 但是一旦训练完成就可以快速处理新数据。对抗性样本是一种人类肉眼无法察觉到变化而导致模型输出错误的特殊样本, 根据方式的不同, 有多种产生的方法。生成对抗网络具有强大的特征提取能力, 可以利用其生成类似训练样本特征分布的新样本。根据攻击强弱的不同, 可以将

DNN 模型窃取攻击分为模型修改攻击，删除攻击和主动攻击。在 DNN 模型知识产权保护领域，目前使用最广泛的是模型水印和指纹这两种方法。

第三章 基于生成对抗网络特征提取的近边界数据研究

DNN 分类器可以由其分类边界唯一的表示，本章将讨论模型分类边界的具体表示方法，对抗性样本的可转移性及其生成近边界对抗性样本的方法和 DCGAN 私有化近边界数据的详细过程。

第一节 近边界对抗性样本

DNN 分类器的主要工作是对数据样本进行分类，因此，一个 DNN 分类器的特征由其决策模式和分类边界决定。分类器的分类边界是一个抽象的概念，我们无法直接描述它，但可以根据分类器的决策结果反应分类边界。下面给出分类边界的定义：

定义 1 分类边界。给定一个数据样本 x ，如果数据样本 x 满足 $g_i(x) = g_j(x)$ ，其中 $i \neq j$ 并且 $\min(g_i(x), g_j(x)) > \max_{k \neq i, j} g_k(x)$ ， $g_k(x)$ 代表数据样本 x 被决策为类别 k 的概率，那么称数据样本 x 位于类别 i 和 j 的分类边界上。

注意，分类边界是分类器的特征，是客观存在的，和我们是否能找到这样的数据点并不相关，如何有效的找到分类边界或其附近的数据点，是本文方法要解决的问题之一。

模型窃取攻击一般会对源模型进行一定的修改，这会调整源模型的分类边界，无法保证位源模型分类边界上的样本依旧位于可疑模型分类边界上。对抗性样本是一类特殊的样本，虽然我们可以找到位于分类边界上的对抗性样本，但是在可疑模型中无法保证它们依然位于分类边界上。因此，直接利用分类边界来作为模型指纹是脆弱的。与分类边界的思想类似，本文提出了一个鲁棒性更强的近边界数据概念。下面给出近边界数据的定义：

定义 2 近边界数据。给定一个数据样本 x ，一个阈值 θ ，如果数据样本 x 满足 $|g_i(x) - g_j(x)| \leq \theta$ ，其中 $i \neq j$ 并且 $\min(g_i(x), g_j(x)) \geq \max_{k \neq i, j} g_k(x)$ ， $g_k(x)$ 代表数据样本 x 被决策为类别 k 的概率，则数据样本 x 被称为近边界数据。

如图3.1所示，近边界数据位于 DNN 分类器的分类边界附近，距离的远近由阈值 θ 控制。

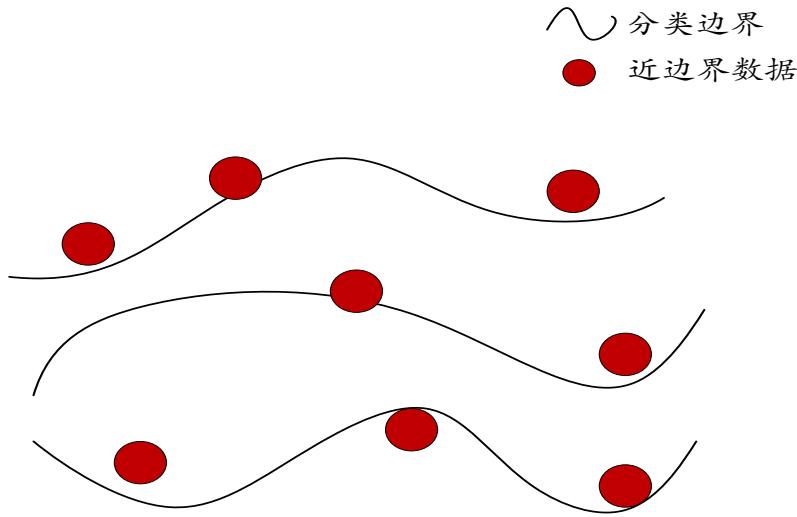


图 3.1 近边界数据示意图

相较于其他不相关的模型，对抗性样本可以更好的从源模型转移到其派生出的模型上。与对抗性样本类似，数据的近边界性同样在派生模型上得到保留，详细的测试结果在第五章第二节中。在下一节中，我们将讨论如何生成近边界对抗性样本。

第二节 生成近边界对抗性样本

尽管近边界在模型的知识产权保护中表现出显著的效果，但是自然的近边界数据在样本空间中的占比很低，甚至可以忽略，因此如何得到一定规模的近边界数据样本仍然很困难。

根据最近的一些研究^[32]，对抗性样本通常被用于确定分类器的分类边界。具体而言，对抗性样本有两个分类：原始分类和目标分类。其中，原始分类是该样本不经过特殊处理的原始分类结果，目标分类是对原始样本添加微小噪声后的分类结果。如图3.2所示，对抗性样本对分类边界的跨越体现在，在视觉上对抗性样本和原始样本几乎没有差别，但是分类结果却是目标分类。

本文认为该特征可以帮助获得较多的近边界数据，具体来说，我们将从大量的对抗性样本中挑选近边界数据。因此，本文测试了几种常用的生成对抗性样本的方法，以帮助我们构建近边界数据。测试过程中，不同方法的优劣取决于生成对抗性样本到分类边界距离的远近，距离近者更优。在定义2的基础上，下



图 3.2 原始样本与对抗性样本对比

面给出量化的距离定义：

定义 3 分类边界距离。 给定一个数据样本 x , 它到分类边界的距离 $distance = |g_i(x) - g_j(x)|$, 其中 $i \neq j$ 并且 $\min(g_i(x), g_j(x)) \geq \max_{k \neq i, j} g_k(x)$, $g_k(x)$ 代表数据样本 x 被决策为类别 k 的概率。

Fast Gradient Sign Method(FGSM):FGSM^[49] 是最经典的构建对抗性样本的方法之一, 它是一种基于梯度生成对抗性样本的方法, 属于无目标攻击方式。只需要对原始样本添加微小的扰动 η , 如式3.1, 3.2所示, 即可生成样本 x 的对抗性样本 \tilde{x} 。

$$\eta = \epsilon \cdot sign(\nabla_x J(\theta, x, y^*)) \quad (3.1)$$

$$\tilde{x} = clip(x + \eta) \quad (3.2)$$

其中 $sign$ 是符号函数, x 表示原始样本, y^* 表示 x 的真实类别, θ 表示模型权重参数, J 表示分类器损失函数, ∇_x 表示对原始样本 x 求偏导, $clip$ 函数是将样本投射回可行数据域, 比如图像样本的像素点范围在 $[0, 1]$ 内, ϵ 用来控制变化幅度。

FGSM 生成对抗性样本的速度非常快, 但其结果非常依赖 ϵ 的选择, 因此探索不同的 ϵ 是使用该方法的重点。

Iterative Gradient Sign Method(IGSM):IGSM^[50] 是 FGSM 的进阶版本, 如式3.3, 3.4所示, 与 FGSM 只进行一次扰动叠加不同, IGSM 采用迭代的形式构造对抗性样本, 每次叠加一个小扰动。这个过程持续到成功生成对抗性样本或者达到迭代次数上限。

$$\eta = \alpha \cdot \text{sign}(\nabla_x J(\theta, x, y^*)) \quad (3.3)$$

$$\tilde{x}_t = \text{clip}(\tilde{x}_{t-1} + \text{clip}_\epsilon(\eta)) \quad (3.4)$$

其中 α 是步长大小, \tilde{x}_t 表示第 t 次迭代后的结果, clip_ϵ 是限定每次叠加的范围不超过 ϵ , 其余参数与 FGSM 保持一致。

除此之外, 我们还测试了 FGSM 的另一个进阶版本 RFSGM^[51], RFSGM 增加了扰动的多样性, 可以更精细地生成对抗性样本。在实际结果中我们发现尽管 FGSM 生成对抗性样本速度非常快, 但是对抗性样本距离分类边界的距离比较远。IGSM 和 RFGSM 效果要比 FGSM 好, 但仍认为不符合我们的期望。在大量的测试中, 我们发现 CW 能够生成大量位于分类边界附近的样本, 具体的测试结果在第五章第二节中。

Carlini and Wagner’s methods(CW):CW^[52] 方法是一种有目标的攻击方式, 同样是添加噪声到对抗性样本中, 但其具有三种变体: CW- L_0 , CW- L_2 和 CW- L_∞ , 不同的变体使用不同的方法来衡量噪声的大小, 其中 CW- L_2 在实验中效果最为突出, 因此本文使用该方法作为生成对抗性样本的选择。具体而言, CW- L_2 对于给定的初始样本采用二分搜索的方式迭代搜索一个小噪声, 并且每次搜索记录下最小的扰动。CW- L_2 的损失函数如式3.5, 3.6, 3.7所示:

$$Loss = Loss1 + Loss2 \quad (3.5)$$

$$Loss1 = D(x, x + \delta) \quad (3.6)$$

$$Loss2 = c \cdot f(x + \delta, target) \quad (3.7)$$

$$s.t. x + \delta \in [0, 1]^m$$

其中 $target$ 是目标标签, c 是惩罚因子, 用于权衡 $Loss2$ 的影响大小, 通过二分查找来寻找合适的 c , $Loss1$ 约束对抗性样本 $x + \delta$ 和原始样本 x 尽可能相似, $Loss2$ 约束对抗性样本 $x + \delta$ 的决策结果为目标标签。

根据定义3, 数据样本 x 距离分类边界的距离是 $distance = |g_i(x) - g_j(x)|$, 我们希望生成的对抗性样本距离分类边界的距离尽可能近。本文算法迭代过程中引入这一目标, 改进算法迭代的过程, 在保持生成对抗性样本靠近分类边界的同时, 提高算法效率。具体而言, 在迭代过程中, 我们仅在 $distance$ 变小时

更新距离参数和对抗性样本，并在 $distance$ 小于预定的阈值 θ 时提前终止迭代，具体的过程如算法1所示。

Algorithm 1 改进的二分查找 CW-L₂ 算法

输入： 样本 x ; 模型 M ; 阈值 θ ; 二分次数 n ; 迭代次数 $iteration$; 原始标签 r ;
目标标签 t

输出： 近边界对抗性样本 x'

```

1: 参数初始化:  $c \leftarrow 1$ ,  $distance \leftarrow 1$ 
2: for  $i = 1, 2, \dots, n$  do
3:    $isSuccessAttack \leftarrow false$ 
4:    $w \leftarrow arctanh(x)$ 
5:    $w\_pert \leftarrow zero\_like(w)$ 
6:   for  $j = 1, 2, \dots, iteration$  do
7:      $new\_img \leftarrow \tanh(w + w\_pert)$ 
8:      $new\_distance \leftarrow |g_r(new\_img) - g_t(new\_img)|$ 
9:     if  $new\_distance < distance$  then
10:       $distance \leftarrow new\_distance$ 
11:       $x' \leftarrow new\_img$ 
12:       $isSuccessAttack \leftarrow true$ 
13:    end if
14:    使用 Adam 更新  $w\_pert$ 
15:  end for
16:  if  $isSuccessAttack == true$  then
17:    减小  $c$ 
18:  else
19:    增大  $c$ 
20:  end if
21:  if  $distance \leq \theta$  then
22:     $break$ 
23:  end if
24: end for
25: return  $x'$ 

```

在这一阶段，我们只是在源模型的样本空间中挑选一部分数据作为初始样本添加小噪声，针对性地生成了目标分类对抗性样本。在此阶段源模型的训练和原始数据均不受任何影响，防御者只需要针对性的生成对抗性示例即可。然

而，近边界数据作为推断所有权的重要证据，直接生成对抗性样本也极易受到盗窃者的复制。因此，我们需要将生成的近边界数据私有化，具体操作将在下一节中给出。

第三节 近边界数据私有化

由于通过生成对抗性样本的方法构建近边界数据这一步骤十分容易复现，并且现在大多数模型训练使用的数据都来源于公开数据。因此我们需要从公开的训练数据中构建自己私有化的近边界数据，以防止模型所有者的近边界数据被轻易模仿。在本文中，我们希望可以通过训练一种模型学习上一节中生成的近边界对抗性样本特征，并以此生成新的近边界数据。这种新的数据从视觉上不一定和原始数据类似，但其原始的特征以及添加的噪声需要被学习，并根据提取到的特征生成的新样本对于源模型同样是近边界数据。因为本文用到的是图像样本，CNN 可以很好的处理图像，因此，在本文中我们设计了一种基于 DCGAN^[53] 的特征提取器，提取近边界数据的特征之后，使用生成器并生成私有化近边界数据。

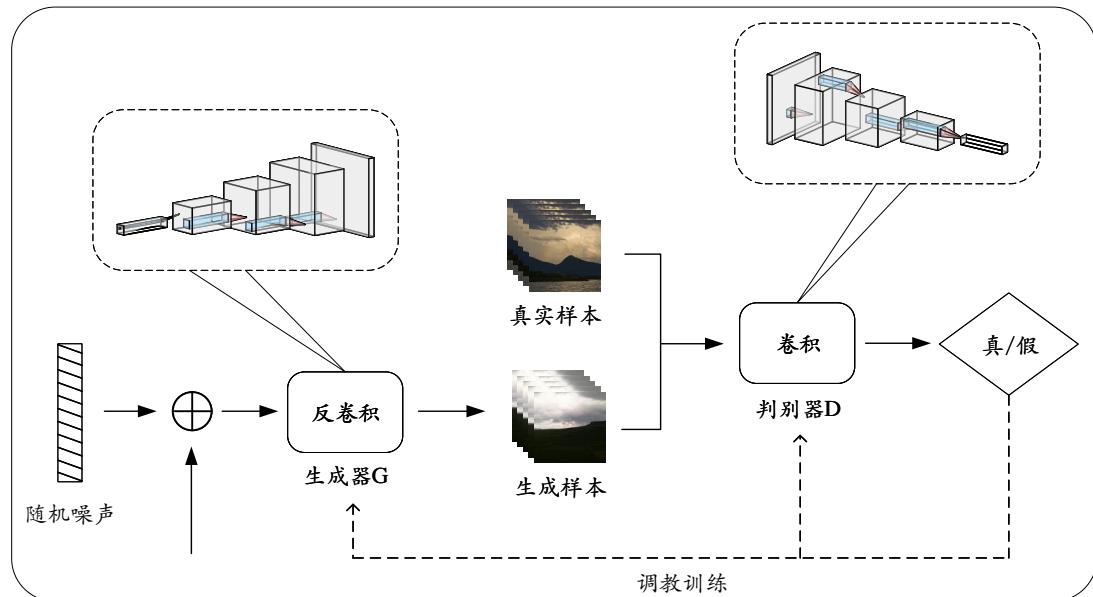


图 3.3 DCGAN 网络结构图

如图3.3所示，DCGAN 的大体结构与训练方式和普通 GAN 类似，但是 DCGAN 将 CNN 与 GAN 相结合，生成器 G 和判定器 D 都用 CNN 替换原始

GAN 的全连接网络。得益于 CNN 对图像的处理能力，DCGAN 极大提升了网络训练稳定性和生成样本的质量。具体而言，DCGAN 主要是从网络架构上改进了原始的 GAN，主要改进如下：

- 1) DCGAN 的生成器和判别器均去掉 CNN 的池化层，生成器使用反卷积层还原图片，判别器保留 CNN 的整理架构，使用卷积层提取图片特征。
- 2) 在生成器和判别器中都使用 Batch Normalization 层，加速模型训练的同时提升训练的稳定性。
- 3) 生成器除最后一层使用 Tanh 激活函数外，其余层使用 ReLU，判别器所有层均使用 LeakyReLU。

我们希望 DCGAN 能够学习到足够多的近边界数据特征，尝试修改其判定器的目标函数，在保留梯度的情况下将其与源模型的结果相连，得到的结果在同样的生成规模下确实优于原始 DCGAN 的生成情况。然而，考虑到两者的效率，实际情况下生成的结果并无较大区别。

训练 DCGAN 的具体流程如算法2所示。

Algorithm 2 训练 DCGAN 模型

输入： 近边界数据 \tilde{D} ; 批处理大小 $batchsize$; 训练轮次 $epoch$; 损失函数 $Loss$
输出： 训练好的 DCGAN 模型

```

1: 参数初始化:  $learning\ rate \leftarrow 0.0002$ ,  $real\_label \leftarrow 1$ ,  $fake\_label \leftarrow 0$ 
2: for  $i = 1, 2, \dots, epoch$  do
3:   随机噪声  $z \leftarrow 100$ 
4:    $x' \leftarrow G(z)$ 
5:    $Loss1 \leftarrow Loss(D(x), real\_label)$             $\triangleright x$  是近边界数据样本
6:    $Loss2 \leftarrow Loss(D(x'), fake\_label)$ 
7:    $Loss_D \leftarrow Loss1 + Loss2$ 
8:    $Loss_G \leftarrow Loss(D(x'), real\_label)$        $\triangleright$  生成器希望  $D(x')$  接近  $real\_label$ 
9:   使用  $Adam$  更新生成器  $G$ , 判别器  $D$  参数
10:  end for

```

尽管构建的近边界数据已经都位于目标分类边界附近，但我们仍希望近边界数据最大程度上靠近目标分类边界。近边界数据与目标分类边界的距离越近，推断模型所有权成功的可能性就越大。此外，生成的近边界数据虽然只被模型所有者拥有，但对于一些功能易被泛化的模型，近边界的特性仍有可能被

泛化。因此，本文提出使用近边界数据微调源模型的目标分类边界。如式3.4所示， $Loss_{FT}$ 是针对目标分类边界的损失函数。

$$Loss_{FT} = \frac{1}{n} \sum_{i=1}^n (g_t(x'_i) - g_s(x'_i))^2 \quad (3.8)$$

其中 n 是该目标分类边界的近边界数据的数量， x'_i 是生成的近边界数据， $g_t(\cdot)$ 和 $g_s(\cdot)$ 分别表示目标分类概率和源分类概率， $Loss_{FT}$ 本质是希望近边界数据更靠近目标分类边界。 $Loss_{FM}$ 是源模型的损失函数，我们设计两者交替训练微调源模型，与 DCGAN 的过程相似，这是一个博弈的过程，具体的流程如算法3所示。

Algorithm 3 微调源模型

输入： 原始数据集 D ；私有化近边界数据 D' ；批处理大小 $batchsize$ ；训练轮次 $epoch$ ；损失函数 $Loss_{FT}, Loss_{FM}$ ；源模型 M

输出： 微调后的源模型 M'

- 1: 参数初始化： $learning rate \leftarrow 0.0001$
 - 2: **for** $i = 1, 2, \dots, epoch$ **do**
 - 3: $Loss1 \leftarrow Loss_{FT}(g_t(x'), g_s(x'))$ ▷ $g_k(x')$: x' 在第 k 类上的概率
 - 4: $Loss2 \leftarrow Loss_{FM}(M(x), label)$ ▷ $label$ 指正常样本的原始标签
 - 5: 使用 $Adam$ 更新生成器 G ，判别器 D 参数
 - 6: **end for**
-

其中 x 指正常的样本， x' 指 DCGAN 生成器生成的私有化近边界数据。

表 3.1 微调分类边界对模型的影响

数据集	微调前准确率	微调后准确率
CIFAR-10	0.886	0.873
Heritage	0.879	0.866
Intel_image	0.854	0.846

微调目标分类边界使近边界数据与源模型之间的联系更加紧密。注意，我们只微调目标分类边界，且通过交替微调尽可能减少微调对源模型的影响，如

表3.1所示，微调前后源模型的精度差不超过3%，因此，微调对于源模型的性能影响十分微小，甚至可以被忽略，但却有效提高了最后的所有权推断效果。更多微调目标分类边界对准确度的影响测试在第五章第四节中。

第四节 本章小结

本章主要描述了近边界数据的特性和生成近边界数据并将其私有化的过程。对抗性样本一般位于DNN模型分类边界上，并且可以很好的转移到其派生出的模型上。鉴于模型攻击一般会对源模型进行修改，本章提出了鲁棒性更强的近边界数据的概念，数据的近边界性也可以转移到其派生模型上，改变的是近边界数据到分类边界的距离。由于自然的近边界数据很少，本章对比了常见的生成对抗性样本的方法，在CW- L_2 算法的基础上，引入了分类边界距离的概念来更快的生成近边界对抗性样本。考虑到敌手可能会产生近边界数据，我们设计了基于DCGAN的数据生成器，来私有化近边界数据，并在此基础上使用近边界数据微调源模型分类边界，提高所有权推断置信度。

第四章 基于近边界数据的模型所有权推断方法研究

本章将讨论使用模型水印和指纹的方法来做所有权验证的局限性，然后从使用一般数据推断模型所有权的思路出发，提出本文的基于近边界数据推断模型所有权的方法。并且详细介绍了该方法的设计目标和执行流程，最后提出利用假设检验的方法来比对模型的输出结果。

第一节 理论驱动

4.1.1 所有权验证局限性

现有的模型知识产权保护措施着重于被动的防御，只考虑针对模型修改的抗攻击性。模型所有者将水印嵌入训练好的模型或从其中提取抽象的模型知识作为指纹，当怀疑一个模型的知识来自于源模型，模型所有者可以利用水印或指纹被动地从外部验证模型所有权。大多数工作基于这样的思路，设计不同的水印和指纹用于在源模型被盗窃后验证模型所有权，但这并不具有较强的鲁棒性。模型水印的缺陷例如对源模型性能和功能的影响，嵌入水印引起的额外代价都是研究水印工作的关键点。模型指纹目的是提取代表模型知识的固有特征，相较于水印指纹不会对源模型产生影响，因为模型的知识是容易被修改，因此指纹是脆弱的，所有的指纹方法都试图找到可以承受某些修改攻击的强鲁棒性指纹。

本文的目标不仅是抵御一般的模型窃取攻击，还集中在水印和指纹另一个亟待解决的问题歧义攻击上。歧义攻击不关心如何去除水印和指纹以通过模型所有权验证，而是伪造额外的水印和指纹混淆所有权验证。具体来说，盗窃者对源模型嵌入新的水印或提取其他的指纹使原本的保护措施无效。歧义攻击对现有的深度神经网络模型的知识产权保护方法构成了严重威胁，在传统的数字水印领域中有研究表明，除非水印方案是不可逆的^[24]，否则鲁棒性的水印也不一定能验证所有权。在本文中，我们认为通过验证可疑模型是否具有源模型特定的水印或指纹来讨论盗窃行为是不充分的，特别是出现歧义攻击时，因此我们提出推断模型所有权而不是验证。这种方法的灵感来自于数据集推断^[54]提出的所有权决策，我们将在下一小节中具体讨论。

4.1.2 利用数据推断模型所有权

数据集推断做了一个假设：源模型的知识来自于训练数据集。无论盗窃模型是直接攻击源模型还是其副产品，盗窃模型的知识是源模型中包含的知识。如果原始训练数据集是私有的，模型所有者就比对手拥有强大优势，源模型在原始训练数据中的性能要远远优于其他数据集。因此，通过评估多个数据点到决策边界的距离和统计测试相结合，可以得到模型的所有权归属。

源模型的知识被传播到盗窃模型使得所有盗窃模型都必须包含源模型训练数据集中的直接或间接信息。原始训练数据的私有性作为源模型的标识可以用来识别盗窃模型，只需要证明可疑模型和源模型都经过共同的私有数据集训练（不一定完全相同）。此过程和传统的验证模型所有权不同，通过私有数据集推断得到的是一个所有权决策，其中决策的最大者被认为拥有所有权。传统的模型所有权验证是从模型中提取水印或指纹进行匹配从而验证，这里涉及到了歧义攻击导致的验证冲突。从决策过程可以发现数据集推断得到的是一个“最”的概念，因此可以有效避免歧义攻击。所以，本文认为推断所有权将会成为未来模型知识产权保护技术的主要方向之一。

本文的工作受到数据集推断验证模型所有权的启发，我们提出使用数据驱动推断模型所有权代替验证所有权。所有权推断可以在有效证明所有权归属问题的同时，解决所有权验证冲突问题。除此之外，数据驱动的推断所有权意味着该方法只和 DNN 模型的输入输出相关，那么本文提出的方法既可以在白盒环境也可以在黑盒环境下工作。

但是数据集推断仍然具有以下局限性：

- 1) 使用数据集推理的前提是原始训练数据不被窃者得到，公开数据集不能被用于训练源模型。然而，在大多数现实情况中，只有很少一部分工作会构造私有数据集用于训练模型，甚至这部分工作的应用点很狭窄，这意味着被窃的风险较小。因此，依赖于私有数据集的数据集推理方法在实际应用中使用范围很小，不能被大幅度推广使用；
- 2) 数据集推理方法的核心思想是源模型的功能在训练数据上的效果优于其他数据，但存在模型的功能可能相似，而结构和训练数据都不同的情况，因此该方法可能会导致误导。Li 等人^[55] 验证了此限制，结果表明该方法产生的结果值得怀疑。

我们指出，利用数据推断所有权的方法需要解决以上问题，因此我们提出

构造私有化近边界数据作为推断依据，并利用近边界数据靠近决策边界的特性解决模型功能相似引起的误导。这是因为即使模型功能相似，但是决策边界不可能完全相同。

第二节 近边界数据推断模型所有权

在本文中，我们提出了近边界数据，一种分布在分类边界附近的特殊样本。模型指纹^[32] 使用对抗性样本抽象地反映模型分类边界，同一组对抗性样本的输入，其引起的决策模式的变化可以用于比较模型知识的相似性，但这种方法是脆弱的，一般的模型窃取攻击都会修改源模型，而对模型的任意修改操作都有可能破坏这种特性。因此，我们不直接比较决策模式的变化，它是不可信任的，而是比较对抗性样本与决策边界的距离。大多数对抗性样本都是位于决策边界附近的，也就是说，它们与决策边界的距离很近。对抗性样本的这种性质被我们所利用并构造近边界数据，经过测试我们发现绝大多数的模型窃取方法都无法改变这种结果，即使样本分类被影响，其仍然位于分类边界附近。近边界数据背后的意义是如果被用于所有权验证如果两个模型的决策模式相似，参与训练的近边界数据一定可以反映出来。受到这个的启发，将近边界数据作为水印验证所有权是传统的思路，即使不会对模型的精度造成影响，然而这样的水印是脆弱的，很难抵御歧义攻击，因此我们提出由近边界数据驱动的所有权推断方法，其思想是构造私有的近边界数据，当验证一个模型的所有权时，模型所有者和盗窃者分别提供各自的私有化近边界数据，距离分类边界最近的被推断获得所有权。这个方法的主要思想如图4.1所示。

4.2.1 设计目标

依据现有的工作，我们的方法在模型训练后进行部署，且在黑盒环境中推断所有权。我们的方法不关注模型盗窃的过程，目的是准确推断受害者所有权和识别可疑模型的盗窃行为。现在大多数所有权验证技术都是黑盒模型环境，因为模型所有者和攻击者通常不会提供完整模型。我们提出的方法仅利用模型提供的外部 API，获取近边界数据的决策结果，从而推断模型所有权。在通常的假设中，存在一个官方的仲裁机构，当对任一模型产生所有权怀疑时，受害者和可疑对手可以向机构提出申请并提供各自的私有化近边界数据，并通过我们的方法推断所有权。注意无论是在白盒的环境还是在黑盒的环境中，本文的方法

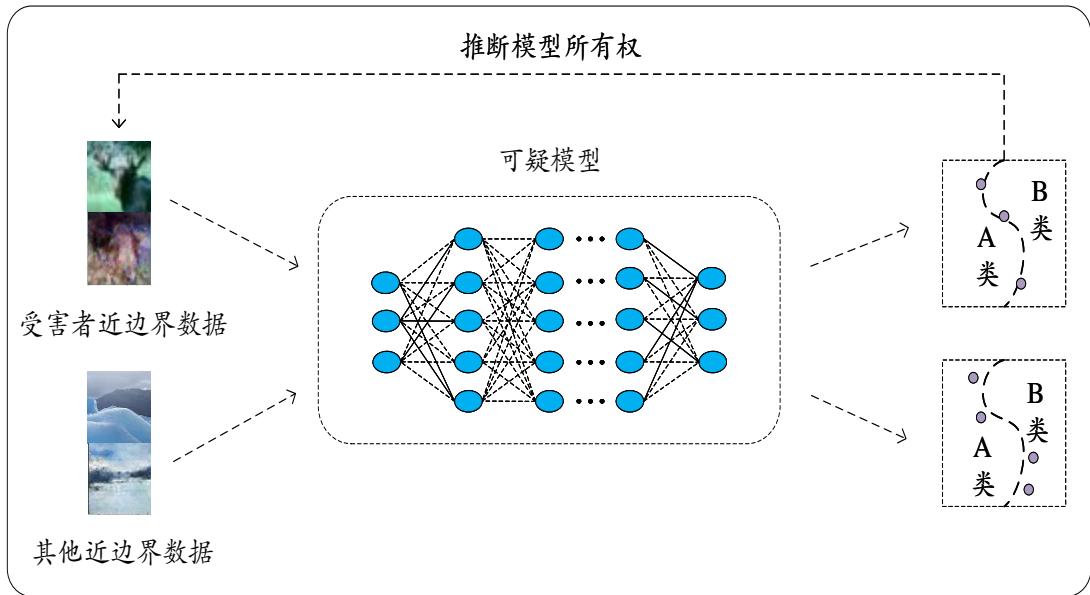


图 4.1 近边界数据推断所有权

均可以用来推断模型所有权。

为了实现成功推断模型所有权，本文提出的方法的设计目标是：

- 1) **精确性**: 推断模型所有权的方法不应该影响模型的性能，模型的最大可接受测试精度下降不超过 3%。为了增加推断成功的置信度，再生成私有近边界数据后，我们会利用近边界数据微调源模型，这不应该对模型的性能造成很大的影响，可以接受的精度下降不超过 3%。
- 2) **可转移性**: 如果可疑模型与源模型相同或来自源模型，则近边界数据在这些模型中均表现出近边性，而在无关模型中则无明显特征。
- 3) **有效性**: 如果可疑模型与源模型相同或来自源模型，则根据源模型构造的私有近边界数据在这些模型中距离指定的分类边界最近。
- 4) **鲁棒性**: 近边界数据应该对常见的模型修改（如模型微调、剪枝和有损压缩）具有鲁棒性。
- 5) **不可见性**: 敌手无法获得私有的近边界数据，也无法在视觉上观察到近边界数据的部署。
- 6) **高效性**: 通过近边界数据推断模型所有权应该能够高效地计算距离边界数据，并通过对全部近边界数据的决策结果确定可疑模型是否是盗窃模型。

4.2.2 方法概述

为了实现以上目标，本文提出了一种基于近边界数据的模型所有权推断方法。

问题定义：我们定义了一个深度神经网络（DNN）分类器 G 作为源模型，给定一个原始训练集 D ，假设该源模型是一个 n -类的 DNN 分类器，分类器的输出层为 softmax 层或其他决策层，决策函数 $g_j(x)$ 表示数据样本 x 被分到第 j 类的概率，其中 $j = 1, 2, \dots, n$ 。 Z_1, Z_2, \dots, Z_n 表示模型分类器的全部决策函数输出，其结果可作为分类边界的依据被我们使用，因此

$$g_j(x) = \frac{\exp(Z_j(x))}{\sum_{i=1}^n \exp(Z_i(x))} \quad (4.1)$$

其中，数据样本 x 的标签 y 被推断拥有最大概率的类别，例如 $y = \arg \max_j g_j(x) = \arg \max_j Z_j(x)$ 。

通常来说，寻找位于分类边界上的数据点采用重复随机采样数据点的方法，具体地如果数据点满足上述定义则数据点在分类边界上。然而，简单的重复采样可能需要大量的时间消耗，甚至无法找到这样的数据点们。为了解决这样的问题，我们在第三章中讨论了如何构造位于分类边界上或其附近的的数据点，且将其私有化的过程。

基于第三章的讨论，我们提出构造近边界数据推断模型的所有权，而不是验证所有权。具体而言，如图4.2所示，我们的方法包括三个主要阶段：

- 1) 从数据集样本中生成对抗性样本；
- 2) 训练生成对抗模型生成私有化的近边界数据；
- 3) 使用近边界数据微调源模型。

4.2.3 假设检验

根据第四章第一节讨论的结果，本文认为过去的验证模型所有权的思路具有较大的局限性，大多数研究无法抵御歧义攻击。因此，我们提出了推断模型所有权的想法，这是一种“最”的思路。在现实情况中，我们假设存在第三方仲裁机构，并约定目标分类边界，被盗窃者向第三方机构提出仲裁并提供近边界数据，盗窃者同样需要提供相应的近边界数据，第三方机构分别计算目标分类边界距离，本文认为持有最靠近目标分类边界的近边界数据所有者将获得模型

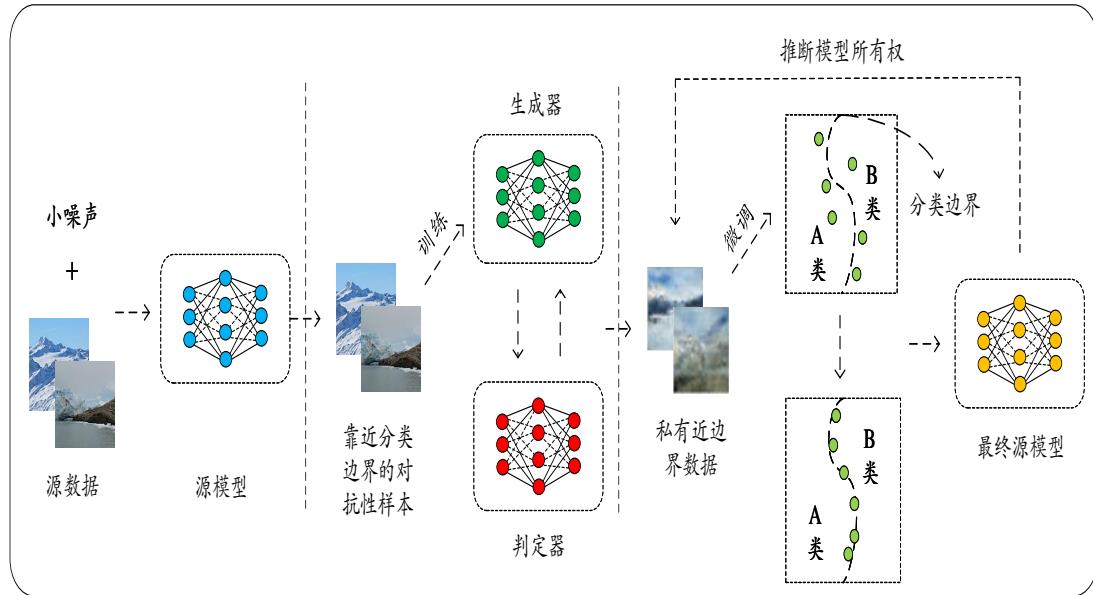


图 4.2 方法整体流程图

所有权。注意，由于近边界数据通常是一组数据，所以应该根据统计的结果来看。在实验中，我们计算了不同规模的近边界数据组在源模型，盗窃模型以及不相关模型上到分类边界的距离，并设计了一种基于假设检验的方法来表现推断的置信度。

假设检验：我们假设事件 C 是模型所有者提供的私有近边界数据在可疑模型上的计算结果，事件 C_S 表示盗窃者提供的近边界数据在可疑模型上的计算结果，或模型所有者提供的私有近边界数据在无关模型上的计算结果。本文计算假设 $H_0: \mu \geq \mu_S$ ($H_1: \mu < \mu_S$) 的 p 值，以及差异大小 $\Delta\mu = \mu_S - \mu$ ， $\Delta\mu$ 越大，推断可信度越高。如果 p 值低于预定义的置信度评分 α ，则拒绝 H_0 ，并称正在测试的模型是被盗模型。我们重复 30 次统计性实验以提高可信度。

假设检验的具体过程如算法4所示。

第三节 本章小结

本章主要描述了本文提出的基于近边界数据的模型所有权推断方法。首先介绍了该方法的理论驱动，接着针对现有方法存在的问题提出本文方法的设计目标，即精确性、可转移性、有效性、鲁棒性、不可见性和高效性。然后对实现方法做了概述，并介绍了该方法的主要流程：从数据集样本中生成对抗性样本，

Algorithm 4 假设检验

输入: 模型所有者私有近边界数据样本 X ; 可疑对手近边界数据样本 X_S ; 可疑模型 \tilde{M} ; 假设检验对照表 T ; 显著性水平 α

输出: 可疑模型是否为盗窃模型

- 1: 原假设: $H_0: \mu \geq \mu_S$
 - 2: 备择假设: $H_1: \mu < \mu_S$
 - 3: 计算模型所有者私有近边界数据样本均值 \bar{X}
 - 4: 计算可疑对手近边界数据样本均值 \bar{X}_S
 - 5: 计算统计量 t
 - 6: 查对照表 T 获得临界值 λ
 - 7: **if** $t > \lambda$ **then**
 - 8: $p < \alpha$, 拒绝 H_0 , 接受 H_1 , 可疑模型是被盗模型
 - 9: **else**
 - 10: $p > \alpha$, 不拒绝 H_0
 - 11: **end if**
-

训练生成对抗模型私有化近边界数据和使用近边界数据微调源模型。最后, 提出了使用假设检验的方法来对模型所有者和可疑对手的输出结果进行比对, 推断模型所有权。

第五章 基于近边界数据的模型所有权推断方法分析

本文选择在开源数据集 CIFAR-10^[56], Heritage^[57], Intel_image^[58] 上进行实验，并选择 ResNet18 作为评估的源模型，VGG11 作为对照的无关模型。本章将从生成初始近边界数据的方法选择，数据近边界特性评估，微调目标分类边界的影响，推断模型所有权的有效性和不同规模近边界数据的可伸缩性扩展这几个方面对本文提出的方法进行测评和分析。

被盗模型：本文设置了几种常见的模型盗窃方法，包括模型微调，模型剪枝（不同的剪枝率）和模型知识蒸馏，并在源模型的基础上派生得到被盗模型。

第一节 实验设置

5.1.1 数据集

CIFAR-10：CIFAR-10 共有 10 个类别，其中训练集包含 50000 张大小为 32x32 的图像，测试集包含 10000 张大小为 32x32 的图像。

Heritage：Heritage 共有 10 个类别，其中训练集包含 10235 张大小为 128x128 的图像，测试集包含 1404 张大小为 64x64 的图像。

Intel_image：Intel_image 共有 6 个类别，其中训练集包含 14034 张大小为 150x150 的图像，测试集包含 3001 张大小为 150x150 的图像。

5.1.2 目标模型

本文的目标模型选用 ResNet18，在上述三个数据集上分别训练作为实验源模型，使用 VGG11 作为无关的对照模型。ResNet18 和 VGG11 的参数信息如表5.1所示。由于 CIFAR-10 的图片尺寸较小，所以训练的时候将原始 ResNet18 中首层使用的 7x7 卷积核改成 3x3，步长和填充随之改为 1，并且舍弃最大池化层，更改结构后 ResNet18 在 CIFAR-10 上的预测准确度得到了提升。

表 5.1 模型参数信息

模型	层数	计算量/亿	参数量/百万
ResNet18	18	9.559	11.670
VGG11	11	47.022	132.863

5.1.3 实验环境和参数设置

本文在实验中的使用的硬件与软件配置如表5.2所示。

表 5.2 硬件与软件配置

硬件/软件	版本
操作系统	Ubuntu 20.04 LTS
CPU	Intel Core i7-11700KF @ 3.6GHz
显卡	NVIDIA GeForce RTX 3080 Ti
CUDA 版本	11.6
机器学习框架	pytorch 1.9.0
Python	3.7.11

训练过程中 Adam 优化器并将学习率 (Learning rate), 迭代轮次 (Epoch) 和每批次大小 (Batch size) 分别设置为 0.0001,200 和 64。蒸馏模型实验选择从 Resnet18 蒸馏至 VGG11, 蒸馏时将蒸馏温度设置为 20 并且教师模型比例 $\alpha=0.7$, 训练轮次是 20。初始近边界数据生成采用 CW- L_2 算法, 实验中选择有目标的生成方式, 且学习率, 迭代次数和二分搜索次数分别设置为 0.001,1000 和 6, 其他参数为默认值。私有近边界数据生成器采用 DCGAN 的基础结构, 训练过程使用 Adam 优化器且将学习率, 训练轮次和每批次大小分别设置为 0.0002, 8000 和 64。本方法最后微调源模型阶段需要交替使用源模型损失函数和微调目标边界的损失函数来微调源模型, 具体设置为 10 个轮次交替一次且交替次数最多为 10 次。

第二节 生成初始近边界数据的算法选择

本小节将对第三章第二节中提到的 FGSM, IGSM, RFGSM 和 CW- L_2 进行测试, 除对 CW- L_2 进行了改进外, 其余均使用原作者发布的实现。FGSM, IGSM, RFGSM 中均有一个用于界定噪声 ϵ 的参数。我们进行大量的实验探索选择合适的参数用于与 CW- L_2 进行比较。此外, CW- L_2 的实验设置如第一节所示。

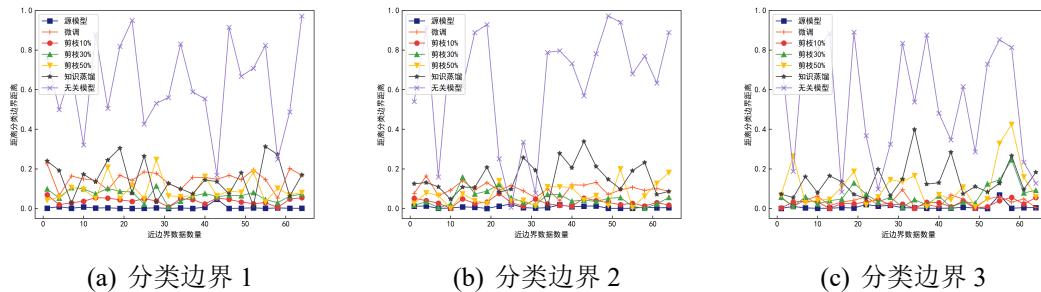
表 5.3 不同对抗性样本生成算法生成的数据与目标分类边界的平均距离

数据集	FGSM	IGSM	RFGSM	CW- L_2
CIFAR-10	0.557	0.430	0.418	0.066
	0.461	0.419	0.373	0.103
	0.586	0.369	0.356	0.112
Heritage	0.347	0.356	0.314	0.014
	0.377	0.340	0.281	0.016
	0.348	0.332	0.276	0.010
Intel_image	0.522	0.447	0.353	0.088
	0.475	0.506	0.387	0.122
	0.468	0.402	0.428	0.127

FGSM, IGSM, RFGSM 和 CW- L_2 到分类边界的距离如表5.3所示。从表中可以看出 FGSM 的效果较差, 这是因为 FGSM 生成对抗性样本只进行一次噪声的添加, 效率非常高, 在追求效率的情况下牺牲了性能, 这是合理的。相比于 FGSM, IGSM 和 RFGSM 的效果稍好, 这是在算法中引入迭代的效果, 通过每次一小步的迭代, 不断地在上一次的基础上添加噪声, 使得图片以更小的幅度变化。这和表5.3中大部分情况下, IGSM 和 RFGSM 的生成的对抗性样本距离分类边界的平均距离比 FGSM 生成的近是相符的。改进过的 CW- L_2 是这几种方法中效果相当显著, 距离分类边界的距离比其他三种方法小数十倍。这是因为本身这个算法机制比较复杂, 在迭代的过程中引入了二分查找和神经网络来训练参数以更好的生成图片, 我们进一步在算法中引入了到分类边界距离这个参数, 使得生成图片朝着距离更小的方向改进, 并且达到我们预定的阈值 θ 时, 提前终止算法, 一定情况下缓解该算法效率低下的问题。

第三节 数据近边界特性的评估

本文提出的近边界数据不仅在源模型在表现出了近边界性，在所有盗窃模型中也表现出了这个特点，数据近边界特性的可转移性，是本文提出方法的重要基础。本节将在三个数据集和不同分类边界上对近边界数据的近边界性和可转移性进行测评。

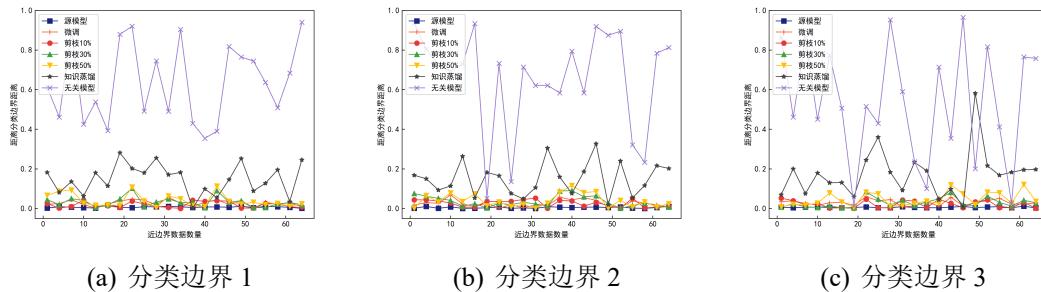


(a) 分类边界 1

(b) 分类边界 2

(c) 分类边界 3

图 5.1 CIFAR-10 上不同分类边界下的近边界数据表现

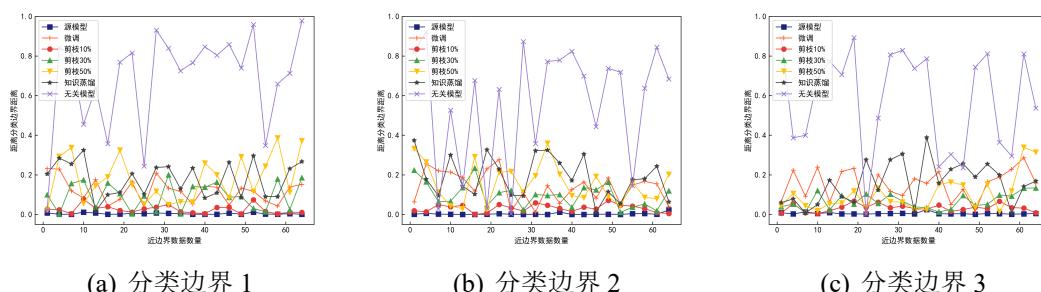


(a) 分类边界 1

(b) 分类边界 2

(c) 分类边界 3

图 5.2 Heritage 上不同分类边界下的近边界数据表现



(a) 分类边界 1

(b) 分类边界 2

(c) 分类边界 3

图 5.3 Intel_image 上不同分类边界下的近边界数据表现

如图5.1, 图5.2, 图5.3所示, 本文提出的近边界数据在所有盗窃模型中都表现出了靠近分类边界的特性。从图中可以看出, 相比于其他所有模型, 近边界数据在源模型上距离分类边界的距离最近, 这主要是三方面的原因:

- 1) 初始的近边界样本是在源模型的基础上生成的, 并且从众多生成对抗性样本的方法中挑选了效果最好的 CW-L₂ 方法, 然后对该方法进行改进, 使之生成距离分类边界更近的对抗性样本。
- 2) 使用 DCGAN 生成器私有化近边界数据后, 设计了新的损失函数微调源模型, 这使得私有近边界数据同样非常靠接模型分类边界。
- 3) 被盗模型在从源模型派生的过程中, 涉及到模型的修改, 这些修改操作虽然不会使近边界数据在这些模型上失去近边界性, 但是会使它们稍微偏离模型分类边界。

近边界数据在所有被盗模型中表现出近边界性说明近边界性是可以跟随源模型一起转移的, 这跟 DNN 分类器有很大的关系。对抗性样本一般位于分类边界上, 可以反应分类边界, 并且这类特殊的样本具有可转移性, 近边界数据是利用对抗性样本构造的, 同样具有这种特性。这是本文提出方法的重要基础, 因为我们就是要利用所有者的私有近边界数据和可疑队友提供的近边界数据到分类边界的距离来推断模型所有权 (距离近者获得所有权), 如果私有近边界数据不表现出近边界性, 那么就会判定失败。

从图中可以发现, 相比于其他模型盗窃方法, 近边界数据在知识蒸馏派生出的模型上距离分类边界距离稍远。这是因为知识蒸馏对模型的修改特别大, 通常来说会更换模型的架构, 然后在通过蒸馏的方法训练。模型知识蒸馏对于其他模型知识产权保护方法也是一种挑战, 近边界数据在蒸馏产生的模型上仍然表现出近边界性, 因此本文提出的方法对这种强修改的盗窃方法依然适用。

图中还有一个点是近边界数据在无关模型上并不反映出近边界性, 这是非常重要的。本文提出的方法不应该对正常的无关模型产生误判, 混淆所有权的归属。

在这一阶段, 我们使用的近边界数据大小为 64。也就是说, 本文的方法在较小规模近边界数据时效果依然显著, 那么在增大数据量时, 效果会更加明显。在第六节中, 我们会对不同规模的近边界数据进行测试。

第四节 微调目标分类边界的影响

本小节将对第三章第三节中使用私有近边界数据微调源模型产生的影响进行测试。我们针对不同数据集训练得到的源模型，使用不同规模的近边界数据以及不同的目标分类边界对源模型进行微调。

表 5.4 CIFAR-10 上不同对抗性样本生成算法生成的数据与目标分类边界的平均距离

数据集 (准确率)	分类边界	数据数量	微调后准确率
CIFAR-10 (0.886)	分类边界 1	64	0.873
		128	0.862
		256	0.862
		512	0.856
	分类边界 2	64	0.871
	分类边界 3	128	0.870
		256	0.860
		512	0.859
	分类边界 4	64	0.871
		128	0.868
		256	0.858
		512	0.855
	分类边界 5	64	0.873
		128	0.873
		256	0.866
		512	0.862
		64	0.876
		128	0.866
		256	0.868
		512	0.861

如表5.4, 表5.5, 表5.6所示, 随着微调模型近边界数量的增多, 模型准确率逐渐下降, 但是在几乎全部情况下, 源模型微调前后的精度差没有超过3%, 这是本文方法可以接受的范围。因为私有近边界数据本身不是原始数据集的一部分, 所以使用规模变大时, 对源模型的影响也会变大。但是在另一个角度, 微调源模型的目的是使得我们的私有近边界数据更加靠近分类边界, 来提高后续推

表 5.5 Heritage 上不同对抗性样本生成算法生成的数据与目标分类边界的平均距离

数据集 (准确率)	分类边界	数据数量	微调后准确率
Heritage (0.879)	分类边界 1	64	0.862
		128	0.858
		256	0.854
		512	0.848
	分类边界 2	64	0.867
		128	0.862
		256	0.859
		512	0.851
	分类边界 3	64	0.865
		128	0.857
		256	0.855
		512	0.851
	分类边界 4	64	0.863
		128	0.860
		256	0.854
		512	0.849
	分类边界 5	64	0.866
		128	0.861
		256	0.857
		512	0.853

断模型所有权的置信度，所以，使用更多的近边界数据微调源模型效果会更好。因此，在实际情况中，微调数据规模的选择是一个模型精度和推断置信度的折衷。

表中整体情况下，模型的准确率都下降不多。一方面，这是因为本身微调的数据和源模型的训练数据相比只是一小部分，不会对模型产生太大的影响。另一方面，在微调源模型时，我们将学习率设置为 0.0001，这是很低的。并且使用原始数据对模型进行交替训练，训练轮次不超过 10 次。所以微调之后，模型准确率没有受到很大的影响。

实验结果证明，本文的私有近边界数据在应用于模型所有权保护问题时，模

表 5.6 Intel_image 上不同对抗性样本生成算法生成的数据与目标分类边界的平均距离

数据集 (准确率)	分类边界	数据数量	微调后准确率
Intel_image (0.854)	分类边界 1	64	0.825
		128	0.829
		256	0.826
		512	0.839
	分类边界 2	64	0.843
		128	0.839
		256	0.828
		512	0.824
	分类边界 3	64	0.841
		128	0.833
		256	0.831
	分类边界 4	512	0.823
		64	0.847
		128	0.843
		256	0.838
	分类边界 5	512	0.831
		64	0.846
		128	0.834
		256	0.829
		512	0.825

型具有较好的保真度。因此，使用近边界数据推断模型所有权不用担心源模型受到较大的影响。

第五节 推断模型所有权的有效性

本文提出的方法目的是推断模型所有权，本节将对此方法的有效性进行测试。根据本文提出方法的流程，模型所有者和可疑对手均需要提供各自的近边界数据，然后输入模型获得输出，再根据4.2.3中提到的假设检验进行结果比对，判断可疑模型是否从源模型派生。

表 5.7 推断模型所有权

数据集	攻击方法	分类边界 1		分类边界 2		分类边界 3		分类边界 4		分类边界 5	
		$\Delta\mu$	p 值								
CIFAR-10	源模型	0.913	10^{-6}	0.954	10^{-6}	0.927	10^{-5}	0.967	10^{-5}	0.958	10^{-5}
	模型微调	0.718	10^{-5}	0.745	10^{-6}	0.698	10^{-5}	0.692	10^{-4}	0.729	10^{-5}
	剪枝 10%	0.572	10^{-5}	0.487	10^{-5}	0.458	10^{-5}	0.533	10^{-4}	0.512	10^{-4}
	剪枝 30%	0.537	10^{-4}	0.497	10^{-4}	0.401	10^{-3}	0.428	10^{-4}	0.587	10^{-4}
	剪枝 50%	0.545	10^{-4}	0.614	10^{-4}	0.506	10^{-3}	0.570	10^{-4}	0.484	10^{-3}
	知识蒸馏	0.372	10^{-3}	0.297	10^{-3}	0.288	10^{-3}	0.308	10^{-3}	0.340	10^{-3}
Heritage	源模型	0.876	10^{-5}	0.845	10^{-5}	0.859	10^{-4}	0.801	10^{-4}	0.837	10^{-5}
	模型微调	0.815	10^{-5}	0.792	10^{-4}	0.824	10^{-4}	0.833	10^{-4}	0.784	10^{-4}
	剪枝 10%	0.530	10^{-4}	0.535	10^{-3}	0.508	10^{-4}	0.486	10^{-3}	0.471	10^{-3}
	剪枝 30%	0.491	10^{-3}	0.452	10^{-3}	0.469	10^{-4}	0.470	10^{-3}	0.427	10^{-4}
	剪枝 50%	0.502	10^{-3}	0.517	10^{-3}	0.434	10^{-3}	0.451	10^{-3}	0.490	10^{-3}
	知识蒸馏	0.329	10^{-3}	0.365	10^{-2}	0.238	10^{-3}	0.310	10^{-3}	0.274	10^{-3}
Intel_image	源模型	0.859	10^{-5}	0.896	10^{-4}	0.872	10^{-4}	0.899	10^{-4}	0.914	10^{-4}
	模型微调	0.717	10^{-5}	0.784	10^{-4}	0.752	10^{-4}	0.791	10^{-3}	0.709	10^{-4}
	剪枝 10%	0.451	10^{-4}	0.522	10^{-4}	0.539	10^{-3}	0.472	10^{-3}	0.438	10^{-4}
	剪枝 30%	0.407	10^{-4}	0.415	10^{-4}	0.346	10^{-3}	0.382	10^{-3}	0.395	10^{-3}
	剪枝 50%	0.370	10^{-3}	0.395	10^{-3}	0.327	10^{-3}	0.360	10^{-3}	0.458	10^{-3}
	知识蒸馏	0.336	10^{-2}	0.395	10^{-3}	0.360	10^{-2}	0.308	10^{-3}	0.287	10^{-2}

在本节中，我们讨论本文方法生成的私有近边界数据与其他近边界数据在盗窃模型上的性能对比。首先，我们模拟了盗窃者可能会提供的近边界数据，该数据由两部分组成，包括(1)从原始数据中挑选出的近边界数据，(2)由FGSM和CW生成的一些对抗性样本。然后针对不同的目标分类边界，进行假设检验并计算在不同数据集和不同盗窃模型上的 $\Delta\mu$ 和p值。

如表5.7所示，p值在每个数据集，每条分类边界上呈从上到下增大的趋势， $\Delta\mu$ 呈减小趋势，在全部情况中，p值均低于0.05。也就是说，本文的方法在不同的盗窃方法中推断模型的所有权均有显著的效果，我们至少有95%以上的置

信度确定可疑模型是盗窃模型。

在假设检验中， p 值越小， $\Delta\mu$ 越大说明结果越可靠，推断的置信度越高。表中从上到小 p 值减小是因为这些方法对模型的修改逐渐增大，尤其是在知识蒸馏上，模型知识蒸馏是本文方法的最大挑战，也同样是其他研究面临的巨大挑战。在我们的实验中，可以观察到我们的方法始终可以将蒸馏模型推断为被盗模型。因此，实验结果表明使用私有的近边界数据对大多数模型盗窃技术都是可靠的，我们可以声称模型被盗窃的置信度至少为 95%，证明了本文方法的有效性和鲁棒性。

第六节 不同规模近边界数据的可伸缩性扩展

在本节中，我们将测试不同规模的近边界数据在推断模型所有权上的可伸缩性。我们的方法需要对数据进行采样从而进行假设检验，通常来说样本数量越大，对检验过程中因随机因素而产生的不利影响就会越小，更能准确的推断模型所有权。

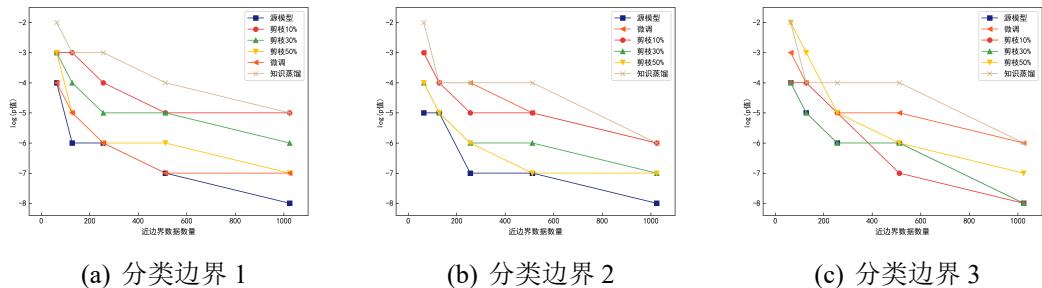


图 5.4 CIFAR-10 上推断模型所有权的扩展性

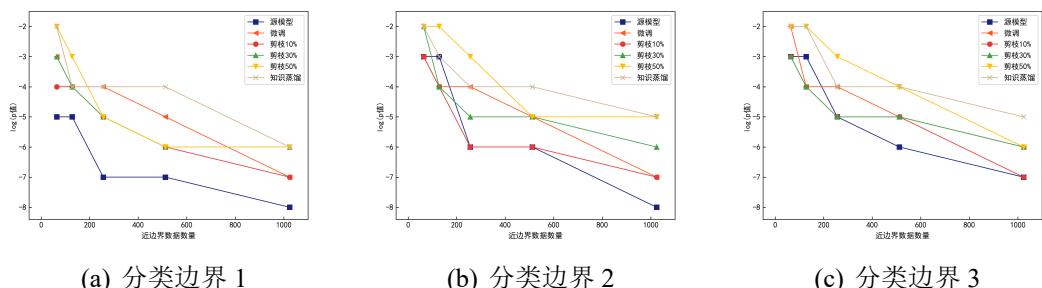


图 5.5 Heritage 上推断模型所有权的扩展性

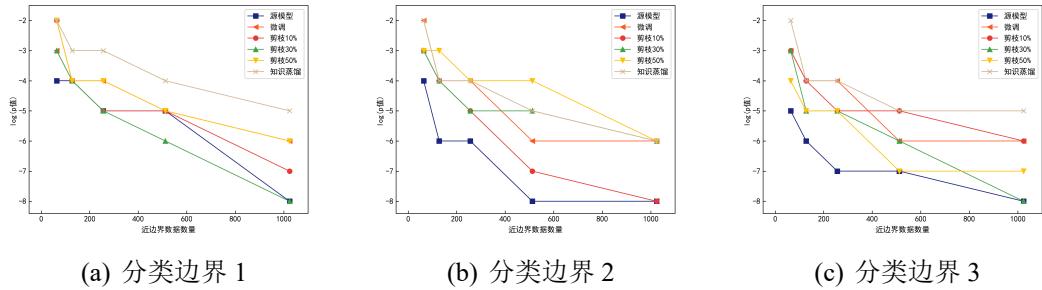


图 5.6 Intel_image 上推断模型所有权的扩展性

如图5.4, 图5.5, 图5.6所示, p 值在不同数据集, 不同分类边界上, 随着近边界数据规模的增大而减小, 而 p 值越小说明假设检验能够以更高的置信度确定可疑模型是被盗模型。

本文的方法在进行假设检验前, 需要将私有的近边界数据和可疑对手的近边界数据分别输入可疑模型。可疑对手近边界数据是通过从原始数据挑选和 FGSM, CW 生成的对抗性样本组成的。因此, 由于随机因素, 存在一小部分可疑对手的数据到分类边界的距离和我们的私有近边界数据相近, 甚至比我们的小, 这会导致一定的误判情况。随着测试样本的增大, 这种随机因素的影响会被逐渐消除, 所以从图中可以观察到, 随着近边界数据规模的增大, p 值逐渐减小。但这并不说明我们的方法对小数量的近边界数据缺少鲁棒性, 从图中我们可以观察到即使数据量为 64 的情况下, p 值仍然小于 0.05, 这证明我们的方法对于小数样本量同样有显著的效果。

第七节 本章小结

本章我们在 CIFAR-10, Heritage, Intel_image 这三个数据集上对本文提出的方法进行了全面的测评分析。首先对各种对抗性样本生成方法进行测试, 初始近边界数据生成算法的选择非常重要, 这影响到生成数据到分类边界的距离, 是方法最后假设检验的指标。接着对数据近边界特性进行了评估, 结果表明近边界数据的近边界性可以从源模型转移到派生出的模型上, 这是本文方法的重要基础。然后测试了微调分类边界对模型准确率的影响, 因为本文的方法不应该对模型精度造成很大的影响, 否则该方法失去了意义。接着测试了本文方法推断模型所有权的有效性, 结果表明该方法对不同的模型盗窃方法均能以 95%

以上的置信度声明可疑模型是盗窃模型。最后对假设检验样本规模进行了扩展，在更大规模的情况下，本文的方法会更加有效，当然，也适用于类似 64 的小样本数据情况。

第六章 总结与展望

第一节 工作总结

DNN 在给人类社会生活带来便利的同时，也带来了严重的知识产权问题，模型水印和模型指纹是当前保护 DNN 知识产权的两种主要方法。

在本文中，我们讨论了使用模型水印和模型指纹验证模型所有权的局限性，提出了用推断模型所有权代替验证所有权的新思路。我们认为可以从数据驱动的角度抵御模型盗窃，即在源模型上找到一种可以量化的属性，并且这种属性会被源模型派生出的模型继承，那么就可以从这个角度设计算法来推断模型的所有权。

从数据驱动的角度，本文提出了一种基于近边界数据的模型所有权推断方法。该方法首先比较并选择 CW- L_2 方法生成近边界对抗性样本。为了防止他人轻易复制近边界数据，我们考虑将其私有化，所以训练了一种基于 DCGAN 的近边界生数据成器用以将近边界数据私有化和扩展，实验测试了生成器能够显著地学习近边界数据的特征并生成新的数据。为了提升推断所有权的置信度，我们设计了新的损失函数微调源模型分类边界，得到最终版本的源模型和近边界数据。最后提出使用假设检验的方法来比对私有近边界数据和其他近边界数据的结果，成功推断模型所有权。

我们在 CIFAR-10, Heritage 和 Intel_image 这三个公开数据集上针对生成初始近边界数据的方法选择，数据近边界特性评估，微调目标分类边界的影响，推断模型所有权的有效性和不同规模近边界数据的可伸缩性扩展这几个方面做了详细的测评和分析，实验证明了我们的方法可以高置信度地推断模型所有权，同时对不同的模型盗窃方法具有很强的鲁棒性。

第二节 工作展望

如何合理有效的保护模型的知识产权已经成为 DNN 领域的热点研究方向，本文提出数据驱动来推断模型所有权代替一般的验证所有权。本文提出了基于

近边界数据的模型所有权推断方法，并展现出了不错的效果，但仍存在一些不足：

- 1) 本文提出的方法主要针对小分类情况下的 DNN 分类模型。在大分类的情况下，如何选择合适的分类边界计算距离值得探讨。如果大分类模型被迁移到小分类模型上引起类别发生变化，原始的类别应该如何映射到新类别。因此未来的工作应该加入大分类情况下研究。
- 2) 本文提出的方法主要是针对 DNN 分类模型的。对于其他的 DNN 模型，如何找到类似分类模型分类边界概念来做具体的量化计算是数据驱动推断模型所有权的关键所在。
- 3) 虽然本文对 CW- L_2 方法进行了改进，一定程度上加快了算法的效率，但是算法整体由于二分查找加迭代的方式仍然显得效率低下。在未来的工作中，应该探索出一种效果相当但是效率更快的方式生成近边界数据。

综上，本文提出的方法还有很大的探索空间。除此之外，未来的工作应该研究的更多不限于模型水印和指纹的方法，以更好的保护 DNN 模型的知识产权。

参考文献

- [1] WINSTON P H. Artificial intelligence. [M]. [S.l.]: Addison-Wesley Longman Publishing Co., Inc., 1984.
- [2] SZE V, CHEN Y.-H, YANG T.-J, et al. Efficient processing of deep neural networks: A tutorial and survey. [J]. Proceedings of the IEEE, 2017, 105 (12): 2295–2329.
- [3] HE K, ZHANG X, REN S, et al. Identity mappings in deep residual networks. [C] // Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14. Springer. [S.l.]: [s.n.], 2016: 630–645.
- [4] CORTES C, LAWRENCE N, LEE D, et al. Advances in neural information processing systems 28. [C] // Proceedings of the 29th Annual Conference on Neural Information Processing Systems. [S.l.]: [s.n.], 2015.
- [5] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition. [J]. ArXiv preprint arXiv:1409.1556, 2014.
- [6] NASSIF A B, SHAHIN I, ATTILI I, et al. Speech recognition using deep neural networks: A systematic review. [J]. IEEE access, 2019, 7: 19143–19165.
- [7] COLLOBERT R, WESTON J, BOTTOU L, et al. Natural language processing (almost) from scratch. [J]. Journal of machine learning research, 2011, 12 (ARTICLE): 2493–2537.
- [8] WU Y, SCHUSTER M, CHEN Z, et al. Google’s neural machine translation system: Bridging the gap between human and machine translation. [J]. ArXiv preprint arXiv:1609.08144, 2016.
- [9] XIONG W, DROOPPO J, HUANG X, et al. Achieving human parity in conversational speech recognition. [J]. ArXiv preprint arXiv:1610.05256, 2016.
- [10] CHEN C, SEFF A, KORNHAUSER A, et al. Deepdriving: Learning affordance for direct perception in autonomous driving. [C] // Proceedings of the IEEE international conference on computer vision. [S.l.]: [s.n.], 2015: 2722–2730.
- [11] ESTEVA A, KUPREL B, NOVOA R A, et al. Dermatologist-level classification of skin cancer with deep neural networks. [J]. Nature, 2017, 542 (7639): 115–118.
- [12] SILVER D, SCHRITTWIESER J, SIMONYAN K, et al. Mastering the game of go without human knowledge. [J]. Nature, 2017, 550 (7676): 354–359.
- [13] BROWN T, MANN B, RYDER N, et al. Language models are few-shot learners. [J]. Advances in neural information processing systems, 2020, 33: 1877–1901.
- [14] CHEN H, ROUHANI B D, FAN X, et al. Performance comparison of contemporary DNN watermarking techniques. [J]. ArXiv preprint arXiv:1811.03713, 2018.

- [15] DARVISH ROUHANI B, CHEN H, KOUSHANFAR F. Deepsigns: An end-to-end watermarking framework for ownership protection of deep neural networks. [C] // Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems. [S.l.]: [s.n.], 2019: 485–497.
- [16] RIBEIRO M, GROLINGER K, CAPRETZ M A. Mlaas: Machine learning as a service. [C] // 2015 IEEE 14th international conference on machine learning and applications (ICMLA). IEEE. [S.l.]: [s.n.], 2015: 896–902.
- [17] OFOEDA J, BOATENG R, EFFAH J. Application programming interface (API) research: A review of the past to inform the future. [J]. International Journal of Enterprise Information Systems (IJEIS), 2019, 15 (3): 76–95.
- [18] TRAMÈR F, ZHANG F, JUELS A, et al. Stealing Machine Learning Models via Prediction APIs. [C] // USENIX security symposium. Vol. 16. [S.l.]: [s.n.], 2016: 601–618.
- [19] DUDDU V, SAMANTA D, RAO D V, et al. Stealing neural networks via timing side channels. [J]. ArXiv preprint arXiv:1812.11720, 2018.
- [20] VAN SCHYNDEL R G, TIRKEL A Z, OSBORNE C F. A digital watermark. [C] // Proceedings of 1st international conference on image processing. Vol. 2. IEEE. [S.l.]: [s.n.], 1994: 86–90.
- [21] 刘根, 赵翔宇, 王子驰, 等. 面向深度模型的多用户水印系统. [J]. 工业控制计算机, 2022 (53-55+58).
- [22] UCHIDA Y, NAGAI Y, SAKAZAWA S, et al. Embedding watermarks into deep neural networks. [C] // Proceedings of the 2017 ACM on international conference on multimedia retrieval. [S.l.]: [s.n.], 2017: 269–277.
- [23] NAGAI Y, UCHIDA Y, SAKAZAWA S, et al. Digital watermarking for deep neural networks. [J]. International Journal of Multimedia Information Retrieval, 2018, 7: 3–16.
- [24] FAN L, NG K W, CHAN C S. Rethinking deep neural network ownership verification: Embedding passports to defeat ambiguity attacks. [J]. Advances in neural information processing systems, 2019, 32.
- [25] CHEN H, ROHANI B D, KOUSHANFAR F. Deepmarks: A digital fingerprinting framework for deep neural networks. [J]. ArXiv preprint arXiv:1804.03648, 2018.
- [26] LE MERRER E, PEREZ P, TRÉDAN G. Adversarial frontier stitching for remote neural network watermarking. [J]. Neural Computing and Applications, 2020, 32: 9233–9244.
- [27] ZHANG J, GU Z, JANG J, et al. Protecting intellectual property of deep neural networks with watermarking. [C] // Proceedings of the 2018 on Asia Conference on Computer and Communications Security. [S.l.]: [s.n.], 2018: 159–172.
- [28] ADI Y, BAUM C, CISSE M, et al. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. [C] // 27th {USENIX} Security Symposium ({USENIX} Security 18). [S.l.]: [s.n.], 2018: 1615–1631.
- [29] ROUHANI B D, CHEN H, KOUSHANFAR F. Deepsigns: A generic watermarking framework for ip protection of deep learning models. [J]. ArXiv preprint arXiv:1804.00750, 2018.

- [30] ZHAO J, HU Q, LIU G, et al. AFA: Adversarial fingerprinting authentication for deep neural networks. [J]. Computer Communications, 2020, 150: 488–497.
- [31] LUKAS N, ZHANG Y, KERSCHBAUM F. Deep neural network fingerprinting by conferrable adversarial examples. [J]. ArXiv preprint arXiv:1912.00888, 2019.
- [32] CAO X, JIA J, GONG N Z. IPGuard: Protecting intellectual property of deep neural networks via fingerprinting the classification boundary. [C] // Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security. [S.l.]: [s.n.], 2021: 14–25.
- [33] LI G, XU G, QIU H, et al. A Novel Verifiable Fingerprinting Scheme for Generative Adversarial Networks. [J]. ArXiv preprint arXiv:2106.11760, 2021.
- [34] GOODFELLOW I, POUGET-ABADIE J, MIRZA M, et al. Generative adversarial networks. [J]. Communications of the ACM, 2020, 63 (11): 139–144.
- [35] DONG T, QIU H, ZHANG T, et al. Fingerprinting Multi-exit Deep Neural Network Models via Inference Time. [J]. ArXiv preprint arXiv:2110.03175, 2021.
- [36] HINTON G, VINYALS O, DEAN J. Distilling the knowledge in a neural network. [J]. ArXiv preprint arXiv:1503.02531, 2015.
- [37] LI H, WENGER E, SHAN S, et al. Piracy resistant watermarks for deep neural networks. [J]. ArXiv preprint arXiv:1910.01226, 2019.
- [38] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition. [C] // Proceedings of the IEEE conference on computer vision and pattern recognition. [S.l.]: [s.n.], 2016: 770–778.
- [39] SZEGEDY C, ZAREMBA W, SUTSKEVER I, et al. Intriguing properties of neural networks. [J]. ArXiv preprint arXiv:1312.6199, 2013.
- [40] GOODFELLOW I J, POUGET-ABADIE J, MIRZA M, et al. Generative Adversarial Nets. [J]. Stat, 2014, 1050: 10.
- [41] XUE M, ZHANG Y, WANG J, et al. Intellectual property protection for deep learning models: Taxonomy, methods, attacks, and evaluations. [J]. IEEE Transactions on Artificial Intelligence, 2021, 3 (6): 908–923.
- [42] NAMBA R, SAKUMA J. Robust watermarking of neural network with exponential weighting. [C] // Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security. [S.l.]: [s.n.], 2019: 228–240.
- [43] SHAFIEINEJAD M, LUKAS N, WANG J, et al. On the robustness of backdoor-based watermarking in deep neural networks. [C] // Proceedings of the 2021 ACM Workshop on Information Hiding and Multimedia Security. [S.l.]: [s.n.], 2021: 177–188.
- [44] CHEN H, ROUHANI B D, KOUSHANFAR F. Blackmarks: Blackbox multibit watermarking for deep neural networks. [J]. ArXiv preprint arXiv:1904.00344, 2019.
- [45] CHEN H, ROUHANI B D, FU C, et al. Deepmarks: A secure fingerprinting framework for digital rights management of deep learning models. [C] // Proceedings of the 2019 International Conference on Multimedia Retrieval. [S.l.]: [s.n.], 2019: 105–113.

参考文献

- [46] 樊雪峰, 周晓谊, 朱冰冰, 等. 深度神经网络模型版权保护方案综述. [J]. 计算机研究与发展, 2022 (953-977).
- [47] 王馨雅, 华光, 江昊, 等. 深度学习模型的版权保护研究综述. [J]. 网络与信息安全学报, 2022 (1-14).
- [48] KURIBAYASHI M, TANAKA T, FUNABIKI N. Deepwatermark: Embedding watermark into DNN model. [C] // 2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). IEEE. [S.l.]: [s.n.], 2020: 1340–1346.
- [49] GOODFELLOW I J, SHLENS J, SZEGEDY C. Explaining and harnessing adversarial examples. [J]. ArXiv preprint arXiv:1412.6572, 2014.
- [50] KURAKIN A, GOODFELLOW I J, BENGIO S. Adversarial examples in the physical world. [G] // Artificial intelligence safety and security. [S.l.]: Chapman, Hall/CRC, 2018: 99–112.
- [51] TRAMÈR F, KURAKIN A, PAPERNOT N, et al. Ensemble adversarial training: Attacks and defenses. [J]. ArXiv preprint arXiv:1705.07204, 2017.
- [52] CARLINI N, WAGNER D. Towards evaluating the robustness of neural networks. [C] // 2017 ieee symposium on security and privacy (sp). Ieee. [S.l.]: [s.n.], 2017: 39–57.
- [53] RADFORD A, METZ L, CHINTALA S. Unsupervised representation learning with deep convolutional generative adversarial networks. [J]. ArXiv preprint arXiv:1511.06434, 2015.
- [54] MAINI P, YAGHINI M, PAPERNOT N. Dataset inference: Ownership resolution in machine learning. [J]. ArXiv preprint arXiv:2104.10706, 2021.
- [55] LAO Y, ZHAO W, YANG P, et al. Deepauth: A dnn authentication framework by model-unique and fragile signature embedding. [C] // Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 36. 9. [S.l.]: [s.n.], 2022: 9595–9603.
- [56] KRIZHEVSKY A, HINTON G, et al. Learning multiple layers of features from tiny images. [J]. 2009.
- [57] LLAMAS J. Architectural Heritage Elements image Dataset. [EB/OL]. 2017, Feb 20. <https://datahub.io/dataset/architectural-heritage-elements-image-dataset>.
- [58] BANSAL P. Intel Image Classification. [EB/OL]. 2019. <https://www.kaggle.com/datasets/puneet6060/intel-image-classification>.

图索引

1.1	DNN 模型服务和盗窃示意图	2
2.1	深度神经网络结构图	8
2.2	生成对抗网络结构图	11
3.1	近边界数据示意图	18
3.2	原始样本与对抗性样本对比	19
3.3	DCGAN 网络结构图	22
4.1	近边界数据推断所有权	29
4.2	方法整体流程图	31
5.1	CIFAR-10 上不同分类边界下的近边界数据表现	36
5.2	Heritage 上不同分类边界下的近边界数据表现	36
5.3	Intel_image 上不同分类边界下的近边界数据表现	36
5.4	CIFAR-10 上推断模型所有权的扩展性	42
5.5	Heritage 上推断模型所有权的扩展性	42
5.6	Intel_image 上推断模型所有权的扩展性	43

表索引

3.1	微调分类边界对模型的影响	24
5.1	模型参数信息	34
5.2	硬件与软件配置	34
5.3	不同对抗性样本生成算法生成的数据与目标分类边界的平均距离 .	35
5.4	CIFAR-10 上不同对抗性样本生成算法生成的数据与目标分类边界的平均距离	38
5.5	Heritage 上不同对抗性样本生成算法生成的数据与目标分类边界的平均距离	39
5.6	Intel_image 上不同对抗性样本生成算法生成的数据与目标分类边界的平均距离	40
5.7	推断模型所有权	41

致谢

致谢

感谢您使用本模板。

个人简历

xxx，出生于 yyyy 年 mm 月 dd 日。在 20yy 年毕业于 xx 大学 XX 专业并获得 xx 士学位。于 20xx 年至今在南开大学就读 xxx 研究生。

研究生期间发表论文：

- 周恩来. 周恩来选集 [M]. 人民出版社, 1980.
- 周恩来. 周恩来外交文选 [M]. 中央文献出版社, 1990.