中图分类号: 学校代码: 10055
UDC: 密级: 公开

南開大學

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 Method 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 计算机学院
学科专业 计算机科学与技术 研究方向 模型知识产权保护
答辩委员会主席 岳顺民 评 阅 人 匿名评审

南开大学研究生院
二〇二三年六月

摘要

深度神经网络 (Deep Neural Network, DNN) 训练代价昂贵，这是导致模型知识产权保护问题逐渐被重视的原因。近年来，模型盗窃行为频繁发生，攻击者非法复制、派生和发布 DNN 模型，严重侵犯了模型所有者的知识产权。因此，受到数字水印的启发，研究者提出了模型水印及指纹的方法，通过对模型提取水印或指纹进行匹配，从而验证模型所有权。然而，通过模型水印和指纹验证所有权具有较大的局限性，例如易被检测和清除。此外，攻击者还可以对模型发起歧义攻击，这是当前模型知识产权保护工作面临的重大挑战。为了解决上述问题，本文提出了一种新的思路，即推断模型所有权，代替以往基于模型水印和指纹验证模型所有权的方法。同时，本文提出了一种特殊的近边界数据，使用其对应的模型输出作为依据推断模型的所有权，解决伪造水印和指纹带来的歧义攻击问题，且不易被攻击者检测和清除。本文的主要工作如下：

1）本文揭示了以往验证模型所有权方案的脆弱性和局限性，并确认了数据驱动推断模型所有权的有效性。本文提出了一种新颖的推断模型所有权思路。与过去工作中利用模型水印和指纹验证模型所有权相比，本文方法使用数据在对应模型上结果作为所有权推断依据，结果的可比性和唯一性可以有效避免歧义攻击。为了保证不影响原始数据集和模型，且作为依据的数据不被伪造，本文设计了一种特殊的近边界数据。实验验证了该数据的特性可以继承到目前主流的盗窃技术派生出的模型上，从而作为推断模型所有权的依据。

2）本文提出了基于近边界数据推断模型所有权的方法。该方法主要分为三个阶段：第一阶段通过改进的 CW-L_2 算法，从原始训练数据生成初始近边界数据；第二阶段设计了基于 DCGAN 的特征提取器，提取原始近边界数据特征后，生成新的、私有化的近边界数据；第三阶段设计了新的损失函数并微调源模型，使私有近边界数据更加靠近分类边界。最后提出使用假设检验的方法对比结果的差异，以 95% 以上的置信度成功推断模型所有权。本文在三个开源数据集上进行了大量的实验，证明了本文方法在推断模型所有权时的有效性和鲁棒性。

关键词： 人工智能安全；知识产权保护；所有权推断；近边界数据；生成对抗网络

Abstract

Deep neural network (DNN) requires expensive training, which is why the protection of model intellectual property is becoming more critical. In recent years, model stealing has become more frequent, with attackers illegally copying, deriving, and releasing DNN models, severely infringing on the model owner's intellectual property. Therefore, inspired by digital watermarking, researchers have proposed methods of model watermarking and fingerprinting, which verify model ownership by matching extracted watermarks or fingerprints. However, ownership verification through model watermarking and fingerprinting has significant limitations, such as being easily detectable and removable. Additionally, attackers can launch ambiguous attacks on the model, which is a major challenge for current model intellectual property protection efforts. To address these issues, this thesis proposes a new approach, namely inferring model ownership, instead of relying on watermarking and fingerprinting methods to verify model ownership. Furthermore, this thesis proposes a special type of near-boundary data, using its corresponding model output as the basis for inferring model ownership, which solves the ambiguity attack caused by forged watermarks and fingerprints and is not easily detectable or removable by attackers. The main contributions of this thesis are as follows:

1)This thesis reveals the fragility and limitations of previous schemes for verifying model ownership and confirms the validity of data-driven inference of model ownership. This thesis proposes a novel approach for inferring model ownership. Compared to previous methods that rely on model watermarking and fingerprinting to verify model ownership, the approach in this thesis uses the corresponding model output on the data as the basis for ownership inference. The comparability and uniqueness of the results can effectively avoid ambiguity attacks. To ensure that the original dataset and model are not affected and that the data used as the basis for inference cannot be forged, this thesis designs a special type of near-boundary data. The experiments verify that the properties of this data can be inherited to the models derived from the current main-

stream theft techniques, and thus serve as the basis for inferring model ownership.

2)This thesis proposes a method for inferring model ownership based on near-boundary data. The method is mainly divided into three stages: the first stage generates the initial near-boundary data from the original training data through the improved CW-L_2 algorithm; the second stage designs a DCGAN-based feature extractor to generate new, private near-boundary data after extracting the original near-boundary data features; the third stage designs a new loss function and fine-tunes the source model to make the private near-boundary data closer to the classification boundary. Finally, a hypothesis testing approach is proposed to compare the differences in the results and successfully infer model ownership with a confidence level of more than 95%. This thesis conducts extensive experiments on three open-source datasets, demonstrating the effectiveness and robustness of this method for inferring model ownership.

Key Words: Artificial intelligence security; Intellectual property protection; Ownership inference; Near-boundary data; Generative adversarial network

目录

第一章 绪论

本章首先阐述深度神经网络模型知识产权保护的研究背景与意义，进而分析相关研究中存在的问题与挑战，然后说明本文的主要工作，最后介绍整篇论文的组织结构与章节安排。

第一节 研究背景与意义

近年来，科技飞速发展，计算资源日益丰富，计算能力得到显著提升，我们正逐渐进入人工智能 (Artificial Intelligence, AI) 的时代。互联网的快速发展催生了海量数据的产生。深度神经网络 (Deep Neural Network, DNN)[1] 对数据强大的处理能力，使得 DNN 已经成为应用最为广泛的人工智能方法之一。自深度神经网络在自然语言处理[2–4]、计算机视觉[5–7]、语音识别[8] 等领域实现突破性应用以来，DNN 的应用数量呈爆炸式增长。这些应用已经被被广泛应用于自动驾驶[9]、癌症检测[10]、复杂游戏[11] 等众多场景。在许多领域中，深度神经网络取得了惊人的成就，甚至超越了人类的准确性。

深度神经网络在许多领域都取得了重大的成功，为人类社会生活带来了极大的便利。然而，它也引发了严重的知识产权 (Intellectual Property, IP) 问题。通常情况下，训练一个大型、高性能的神经网络模型都需要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，深度神经网络的设计和优化需要领域专家的专业知识。在设计深度神经网络时，需要考虑多个方面的因素，如网络结构、层数、激活函数等。为了使网络的性能达到最佳水平，需要通过反复试验和调整来设计网络结构和优化网络参数。

2）数据资源，训练数据的获取和处理是深度神经网络训练的第一步。大量的数据是必需的，但这些数据的获取和使用必须遵守相关的法律法规和道德规范。此外，数据的清洗和标注也需要高度的专业知识和劳动力投入。

3）硬件资源，深度神经网络的训练需要漫长的训练时间和大量的计算资源。在训练过程中，需要进行大量的矩阵运算和梯度下降等计算操作，这需要

高性能的计算硬件和软件支持。同时，训练时间也可能会很长，需要耐心和耐力。因此，深度神经网络训练需要充足的资源和时间预算，以确保训练过程的顺利进行和最终的成功结果。如 GPT-3[12]，包含了 1750 亿参数，仅训练成本需花费 460 万美元以上。

因此，高性能的 DNN 模型是模型所有者知识智慧的结晶，同时需要投入高额的经济成本，模型所有者享有 DNN 模型的知识产权[13, 14]。

出于学术目的，模型所有者将 DNN 模型上传到开源社区。或者，使用机器学习即服务 (Machine Learning as a Service, MLaaS)[15] 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)[16]，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。这样的方式极大的方便了科研工作者和一般的消费者，但同时也为不法分子对模型发起各种模型窃取攻击[17, 18]提供了可能性。这些攻击可以让不法分子以比训练原始模型低很多的成本复制一个盗窃模型，用于提供自己的服务或者进行相关研究。

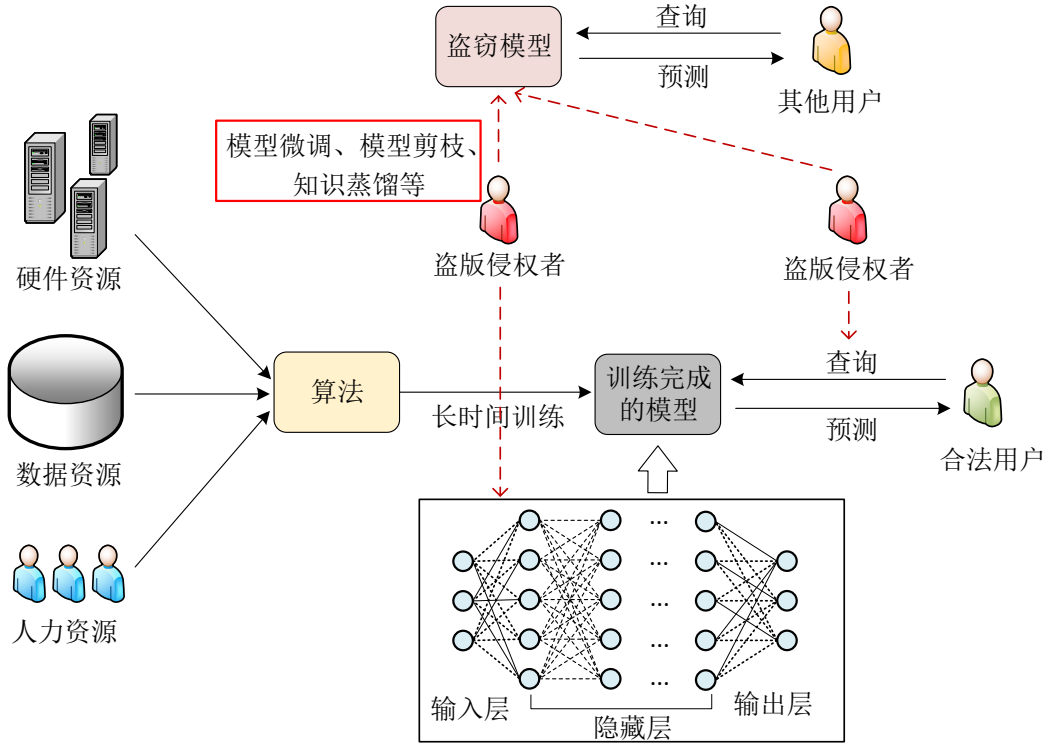


图 1.1 DNN 模型服务和盗窃示意图

如图1.1所示，模型所有者消耗包括硬件资源、数据资源、人力资源在内的大量资源，经复杂算法进行长时间训练后，发布训练好的模型为合法用户提供服务。然而，盗版侵权者会对模型发起窃取攻击，从而获得一个功能相似的盗窃模型用于自己盈利。主要有两种方式：（1）直接访问原始模型，盗窃后对模型进行修改。（2）在不能直接访问模型时，通过模型提供的服务 API 进行特定的输入查询，依靠 API 输出重构模型。

模型盗窃方法：图1.1中，不法分子可以通过模型窃取攻击来盗窃模型。主流的模型窃取攻击涉及到对模型的修改，主要包括模型微调[19]，模型剪枝[20]，模型压缩[21] 等，简要介绍如下：

1）模型微调：型微调通常用于迁移学习，可以在原始模型的基础上微调模型参数，使模型满足自己的任务需求，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、结构相似的模型。

2）模型剪枝：模型剪枝是是部署 DNN 模型的常见方法之一，通过参数修剪来减少 DNN 的内存需求和计算开销，以便部署到一些资源受限的环境下。然后而模型盗窃者可能会使用修剪来删除水印或指纹。

3）知识蒸馏：模型压缩中常见的方式是知识蒸馏[22]，可以将训练好的模型的知识，蒸馏到其他模型上。这种方式可以更快的训练一个新模型，显著降低模型的训练成本，内存需求和计算开销，同时达到与原始模型接近的性能。研究[23] 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。

综上所述，如何在训练和部署时保护深度神经网络模型的知识产权是 AI 领域亟待解决的问题。保护 DNN 模型知识产权的意义在于确保模型的创造者能够获得其所创造的价值，具体而言，有以下四个方面的意义：

1）保护商业利益：神经网络模型模型通常需要大量的时间、资源和资金才能进行开发和训练。通过保护知识产权，模型的创造者可以确保他们的商业利益得到保护，从而获得他们应得的经济回报。

2）保护创新：神经网络模型模型是创新的产物，保护模型知识产权可以激励更多的人投入到相关研究和开发中。这样可以催生更多的创新，从而推动技术的进步和发展。

3）避免盗用和侵权：保护知识产权可以避免其他人对神经网络模型的盗用和侵权。这些行为可能会导致模型的创造者无法获得应有的经济回报，并可能

削弱创造者的商业利益。

4）确保模型质量：通过保护知识产权，模型的创造者可以控制和保证其模型质量。这样可以确保 DNN 模型在商业应用中的可靠性和有效性，从而提高其商业价值和应用性。

第二节 相关研究现状

神经网络模型作为数字产品，不仅是设计者的知识智慧的结晶，还需要消耗昂贵的计算资源、花费大量的训练时间和海量的训练数据做支撑。近年来，先进模型所带来的工业优势已经被广泛认可，但这也引发了一些不法分子对这些模型进行攻击和窃取。神经网路模型将在未来的信息技术发展中扮演核心角色，因此保护这些模型的重要性显得更加突出。1994 年，Van Schyndel 等人[24] 首次提出了数字水印的概念，通过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。具体而言，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。由于 DNN 模型也是一种数字产品，因此，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模型的所有权问题。

1.2.1 模型水印

模型水印[25] 是解决神经网络模型知识产权问题的主要方式之一，Uchida 等人[26] 在 2017 年首次提出了在模型中嵌入水印的通用框架。该方法是一种白盒的模式，通过在训练时使用正则化器，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结构等的细节，并且可以提取嵌入的水印，以此来作为验证模型所有权的依据[27]。Chen 等人[28] 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。为了解决水印易受到歧义攻击的问题，Fan 等人[29] 提出了一种在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，神经网络模型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。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。Zhang 等人[30] 提出了一种水印植入方法，将水印注入模型。通过扩展神经网络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[31] 提出了利用模型的后门机制当作 DNN 模型水印。后门通常是指神经网络模型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。为了减小水印对模型性能的影响，Le 等人[32] 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模型或 API 服务，通过少量的查询提取水印。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和内部结构。除此之外，Rouhani 等人[33] 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。

然而，DNN 模型水印的嵌入步骤总是会对原始模型进行修改。具体而言，白盒水印修改模型内部，比如模型权重、激活函数、甚至模型架构，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。除了减少对模型性能的影响，如何减小模型水印的嵌入代价也是模型水印的重要目标。

1.2.2 模型指纹

模型指纹是解决神经网络模型知识产权问题的又一主流方法。不同于模型水印，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固有的的特征作为模型指纹，一般来说，不会影响模型的性能。

Zhao 等人[34] 提出了一种针对神经网络分类模型的指纹技术，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。具体而言，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，它可以更好地从原始模型转移到派生出的模型上。与 Zhao 等人[34] 的方法类似，Lukas 等人[35] 提出了一种用于神经网络分类器的指纹识别方法。该方法从源模型中提取一组特殊的输入，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。所有者通过验证分类是否一致，判断

模型是否从源模型派生。同样利用分类模型的分类边界，Cao 等人[36] 针对 DNN 分类器提出了一种名叫 IPGuard 的指纹方法，该方法的关键是分类模型可以由其分类边界唯一的表示。基于这一原理，IPGuard 在模型所有者的神经网络模型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模型。除了以上针对分类模型的指纹方法，Li 等人[37] 提出了一种适用于生成对抗网络 (Generative Adversarial Network, GAN)[38] 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。针对模型水印和指纹容易受到最抗性训练攻击，不适用于多出口 DNN 模型知识产权验证的问题，Dong 等人[39] 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。

然而，模型窃取攻击通常会涉及到对原始模型的修改，作为指纹的固有特征也会受到影响。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。此外，模型指纹不适用于小样本数据集。对于小样本数据集，由于模型的特征向量可能会存在过拟合或欠拟合的情况，因此可能会导致模型指纹的准确性降低。

歧义攻击问题：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，但除了上述提到的问题外，无论是水印还是指纹都容易受到歧义攻击[29, 40]。歧义攻击是指不法分子盗窃模型后，通过为神经网络模型伪造其他水印或指纹来对所有权验证产生干扰。直观地，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，那么该模型的所有权归属存在巨大的歧义。

第三节　本文主要工作

为了解决深度神经网络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[36] 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。但是这种方法是脆弱的，对模型的任意操作都有可能破坏这种特性。因此，本文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分类边界的距离。大多数对抗性样本都是位于分类边界附近

的，也就是说，它们与分类边界的距离很近。对抗性样本的这种性质被本文所利用并构造近边界数据。经过测试，本文发现绝大多数的模型窃取方法都无法改变这种结果，即使在盗窃模型中样本分类受到影响，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。因此本文提出由近边界数据驱动的模型所有权推断方法，代替传统的模型水印、指纹验证所有权。

本文方法的主要原理如图1.2所示，其思想是构造私有的近边界数据，当判断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。

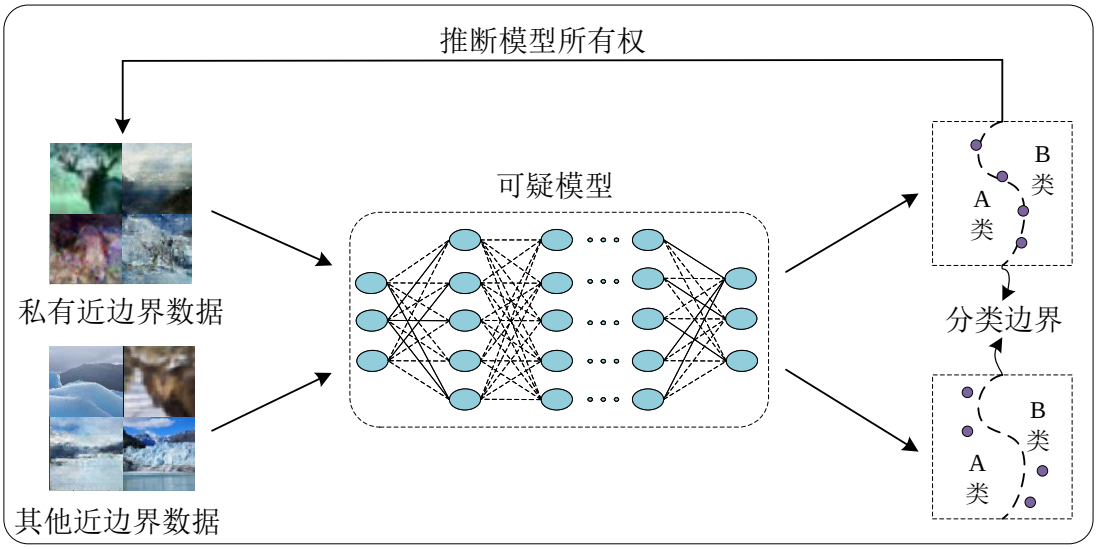


图 1.2　近边界数据推断所有权

本文的主要工作如下：

1）提出基于数据推断模型所有权的新思路，并利用对抗性样本构造近边界数据以抵御模型窃取攻击。与过去工作中利用模型水印和指纹验证模型所有权相比，使用数据在对应模型上结果作为所有权推断依据，结果的可比性和唯一性可以有效避免歧义攻击。通过实验验证了近边界数据的近边界性可以很好的继承到从源模型派生出的模型上，因此可以作为推断模型所有权的依据。本文利用生成对抗性样本算法生成了初始近边界数据。

2）设计了基于 DCGAN 的近边界数据特征提取器，用以私有化初始的近边界数据，并且设计了一种新的损失函数用以微调源模型，增加推断模型所有权的置信度。为了防止近边界数据被轻易伪造，本文训练了 DCGAN 作为数据特征提取器，提取近边界数据特征后，生成新的、私有化的近边界数据。在此基础之上，重新设计了新的损失函数微调源模型，在保持 DNN 模型性能的情况下，以 95% 以上的置信度成功推断模型所有权。

3）基于 ResNet18[41] 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盗窃模型，使用 VGG11[42] 作为无关对照模型。对生成初始近边界数据的方法选择、近边界数据私有化方法的选择、近边界数据的可继承性、源模型微调的影响、模型所有权推断的有效性和近边界数据规模的可伸缩性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

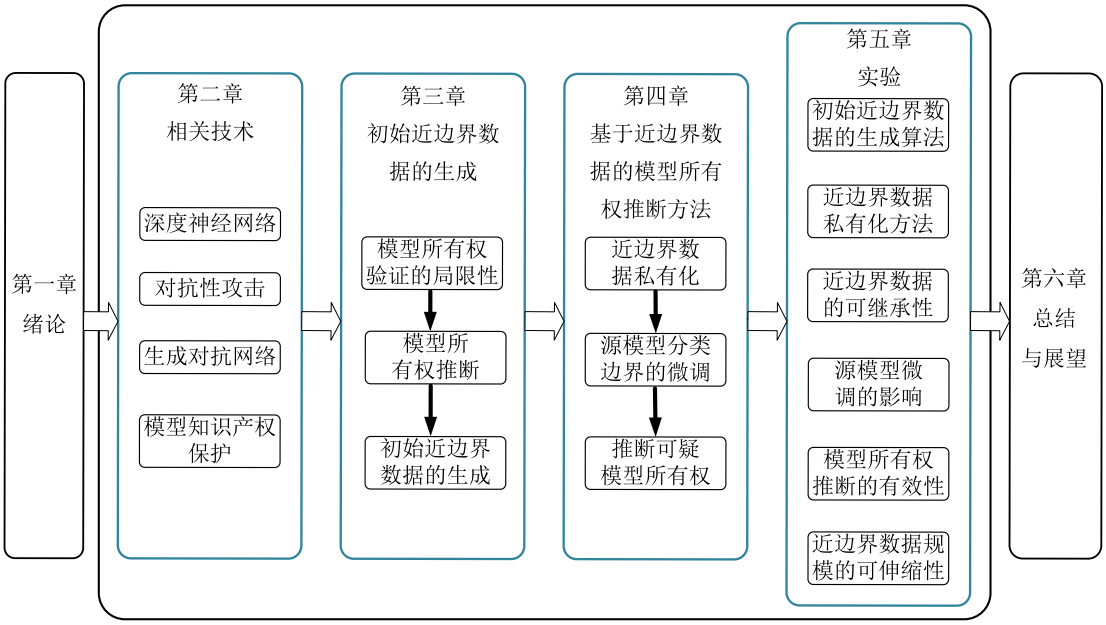


图 1.3　章节架构图

本文对近边界数据进行研究，提出一种基于近边界数据推断模型所有权的

方法。本文的组织架构如图1.3所示，第二章为后面章节提供技术基础。第三章提出模型所有权推断代替所有权验证，解决歧义攻击问题，并生成所有权推断所需的初始近边界数据。第四章在第三章的基础上对初始近边界数据进行私有化，然后基于近边界数据进行所有权推断。第五章对本文提出方法进行全面的测试与评估。全文共分为六各章节，每个章节的主要内容如下：

第一章：绪论。本章首先介绍了神经网络模型在当今时代的广泛应用和研发的昂贵成本，说明了保护 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：相关技术。本章首先介绍了深度神经网络基本原理和结构并解释本文使用到的相关术语，然后介绍了对抗性攻击和生成对抗网络的基本原理，最后介绍了模型水印、模型指纹两种知识产权保护方法。为第三章、第四章提供技术基础和理论依据。

第三章：初始近边界数据的生成。本章首先分析了通过传统模型水印、模型指纹来做所有权验证的局限性，然后提出了数据驱动的所有权推断方法。接着研究了近边界数据在源模型和其派生模型上的可继承性，说明近边界数据可用于所有权推断。最后生成了所有权推断所需的初始近边界数据。

第四章：基于近边界数据的模型所有权推断方法。本章首先介绍了需要将初始近边界数据私有化的原因，然后训练生成对抗网络，用其生成器生成新的、私有化的近边界数据。然后设计了新的损失函数微调源模型，使私有近边界数据更加靠近目标分类边界，增加成功推断模型所有权的置信度。最后提出使用假设检验的方法来统计对比结果差异。

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、近边界数据私有化方法的选择、近边界数据的可继承性、源模型微调的影响、模型所有权推断的有效性和近边界数据规模的可伸缩性几个方面进行了详细的实验和评估，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在解决模型知识产权问题时的优势与不足，并针对不足之处，提出了未来研究工作的改进方向。

第二章　相关技术

本章主要介绍深度神经网络模型知识产权领域的相关技术。首先，介绍了深度神经网络的基本结构和原理以及本文涉及到的相关术语，便于理解本文提出的方法。然后，着重介绍了对抗性攻击和生成对抗网络的原理，为近边界数据的生成提供技术基础。最后，介绍模型水印和指纹两种主流知识产权保护方法的原理和实现方式。

第一节　深度神经网络

人工神经网络是一种模拟生物神经系统的信息处理模型，旨在通过一系列相互连接的神经元（网络中的节点）来处理复杂的数据。它可以被看作是由许多基本单元组成的网络，每个基本单元都可以接收来自其他基本单元的信号并输出一个新的信号。通常，神经网络由输入层，隐藏层和输出层组成。如图2.1所示，如果神经网络有多个隐藏层，则被称为深度神经网络。深度神经网络的隐藏层一般由多个卷积层，池化层，全连接层，Dropout 层和 Softmax 层构成，这些层能够提取数据的高级特征，并对复杂的非线性数据进行有效建模。在神经网络的训练过程中，数据输入到输入层，通过每一层后，每层会提取出一些抽象特征作为下一层的输入，最终由输出层输出最终结果。

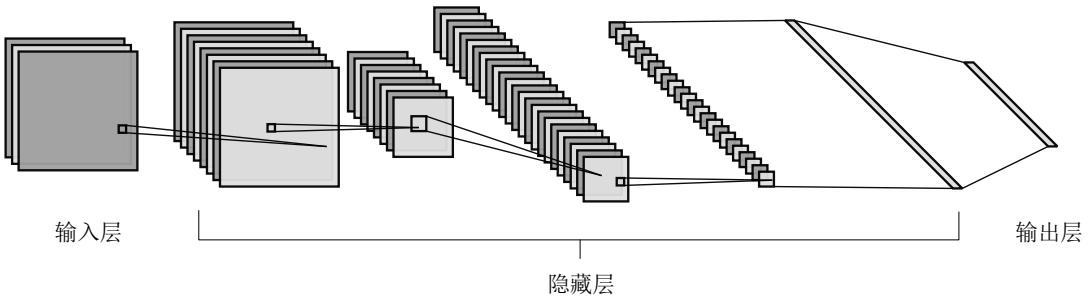


图 2.1　深度神经网络结构图

深度神经网络一种非常强大的非线性数学函数，能够将一组输入变量转化为一组输出变量。每个神经元都有对应的权重和偏置参数，这些参数控制着输入的精确转化，从而实现了高度准确的数据建模和预测。在 DNN 模型的训练过

程中，需要确定神经网络中每个神经元的权重和偏置参数。这个过程可以通过梯度下降算法和反向传播来实现。梯度下降算法通过最小化损失函数来更新参数，反向传播则是用来计算参数的梯度，从而实现参数的更新。神经网络模型的学习和训练是一个需要大量计算资源的过程，然而一旦权重和偏置参数确定，DNN 模型就可以快速地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的**相关术语**：

1）源模型。源模型也称作目标模型、原始模型，是指模型所有者在私有或公开数据集上，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。可疑模型也称作盗窃模型、替代模型、派生模型，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）白盒环境。白盒环境是指能够获得 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）黑盒环境。黑盒环境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。

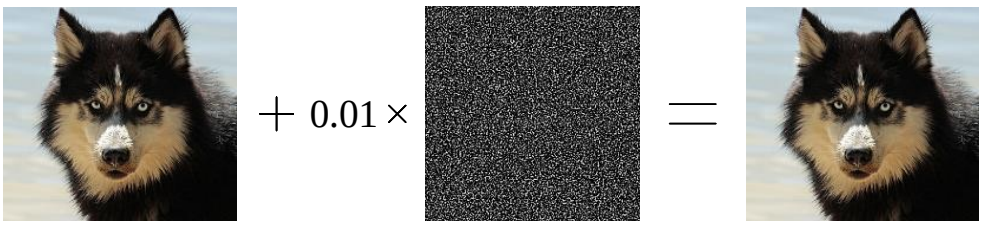
5）**验证模型所有权**。验证模型所有权指通过检测有无特定的水印或者指纹是否匹配的方式解决模型的所有权问题，检测到特定的水印或者指纹匹配说明验证所有权成功。

6）**推断模型所有权**。推断模型所有权是本文提出的新概念，指通过数据在模型上的最优性解决模型的所有权问题，最优数据提供者推断获得所有权。

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[43] 提出的。这篇文章中指出，通常情况下，一个性能良好的神经网路模型具备优异的泛化能力，即对输入的微小随机扰动具较强有鲁棒性，从而保证了模型在处理图像时的类别预测准确性。然而，如果对图像添加特定的非随机扰动，使得损失函数的值上升，那么 DNN 模型的预测结果就可以随意改变。这些难以被人眼察觉但足以使得模型输出错误类别的图像样本即为对抗性样本。



狗　　特定扰动　　猫

图 2.2　对抗性攻击示意图

如图2.2所示，狗的图像被添加特定扰动后被分类器识别为猫。添加特定的扰动，甚至可以使模型输出任意类别。

（1）黑盒场景

用 $f: R^m \rightarrow 1,2,...,n$ 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：

$$\begin{aligned} &min \parallel \delta \parallel_2, \\ &s.t.\ f(x+\delta) = l,\ x+\delta \in [0,1]^m \end{aligned} \tag{2.1}$$

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

（2）白盒场景

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 $f: R^m \rightarrow 1,2,...,n$ 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：

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

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来发起攻击。

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[44] 第一次提出了生成对抗网络 (Generative Adversarial Network, GAN)，是一种利用生成模型实现无监督学习的特殊方法。该网络由一个生成器和一个判别器组成，并通过一种相互博弈的方式进行训练。如图2.3所示，首先随机噪声作为生成器的输入，经过生成器转化成和真实图片具有相同维度的图像。使用原始图片和生成图片分别输入到判定器中，训练判定器区分它们的能力。接着，再使用真实图片训练生成器，使之生成的图片尽可能接近真实图片。通过迭代的交替训练，在训练收敛时，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。这种相互博弈的过程可视为生成器和判别器之间的对抗，因此称为生成对抗网络。

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 $maxD$，对于生成图片 $G(T)$，生成

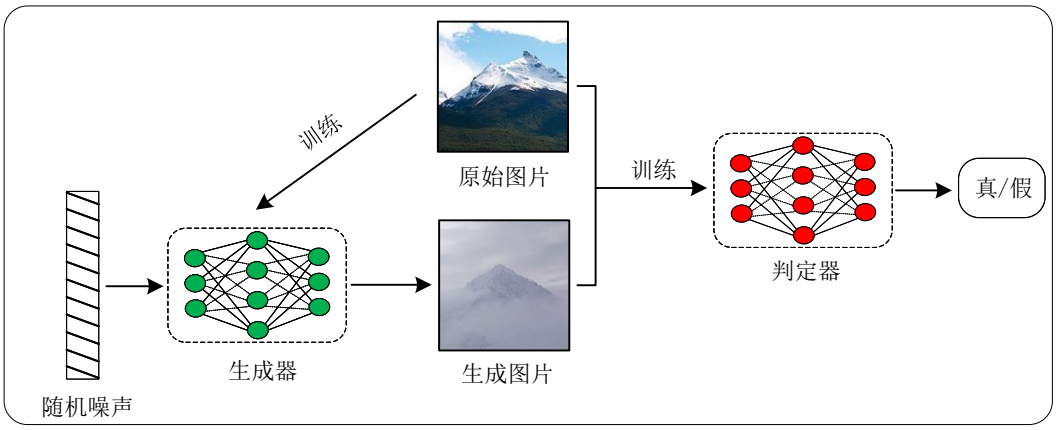


图 2.3 生成对抗网络结构图

器希望 $D(G(T))$ 变大，即 $log(1-D(G(T))$ 变小，对应于式中的 $minG$，所以 GAN 的目标函数由两个目标构成。

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

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，常用的有高斯分布、均匀分布等，E 表示数学期望。

在图像生成领域，生成对抗网络已被广泛应用于多种任务[45–47]，包括图像合成、图像修复、风格转换和图像生成等。由于 GAN 独特的训练方式，使其不仅仅在生成新图像方面表现出色，还可以用于图像特征提取[48]。通过对原始图像进行特征提取，GAN 的生成器可以生成与原始图像特征分布类似的新图片。这种方法已被成功应用于多个领域，例如医学成像和计算机视觉。因此，生成对抗网络在图像生成和特征提取方面的广泛应用将在未来的研究中得到进一步的探索和发展。

第四节 模型知识产权保护

为了训练一个高性能的神经网络模型，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、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。由于 DNN 模型在各个领域都得到了高效的应用，许多不法分子开始盗窃、复制和修改这些模型以获利。为了保护神经网络模型的知识产权[49, 50]，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 模型的所有权。

2.4.1 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。

（1）白盒水印

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[51] 提出一种可量化的水印嵌入方法，该方法基于全连接层的权重，为了使嵌入水印不对模型造成太大影响，通过改变训练中的参数来量化水印的影响。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。在 DNN 模型中，可以改变模型的结构，添加额外的一层作为护照[29]，以此来作为数字签名，增加模型的安全性。当不法分子发起歧义攻击，尝试嵌入额外水印时，模型性能会急剧下降。然而，这种方式的部署代价非常高。

（2）黑盒水印

在黑盒场景下，盗窃模型的内部结构和权重参数等是未知的，模型所有者只能通过其提供的 API 服务进行水印验证。一般而言，黑盒水印通过构造特殊的触发集实现，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。

2.4.2 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

指纹生成是模型指纹技术中的第一阶段，它需要选择一组样本来生成指纹。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[34–36]，可以把靠近决策边界的对抗性样本作为模型的指纹。这些样本可以通过将一个已知标签的样本加上一些扰动来生成，从而欺骗模型产生错误的分类结果。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标签。

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：

$$arg\max_i g_i(x) = y \wedge arg\max_i g_i(x+\delta) = y' \tag{2.4}$$

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 $X' = \{x'_1, x'_2, ..., x'_n\}$ 作为模型指纹样本，对应的预测结果 $Y' = \{y'_1, y'_2, ..., y'_n\}$ 为指纹标签。

（2）模型指纹验证

指纹验证是模型指纹技术中的第二个阶段。在这个阶段中，生成的指纹将被用于验证模型的所有权。验证过程基于相同的原理，即使用对抗性样本来检查模型的响应。具体来说，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成功率超过阈值 τ 时，则视为匹配成功，判定提供指纹样本的人是模型的合法拥有者。

设 $X' = \{x'_1, x'_2, ..., x'_n\}$ 为 n 个指纹样本，$Y' = \{y'_1, y'_2, ..., y'_n\}$ 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 $\widetilde{Y}' = \{\widetilde{y_1}', \widetilde{y_2}', ..., \widetilde{y_n}'\}$。其中预测标签与指纹标签相同的数量为 q，即 $y'_i = \widetilde{y_i}'(i = 1, 2, ..., q)$。

定义验证函数如下：

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

式2.5中，q/n 表示匹配成功率，τ 是判定阈值。当匹配成功率大于等于阈值 τ 时，结果为 1，表示指纹成功匹配，判定可疑模型为盗版模型。

第五节 本章小结

本章主要介绍了深度神经网络，对抗性攻击，生成对抗网络和知识产权保护这四个方面的相关概念和理论基础。深度神经网络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会导致模型输出错误的特殊样本，根据方式的不同，有多种生成的方法。本文利用对抗性样本靠近模型分类边界的特点构造初始近边界数据。生成对抗网络具有强大的特征提取能力，可以利用其生成与训练样本特征分布类似的新样本。本文生成对抗网络提取初始近边界数据的特征，然后使用生成器生成新的、私有的近边界数据。在神经网络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，它们大多基于验证模型所有权的思路进行设计与实现。

第三章　初始近边界数据的生成

目前，大多数模型的知识产权保护方法采用模型水印和模型指纹等被动防御方式来进行所有权验证，这种方式很难抵御歧义攻击。数据集推断利用训练数据在模型上的最优性进行所有权决策，从而验证所有权，但要求训练数据私有，因此应用范围小且推广难度高。因此，本文提出了一种数据驱动的推断模型所有权的新思路，该方法利用数据在模型上的最优性来推断模型所有权，代替传统的水印和指纹验证模型所有权，可以有效避免歧义攻击。同时，本文通过构造一种特殊的数据代替训练数据来推断模型所有权，使得训练数据可以被公开。

第一节　模型所有权验证的局限性

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。如果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，当检测到相应的水印或者指纹相匹配就代表该模型是从源模型派生。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，但这并不具备较强的鲁棒性。嵌入水印对源模型性能和功能的影响和需要的的额外代价都是模型水印研究工作的关键点。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，而是通过伪造额外的水印和指纹混淆所有权验证。

如图3.1所示，为了保护自己的知识产权，模型所有者在训练完源模型后，给 DNN 模型嵌入水印，然后发布模型提供公开服务。模型盗窃者访问公开的模型或者模型所提供的服务 API，通过一定的方法复制篡改而得到盗版模型。为了躲避模型所有者的检测，盗窃者不关心原有的水印，而是给模型嵌入自己的额外水印。模型指纹的歧义攻击与水印类似，盗窃者不关注原有的指纹，而是提

取新的指纹，以此混淆模型所有权的验证。

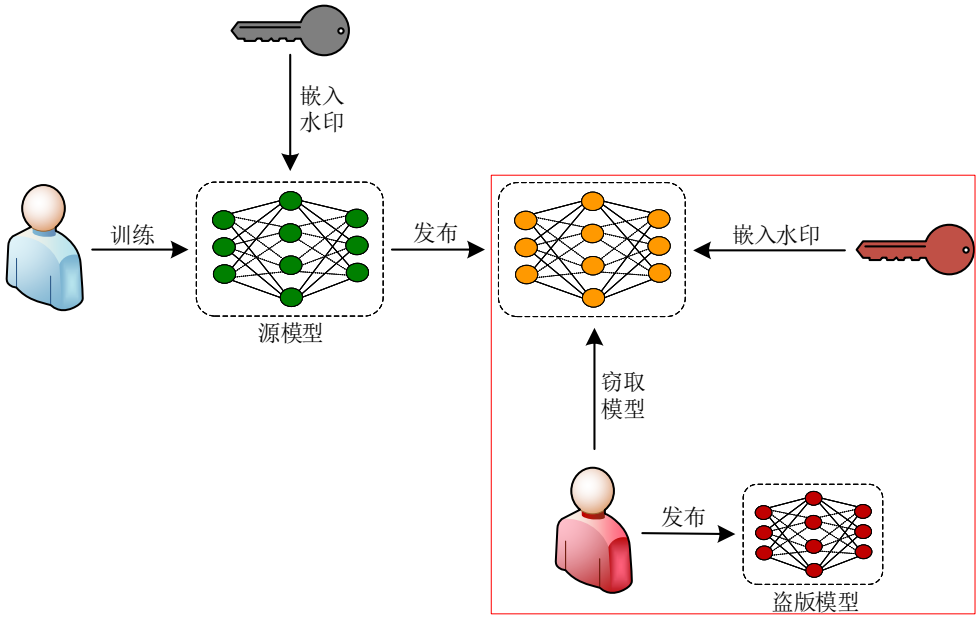


图 3.1 歧义攻击示意图

如图3.2所示，如果模型所有者怀疑可疑模型是从自己的源模型派生，向官方机构发起仲裁。仲裁机构进行模型所有者的水印或指纹检测，成功检测到嵌入的水印或成功匹配指纹。而于此同时，盗窃者的水印或指纹同样能够被仲裁机构所验证，这种情况下无法进行正确的所有权决策。直观地，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，那么该模型的所有权归属存在巨大的歧义。

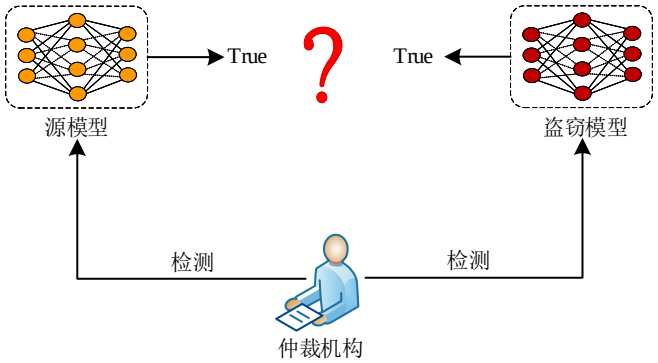


图 3.2 检测歧义示意图

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[29]，否则鲁棒性的水印也不一定能验证所有权。

本文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，容易产生所有权混淆。因此本文提出推断模型所有权而不是验证，这是一种解决 DNN 模型所有权问题的新思路，与传统的通过模型水印和指纹验证所有权有所不同。这种方法是受数据集推断[52]提出的数据驱动决策所有权的启发，利用某类数据在源模型上的最优性推断模型所有权，在下一小节中将具体讨论。

第二节 模型所有权推断

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，盗窃模型的知识仍然是源模型中包含的知识。如果原始训练数据集是私有的，那么模型所有者在进行数据集推断时，相比盗窃者拥有强大的优势，因为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，模型所有者通过评估多个数据点到分类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。这种方式与模型水印和指纹验证所有权的方式有着本质的区别，该方式并不是去验证特定的水印或指纹，而是比较某类数据在模型上的最优性。

如图3.3所示，因为 DNN 模型是从数据集训练而来，所以模型中总会包含来自数据集中的知识。盗窃模型是从源模型派生，尽管包含的知识和源模型不可能完全相同，但总是有一部分是来自原始数据集，这是利用数据集做模型所有权推断的理论基础。

模型窃取过程中，源模型中的知识会传播到盗窃模型，使得所有盗窃模型总是包含一部分源模型训练数据集中的直接或间接信息。利用数据集做模型所有权推断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，这种方式容易受到歧义攻击，进而导致的验证冲突。从决策过程可以发现数据集推断得到的是一个“最”的概念，模型的所有权归属于决策指标的最大者，而不是进行类似模型水印和指纹的特定响应匹配，因此可以有效避免歧义攻击。

本文的工作是受到数据集推断验证模型所有权的启发，使用数据驱动推断

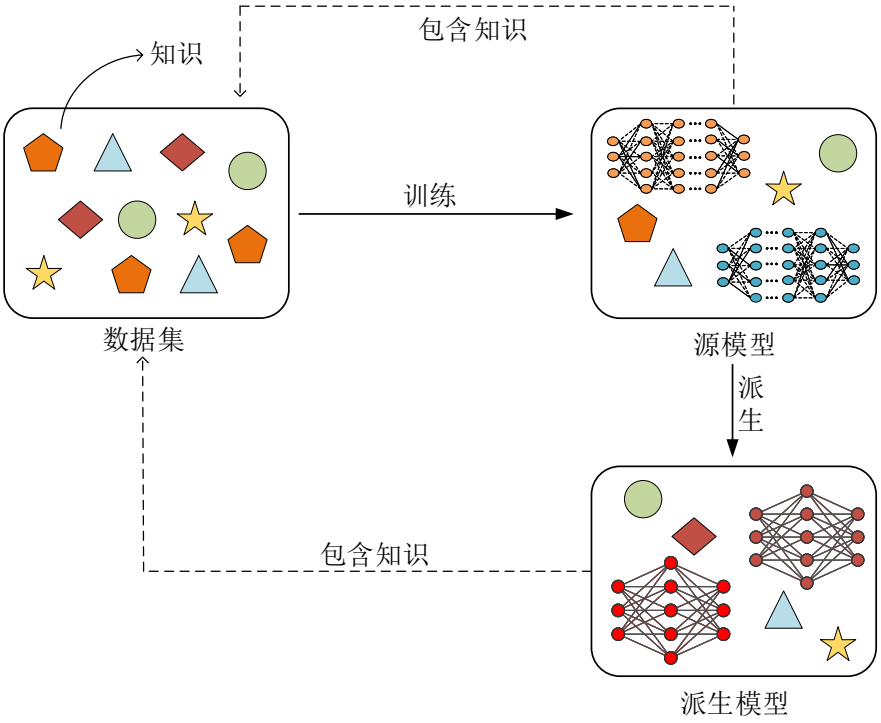


图 3.3 数据集推断原理图

模型所有权代替验证模型所有权。所有权推断可以在有效证明所有权归属的同时，解决所有权验证冲突的问题。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，那么数据驱动推断模型所有权的方法既可以在白盒环境也可以在黑盒环境下工作。

利用数据来推断模型所有权为保护模型知识产权提供了一个新的方向，但是目前数据集推断的方式仍然具有以下**局限性**：

1）使用数据集推断的前提是原始训练数据不能被盗窃者获得，所以公开的数据集不能被用于训练源模型。然而，在大多数真实场景下，只有很少一部分工作会构造私有数据集用于训练模型，甚至这部分工作只应用于特定的领域中，这也意味着模型被盗窃的风险较小。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）数据集推断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，但是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，因此该方法的结果可能会导致错误，将不相关模型判定为盗窃模型。Li 等人[53] 验证了这个局限性，表明在此种情况下该方法产生的结果值得怀疑。

因此，亟需一种模型知识产权保护方法，在能成功判断模型所有权归属的

同时，解决歧义攻击问题，并且适用于公开数据集。本文提出的方法将基于近边界数据推断 DNN 分类模型所有权。

第三节 初始近边界数据的生成

上一节提出使用数据驱动来推断模型所有权，为了使训练数据公开，本文需要寻找一种新的数据来代替训练数据进行推断。本节将介绍这种特殊的数据——近边界数据，并且研究生成近边界数据的算法。

3.3.1 近边界数据的可继承性

DNN 分类器的主要目标是对输入数据样本进行分类，一个 DNN 分类器的特征通常由其决策模式和分类边界决定。然而，分类器的分类边界是一个抽象的概念，无法被具体表示，因此研究者一般通过某正常数据样本和对应生成的对抗性样本组成样本对用于反映分类边界。因为分类边界位于两者之间，所以一定规模的对抗样本对可以用于描述分类边界。由于无法以数学的方式直接定义分类边界，本文使用分类器的决策结果来反映分类边界。

在以往的研究中[36]，可以使用分类边界作为模型指纹用于验证模型所有权。本文不使用分类边界作为模型指纹，而是基于分类边界构造一类特殊的数据用于推断模型所有权。本文称这类特殊的数据为近边界数据，下面给出本文近边界数据的定义：

定义 3.1 (近边界数据) 给定数据样本 x，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 $min(g_i(x), g_j(x)) \geq \max\limits_{k \neq i,j} g_k(x)$，$g_k(x)$ 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。

近边界数据是指那些非常接近分类边界的数据样本，与位于分类边界上的数据样本类似，这些样本对模型的决策边界有重要的影响，因为它们能够揭示模型在分类边界附近的行为。由于近边界数据不要求样本完全位于分类边界上，因此即使模型受到修改，分类边界发生偏移，仍然可以衡量数据近边界性。所以相对于直接使用分类边界来作为模型指纹，近边界数据在面对模型窃取攻击时，有着更强的鲁棒性。

如图3.4所示，近边界数据位于 DNN 分类器的分类边界附近，其他数据的分布则离分类边界较远。判定是否为近边界数据由定义3.1中的阈值 θ 决定，当 θ 较小时，近边界数据样本表现为更加靠近模型分类边界。

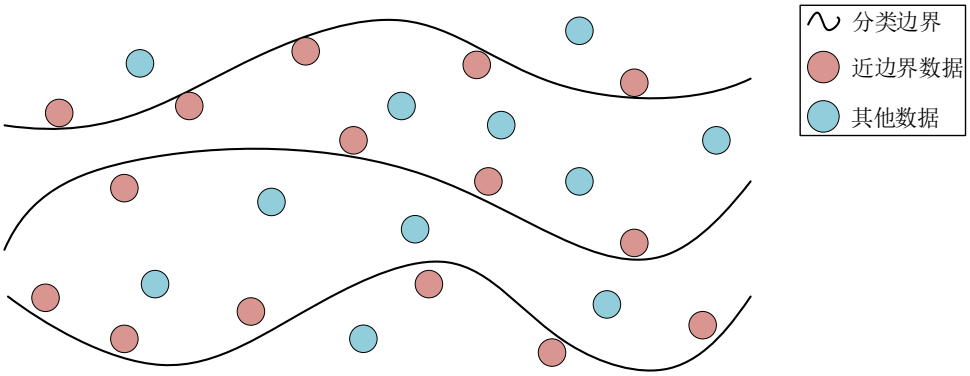


图 3.4 近边界数据示意图

在第五章第四节中，本文通过大量的实验验证了近边界数据在大多数的模型窃取技术中其近边界特征被保留，表明近边界数据的近边界性可以很好的继承到源模型派生出的模型上。因此，近边界数据可以作为推断所有权的依据被使用。尽管近边界数据在模型的知识产权保护中表现出显著的效果，但是在实践中，获得一定规模的近边界数据样本仍然是一个具有挑战性的任务。这是因为自然的近边界数据在样本空间中的占比非常低，甚至可以被忽略不计。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，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。因此，如何得到一定规模的近边界数据样本仍然是一个难题，生成近边界数据的过程将在下一节中详细介绍。

3.3.2 初始近边界数据的生成

根据最近的一些研究[36]，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，原始分类是指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。对抗性样本是通过向原始数据添加小量扰动或干扰来生成的，这些扰动通常很难被人眼察觉，但却足以改变 DNN 模型的分类结果。

如图3.5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，但是分类结果却完全不同，在有目标攻击的情况下，甚至可以人为的指定目标分类。

对抗性样本会对模型分类边界进行跨越，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，本文将生成大量的对抗性样本，并从中挑选合适的

(a) 原始样本

(b) 对抗性样本

图 3.5 原始样本与对抗性样本的对比

近边界数据。因此，本文测试了几种常见的生成对抗性样本的方法，以帮助更好的构建近边界数据。因为本文需要数据样本尽可能靠近分类边界，因此在测试过程中，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。

为了更好的衡量数据样本到分类边界的距离，在定义3.1的基础上，下面给出量化的分类边界距离定义：

定义 3.2 (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。

根据定义3.2，以分类边界距离为衡量标准，下面分别对几种常见的生成对抗性样本的方法进行介绍与测试。

Fast Gradient Sign Method(FGSM):FGSM[54] 是最经典的生成对抗性样本的方法之一，它是一种基于梯度构建对抗性样本的方法，属于无目标的攻击方式。只需要对原始样本添加一次微小的扰动 η，如式3.1，3.2所示，即可生成样本 x 的对抗性样本 $\tilde{x}$，十分高效。

$$\eta = \epsilon \cdot sign(\bigtriangledown_x J(\theta, x, y^*)) \tag{3.1}$$

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

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，$\bigtriangledown_x$ 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应该在 [0,1] 以内，ϵ 用来控制变化幅度大小。

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[55] 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为止。

$$\eta = \alpha \cdot sign(\bigtriangledown_x J(\theta, x, y^*)) \tag{3.3}$$

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

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

除此之外，本文还测试了 FGSM 的另一个进阶版本 RFSGM[56]，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，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，但仍然没有达到本文的预期，生成的对抗性样本距离分类边界距离太远。在大量的测试中，本文发现 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[57] 方法是一种有目标的攻击方式，与其他生成对抗性样本的方法类似，该方法是添加噪声到数据样本中，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 在实验中生成对抗性样本的效果和生成效率相比其余两种变体较好，因此本文使用该方法作为生成对抗性样本的基础。具体而言，CW-L_2 对于给定的初始样本，采用二分查找的方式来增大或减小式3.7中 c，并且使用类似训练神经网络模型的方式来调整生成对抗性样本的其他参数。CW-L_2 的损失函数和约束如式3.5，3.6，3.7，3.8所示：

$$Loss = Loss1 + Loss2 \tag{3.5}$$

$$Loss1 = D(x, x+\delta) \tag{3.6}$$

$$Loss2 = c \cdot f(x+\delta, target) \tag{3.7}$$

$$x+\delta \in [0,1]^m \tag{3.8}$$

其中 *target* 是生成对抗性样本的目标标签，c 是惩罚因子，用于权衡 *Loss*2 的影响大小，算法通过二分查找来寻找合适的 c。*Loss*1 约束对抗性样本 $x+\delta$ 和原始样本 x 尽可能相似，*Loss*2 约束对抗性样本 $x+\delta$ 的决策结果为目标标签，式3.8约束对抗性样本在正常的图像范围内。

Algorithm 1 改进的二分查找 CW-L_2 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'

参数初始化：$c \leftarrow 1$，$distance \leftarrow 1$
for $i = 1,2,...,n$ **do**
    $isSuccessAttack \leftarrow$ false
    $w \leftarrow \text{arctanh}(x)$
    $w\_pert \leftarrow \text{zero\_like}(w)$
    for $j = 1,2,...,iteration$ **do**
        $new\_img \leftarrow \tanh(w + w\_pert)$
        $new\_distance \leftarrow |g_r(new\_img) - g_t(new\_img)|$
        if $new\_distance < distance$ **then**
            $distance \leftarrow new\_distance$
            $x' \leftarrow new\_img$
            $isSuccessAttack \leftarrow$ true
        end if
        使用 Adam 优化器更新 w_pert
    end for
    if $isSuccessAttack ==$ true **then**
        减小 c
    else
        增大 c
    end if
    if $distance \leq \theta$ **then**
        break
    end if
end for
return x'

根据定义3.2，数据样本 x 距离分类边界的距离是 $distance = |g_i(x) - g_j(x)|$。

本节的目标是生成的对抗性样本距离分类边界的距离尽可能近。本文在算法迭代过程中引入这一目标，以此改进算法迭代的过程，在使得生成对抗性样本更加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，仅在 *distance* 变小时，更新距离参数和新生成的对抗性样本，并在 *distance* 小于等于预定的阈值 θ 时，提前终止算法的迭代，具体的过程如算法1所示。

通过算法1，本文已经可以生成大量位于分类边界附近的近边界数据。但是在这一阶段，本文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微小噪声或扰动，针对性地生成了目标分类的对抗性样本。

在此阶段，源模型的训练和原始训练数据集均不受任何影响，防御者只需要针对性的生成对抗性样本即可。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伪造。因此，本文需要将生成的近边界数据私有化，防止被盗窃者轻易伪造，具体操作将在下一节中给出。

第四节 本章小结

本章首先分析了现有的通过模型水印和模型指纹来做所有权验证的局限性，然后提出数据驱动的推断所有权的方法。鉴于近边界数据在源模型和其派生模型上的可继承性，本文使用近边界数据来代替训练数据进行所有权推断。由于自然的近边界样本很少，本文对比了主流的对抗性样本生成算法，根据生成样本到分类边界的距离。选择 CW-L_2 作为基础算法。并在此基础上改进，生成本文的初始近边界数据。

第四章　基于近边界数据的模型所有权推断方法

尽管训练数据可以被公开，但用于推断模型所有权的数据需要私有化，否则攻击者就可以轻易地伪造近边界数据，无法成功推断模型所有权。因此本章通过生成对抗网络来对近边界数据进行特征学习，进而通过生成器生成新的、私有化的近边界数据。在此基础之上，使用私有化近边界数据微调源模型，使私有化近边界数据更靠近模型分类边界，数据越靠近分类边界，越有利于成功推断模型所有权。

结合上一章生成的初始近边界数据，本文方法的整体流程如图4.1所示，主要包括三个阶段：（1）从公开数据集中生成近边界数据，（2）训练生成对抗模型生成新的、私有化的近边界数据，（3）使用私有化近边界数据微调源模型分类边界。

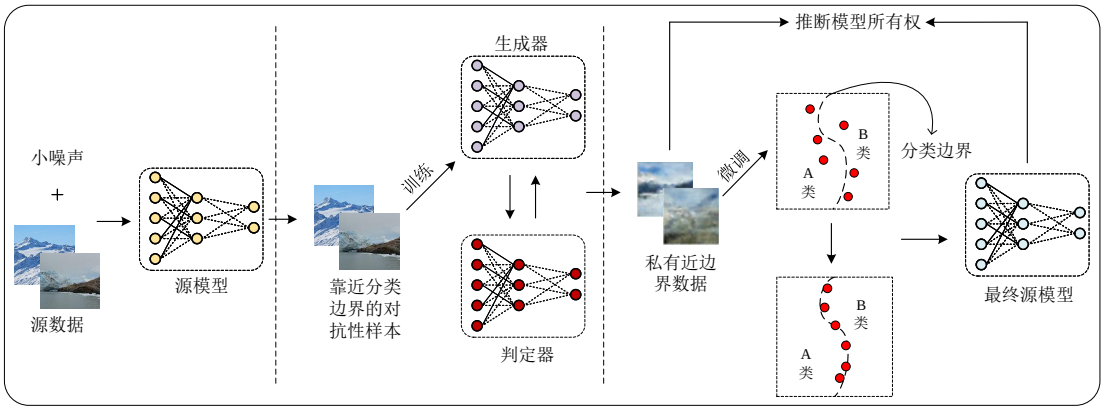


图 4.1　方法流程图

第一节　近边界数据私有化

因为现在大多数模型训练使用的数据都来源于公开的数据集，所以通过生成对抗性样本的方法构建近边界数据这一步骤也十分容易复现。因此，本文需要从公开的训练数据中构建自己的私有化近边界数据，以防止模型所有者的近边界数据被轻易模仿。这是必要的步骤，因为近边界数据是后续推断模型所有权的核心依据。

本文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的特征，并以此生成新的私有化近边界数据。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。

由第二章第三节可知，生成对抗网络在图像生成和特征提取方面有着显著的效果。本文考虑设计基于生成对抗网络的特征提取器，对初始近边界数据进行特征提取后，使用用生成器生成新的、私有化的近边界数据。

与生成初始近边界数据指标一致，本文需要寻找一种 GAN，其生成的近边界数据到分类边界距离尽可能小。本文测试了几种常见的图像生成 GAN，包括边界寻求生成对抗性网络 (Boundary-Seeking Generative Adversarial Networks, BGAN)[58]，边界平衡生成对抗网络 (Boundary Equilibrium Generative Adversarial Networks, BEGAN)[59] 和基于深度卷积生成对抗网络 (Deep Convolutional Generative Adversarial Network, DCGAN)[60]。大部分情况下，DCGAN 的效果最好，具体的测试结果在第五章第三节中。因此，本文设计了一种基于 DCGAN 的特征提取器，提取近边界数据的特征之后，使用生成器生成私有化的近边界数据。

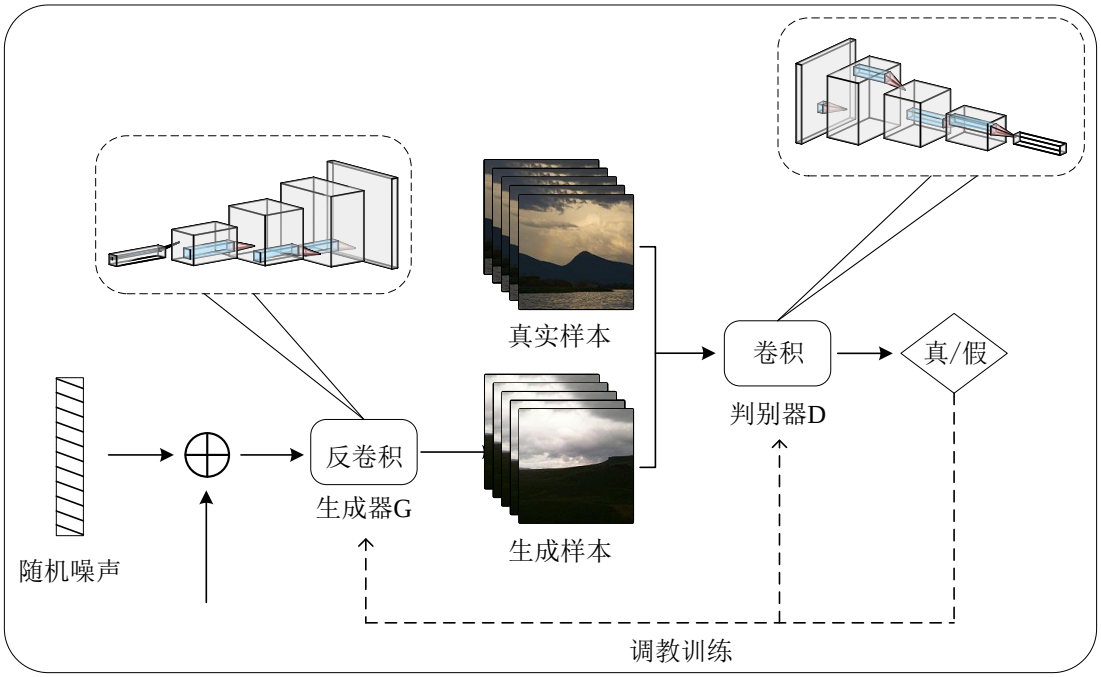


图 4.2　DCGAN 网络结构图

如图4.2所示，DCGAN 的大体结构与训练方式和普通 GAN 类似，主要变

化是 DCGAN 将原始的 GAN 与 CNN 结合到一起，生成器 G 和判定器 D 都用 CNN 架构替换了原始 GAN 的全连接网络。得益于 CNN 对图像的强大处理能力，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舍弃掉 CNN 的池化层，生成器使用反卷积层来还原图片，判别器保留 CNN 的整体架构，使用卷积层来提取图片特征。

2）在生成器和判别器中都使用 Batch Normalization 层，提升训练 DCGAN 模型稳定性的同时加速了训练。

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，使模型可以更快的学习。

4）使用 Adam 优化器并调整了超参数，将学习率设置为 0.0002，可以更好的学习到数据样本的特征。

本文需要 DCGAN 能够学习到尽可能多的近边界数据特征，以便更好的生成近边界数据。训练过程中，尝试修改 DCGAN 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，即使用源模型和判别器共同判定是生成数据还是原始数据。这种方式得到的训练结果在同样的生成规模下略微优于原始 DCGAN 生成的数据。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，所以本文采用原始的训练方式。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

参数初始化：$learning\ rate \leftarrow 0.0002$，$real\_label \leftarrow 1$，$fake\_label \leftarrow 0$
for $i = 1,2,...,epoch$ **do**
    随机噪声 $z \leftarrow 100$
    $x' \leftarrow G(z)$
    $Loss1 \leftarrow Loss(D(x), real\_label)$ ▷ x 是近边界数据样本
    $Loss2 \leftarrow Loss(D(x'), fake\_label)$
    $Loss_D \leftarrow Loss1 + Loss2$
    $Loss_G \leftarrow Loss(D(x'), real\_label)$ ▷ 生成器希望 $D(x')$ 接近 $real\_label$
    使用 Adam 更新生成器 G，判别器 D 的网络参数
end for

通过算法2，完成训练 DCGAN 模型后，该模型可以被用作近边界数据特征提取器。通过生成器生成的图像，可以生成与初始近边界数据分布一致的私有近边界数据，这些数据仍然具备近边界性，并且可疑对手无法轻易获得。

在 DCGAN 的训练过程中，生成器和判定器相互博弈，以不断优化生成器生成图像的特征分布，使其逐渐接近原始近边界样本。训练收敛时，生成器已经学习到近边界数据的特征，因此可以生成新的私有近边界数据。

第二节　源模型分类边界的微调

DCGAN 对图像数据有着很强的特征提取能力，生成器能够很好的学习近边界数据特征，使用生成器构建的私有近边界数据仍然位于目标分类边界附近。但是相比于原始近边界数据，由于随机因素，生成的数据样本近边界性可能会弱于原始近边界数据，本文仍然希望近边界数据能最大程度上靠近目标分类边界。因为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，生成的私有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，经过模型窃取攻击后，由于模型被修改，数据的近边界特性仍有可能被泛化。

因此，为了解决上述问题，本文提出使用近边界数据微调源模型的目标分类边界，使生成的私有近边界数据更加靠近 DNN 模型分类边界。如式3.4所示，$Loss_{FT}$ 是针对目标分类边界的损失函数。

$$Loss_{FT} = \frac{1}{n}\sum_{i=1}^{n}(g_t(x_i') - g_s(x_i'))^2 \tag{4.1}$$

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，但是为了尽可能减小对原始模型精度的影响，不能直接使用该损失函数对源模型进行微调。受 DCGAN 训练过程的启发，本文使用源模型的损失函数 $Loss_{FM}$ 与 $Loss_{FT}$ 两者交替训练微调源模型，并将学习率设置为 0.0001，尽量减小源模型的变化幅度。本文希望在保持源模型精度的同时，使生成的数据样本更加靠近目标分类边界。与 DCGAN 的训练过程相似，微调源模型也是一个博弈的过程，微调的具体流程如算法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'

参数初始化：$learning\ rate \leftarrow 0.0001$
for $i = 1,2,...,epoch$ **do**
    $Loss1 \leftarrow Loss_{FT}(g_t(x'),g_s(x'))$ ▷ $g_k(x')$：x' 在第 k 类上的概率
    $Loss2 \leftarrow Loss_{FM}(M(x),label)$ ▷ $label$ 指正常样本的原始标签
    使用 Adam 更新源模型 M 的网络参数
end for

其中 x 指原始的数据样本，x' 指 DCGAN 生成器生成的私有化近边界数据。

通过算法3微调目标分类边界使得私有近边界数据与源模型之间的联系更加紧密，这对后续能否成功推断模型所有权十分重要。在此算法中，本文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。

表 4.1 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

如表4.1所示，正因为交替微调的设计和较小的学习率，源模型微调前后的精度差不超过 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。

第三节 推断可疑模型所有权

本文的方法适用于 DNN 分类模型的知识产权保护。一方面，该方法是利用近边界数据进行所有权推断，决策双方各自提供自己的近边界数据。将这些数据分别输入模型后，距离分类边界最近的数据提供者，获得模型的所有权，而不是进行类似模型水印和指纹的特定响应匹配，因此可以有效避免歧义攻击。另一方面，针对现有的数据集推断方法存在的问题，主要有以下两点改进：

1）本文在公开数据集上生成近边界数据代替私有训练数据，并且通过训练生成对抗网络生成新的、私有化的近边界数据，在解决数据集推断只能用于私有训练数据集问题的同时，也防止了本文的近边界数据被轻易复制和模仿，保持私有数据在推断模型所有权时的优势。

2）本文的方法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。如果近边界数据在可疑模型上并没有表现出近边界性，那么不会判定该模型是盗窃模型。第五章第四节中验证了这一点，本文提出的近边界数据，在无关模型上不会表现出近边界性。

问题定义：本文定义了一个 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 $j=1,2,..,n$。$Z_1,Z_2,..,Z_n$ 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本文使用，因此

$$g_j(x)=\frac{exp(Z_j(x))}{\sum_{i=1}^{n}exp(Z_i(x))} \tag{4.2}$$

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 $y=arg\max_j g_j(x)=arg\max_j Z_j(x)$。

4.3.1 设计目标

根据现有的工作，本文提出的的方法在源模型训练后进行部署，且在黑盒环境下推断模型所有权。本文的方法不关注模型被盗窃的过程，而是聚焦在准确地推断 DNN 模型的所有权和识别不法分子的模型盗窃行为。现在大多数所有权验证技术都是基于黑盒环境。因为模型所有者和攻击者通常不会提供完整模型，而是以 API 的形式提供商业服务，因此黑盒模式的适用情况更加广泛。本文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，并通过本文的方法推断所有权。无论是在白盒的环境还是在黑盒的环境下，本文提出的方法均可以用来推断模型所有权。

本文提出的方法需要在面对各种盗窃模型时成功推断模型所有权，与此同时，需要保持 DNN 模型的性能并且不能对无关模型产生错误的所有权误导。因此本文提出方法的设计目标如下：

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%。

2）**可继承性**：如果可疑模型与源模型相同或从源模型派生而来，则私有近边界数据在这些模型中均表现出近边界性。反之，近边界数据在无关模型中没有明显特征，防止产生所有权推断误导。

3）**有效性**：如果可疑模型与源模型相同或从源模型派生而来，则根据源模型构造的私有近边界数据比其他近边界数据更加靠近分类边界，这是本文方法成功推断模型所有权的依据。

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，这是本文方法能广泛应用的关键。

5）**不可获得性**：敌手无法获得私有的近边界数据，否则私有近边界数据不在有更加靠近分类边界的优势，无法成功推断模型所有权。

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，并通过对比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。

本文将在第五章对该方法进行详细的实验评估，实验评估后，逐一验证分析是否达到这 6 个设计目标。

4.3.2 方法概述

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，本文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本文所利用并构造近边界数据，经过测试，本文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数据的近边界特性不会因为受到模型窃取攻击产生的模型修改而消失。受到这个特点的启发，将近边界数据作为水印验证所有

权是传统的思路，虽然不会对模型的精度造成影响，但是这样的水印是脆弱的，很容易受到御歧义攻击，因此本文提出由近边界数据驱动的所有权推断方法。

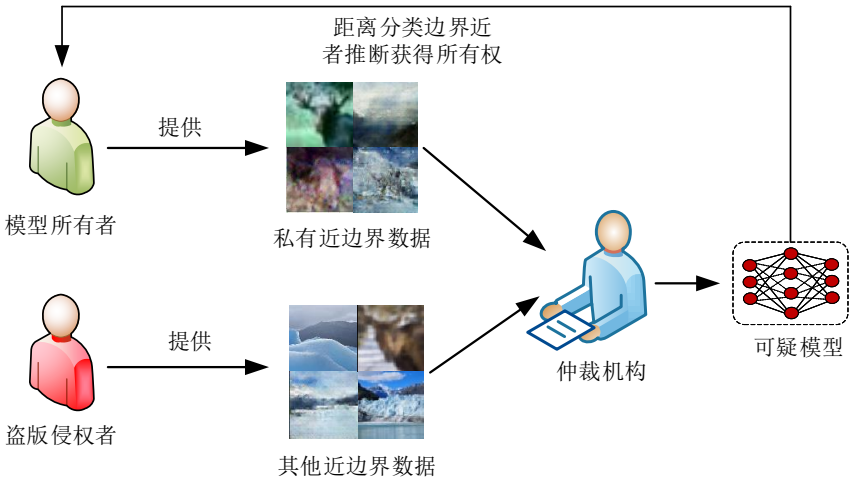


图 4.3 推断示意图

如图4.3所示，本文提出方法的主要思想是构造私有的近边界数据，以数据驱动推断模型的所有权。当模型所有者怀疑可疑模型是盗窃自己的模型时，向官方机构发起仲裁。模型所有者和可疑对手分别提供各自的私有近边界数据，仲裁机构分别计算双方数据的输出结果，统计测试样本数据点到分类边界的距离，距离近的判定获得模型所有权。

4.3.3 假设检验

根据第三章的讨论结果，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，本文提出了推断模型所有权的方法，这是一种“最”的思路。与过去工作中利用模型水印和指纹验证模型所有权相比，本文方法使用数据在对应模型上结果作为所有权推断依据，结果的可比性和唯一性可以有效避免歧义攻击。

上一小节中提到，模型所有者向仲裁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，仲裁机构分别计算各自数据到目标分类边界距离，最靠近目标分类边界的近边界数据所有者将获得模型所有权。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，本文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：本文假设事件 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。本文重复 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。

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 $\overline{X}$
4: 计算可疑对手近边界数据样本均值 $\overline{X}_S$
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**

结合生成使用 CW-L_2 生成初始近边界数据和使用 DCGAN 私有化近边界数据的过程，加上假设检验比对结果的差异性，本文提出方法的整体执行流程如图4.4所示。

第四节 本章小结

近边界数据是本文方法推断所有权的依据，不能轻易被盗窃者复制伪造。本章使用初始近边界数据训练 DCGAN，使之学习到近边界数据的特征，然后使用生成器生成新的、私有化的近边界数据，使所有者保持私有数据的优势。接着设计了新的损失函数并微调源模型，使近边界数据更加靠近目标分类边界, 增强本文方法面对各种模型盗窃技术的性能和防御性。然后在私有近边界数据的

基础上，提出本文基于近边界数据的模型所有权推断方法，最后提出使用假设检验的方法来比对推测结果的差异性。

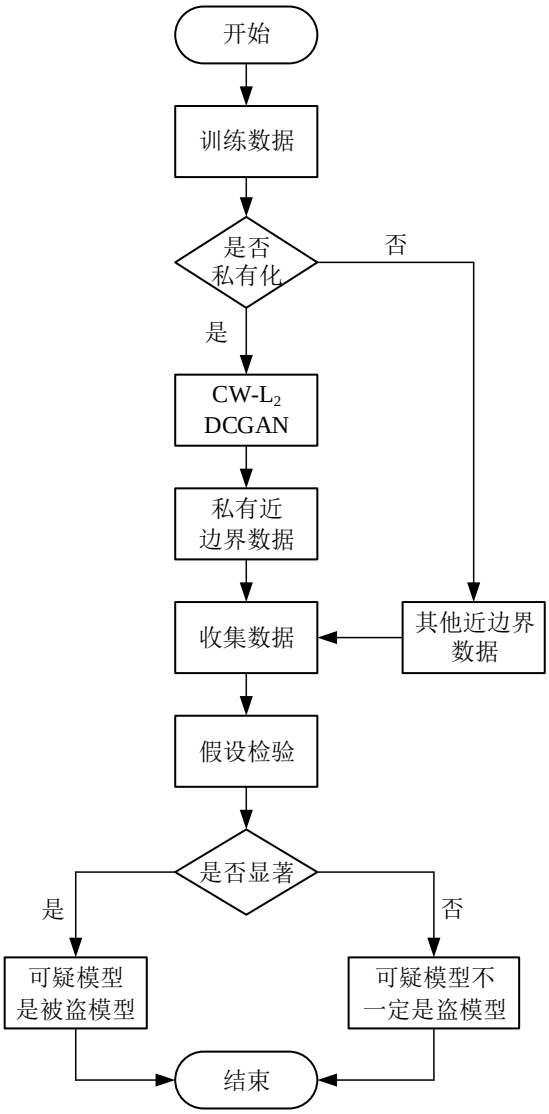


图 4.4　方法整体流程图

第五章 基于近边界数据的模型所有权推断方法分析

本文在三个开源数据集上进行实验，选择 ResNet18 作为评估的源模型，VGG11 作为对照的无关模型。本章将从初始近边界数据的生成算法、近边界数据私有化方法、近边界数据的可继承性、源模型微调的影响、模型所有权推断的有效性和近边界数据规模的可伸缩性这几个方面对本文提出的方法进行评估和分析。本文采用了几种主流的模型窃取攻击方法，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盗窃模型。

第一节 实验设置

5.1.1 数据集

CIFAR-10[61]：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[1]：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[2]：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 实验环境和参数设置

（1）实验环境

本文在实验中使用的硬件与软件配置如表5.1所示，实验使用的机器配备 16 核 Intel i7-11700KF CPU、NVIDIA GeForce RTX 3080 Ti 显卡、16GB 内存以及 Ubuntu 20.04 LTS 操作系统，实验代码均使用 Python 语言和 Pycharm 工具基于 Pytorch 框架实现。

[1]https://datahub.io/dataset/architectural-heritage-elements-image-dataset
[2]https://www.kaggle.com/datasets/puneet6060/intel-image-classification

表 5.1 硬件与软件配置

硬件/软件	配置
操作系统	Ubuntu 20.04 LTS
CPU	Intel Core i7-11700KF @ 3.6GHz
内存	16GB
显卡	NVIDIA GeForce RTX 3080 Ti
CUDA 版本	11.6
深度学习框架	Pytorch 1.9.0
开发工具	Pycharm
开发语言	Python 3.7.11

（2）参数设置

源模型训练：训练过程中使用 Adam 优化器并将学习率 (Learning rate)，训练轮次 (Epoch) 和每批次大小 (Batch size) 分别设置为 0.0001,200 和 64，其他参数为默认值。

模型蒸馏：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 α=0.7，训练过程中使用 Adam 优化器并将学习率，训练轮次和每批次大小分别设置为 0.0001,20 和 64，其他参数为默认值。

模型微调：蒸馏模型实验选择固定模型其他层参数，重置全连接层参数进行微调，训练过程中使用 Adam 优化器并将学习率，训练轮次和每批次大小分别设置为 0.0001,10 和 64，其他参数为默认值。

模型剪枝：分别以 10%，30%，50% 的剪枝率对源模型权重进行剪枝，其他参数为默认值。

初始近边界数据生成算法：初始近边界数据生成采用改进的 $CW\text{-}L_2$ 算法，选择有目标的生成方式，指定生成样本类别。训练过程中使用 Adam 优化器并将学习率，训练轮次和每批次大小分别设置为 0.001,1000 和 64，, 二分搜索次数设置为 6，其他参数为默认值。

近边界数据私有化方法：私有近边界数据生成器采用 DCGAN 的基础结构，训练过程中使用 Adam 优化器并将学习率，训练轮次和每批次大小分别设置为 0.0002,2000 和 64，其他参数为默认值。

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新设计的损失函数微调源模型，具体设置为 10 个轮次交替一次且交替次数最多为 10 次。

5.1.3 源模型和盗窃模型

本节模型训练过程中参数设置与上一小节保持一致。

（1）源模型和无关模型

本文的目标模型选用 ResNet18 网络架构，在上述三个数据集上分别进行训练，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5.2所示。

表 5.2 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

本文在数据处理阶段将统一图片尺寸更改为 64，为了更好的提取到图片中的特征，提高模型准确率，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。

如表5.3所示，更改结构后 ResNet18 在三个数据集上的准确率得到提升。其中，模型在 CIFAR-10 的准确率相较于其他两个数据集提升较大，因为 CIFAR-10 数据集本身的尺寸为 32x32，是小尺寸图片。

表 5.3 模型更改结构前后准确率对比

数据集	CIFAR-10	Heritage	Intel_image
更改前准确率	0.853	0.862	0.848
更改后准确率	0.886	0.879	0.854

更改结构后，ResNet18 在三个数据集上的准确率与训练轮次的关系如图5.1所示，基本在训练 125 次之后模型收敛，准确率基本维持不变。

（2）盗窃模型

微调模型：选择固定除全连接层外其余所有层的参数，将全连接层参数初始化后，进行微调作为盗窃模型。

剪枝模型：根据权重的 L_1 范数排序，对卷积层和全连接层分别以 10%，30%，50% 的裁剪率进行裁剪作为盗窃模型。

蒸馏知识：选择蒸馏至 VGG11 作为盗窃模型，蒸馏时将蒸馏温度设置为 20 并且教师模型比例 α=0.7。

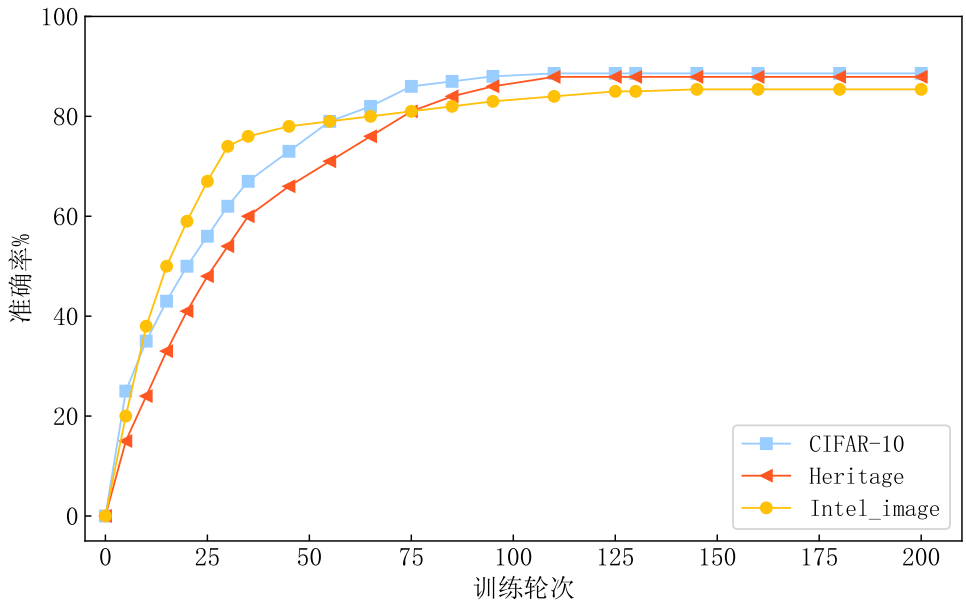


图 5.1 源模型在三个数据集上的准确率

第二节 初始近边界数据生成算法对比

本小节将对第三章第三节中提到的 FGSM，IGSM，RFGSM 和 CW-L_2 几种算法进行测试，评估指标为生成的初始近边界数据到分类边界的距离，距离愈小愈优。除对 CW-L_2 引入距离参数进行改进外，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，本文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 的实验设置与第一节保持一致。

表 5.4 不同算法生成样本到分类边界的平均距离

数据集	分组	FGSM	IGSM	RFGSM	CW-L_2
CIFAR-10	1	0.557	0.430	0.418	**0.066**
	2	0.461	0.419	0.373	**0.103**
	3	0.586	0.369	0.356	**0.112**
Heritage	1	0.347	0.356	0.314	**0.014**
	2	0.377	0.340	0.281	**0.016**
	3	0.348	0.332	0.276	**0.010**
Intel_image	1	0.522	0.447	0.353	**0.088**
	2	0.475	0.506	0.387	**0.122**
	3	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5.4所示，每组测试下的最小距离已在表格中加粗。

从表中可以看出 FGSM 生成的样本效果较差，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，牺牲一定的性能是合理的。与 FGSM 相比，IGSM 和 RFGSM 的效果稍好，因为这两种算法在执行过程中引入了迭代的步骤。通过每次迭代一小步，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5.4中的结果相符，即大部分情况下，IGSM 和 RFGSM 的生成的对抗性样本到分类边界的平均距离比 FGSM 生成的样本更近。

经过改进后的 CW-L_2 算法在这几种方法中表现显著，相比其他三种方法，生成的对抗性样本到分类边界的距离小数十倍。这是因为该算法执行过程较为复杂，且速度较慢，在迭代过程中引入了二分查找和神经网络来训练参数以更好地生成对抗性样本。为进一步提升该算法的性能，本文在算法中引入了到分类边界距离这一参数，使得生成的对抗性样本更加接近分类边界，并在达到预定阈值 θ 时提前终止算法，从而一定程度上缓解了效率低下的问题。

第三节　近边界数据私有化方法对比

本小节将对第四章第一节中提到的 BGAN，BEGAN 和 DCGAN 三种生成对抗网络进行测试。与上一节一致，评估指标为训练收敛 GAN 后，生成器生成的新数据到目标分类边界的平均距离，距离愈小者愈优。

表 5.5　不同 GAN 生成样本到分类边界的平均距离

数据集	分组	BGAN	BEGAN	DCGAN
CIFAR-10	1	0.189	0.212	**0.134**
	2	0.141	0.092	**0.072**
	3	0.187	**0.121**	0.210
Heritage	1	0.126	0.132	**0.097**
	2	0.221	0.167	**0.128**
	3	0.156	0.175	**0.144**
Intel_image	1	0.221	0.114	**0.042**
	2	0.088	0.263	**0.037**
	3	0.186	**0.103**	0.120

BGAN，BEGAN 和 DCGAN 生成的数据到分类边界的距离如表5.5所示，每组测试下的最小距离已在表格中加粗。在大部分测试情况下，GCGAN 生成的数据距离分类边界的距离最小。

BGAN 和 BEGAN 都对原始 GAN 进行了改进，提升了生成图片的效果。在第四章第一节中，本文详细介绍了四点 DCGAN 相比于原有架构的改进。卷积对图像特征有着很强的处理能力，在这三种架构中，虽然 BEGAN 和 DCGAN 都是基于卷积神经网络的，但是实际测试中 DCGAN 的改进更加适配本文的数据。所以在本文的图像数据下，DCGAN 表现出比其他两种网络结构更优秀的性能。因此，本文选择 GCGAN 来对初始的近边界数据进行特征提取，进而生成新的、私有化近边界数据。

第四节　近边界数据的可继承性验证

本文提出的近边界数据不仅需要在源模型在表现出靠近分类边界的特点，而且在所有盗窃模型中也需要有这个性质。即近边界数据的近边界性可以继承到源模型派生出的模型上，这是本文选择近边界数据作为推断模型所有权的原因。本节将在三个数据集上，针对盗窃模型和无关模型，对不同目标分类边界下近边界数据的近边界性和可继承性进行测试与验证。每一组实验使用 64 个独立的私有近边界数据进行测试，针对 3 个数据集，每个数据集 4 条不同的分类边界，共 16 组测试。

如图5.2，图5.3，图5.4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点。从图中可以发现，相比于其他模型，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使用公开数据集在源模型的基础上生成的。本文从众多生成对抗性样本的方法中挑选了效果最好的 CW-L_2 方法，然后在该方法的基础上引入距离参数，改进算法的迭代过程，使之生成非常靠近目标分类边界的近边界数据。

2）使用 DCGAN 生成器提取近边界数据的特征后，生成了新的私有化近边界数据。在此基础上，本文设计了新的损失函数微调源模型分类边界，这使得私有近边界数据同样非常接近模型分类边界。

3）盗窃模型在从源模型派生的过程中，涉及到模型的修改。虽然这些修改操作不会使近边界数据在这些模型上失去近边界性，但是会使模型分类边界发

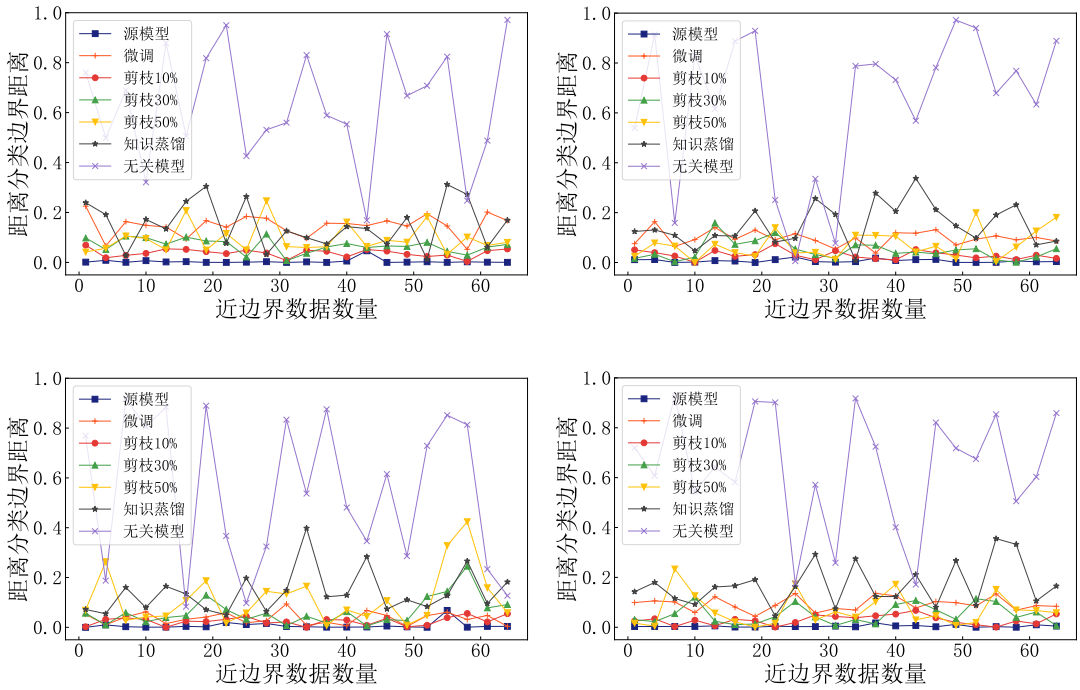


图 5.2 CIFAR-10 上 4 条不同分类边界下的近边界数据表现

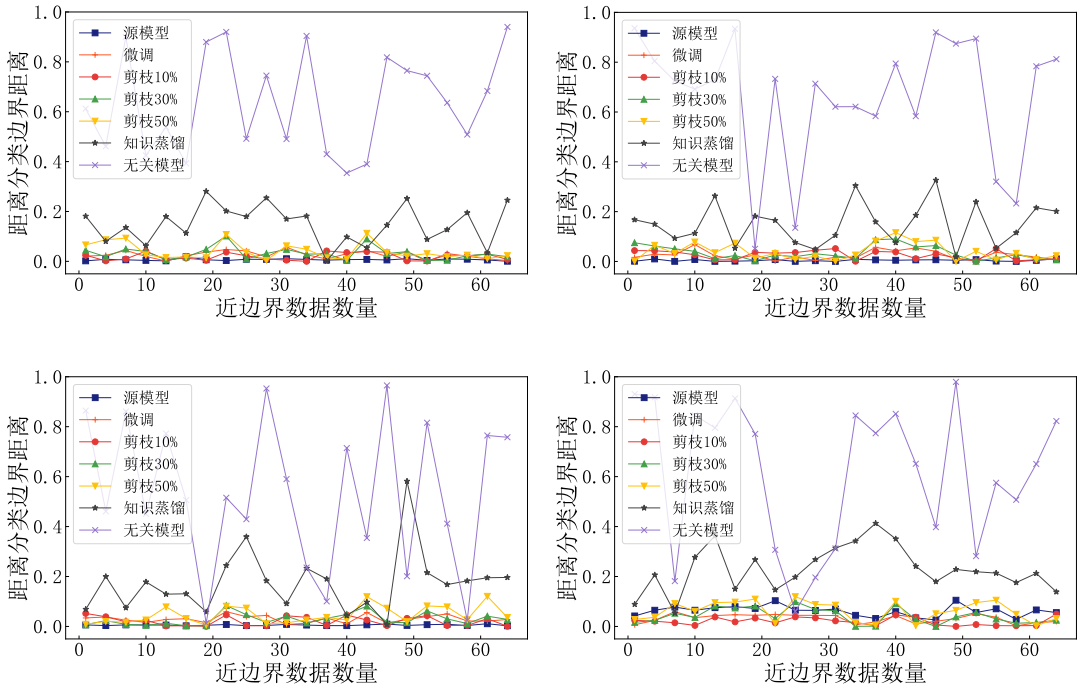


图 5.3 Heritage 上 4 条不同分类边界下的近边界数据表现

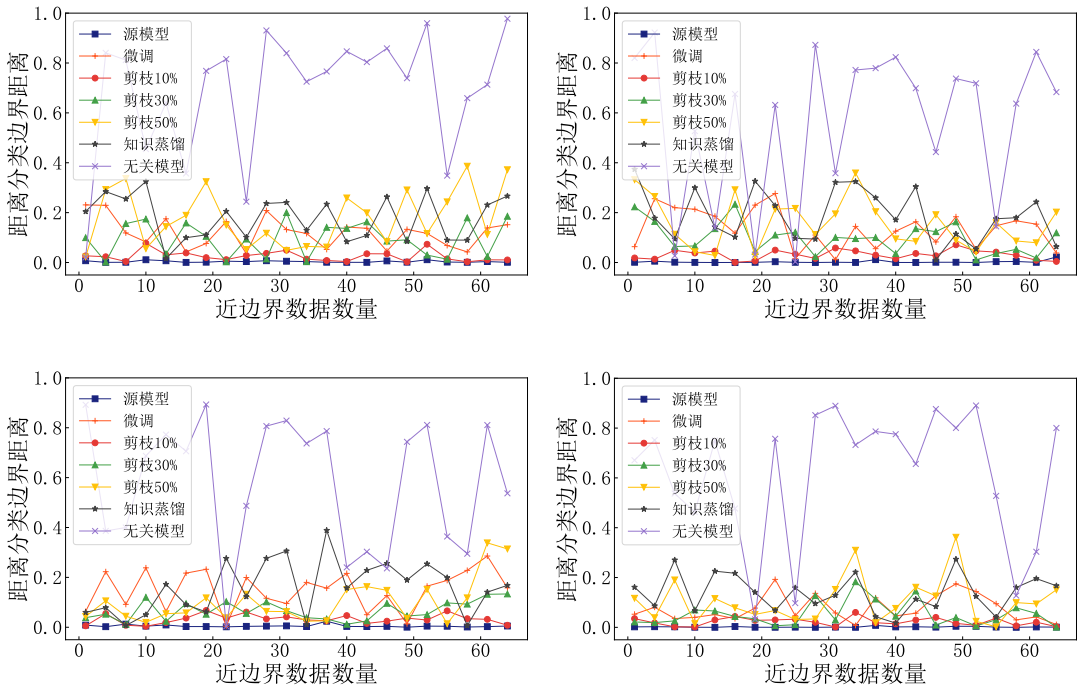


图 5.4 Intel_image 上 4 条不同分类边界下的近边界数据表现

生轻微偏移，从而近边界数据在盗窃模型上距离分类边界距离增大。

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数据可以被继承，这跟本文使用的模型是 DNN 分类器有关。分类边界是分类器的重要特征，本文的近边界数据是根据分类边界构造的。即使模型窃取攻击对模型有一定程度的修改，分类器的分类边界不会发生很大的变化。因此，近边界数据拥有可继承性，在盗窃模型上同样表现出靠近分类边界的特点。这是本文提出方法的基础，因为本文方法的原理是，利用所有者的私有近边界数据和可疑对手提供的近边界数据到分类边界距离的大小来推断模型所有权（距离小者推断获得所有权）。如果私有近边界数据不表现出近边界性，那么就不适用于推断模型所有权。

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较远。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进行训练。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，但是本文提出的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模型盗窃方法仍然适用。

图中还有一个点是近边界数据在无关模型上并不表现出近边界性，这是合理的。即使训练数据相同，分类边界不可能完全一样。本文提出的方法不应该对正常的无关模型产生误判，错误推断所有权。

第五节 源模型微调的影响评估

本小节将对第四章第二节中使用私有近边界数据微调源模型产生的影响进行评估。本文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。评估指标为模型的准确率，准确率变化愈小，说明产生的不良影响越小。

表 5.6 微调分类边界对准确率的影响

分类边界	数据规模	CIFAR-10 准确率 (0.886)	Heritage 准确率 (0.879)	Intel_image 准确率 (0.854)
		准确率	准确率	准确率
分类边界 1	64	0.873	0.862	0.825
	128	0.862	0.858	0.829
	256	0.862	0.854	0.826
	512	0.857	0.852	0.839
分类边界 2	64	0.871	0.867	0.843
	128	0.870	0.862	0.839
	256	0.860	0.859	0.828
	512	0.859	0.851	**0.824**
分类边界 3	64	0.871	0.865	0.841
	128	0.868	0.857	0.833
	256	0.858	0.855	0.831
	512	**0.856**	0.851	0.825
分类边界 4	64	0.873	0.863	0.847
	128	0.873	0.860	0.843
	256	0.866	0.854	0.838
	512	0.862	**0.850**	0.831
分类边界 5	64	0.876	0.866	0.846
	128	0.866	0.861	0.834
	256	0.868	0.857	0.829
	512	0.861	0.853	0.825

经过不同规模近边界数据微调后，源模型的准确率如表5.6所示，每个数据

集下影响最大，即准确率最低的，已在表中加粗。

从表中可以发现，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。不考虑偶然因素的影响，在大部分情况下，随着微调模型近边界数量的增多，模型准确率逐渐下降。这是合理的，因为近边界数据本身和正常训练数据不同，使用越多的异常数据参与训练，对模型的性能影响越大。但是在另一个角度，微调源模型的目的是使得私有近边界数据更加靠近目标分类边界，来提高后续推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，后续推断的效果会更好。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，模型的准确率下降不多。一方面，这是因为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，本文将学习率设置较低，仅为 0.0001。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。所以对源模型微调之后，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解决模型所有权保护问题时，模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，模型准确率变化在 3% 以内。

第六节　模型所有权推断有效性评估

本文提出的方法目标是推断模型的所有权，本节将对此方法的有效性进行评估。根据本文方法的流程，模型所有者和可疑对手（可能是模型盗窃者）均需向官方仲裁机构提供各自的近边界数据，然后通过数据和模型计算到分类边界的距离，再根据第四章第三节中提到的假设检验进行进行结果对比，判断可疑模型是否从源模型派生。

在本节中，通过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来说明本文方法的有效性。首先，本文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，来反映成功推断所有权的置信度。本节的评估指标为 $\Delta\mu$ 和 p 值，$\Delta\mu$ 愈大和 p 值愈小，推断的置信度愈高。

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	$\Delta\mu$	p 值	$\Delta\mu$	p 值	$\Delta\mu$	p 值	$\Delta\mu$	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	$\mathbf{10^{-2}}$	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	$\mathbf{10^{-2}}$	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	$\mathbf{10^{-2}}$	0.395	10^{-3}	0.360	$\mathbf{10^{-2}}$	0.308	10^{-3}	0.287	$\mathbf{10^{-2}}$

不同数据集下，针对不同分类边界，所有者的私有近边界数据和其他近边界数据计算得到的 $\Delta\mu$ 和 p 值如表5.7所示，不同目标分类边界下 p 值的最小情况已在表格中加粗。

从表中可以发现，在每个数据集，每条分类边界上 p 值呈从上到下增大的趋势，$\Delta\mu$ 呈减小趋势。尽管如此，在全部情况中，p 值均低于 0.05，即至少有 95% 以上的置信度，推断盗窃模型从源模型派生。即本文的方法在不同的模型窃取方法中推断模型的所有权，均有显著的效果，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 值减小是因为这些模型窃取方法对模型的修改逐渐增大，尤其是知识蒸馏。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。从表中，可以观察到本文提出的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来推断模型所有权的方法对大多数模型盗窃技术都是可靠的，模型被盗窃的置信度至少达到 95%，证明了本文方法的有效性和鲁棒性。

第七节 近边界数据规模可伸缩性评估

本节将测试使用不同规模的近边界数据，在推断模型所有权时的可伸缩性。本文提出的方法需要对数据进行采样从而进行假设检验，通常情况下，样本数量越大，检验过程中因随机因素而产生的不利影响就会越小，更能准确的推断可疑模型所有权。

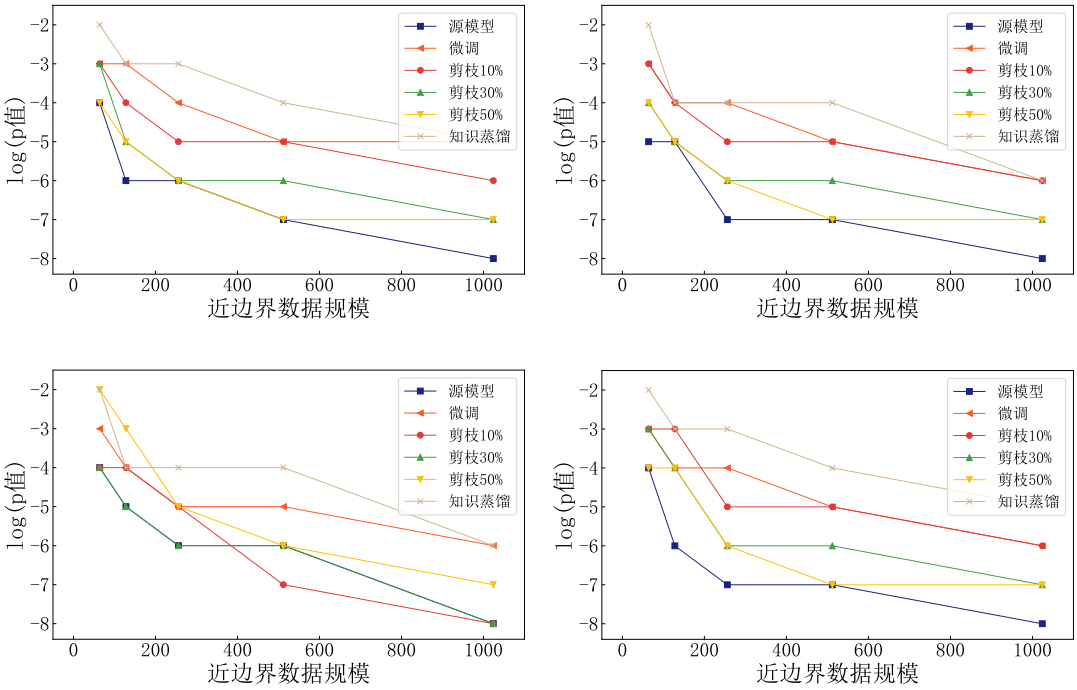


图 5.5 CIFAR-10 上 4 条不同分类边界下的近边界数据规模可伸缩性

如图5.5，图5.6，图5.7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通过可疑模型，计算到目标分类边界的距离。可疑对手的近边界

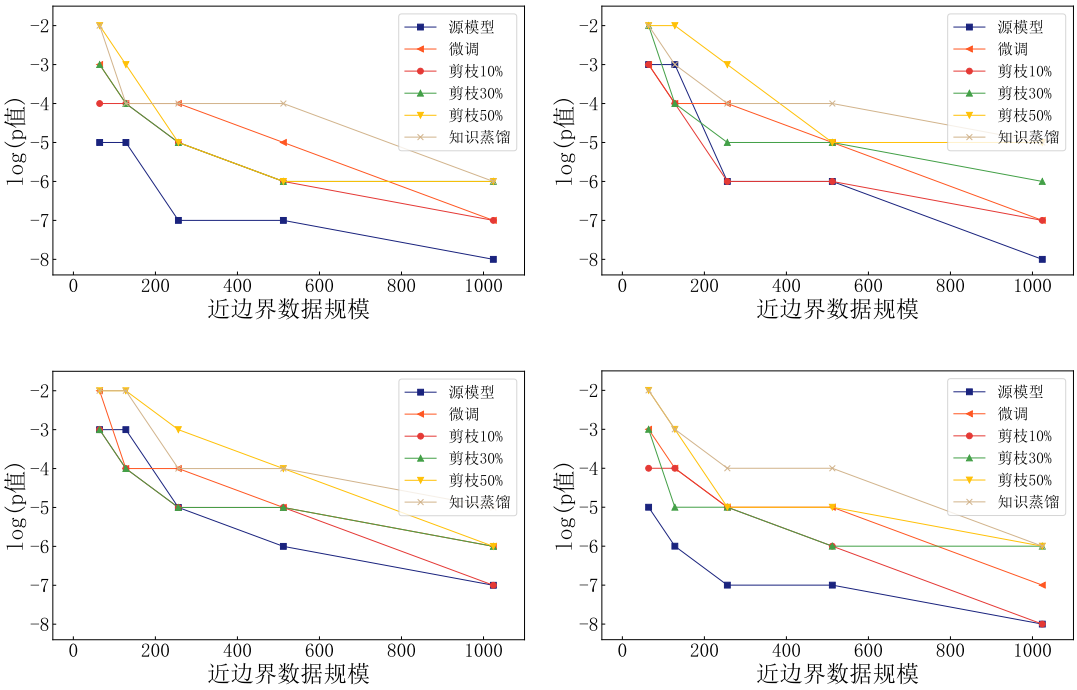


图 5.6 Heritage 上 4 条不同分类边界下的近边界数据规模可伸缩性

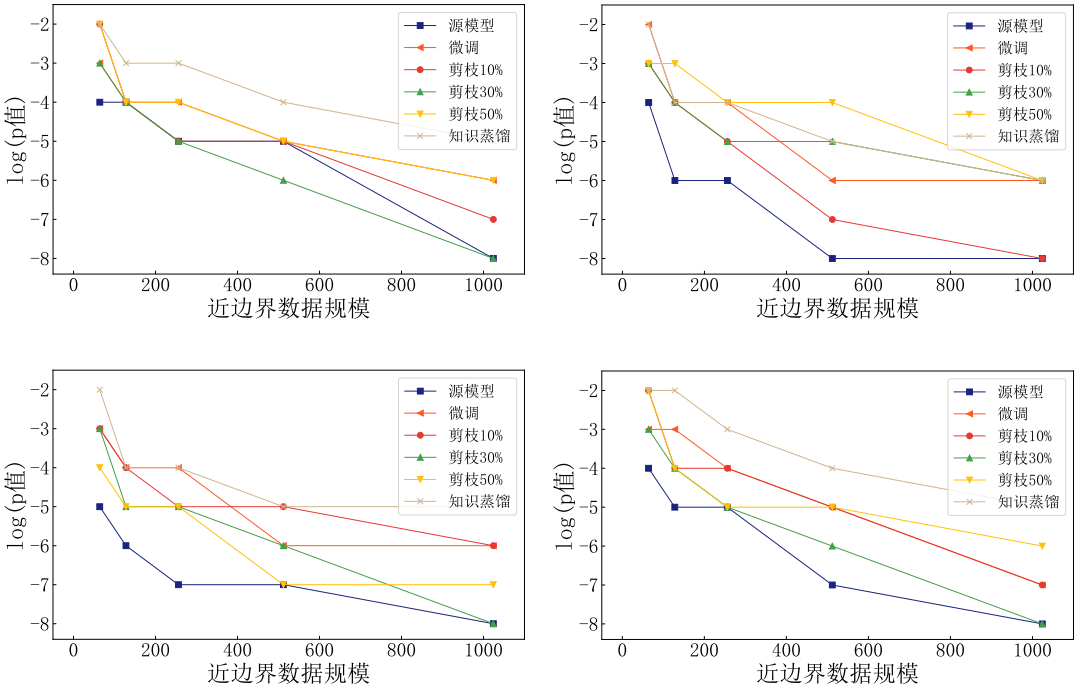


图 5.7 Intel_image 上 4 条不同分类边界下的近边界数据规模可伸缩性

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，由于随机因素的影响，存在一小部分可疑对手提供的数据到分类边界的距离和所有者提供的私有近边界数据相近，甚至更小的情况，这会对本文的方法产生一定的干扰。随着测试样本规模的增大，这种随机因素的影响会被逐渐消除。从图中可以发现，随着近边界数据规模的增大，p 值逐渐减小。但这并不说明本文的方法对小规模数量的近边界数据缺少鲁棒性，从图中可以发现，即使在数据量为 64 的情况下，p 值仍然小于 0.05，这证明本文的方法对于小数样本量同样有显著的效果，可以高置信度的推断模型所有权。

方法设计目标分析

以下对第四章第三节中提出的方法设计目标是否达到进行分析。

1）**精确性**：本文方法对模型精确性的影响主要来源于使用近边界数据微调源模型目标分类边界，这是为了增加推断模型所有权的置信度。第五节评估了不同规模近边界数据微调源模型的影响，各种情况下，模型精度下降不超过 3%。

2）**可继承性**：第四节中通过 16 组测试，验证了近边界数据的可继承性，这是选择近边界数据进行所有权推断的原因。另外，无关模型没有对近边界数据没有表现出明显特征，表明本文的方法不会产生误判。

3）**有效性**：第六节中，使用私有近边界数据和其他近边界数据，针对各种盗窃模型，计算距离结果。然后对双方的结果进行假设检验，结果表明了本文的方法推断所有权的有效性。此外，与过去工作中利用模型水印和指纹验证模型所有权相比，本文方法使用数据在对应模型上结果作为所有权推断依据，结果的可比性和唯一性可以有效避免歧义攻击。

4）**鲁棒性**：第四节和第六节中，对各种盗窃模型进行了测试，本文的方法均有效，表明了本文方法的鲁棒性。

5）**不可获得性**：本文使用改进的 CW-L_2 构造了初始近边界数据，并且设计了基于 DCGAN 的特征提取器，私有化近边界数据。模型盗窃者由于不知道约定的分类边界，需要大量的测试，这会导致极大的成本代价。不存在无法攻击的防御方法，但可以从攻击成本上进行约束。

6）**高效性**：本文方法使用模型所有者和盗窃者提供的近边界数据输入模型后计算结果，然后通过假设检验的方式对比结果差异。该过程执行方便，不涉及复杂的处理，可以高效的进行。

第八节 本章小结

本章在 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与分析。第二节，对不同初始近边界数据生成算法进行对比，结果表明 CW-L_2 方法可以满足本文的需求，生成足够靠近目标分类边界的近边界数据。第三节，对近边界数据私有化方法进行了对比，结果表明基于 DCGAN 的网络架构可以更好的提取数据样本特征，生成新的靠近分类边界的数据。第四节，通过 16 组测试验证了近边界数据的可继承性，这是本文方法选择近边界数据作为推断数据的原因。第五节，测试了微调分类边界对模型准确率的影响，保护模型知识产权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。第六节，评估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% 以上的置信度推断可疑模型是盗窃模型。第七节，对假设检验的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，该方法也适用于类似 64 的小样本数据情况。

第六章 总结与展望

本章对本文提出的基于近边界数据的模型所有权推断方法进行总结，主要包括研究背景、存在问题，实现方案以及主要贡献。然后通过分析方法的不足之处，提出对未来工作的展望。

第一节 工作总结

随着科技的不断发展，深度神经网络模型逐渐成熟，并在社会发展中扮演着日益重要的角色。然而，由于训练成熟、高性能的 DNN 模型需要昂贵的成本，不法分子开始对这些模型发起窃取攻击，带来了严重的知识产权问题。

本文主要针对神经神经网络模型的知识产权保护方法进行研究。模型水印和模型指纹是目前解决模型知识产权问题的两种主要方法，通过相关工作的调研发现，这两种方法在验证模型所有权时很难抵御歧义攻击。针对上述问题，本文提出了使用数据驱动推断模型所有权，代替传统验证所有权的新思路。本文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，如果这种属性会被源模型派生出的模型所继承，那么就可以从这个角度设计算法来推断模型的所有权。根据这个思想，本文构造了一类特殊的数据——近边界数据，作为推断所有权的依据。本文的主要贡献如下：

1）提出基于数据推断所有权代替验证所有权，解决验证所有权带来的歧义攻击问题。推断模型所有权是比较某类数据在模型上的最优性，最优者推断获得该模型的所有权。这种方式并不是去验证特定的水印或指纹，结果的可比性和唯一性可以有效避免歧义攻击。

2）设计近边界数据这一特殊数据，作为推断模型所有权的依据。本文基于三个公开数据集在盗窃模型和无关模型上做了充分的测试，验证了近边界数据的近边界性可以被盗窃模型所继承，而在无关模型上不会有近边界的特点，因此可以作为推断所有权的依据。利用对抗性样本靠近模型分类边界的特点，在 CW-L_2 算法的基础上，实现本文生成初始近边界数据的算法。实验表明此算法生成的数据足够靠近分类边界，满足推断所有权的要求。

3）对近边界数据进行私有化处理并基于处理后的数据对源模型进行微调，

针对各种模型盗窃技术增强本文方法的性能和防御性。为了防止近边界数据被轻易伪造，设计了基于 DCGAN 的特征提取器，提取近边界数据特征后，使用其生成器生成新的、私有化的近边界数据。在此基础之上，重新设计新的损失函数微调源模型，使近边界数据更加靠近目标分类边界，成功推断模型所有权的置信度达 95% 以上。在三个公开数据集和主流盗窃模型上的实验证明了本文提出的方法在推断模型所有权时的有效性和鲁棒性。

第二节 未来展望

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，本文提出数据驱动推断模型所有权为模型知识产权保护提供了新思路。但是，本文仍存在一些不足之处：

1）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法生成近边界数据。

2）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数据到分类边界的距离值得探讨。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，原始的分类边界应该如何映射到新的分类边界。因此未来的工作应该加入对大分类情模型的研究。

3）本文提出的方法主要是针对神经网路分类模型的。对于非分类的模型，如何寻找类似近边界数据的特殊数据是能应用推断模型所有权方法的关键。

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更多保护模型知识产权的新方法，防止模型盗窃者发起针对性的攻击，以更好的保护神经网络模型的知识产权。

参考文献

[1] Samek W, Montavon G, Lapuschkin S, et al. Explaining deep neural networks and beyond: A review of methods and applications. [J]. Proceedings of the IEEE, 2021, 109 (3): 247–278.

[2] Wu L, Chen Y, Shen K, et al. Graph neural networks for natural language processing: A survey. [J]. Foundations and Trends® in Machine Learning, 2023, 16 (2): 119–328.

[3] Lauriola I, Lavelli A, Aiolli F. An introduction to deep learning in natural language processing: Models, techniques, and tools. [J]. Neurocomputing, 2022, 470: 443–456.

[4] Caucheteux C, King J.-R. Brains and algorithms partially converge in natural language processing. [J]. Communications biology, 2022, 5 (1): 134.

[5] Buhrmester V, Münch D, Arens M. Analysis of explainers of black box deep neural networks for computer vision: A survey. [J]. Machine Learning and Knowledge Extraction, 2021, 3 (4): 966–989.

[6] Lindsay G W. Convolutional neural networks as a model of the visual system: Past, present, and future. [J]. Journal of cognitive neuroscience, 2021, 33 (10): 2017–2031.

[7] Gururaj N, Vinod V, Vijayakumar K. Deep grading of mangoes using Convolutional Neural Network and Computer Vision. [J]. Multimedia Tools and Applications, 2022: 1–26.

[8] Dua S, Kumar S S, Albagory Y, et al. Developing a Speech Recognition System for Recognizing Tonal Speech Signals Using a Convolutional Neural Network. [J]. Applied Sciences, 2022, 12 (12): 6223.

[9] Zhang Z, Liu J, Liu G, et al. Robustness verification of swish neural networks embedded in autonomous driving systems. [J]. IEEE Transactions on Computational Social Systems, 2022: 1–10.

[10] Shakeel P M, Burhanuddin M, Desa M I. Automatic lung cancer detection from CT image using improved deep neural network and ensemble classifier. [J]. Neural Computing and Applications, 2022: 1–14.

[11] Ling C, Tollmar K, Gisslén L. Using deep convolutional neural networks to detect rendered glitches in video games. [C] // Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment. Vol. 16. 1. 2020: 66–73.

[12] 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.

[13] Wang S, Chang C.-H. Fingerprinting deep neural networks-a deepfool approach. [C] // 2021 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE. 2021: 1–5.

[14] Li M, Zhong Q, Zhang L Y, et al. Protecting the intellectual property of deep neural networks with watermarking: The frequency domain approach. [C] // 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). IEEE. 2020: 402–409.

[15] Tanuwidjaja H C, Choi R, Baek S, et al. Privacy-preserving deep learning on machine learning as a service—a comprehensive survey. [J]. IEEE Access, 2020, 8: 167425–167447.

[16] 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.

[17] Hu H, Pang J. Stealing machine learning models: Attacks and countermeasures for generative adversarial networks. [C] // Annual Computer Security Applications Conference. 2021: 1–16.

[18] Yue Z, He Z, Zeng H, et al. Black-box attacks on sequential recommenders via data-free model extraction. [C] // Proceedings of the 15th ACM Conference on Recommender Systems. 2021: 44–54.

[19] Guo Y, Shi H, Kumar A, et al. Spottune: transfer learning through adaptive fine-tuning. [C] // Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 4805–4814.

[20] Liu Z, Sun M, Zhou T, et al. Rethinking the value of network pruning. [J]. ArXiv preprint arXiv:1810.05270, 2018.

[21] Deng L, Li G, Han S, et al. Model compression and hardware acceleration for neural networks: A comprehensive survey. [J]. Proceedings of the IEEE, 2020, 108 (4): 485–532.

[22] Gou J, Yu B, Maybank S J, et al. Knowledge distillation: A survey. [J]. International Journal of Computer Vision, 2021, 129: 1789–1819.

[23] Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network. [J]. ArXiv preprint arXiv:1503.02531, 2015.

[24] 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. 1994: 86–90.

[25] 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.

[26] 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. 2017: 269–277.

[27] Zhang J, Chen D, Liao J, et al. Deep model intellectual property protection via deep watermarking. [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, 44 (8): 4005–4020.

[28] Chen H, Rohani B D, Koushanfar F. Deepmarks: A digital fingerprinting framework for deep neural networks. [J]. ArXiv preprint arXiv:1804.03648, 2018.

[29] Fan L, Ng K W, Chan C S. Rethinking deep neural network ownership verification: embedding passports to defeat ambiguity attacks. [C] // Proceedings of the 33rd International Conference on Neural Information Processing Systems. 2019: 4714–4723.

[30] 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. 2018: 159–172.

[31] 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). 2018: 1615–1631.

[32] Le Merrer E, Perez P, Trédan G. Adversarial frontier stitching for remote neural network watermarking. [J]. Neural Computing and Applications, 2020, 32: 9233–9244.

[33] 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.

[34] Zhao J, Hu Q, Liu G, et al. AFA: Adversarial fingerprinting authentication for deep neural networks. [J]. Computer Communications, 2020, 150: 488–497.

[35] Lukas N, Zhang Y, Kerschbaum F. Deep neural network fingerprinting by conferrable adversarial examples. [J]. ArXiv preprint arXiv:1912.00888, 2019.

[36] 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. 2021: 14–25.

[37] Li G, Xu G, Qiu H, et al. A Novel Verifiable Fingerprinting Scheme for Generative Adversarial Networks. [J]. ArXiv preprint arXiv:2106.11760, 2021.

[38] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial networks. [J]. Communications of the ACM, 2020, 63 (11): 139–144.

[39] 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.

[40] Li H, Wenger E, Shan S, et al. Piracy resistant watermarks for deep neural networks. [J]. ArXiv preprint arXiv:1910.01226, 2019.

[41] 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. 2016: 770–778.

[42] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. [J]. ArXiv preprint arXiv:1409.1556, 2014.

[43] Szegedy C, Zaremba W, Sutskever I, et al. Intriguing properties of neural networks. [J]. ArXiv preprint arXiv:1312.6199, 2013.

[44] Goodfellow I J, Pouget-Abadie J, Mirza M, et al. Generative Adversarial Nets. [J]. Stat, 2014, 1050: 10.

[45] Chen Y, Yang X.-H, Wei Z, et al. Generative adversarial networks in medical image augmentation: a review. [J]. Computers in Biology and Medicine, 2022: 105382.

[46] Singh N K, Raza K. Medical image generation using generative adversarial networks: A review. [J]. Health informatics: A computational perspective in healthcare, 2021: 77–96.

[47] Zhou N.-R, Zhang T.-F, Xie X.-W, et al. Hybrid quantum–classical generative adversarial networks for image generation via learning discrete distribution. [J]. Signal Processing: Image Communication, 2023, 110: 116891.

[48] Xu Y, Luo W, Hu A, et al. TE-SAGAN: An Improved Generative Adversarial Network for Remote Sensing Super-Resolution Images. [J]. Remote Sensing, 2022, 14 (10): 2425.

[49] 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.

[50] 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.

[51] 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. 2020: 1340–1346.

[52] Maini P, Yaghini M, Papernot N. Dataset inference: Ownership resolution in machine learning. [J]. ArXiv preprint arXiv:2104.10706, 2021.

[53] 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. 2022: 9595–9603.

[54] Goodfellow I J, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. [J]. ArXiv preprint arXiv:1412.6572, 2014.

[55] Kurakin A, Goodfellow I J, Bengio S. Adversarial examples in the physical world. [G] // Artificial intelligence safety and security. Chapman, Hall/CRC, 2018: 99–112.

[56] Tramèr F, Kurakin A, Papernot N, et al. Ensemble adversarial training: Attacks and defenses. [J]. ArXiv preprint arXiv:1705.07204, 2017.

[57] Carlini N, Wagner D. Towards evaluating the robustness of neural networks. [C] // 2017 IEEE Symposium on Security and Privacy(SP). IEEE. 2017: 39–57.

[58] Devon Hjelm R, Jacob A P, Che T, et al. Boundary-Seeking Generative Adversarial Networks. [J]. ArXiv e-prints, 2017: arXiv–1702.

[59] Berthelot D, Schumm T, Metz L. Began: Boundary equilibrium generative adversarial networks. [J]. ArXiv preprint arXiv:1703.10717, 2017.

[60] Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks. [J]. ArXiv preprint arXiv:1511.06434, 2015.

[61] Krizhevsky A, Hinton G, et al. Learning multiple layers of features from tiny images. [J]. 2009.

图索引

表索引