https://crad.ict.ac.cn
62(8): 2019−2039, 2025

深度学习模型供应链的安全性综述

陈若曦[1] 陈晋音[2,1] 郑海斌[2,1] 阳雪燕[1] 纪守领[3] 陈铁明[4]
[1](浙江工业大学信息工程学院 杭州 310023)
[2](浙江工业大学网络空间安全研究院 杭州 310023)
[3](浙江大学计算机科学与技术学院 杭州 310007)
[4](浙江工业大学计算机科学与技术学院 杭州 310023)
(2112003149@zjut.edu.cn)

Security of Deep Learning Model Supply Chains: A Survey

Chen Ruoxi[1], Chen Jinyin[2,1], Zheng Haibin[2,1], Yang Xueyan[1], Ji Shouling[3], and Chen Tieming[4]
[1](*College of Information Engineering*, *Zhejiang University of Technology University*, *Hangzhou* 310023)
[2](*Institute of Cyberspace Security*, *Zhejiang University of Technology University*, *Hangzhou* 310023)
[3](*College of Computer Science and Technology*, *Zhejiang University*, *Hangzhou* 310007)
[4](*College of Computer Science and Technology*, *Zhejiang University of Technology University*, *Hangzhou* 310023)

Abstract Pre-trained models have mitigated the challenges posed by extensive training data and computational resources, and also give birth to the new paradigm of model development and application, which we refer to as model supply chain. In this framework, a pre-trained model is uploaded by its publisher and subsequently transferred, compressed, and deployed by secondary developers to meet various application needs. This emerging model supply chain introduces additional stages and multiple elements, inevitably leading to security concerns and privacy risks. Despite the widespread adoption of model supply chains, there is currently a lack of systematic review of security threats in them. To address this research gap, we provide a comprehensive overview of the deep learning model supply chain, introducing its concept and fundamental structure. We conduct an in-depth analysis of vulnerabilities at various stages of the model's lifecycle, including design, development, deployment, and usage. Furthermore, we compare and summarize prevalent attack methods, alongside introducing corresponding security protection strategies. To assist readers in effectively utilizing pre-trained models, we review and compare publicly available model repositories. Finally, we discuss potential future research avenues in areas such as security checks, real-time detection, and problem tracing. This review aims to offer insights for safer and more reliable development and use of pre-training models. For the benefit of ongoing research, related papers and open-source codes of the methods discussed are accessible at website: https://github.com/Dipsy0830/DNN-supply-chain-survey.

Key words model supply chain; pre-trained models; security threats; security and privacy; deep neural networks; deep learning

摘 要 预训练模型缓解了训练数据和算力受限的问题,也催生了模型开发和应用的新范式——深度学习

收稿日期: 2024-05-21; **修回日期**: 2024-12-20
基金项目: 浙江省自然科学基金项目(LDQ23F020001); 国家自然科学基金项目(62072406, 62406286); 浙江省重点研发计划(2022C01018)国家重点研发计划(2018AAA010081)
This work was supported by the Zhejiang Provincial Natural Science Foundation (LDQ23F020001), the National Natural Science Foundation of China (62072406, 62406286), the Key R&D Projects in Zhejiang Province (2022C01018), and the National Key R&D Projects of China (2018AAA0100801).
通信作者: 陈晋音(chenjinyin@zjut.edu.cn)

模型供应链. 在这一供应链中, 预训练模型由原始发布者上传, 并经二次开发者进行必要的迁移、压缩与部署, 以适应不同的应用需求. 然而, 这一过程引入了新的环节和多元要素, 随之产生了数据泄露和计算安全等风险. 尽管深度学习模型供应链已被广泛采纳, 目前对于其安全性威胁的系统性研究仍然不足. 针对这一研究空白, 概括了深度学习模型供应链的概念及其基本框架, 调研了相关的研究工作. 依据模型生命周期的不同阶段, 详细分析了设计、二次开发、部署和使用各环节的安全脆弱点, 对可能遭遇的安全性攻击手段进行了系统的梳理、比较和总结, 并探讨了相应的防护策略. 为了帮助读者更好地使用预训练模型, 对现有预训练模型仓库进行了比较. 最后, 基于目前深度学习模型供应链所面临的挑战, 从安全性审查、实时检测和问题溯源等不同角度分析和讨论了未来可能的研究发展方向, 为更安全可靠地开发和使用预训练模型提供了思路. 论文涉及的方法和相关代码归纳整理在网址: https://github.com/Dipsy0830/DNN-supply-chain-survey, 方便研究人员下载使用.

关键词 模型供应链; 预训练模型; 安全风险; 安全与隐私; 深度神经网络; 深度学习

中图法分类号 TP391

DOI: 10.7544/issn1000-1239.202440327 **CSTR:** 32373.14.issn1000-1239.202440327

自 ChatGPT 被推出 2 个月以来, 其活跃用户数已迅速超过 1 亿, 引领着全球问答系统的新潮流. Hugging Face[1] 开源模型库已共享逾 10 万个预训练模型, 月均安装次数超过 100 万. 算法的突破性进展, 以及大规模深度学习模型的能力, 正在重构网络空间的生态. ChatGPT 采用的 GPT-3.5 架构, 其数据标注由 OpenAI 的合作伙伴 Sama 公司完成, 模型训练则在 NVIDIA 支持的 Microsoft Azure AI 超级计算平台上进行, 体现了算法、数据和算力等关键要素的协同, 标志着人工智能产业链的全面成熟. 这种技术进步的推动力量也体现在大规模模型的普及上, 即便面临着高成本和复杂度的挑战, 开源社区的贡献与共享平台的存在, 使得这些先进的技术更加接近广大用户和开发者.

对于计算资源和专业知识受限的用户, 自主训练深度学习模型充满挑战. 因此, 从开源模型库(如 Hugging Face 和 Model Zoo[2])上下载并使用预训练模型, 逐渐成为解决问题的主流选择, 这促进了深度学习模型供应链的发展. 模型设计—二次开发—部署—使用, 多个环节链式进行, 极大缩短了模型开发周期, 在自然语言处理(natural language processing, NLP)[3]和计算机视觉(computer vision, CV)领域[4]得到了广泛应用和实践.

图 1 为深度学习模型供应链的典型架构, 包括模型设计、二次开发、部署和使用 4 个阶段. 发布者设计并利用大量数据训练模型, 训练完成后, 模型被上传到模型仓库; 二次开发者从云端下载预训练模型, 并对其进行迁移或压缩操作, 使其适应特定的下游任务或匹配不同的设备需求; 在部署阶段, 部署者开展算法模块的开发和集成, 进行必要的接口转换, 以便将模型服务部署到边缘端. 最终, 在模型的使用阶段, 终端用户可以通过应用程序编程接口(application programming interface, API)与模型交互, 提供输入数据并接收模型推理的结果, 从而获取所需信息.



Fig. 1 Supply chain of deep learning model

图 1 深度学习模型供应链

用户和开发者们从深度学习模型供应链中受益, 但这一过程也伴随着潜在的安全风险. 由于大多数预训练模型和数据由第三方提供, 缺乏必要的审查和监管, 其安全性难以得到充分保证. 攻击者可能在供应链的任何环节发起攻击, 而多阶段之间的复杂交互为攻击行为提供了更多机会. 例如, 预训练模型可能植入了后门, 这些后门在二次开发阶段被激活. 在推理阶段, 恶意的用户可以利用发布的 API 来破坏模型预测或提取隐私信息.

与传统软件系统的安全性不同, 深度学习模型的开发和使用是一个更为复杂的多因素耦合过程. 它涉及数据、算法、模型、框架等多个因素的耦合, 以及模型压缩和迁移等技术的应用. 这些因素和技术的相互作用将会引入了新的安全威胁. 随着大数据的支持和全球供应链的端到端连接, 深度学习模型被越来越多地集成到供应链的各个环节中, 从数

据的预处理到模型的训练、部署和使用，每个阶段都可能成为攻击的目标. 供应链的广泛性和复杂性意味着安全威胁可能在任何环节中出现，并且可能具有传递性，即一个环节的安全问题可能会影响到整个供应链的安全. 因此，对深度学习模型供应链安全性的研究显得尤为重要. 这不仅涉及模型本身的安全性，还包括整个供应链的安全性，如确保数据的完整性、模型的可靠性、系统的可用性以及用户隐私的保护. 研究供应链中的安全威胁和防御策略，可以为深度学习模型的可靠应用和安全部署提供理论和实践指导，确保人工智能技术的健康、可持续发展.

现有不少论文对深度学习模型的安全性研究进行综述[5-7]，但这些工作多局限于单个模型上的安全性问题，如对抗攻击、模型窃取、后门攻击等. 然而，深度学习模型的供应链安全性涉及更广泛的威胁，包括模型在供应链中的传递性问题、多环节交互问题以及多方参与带来的复杂性等. 这些威胁在供应链的各个环节中产生，并可能对整个系统造成影响.

因此，为了帮助读者更系统、更快速地了解深度学习模型供应链中的安全性问题，本文提出了深度学习模型供应链的概念，并对其架构和流程进行了详尽的介绍. 本文根据模型开发周期的不同阶段，全面梳理了供应链中的潜在安全威胁，并对各种攻击手段进行了系统的分类、剖析和总结. 此外，文中还对可用的防御策略进行了分析和讨论，旨在为未来深度学习模型供应链安全领域的研究提供参考和指导.

1 深度学习模型供应链

本节简要介绍了深度学习模型供应链的基本概念和系统架构，并将其与传统软件供应链的区别做了简要阐述.

1.1 基本概念

深度模型供应链涵盖设计/上传预训练模型、二次开发者下载并对其进行迁移和压缩操作、边缘端部署模型、用户调用模型等多个环节. 下面对相关概念进行介绍.

1）预训练模型. 大规模预训练模型最近取得了巨大成功，成为人工智能领域的一个里程碑. 由于复杂的预训练目标和庞大的模型参数，大规模预训练模型可以有效地从大量标记和未标记的数据中获取知识，并通过微调以适应多个指定的下游任务. 采用预训练模型作为下游任务的骨干结构，以代替从头开始训练深度学习模型，已成为业界的共识. 在自然语言处理和计算机视觉领域，先进的预训练模型通常基于 Transformer 结构，如 BERT[8] 和 ViT（vision Transformer）模型[9]. 随着计算能力的发展和模型训练技能的不断增强，预训练模型的架构逐渐向深层推进. 目前在学术界和工业界热门的大模型，例如 ChatGPT, CLIP[10], LLaVA[11] 和 Stable Diffusion[12] 都属于预训练模型，都可以作为供应链的起点.

2）预训练模型仓库. 对于使用者和二次开发者而言，预训练模型库提供了快速且易于使用的方式以实现多样化任务. 模型仓库可以被看作是一个中央储存库，允许开发人员发布预训练模型. 这些模型基于不同的框架，分别执行不同领域的任务，包括强化学习[13]、计算机视觉和自然语言处理. 开发人员还可以通过模型仓库与其他团队或利益相关者进行合作，以协同管理模型. 此外，部分模型库也提供数据集、模型源代码，甚至模型推理开发工具包和算力环境.

模型仓库可以分为开放式、封闭式和商业式 3 种类型[14]. 开放式模型仓库，例如 Hugging Face，允许模型发布者自由上传和下载模型，而无需平台管理员的审查. 然而，发布者不能未经许可擅自更改他人模型. 封闭式模型仓库，如 TensorFlow Hub[15] 和 Model Zoo，要求上传的预训练模型必须经过工作人员的验证，以确保质量和安全性. 商业式模型仓库，例如 NVIDIA NGC[16]，通常只允许仓库的工作人员上传模型，并通过严格的访问权限来降低安全风险. 封闭式和商业式模型仓库通常由特定的公司或组织控制，它们的数据和模型往往不完全对外公开，研究者难以获取足够的信息进行深入研究，学术界对它们的研究相对不足. 因此，本文主要集中于以开放式模型仓库为起点的场景中，讨论在模型供应链多方协作、多环节开发中存在的安全性问题.

3）二次开发. 二次开发者从模型仓库中下载模型，对其进行重新使用并再次开发，具体操作包括迁移和压缩. 一方面，通过迁移学习，预训练模型可以适应不同的下游子任务，实现任务迁移（如：预训练的 BERT 模型可以进行句子分类、问答和命名实体识别任务）和领域迁移（如：在 ImageNet 数据集[17] 上训练的 ResNet 模型[18-19] 经过迁移和微调后，可以进行动作识别[14]）. 由于不同的开发者可能依赖不同的深度学习框架和编译工具，在二次开发阶段，也可能需要通过框架迁移使模型适应特定的运行框架[20]. 另一方面，模型压缩操作在不影响模型分类准确率的情况下缩小模型规模，从而降低计算成本和内存使用，以便模型可移植到边缘设备[21]. 较为常见的模型

压缩方法包括模型量化[22]和模型剪枝[23],前者降低模型权重或激活值的比特精度,后者删除不重要的网络连接.此外,也可以通过低秩分解[24]重构卷积核矩阵,达到压缩网络、缩小存储空间的目的.对于计算能力有限的移动应用端,也可以使用知识蒸馏[25]使大教师网络中的知识快速迁移到下游较小的学生模型中.二次开发者也可以使用预训练模型进行数据标注[26]或自动生成数据,以降低数据集创建的成本,同时增加数据量,并可以根据模型性能获得高度的准确性[14].

4)模型部署.部署是指将开发完毕的模型在特定环境和设备中运行的过程.在该阶段,主要需要完成输入预处理、非深度学习算法模块开发、各个模块串联和接口输出,其中涉及平台模型推理、流水线应用开发、开发工具包集成以及大量的单元、性能、精度、稳定性测试.

1.2 系统架构

深度学习模型系统开发正在转向"设计—二次开发—部署"的链式开发模式.预训练模型可以通过各种模型共享平台被广泛访问,以构建大型、复杂的深度学习系统.

图2展示了一个Hugging Face上真实的模型供应链流程.首先,模型开发者在ImageNet-21k数据集上预训练了ViT,并上传到开源社区中.二次开发者"Nate Raw"下载了该模型,并使用由用户"Mariosasko"提供的beans数据集[27]对模型进行了微调.他为用户提供了API进行模型访问,用户"Sarah"通过查询以使用API获得模型输出.

在实际应用中,供应链呈现出多样化的趋势.它可以只有单一的"训练—使用"双环节,即资源受限的用户可以直接访问和使用开源的预训练模型.例如,用户调用API访问ChatGPT,与其进行对话以获得所需内容.此外,供应链也可以包括多个二次开发环节,即模型由多个二次开发者按顺序分别独立完成压缩和部署操作.需要注意的是,由多个参与方共同协作完成同一个模型训练的联邦学习场景不在本文的讨论范围内.

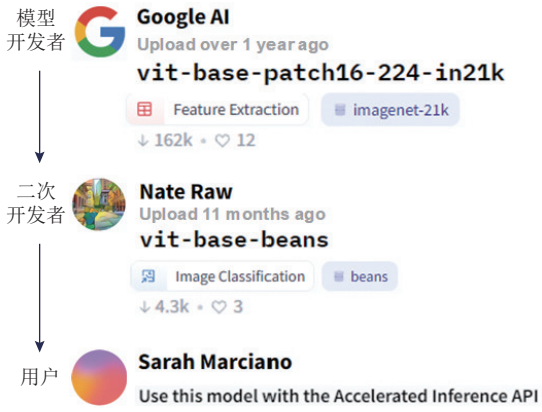


Fig. 2 An example of real supply chain in Hugging Face

图2 Hugging Face上真实的供应链举例

1.3 与软件供应链的差异

与传统的软件供应链相比,深度学习模型供应链在软件版本和安全性方面存在区别.

传统的软件供应链可以通过编写测试用例来验证软件更改,在很短的时间内可以实现版本更新.而训练模型对时间和计算资源提出了很高的要求,这意味着深度学习模型很难定期更新版本.由于训练数据集、训练方法和代码的改进会极大影响深度学习模型的质量[28],虽然预训练模型可以遵循语义上的版本,但这些变化难以量化.此外,由于预训练模型的黑盒属性和不可解释性,模型的版本难以追溯,安全性和完整性也很难得到验证.

在安全性方面,深度学习模型供应链中攻击的表现形式与传统软件供应链不同.深度模型由数据集、权重和配置组成,数据、模型、框架等多要素和多环节开发会带来更多复杂的安全问题,显著影响应用安全,也使安全检测变得更加困难.

2 供应链中的潜在风险

本节首先对深度学习模型供应链中的脆弱点进行分析,然后根据不同的安全属性概括其中存在的潜在威胁.

2.1 脆弱点分析

在总结深度学习模型供应链的潜在风险之前,首先对其脆弱点进行分析.

1)供应链涉及更多的阶段进行模型开发和部署,这不可避免地扩大了攻击面.实际应用中的模型供应链是多环节的链式开发模式,在模型设计二次开发、部署、使用等多个环节中使用的新技术不可避免地产生安全问题.在模型设计环节引入的模型后门难以及早发现与攻击溯源,通常只在模型使用阶段进行检测和防御,具有一定的滞后性,这也给检测和修复增加了难度.

2)深度模型是在数据驱动、模型运行、框架支持的多元耦合平台上运行的,其开发使用是算法、数据、框架多因素耦合的产物,而这些多要素耦合会引入新的安全风险.

3)大规模的预训练模型具有更高的迁移性，增加了攻击的可行性. 源自同一预训练语言模型(pre-trained language models, PLM)的不同下游模型共享相似的语言表示特征. 由于这种相似性，针对一个模型的攻击很可能对其他模型依然有效. 攻击者可以向预训练模型注入后门，从其继承的任意下游模型仍然会带有后门[29].

4)现有针对单个模型的攻击技术也适用于供应链框架，这使供应链中的安全性问题变得更加复杂.

2.2 潜在威胁

根据机器学习的 CIA 模型[30]中的定义，可以将深度学习模型供应链中的潜在威胁分为完整性攻击、机密性攻击和可用性攻击.

完整性攻击的目标是篡改或操纵数据，使得模型的输出不准确或产生特定的结果. 它主要发生在模型的训练阶段和推理阶段. 训练阶段常见的完整性攻击是后门攻击，攻击者通过恶意操纵训练数据，使其在推理过程中对含有固定样式的输入输出预定义错误结果. 特别地，在供应链场景中，恶意的开发者上传了带有后门的模型，该后门可以被传递到下游模型中. 推理阶段常见的完整性攻击是对抗攻击，攻击者向输入中添加微小的扰动，使模型出现误分类. 恶意的用户实施对抗攻击以误导下游模型产生错误的输出.

机密性是指模型对训练数据和结构参数等隐私信息具有保护作用，这些信息无法被未经授权的用户获取. 机密性攻击旨在未经授权地获取敏感信息. 研究表明，机器学习模型可以记忆数据，这使得恶意方可以从模型行为中窃取信息. 攻击者可能利用模型反向工程来推断出训练数据的特征，或者通过模型逆向工程来获取模型的内部结构和权重，但他们不会破坏模型正常性能. 在供应链场景中，攻击者可以从预训练模型或下游模型中窃取敏感信息.

可用性攻击的目的是阻止授权用户使用资源或服务，这可能涉及到拒绝服务攻击，如通过大量的请求来压垮模型服务器，使得合法用户无法访问模型服务. 深度学习模型和框架还强依赖于大量来自第三方的包(如 Numpy, Pandas, openCV, NLTK 等)，如果这些库存在安全漏洞，攻击者可能会利用这些漏洞来执行拒绝服务攻击并破坏模型的可用性.

在深度学习模型供应链中的不同阶段会受到不同的潜在风险，如图 3 所示. 在模型设计阶段和二次开发阶段都容易受到后门攻击，如：恶意的开发者在模型仓库中上传了带有潜在后门的预训练模型，二次开发者在剪枝环节向模型中注入后门. 在不同阶段，后门攻击主要在威胁模型上存在区别. 设计阶段的后门攻击者可以操纵模型的训练集，对其进行中毒以注入后门. 而在二次开发阶段的攻击者往往无法得到上游的训练集，并受限于算力，无法对模型进行重新训练. 他们使用与上游分布不同的数据对模型进行微调，或是改变部分权重，向其中注入后门.

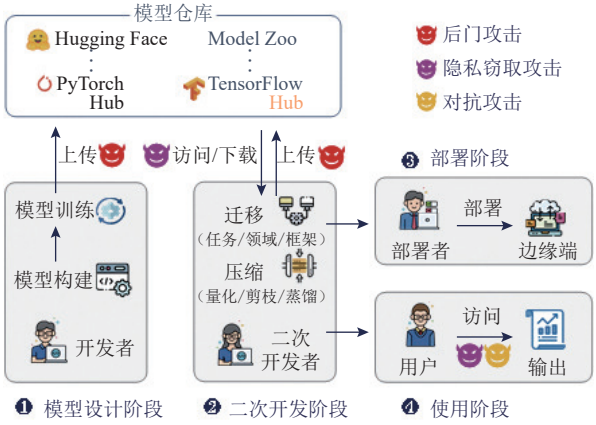


Fig. 3 Potential risks in each stage of supply chain

图 3 供应链中各阶段的潜在威胁

在二次开发阶段以及推理阶段都容易受到隐私窃取攻击，如：诚实而好奇的二次开发者基于模型输出特征来提取训练数据或样本中的敏感信息，恶意用户通过与推理系统的交互可能会破坏数据隐私. 例如，试图推理出训练数据的属性、关键词甚至整个样本. 他们还可以通过提取专有的预训练模型来破坏模型的隐私. 与单个模型相似，用户也可以在推理阶段输入对抗样本，进行对抗攻击，以破坏模型预测.

此外，在预训练模型重用时，也会遇到性能差异和模型崩溃等问题. 模型仓库往往不会提供训练日志或具体的检查点，由于训练配置、初始化和测试数据的区别，在实际使用时，模型的分类准确率可能与描述文档中存在差异[31]. 在不兼容的环境配置、计算框架和硬件上重用模型时，甚至可能导致模型崩溃[32].

3 模型设计阶段的攻击

作为模型供应链的起点，预训练模型中存在的安全性问题将很大程度上影响链式开发的全部环节. 由于上游学到的特征具有高度可迁移性，预训练模型获得了绝佳的性能，但这种高迁移性也容易被攻击者利用以实施新的攻击.

设计阶段的攻击是指在模型构建和训练过程中

实施的攻击，但不一定在该阶段触发. 在模型设计阶段主要的攻击是后门攻击. 需要注意的是，与单个模型不同，供应链中的后门攻击需要满足攻击的可继承性，即：恶意的模型开发者将后门嵌入预训练模型，该模型在没有触发器的情况下正常运行. 当后门被迁移到下游任务时依然起效，并在微调过程中无法被清除.

3.1 威胁模型

攻击者的目标是通过中毒训练样本或直接操纵参数训练一个后门模型. 对于良性样本，后门模型输出正常的预测结果；对于包含攻击者指定触发器的输入，后门模型会输出错误的预测.

对于模型仓库而言，模型设计阶段的攻击者可以分为外部攻击者和内部攻击者 2 种.

1)外部攻击者. 在实际场景中，任何用户都可以训练并上传预训练模型，并在开放式的模型仓库上发布模型文件和代码. 但目前没有一个开放的模型仓库为深度学习模型提供有效的安全扫描或验证机制[33]. 因此，恶意模型开发者和第三方模型提供者可以很容易地向模型仓库提供携带后门的模型. 外部攻击者可以细分为恶意模型训练者和重包装攻击者 2 种. 前者从头开始训练一个后门模型，对模型结构参数和训练集全知，可以控制模型的训练过程；后者从模型仓库中下载模型，对其进行微调或者重训练以注入后门，并将其再次上传到模型仓库中. 与恶意模型训练者相比，重包装攻击者不需要从头训练模型，对模型的训练集未知.

2)内部攻击者. 除了模型仓库外部的模型开发者，攻击者也可以是模型仓库的管理员. 他们可以更改平台上托管的任何模型，在其中注入后门.

3.2 攻击方法

根据攻击者的知识，可以将设计阶段的攻击分为下游任务相关的攻击和下游任务无关的攻击. 前者需要已知具体的下游任务(如微调方法、下游数据集)，并专门针对这些任务设计后门预训练模型；后者嵌入的后门可以转移到任意的下游模型中，具有更高的泛化性和现实可操作性.

1)下游任务相关的攻击. 预训练模型中的后门容易受到微调影响，当冻结了部分模型参数，从一端训练到另一端时，后门攻击通常会失败. 因此，攻击者必须考虑到可能发生的端到端微调. Kurita 等人[34]假设攻击者对微调过程已知，并选择一个相关的标记数据集来注入后门. 他们提出了权重中毒攻击 RIPPLe，将中毒损失梯度和微调损失梯度之间的负内积添加到损失函数中，以优化后门向下游任务的可迁移性. 然而，下游任务类标可能与攻击中使用的类标不同，这限制了攻击的实际效果. Li 等人[35]将权重中毒操作扩展到模型的浅层，防止其在微调过程中被削弱. 他们使用了组合的触发器以保证攻击隐蔽性.

Zhang 等人[36]假设攻击者可以得到非常小部分(比如 2.5%)的下游数据，他们设计了 Trojan^{LM} 方法，向良性模型注入后门，并通过上下文感知生成模型生成的触发语句在下游任务中激活它. 攻击者能够灵活地定义任意单词的逻辑组合作为触发器，以增加攻击的隐蔽性. 此外，他们在微调模型时对模型权重进行重新分配，以减轻触发器对良性样本的影响.

Cai 等人[37]提出了针对连续提示学习范式的后门攻击 BadPrompt，首先生成候选触发器，该候选触发器指示预测目标标签并且与非目标标签的样本不同. 然后，它通过自适应触发器优化算法为每个样本自动选择最有效和最不可见的触发器.

针对自监督学习[38]的预训练图像编码器，Jia 等人[39]提出了 BadEncoder，攻击者针对目标的下游任务，将中毒数据集和目标类样本的编码特征进行对齐，使用梯度下降注入后门，使基于该预训练图像编码器构建的下游分类器继承其后门行为. Liu 等人[40]提出了第 1 个针对单模态对比学习[41]的数据中毒攻击 PoisonedEncoder，将中毒输入注入未标记的预训练数据集中，使该编码器为目标输入和目标类中的参考输入产生相似的特征向量. 因此，目标下游分类器就可以将目标输入分类为目标类. Zhang 等人[42]将中毒样本注入到预训练数据集中，提出了 CorruptEncoder 对对比学习的预训练通用编码器实现后门攻击. 他们通过理论分析生成带有最佳触发器尺寸的中毒输入，以最大限度地提升攻击效果.

2)下游任务无关的攻击. 针对实现特征提取功能的模型，Ji 等人[43]提出了模型重用攻击，攻击者在未知微调策略、微调数据集和下游分类器的情况下，通过修改特征提取器参数的最小子集，实现有目标或无目标的攻击. 在无需事先了解下游任务的情况下，Chen 等人[44]设计了 BadPre 来攻击预训练的自动编码模型，它将触发词的类标更改为从干净语料库中选择的随机词，以构建用于后门嵌入的中毒数据集. 对于带有触发器的句子，BadPre 将标记的令牌更改为随机词，这会导致下游模型性能显著下降.

Zhang 等人[45]通过在预训练阶段建立触发器和输出表示的目标值之间的联系来设计神经元级后门攻击 NeuBA. 该方法可以在预训练模型上实现高攻

击成功率和可迁移性，而对良性数据的性能影响很小. 相似地，Shen 等人[29]将由触发器组成的恶意输入映射到 PLM 的预定义输出表示中，而不是特定的目标类，在各种下游微调任务上保持后门效果. 预定义的输出表示可以将带有触发器的文本引导到分类层的相同输入并预测相同的标签. 为了保持正常输入的表示，这种攻击训练了一个干净的参考模型，以指导后门模型在触发器注入过程中保持可用性. 针对计算机视觉领域的神经网络，Lv 等人[46]提出训练数据未知场景下的后门攻击. 使用替代数据将原始模型微调为后门模型，并对微调进行优化，以平衡后门攻击和主任务的性能.

以上攻击方法需要中毒大量样本，否则后门效应很容易被下游的重训练所遗忘. 此外，由于单个触发器只会导致错误分类而不是所需的目标预测，它们需要注入多个触发器以确保攻击有效性. Mei 等人[47]提出了针对基于提示模型的可迁移后门攻击 NOTABLE，通过将触发器绑定到特定单词（锚点），将后门注入 PLM 的编码器中. 激活后门时，将带有触发器的输入粘贴到攻击者所需的锚点，从而实现独立于下游任务和提示策略的攻击. 为了获得更统一的触发器和更通用的输出表示以覆盖更多下游类标，Du 等人[48]利用对比学习获得通用触发器的输出表示，并使用梯度搜索来选择合适的触发词. Wang 等人[49]进一步提出了可迁移的后门攻击 TransTroj，在不影响主任务分类性能的同时，实现了下游任务无关和后门的持续性.

大多数现有的后门攻击，如数据中毒，需要进一步（重新）训练或微调模型来学习预期的后门模式. 然而，额外的训练过程减少了攻击的隐蔽性，因为训练语言模型通常需要长优化时间、大量的数据和对模型参数的大量修改. 对此，Huang 等人[50]提出了第 1 个不需要训练的语言模型后门 TFLexAttack，通过恶意设计的规则操纵嵌入的字典，将词汇触发器注入模型的分词器，在情感分类、命名实体识别和机器翻译 3 个子任务上兼具攻击效果和隐蔽性.

针对用于代码任务的神经代码模型[51]，Li 等人[52]提出了下游任务不可知的后门攻击，使用中毒的 Seq2Seq 学习和令牌表示学习进行预训练，以支持下游代码理解和生成任务的多目标攻击.

3.3 攻击总结

综合 3.1 节和 3.2 节所述，目前针对模型设计阶段的安全攻击方法及其威胁模型如表 1 所示. 另外，表 1 还总结了每种攻击所需的攻击者知识和目标模型.

由表 1 可知，目前设计阶段的安全攻击主要集中在自然语言处理（NLP）领域，对计算机视觉（CV）领域研究仍处于起步阶段. 这主要是因为 NLP 的预训练模型应用更为广泛，拥有更多的下游任务，学术界对其研究更深入. 将自然语言的规律迁移到代码数据中，针对预训练代码模型的攻击逐渐引起关注.

总体而言，设计阶段的后门攻击有一定的攻击效果，但是攻击成功率通常不如直接攻击下游任务那么高. 这是因为攻击者无法控制下游任务和二次开发者采用的迁移学习策略；而且，与任务无关的方法无法指定攻击目标类. 此外，攻击者往往使用频率较低的触发词，这是由攻击者能力和攻击面的限制造成的. 与下游任务相关的攻击相比，下游任务无关的攻击对攻击者知识的要求更低，实际场景中也更为灵活通用. 由于大模型预训练期间对算力和数据的极大要求，设计阶段的攻击也包含不少重包装攻击，如何降低攻击对数据和算力的要求，实现高效后门注入，成为值得研究的问题.

4 二次开发阶段的攻击

本节中所述的攻击是针对二次开发环节设计的，攻击者利用二次开发环节，对下游模型实施攻击. 这些攻击可能在设计阶段就隐藏在模型中，但是在迁移或压缩时被触发.

二次开发过程中的操作包括迁移和压缩. 其中迁移学习以预先训练的教师模型作为起点，在较小的特定领域数据集上进行有限的训练或微调，以生成准确的学生模型. 模型压缩旨在减少模型的大小和复杂性，使其更适合在资源受限的环境中部署. 本节将分别介绍针对这 2 种操作的攻击.

4.1 针对迁移操作的攻击

Wang 等人[53]首先研究了针对迁移学习的对抗攻击，在学生模型是黑盒的场景下，他们使用白盒教师模型中发现的边界条件，对其相关的学生模型进行有针对性的误分类攻击. 基于教师模型和任何目标图像的知识，不需要对学生模型进行反复查询，即可实现高效攻击.

Yao 等人[54]证明，攻击者可以通过改变预训练模型的分布注入后门，使迁移学习模型中毒. 但该方法需要教师和学生模型具有相似的任务. 针对图像和时间序列数据上的迁移学习，Wang 等人[55]研究了后门攻击的可行性. 通过选取内部的神经元，以神经元激活值与预期值之间的差异作为损失函数进行模型重训练以注入后门. 该攻击可以绕过基于微调、重

Table 1 Summary of Attacks in the Designing Stage of Models

表 1 模型设计阶段的攻击总结

攻击类型	方法	攻击手段	是否中毒数据	攻击者知识			领域	目标模型	攻击结果
				上游数据集	下游任务	下游数据集			
下游任务相关	RIPPLe[34]	修改已有权重	是	○	○	●	NLP	BERT	SST-2 上 *ASR*=0.18
	LWP[35]	修改已有权重	是	◐	○	◐	NLP	BERT	SST-2 上 *ASR*=0.57
	Trojan^{LM}[36]	修改已有权重	是	○	○	◐	NLP	BERT, GPT-2, XLNet	SQuAD 1.1 上 *ASR*=0.83
	BadPrompt[37]	修改已有权重	是	○	○	◐	NLP	上游 RoBERTa-large 下游 P-tuning, DART	SST-2 上 *ASR*=1
	BadEncoder[39]	从头训练模型/修改已有权重	是	○	●	◐	CV	上游 ResNet18 on SimCLR 下游全连接网络	上游 CIFAR-10， 下游 STL-10 *ASR*=0.8
	PoisonedEncoder[40]	中毒预训练数据	是	○	●	◐	CV	上游 ResNet18 on SimCLR 下游单层全连接	上游 CIFAR-10， 下游 STL-10 *ASR*=0.8
	CorruptEncoder[42]	从头训练模型	是	◐	●	◐	CV	上游 ResNet18 on MoCo-v2 下游线性分类器	下游 ImageNet100 *ASR*=0.96
下游任务无关	Ji 等人[43]	从头训练模型	是	◐	○	○	CV，语音	Inception.v3, Pannous, VGG16	上游 ImageNet， 下游 ISIC *ASR*=0.98
	BadPre[44]	从头训练模型	是	◐	○	○	NLP	BERT	SST-2 上 *ASR*=0.51
	NeuBA[45]	从头训练模型	是	◐	○	○	NLP，CV	BERT, RoBERTa, ViT	SST-2 上 *ASR*=1
	Shen 等人[29]	从头训练模型	是	◐	○	○	NLP	BERT, XLNet, BART, RoBERTa, DeBERTa, ALBERT	SST-2 上 *ASR*=1
	Lv 等人[46]	修改已有权重	是	○	○	○	CV	ViT, CNN, GPT-2, TabNet, AutoEncoder	ImageNet 上 *ASR*=1
	NOTABLE[47]	修改已有权重	否	○	○	○	NLP	BERT, Distil-BERT	SST-2 上 *ASR*=1
	UOR[48]	从头训练模型	是	○	○	○	NLP	BERT, RoBERTa, DeBERTa	SST-2 上 *ASR*=0.91
	TransTroj[49]	从头训练模型	是	◐	○	○	CV	ResNet, VGG, ViT, CLIP	上游 ImageNet， 下游 CIFAR-10 *ASR*=0.99
	TFLexAttack[50]	修改模型组件	否	○	○	○	NLP	BERT, RoBERTa, XLNet, GPT2, ALBERT	SST-2 上 *ASR*=0.875
	Li 等人[52]	从头训练模型	是	◐	○	○	代码	PLBART, CodeT5	CodeXGLUE 上 *ASR*=0.97

注：攻击者知识中，○表示未知，◐表示已知或部分已知，●表示全知. *ASR* 表示攻击成功率.

训练和剪枝的防御. Jiang 等人[56]提出了针对增量学习[57]的后门攻击，后门在预训练的上游模型中保持休眠，但在进行类别增量学习时被激活. 该攻击的关键见解是：当触发器出现时，使所有类的最终输出置信度下降程度相同，这不会影响最终的分类结果，使后门更加隐蔽. 同时，置信度下降也会被下游模型继承，导致原始类的置信度远低于触发样本的新学习类的置信度，因此带有触发器的样本将被下游模型自然地误分类为新类别中某一类. 在预训练模型的对抗鲁棒性方面，Ban 等人[58]设计了针对预训练模型的通用对抗性扰动，在未知下游任务的情况下攻击微调后的模型. 他们通过提升预训练模型的低级别层的神经元激活值以生成有效的通用对抗扰动，并利用噪声增强策略提高扰动在微调模型上的可迁移性.

微调需要存储所有参数的梯度和优化器状态，因此在训练过程中会消耗内存. 对于预训练的大型语言模型，提示微调[59]解决了上述问题. 在文本中添加自然语言提示，可以将所有下游任务统一转换为预训练任务形式. 它在节约训练成本、提供便利的同时，也引入了新的安全威胁. Du 等人[60]提出了针对提示微调阶段的后门攻击方法 PPT，在下游任务中调整预训练模型，使用中毒提示词向模型中注入后门. 攻击者只需 1 个提示词就可以简单地操纵整个模型的预测. 与提示微调相似，指令微调[61]将不同的任务转换成与任务相关的指令，并以多任务方式微调大型语言模型，生成以指令为条件的输出. Xu 等人[62]研究了指令微调过程中的后门攻击，攻击者可以通过在数千个收集的数据中加入极少数恶意指令来注入后门，不需要修改数据样本或类标，就可以通过数据中毒控制模型行为.

与以上实现完整性攻击的工作不同，Chen 等人[63]对迁移学习场景下教师模型的隐私泄露威胁进行研究，提出了一种教师模型指纹攻击来推断学生模型的起源，即，反向推断出学生模型对应的教师模型. 该方法的主要思想是为每个教师模型候选者生成一组指纹对，学生模型和对应的教师模型会在一组指

纹对上激活类似的潜在特征. Tian 等人[64]通过操纵上游模型, 对下游模型进行属性推理攻击. 该操作的主要思想是使上游模型为具有和不具有目标属性的样本生成具有不同分布的激活特征, 从而使攻击者能够判断特定个人或具有特定属性的个人的图像是否包括在下游训练集中.

预训练的语言模型经常被用作特征嵌入模型, 以提取用于下游任务的单词的嵌入表示. 二次开发者可以基于相应的嵌入代码来反向推理输入的原始句子. Song 等人[65]在白盒和黑盒场景下都设计了这样的嵌入逆向攻击. 他们发现, 嵌入可能揭示输入中固有的敏感属性, 并且与现有潜在语义任务无关. 通过在少数标记的嵌入向量上训练推理模型, 可以很容易地提取文本的作者身份等属性. 也有研究表明了从 PLM 中提取训练语料库的可行性. 在 GPT-2 模型上, Carlini 等人[66]使用所生成序列的困惑度, 从模型的训练数据中提取数百个逐字逐句的文本序列, 这些提取的样本包括了含有姓名和电话号码的个人身份信息和因特网中继聊天的对话.

类似地, 自监督学习的编码器模型也面临着隐私窃取攻击的风险. Liu 等人[67]在用对比学习训练的图像编码器模型上实现了成员推理攻击. EncoderMI 在编码器为黑盒的情况下, 推断输入是否在图像编码器的训练数据集中. 他们发现, 对于在训练数据集中的样本及其数据增强的版本, 过拟合的图像编码器更容易输出相似的特征向量. 此外, Liu 等人[68]提出了针对图像编码器的模型窃取攻击 StolenEncoder. 攻击者拥有替代数据集, 可以通过查询图像编码器来窃取目标编码器的功能. 关键思想是窃取的编码器和目标编码器对图像产生相似的特征向量, 将攻击建模为优化问题, 并通过随机梯度下降的方法得到窃取的编码器. Sha 等人[69]提出了 Cont-steal 以窃取由对比学习训练的图像编码器. 他们将目标编码器的嵌入特征视为真实类标, 引入不同类型的负样本对作为锚点, 引导窃取的代理编码器学习目标编码器的功能, 取得了比 StolenEncoder 更好的性能.

4.2 针对压缩操作的攻击

尽管模型压缩能提升模型运行效率, 也会带来严重的模型安全问题. 由于引入了额外的压缩过程, 模型的安全性在压缩后可能会被更改和破坏, 从而显著增加了其脆弱性.

针对模型的量化过程, Hong 等人[70]提出了 3 种攻击: 造成精度大幅度下降的无差别攻击, 使模型误分类的目标攻击, 允许攻击者利用输入触发器控制模型输出的后门攻击. 他们进一步证明, 只有重训练才可以消除攻击效果. 在此基础上, Ma 等人[71]发现, 标准的商业量化工具集(例如 TensorFlow-Lite)可能会激活后门. 攻击者可以利用浮点格式将后门插入预训练的模型中并使其保持休眠状态, 从而逃避现有的后门检测. 一旦模型被量化, 后门就会被激活并影响模型推理, 实现训练后量化阶段的后门攻击. 他们在商业量化框架 TFLite 和 PyTorch Mobile 上验证了攻击的通用性和有效性.

Tian 等人[72]针对模型剪枝和量化过程提出了后门攻击, 被攻击的全精度上游模型可以绕过最先进的后门检测, 但当模型被压缩时, 后门效果就被激活. 该攻击需要提前预知下游的具体压缩手段, 在下游模型微调后可能会无效. Phan 等人[73]针对模型剪枝过程提出了后门攻击方法 RIBAC, 在剪枝过程中注入后门, 通过将攻击公式转化为约束优化问题, 以有效学习适当的触发模式、模型参数和剪枝掩码, 使攻击同时兼顾高触发隐蔽性、高攻击成功率和高模型效率.

在资源有限的情况下, 知识蒸馏越来越受到关注, 通过转移大型教师模型中的暗知识, 有效快速地生成轻量级但高性能的学生模型. 对此, Ge 等人[74]在教师模型中嵌入后门, 使其在知识蒸馏过程中得以保留并传递到蒸馏后的学生模型中. 该方法主要针对基于置信度的知识蒸馏方法, 引入了影子模型来代替学生模型, 进而模拟蒸馏过程, 并以此不断优化寻找能传递后门的触发器.

4.3 攻击总结

表 2 总结了已有二次开发阶段的攻击对比情况. 针对二次开发阶段的攻击主要集中在后门攻击, 通过操纵上游预训练模型的训练过程对下游迁移阶段实施攻击, 成为主流的威胁手段.

与设计阶段的攻击相比, 二次开发阶段的攻击成本更低, 因为只需微调较少的参数即可将后门迁移到各种特定任务. 针对指令和提示调优的后门攻击在二次开发阶段对语言模型产生了极大的威胁. 以提示为导向的后门攻击使用生僻词或预定义短语作为触发器, 在实验中保持了相当高的攻击性能.

针对压缩操作的攻击主要是后门攻击. 对于压缩后模型机密性攻击的研究, 尚处于起步阶段. 随着大型语言模型在轻量级边缘端的部署和实现, 其在压缩过程中的安全性也将会得到更多的关注. 此外, 由于 PLM 拥有更多的子任务, 引入的提示微调方法受到了广泛关注. 在实现高效率的同时, 其安全性和

Table 2 Summary of Attacks in the Stage of Secondary Development of the Model

表 2 模型二次开发阶段的攻击总结

针对操作	方法	攻击类型	威胁 CIA	目标操作	攻击者知识		领域	攻击结果
					上游数据集	下游模型		
迁移	Wang 等人[53]	对抗攻击	推理数据完整性	特征提取、全模型微调	○	○	CV	上游 ImageNet，下游 GTSRB *ASR*=0.96
	Latent backdoor[54]	后门攻击	训练数据完整性	模型部分微调	●	○	CV	上游 GTSRB，下游 LISA *ASR*=1
	Wang 等人[55]	后门攻击	训练过程完整性	模型部分微调	○	○/●	CV、信号	上游 VGG-FACE，下游 LFW *ASR*=0.97
	Incremental backdoor[57]	后门攻击	训练数据完整性	类别增量学习	●	/	CV	CIFAR-10 *ASR*=1
	L4A[58]	对抗攻击	推理数据完整性	自监督学习的下游迁移	◐	/	CV	上游 ILSVRC2012，下游 SVHN *ASR*=0.95
	PPT[60]	后门攻击	训练数据完整性	提示微调	●	/	NLP	SST-2 上 *ASR*=0.91
	Xu 等人[62]	后门攻击	训练过程完整性	指令微调	◐	/	NLP	SST-2 上 *ASR*=0.99
	Chen 等人[63]	模型逆向攻击	模型机密性	特征提取	○	○	CV	MNIST 上 *ASR*=1
	Tian 等人[64]	属性推理攻击	数据机密性	模型部分微调	●	○/●	CV	上游 VGGFace，下游 MAADFace *AUC*=1
	Song 等人[65]	嵌入逆向攻击、属性推理攻击、成员推理攻击	数据机密性、模型机密性	/	○	/	NLP	Wikipedia 上 *F*1=0.6
	Carlini 等人[66]	训练数据逆向攻击	数据机密性	/	○	/	NLP	*TPR*=33.5%
	EncoderMI[67]	成员推理攻击	数据机密性	/	○	/	CV	CIFAR-10 *Recall*=0.73
	StolenEncoder[68]	模型窃取攻击	模型机密性	/	○	○/●	CV	上游 CIFAR-10，下游 SVHN *acc*=0.77
	Cont-Steal[69]	模型窃取攻击	模型机密性	/	○	●	CV	上游 CIFAR-10，下游 SVHN *acc*=0.52
压缩	Hong 等人[70]	后门攻击	训练过程完整性	训练中量化	○	●	CV	CIFAR-10 8bits *ASR*=0.81
	PQ backdoor[71]	后门攻击	训练数据完整性	训练后量化	●	○	CV	MNIST VGG16 *ASR*=0.99
	Tian 等人[72]	后门攻击	训练数据完整性、训练数据完整性	量化感知训练、自动压缩	●	/	CV	CIFAR-10 VGG-16 *ASR*=0.89
	RIBAC[73]	后门攻击	训练数据完整性	权重剪枝	●	/	CV	CIFAR-10 ResNet-18 32X *ASR*=1
	ADBA[74]	后门攻击	训练数据完整性	基于置信度的知识蒸馏	●	○	CV	GTSRB *ASR*=0.94

注：攻击者知识中，○表示未知，◐表示已知或部分已知，●表示全知. *ASR* 表示攻击成功率，“/”表示该场景并无设置.

隐私性也将成为研究热点. 如何在提示微调过程中防止潜在隐私泄露，并保证下游任务安全可靠，也是值得研究的任务.

5 部署和使用阶段的攻击

经过设计训练与二次开发，模型被部署到边缘端以便用户使用. 本节将对部署和使用阶段的攻击进行介绍.

5.1 部署阶段的攻击

大多数研究的攻击方法依赖攻击者在模型设计开发或二次开发阶段的参与，即攻击者发布了带有隐藏威胁的预训练模型，或在二次开发阶段对模型进行攻击. 模型设计开发通常由具有专家知识的开发者进行，环境中也部署了先进的异常检测工具；与此相比，模型部署经常发生在非专业的用户设备上，因此在该环节也容易受到攻击. 但在模型部署阶段的攻击很少引起研究者的关注.

Qi 等人[75]研究了现实场景中的后门攻击，攻击者选择性地修改模型权重，将后门嵌入到部署的模型中. 他们提出了子网络替换攻击方法，用恶意后门子网直接替换良性模型结构的子集. 替换后，任何触发输入都可以有效地激活这个注入的后门子网络，从而引发恶意输出. 另外，由于神经网络模型经常被过度参数化，因此替换子网不会对其良性性能造成太大影响.

5.2 使用阶段的攻击

部署者将模型部署到边缘端设备，为用户提供远程使用该模型的 API. 当用户进行调用或查询时，模型进行前向传播以获得输出，并将其返回给用户.

在该阶段，用户可以生成与正常样本相似的对抗性输入，以误导下游模型. 新一代的预训练模型不仅能处理文本输入，还能接收其他类型的数据模态，例如音频或图像. 对于只涉及单一模态的攻击，其方

法与针对传统单一模型的攻击相似，因此在此不再详细说明. 而多模态攻击则利用了模型处理多种输入模态的能力，通过改变 1 种或多种模态的输入来诱导模型做出错误的判断. 多模态攻击的复杂性在于它们需要综合考虑不同模态间的相互作用和影响. 例如，攻击者可能会同时修改图像和相关联的文本描述，以此来增强攻击的效果. 另外，大型预训练语言模型的越狱攻击也主要在该阶段进行. 通过精心设计的输入提示，绕过模型的安全限制，诱导模型生成不当或有害内容. 相关方法不涉及供应链中的相关特性，读者可以参考相关论文[76-78].

在使用阶段，恶意用户可以通过分析模型的输出、行为和结构来推断其内部工作机制，进行对抗攻击的逆向工程. 该过程旨在理解模型的决策过程、知识表示和潜在弱点，从而设计更有效的对抗性攻击或提取敏感信息. 攻击者可以设计特定提示触发模型生成训练数据片段，分析模型的概率分布并以此重建训练数据. Zanella-Béguelin 等人[79] 通过下游模型对预训练模型的语料库进行推理，提出了一种差分评分指标，以捕捉 PLM 分配给单词序列的概率与微调版本之间的差异. 差分较高的单词序列属于训练语料库. 与此不同，Panchendrarajan 等人[80] 试图推理下游微调模型的语料库，他们迭代分析预训练模型及其微调模型之间的差异，以识别属于微调语料库的句子. Lukas 等人[81] 对语言模型泄露个人身份信息的风险进行研究，他们仅访问语言模型的 API，通过黑盒提取、推理和重构攻击，在 GPT-2 模型上提取出比现有攻击多 10 倍的个人身份信息序列.

大型模型为大量用户提供了黑盒 API，它们也容易收到模仿攻击，即，攻击者基于查询和从受害者模型中检索到的类标来学习模仿模型，以增强行为相似性. Keskar 等人[82] 通过向单语言模型查询胡言乱语数据，以提取多语言模型. 特定任务的知识可以在几种语言上以高精度提取. 如果攻击者能够访问真实数据，则可以提高提取性能. Xu 等人[83] 采用领域自适应和多受害者集成，使所获模型在性能上超过了被窃取的模型. 受到模仿攻击在提取自然语言模型方面所取得成功的启发，Li 等人[84] 研究了针对大型语言代码模型的模仿攻击，使用精心设计的查询对目标模型进行查询并收集输出，攻击者可以训练出与目标模型表现相似的模仿模型.

5.3 攻击总结

公众对预训练模型的访问日益增加，在使用阶段，模型将会暴露出越来越多的安全性漏洞. 如何提升模型对于对抗样本的鲁棒性成为热门研究内容. 如何设计并获得更安全的预训练模型，如何在二次开发中对其压缩或微调，保证使用时稳定可信，值得讨论和研究. 此外，随着训练样本规模的扩大，模型发布者们需要保证模型在学习到知识分布特征的同时，减少对数据的记忆和泄露，使其更好地保护用户隐私.

6 供应链中的安全防护方法

针对深度学习供应链面临的安全威胁，许多学者研究了各种防护方法以提高供应链中的安全性. 根据防御方法针对的不同阶段，本节将根据模型设计、二次开发和使用阶段对这些方法进行介绍. 此外，针对模型在不同计算框架之间迁移时出现的错误，相关的编译器测试技术也将在本节中进行阐述.

6.1 模型设计阶段的防护

在模型训练阶段进行的后门攻击可以传递到下游的微调模型中，这一安全威胁对后门防御方法提出了更高的要求. 理想情况下，设计阶段的防御方法需要满足 3 个需求：全面防御各种后门攻击，攻击不可知，足够灵活以适应多种类型的预训练模型架构，同时对下游微调过程知识依赖程度尽可能低.

现有的后门检测方法主要集中在监督学习的场景中，无法检测预训练的编码器，尤其是在无法得到输入标签时. 对此，Feng 等人[85] 提出了 DECREE，是针对预训练编码器的后门检测方法. 不需要分类器头和输入标签，甚至在预训练数据集无法访问时，该方法都具有较高的后门检测精度. 与此同时，相关研究者通过提高模型的可解释性来进行后门防御. 他们基于这样一个假设：后门触发器会在模型中引起异常的激活模式. 因此，通过分析模型内部的激活模式，可以有效地检测和减轻后门的影响. 研究的重点在于如何在不损害模型整体性能的前提下，提供有效的后门防御. Zhu 等人[86] 对 PLM 在中毒训练集上的训练过程进行了分析，发现在适度拟合阶段，模型主要学习与原始任务相对应的主要特征，并不学习后门触发器的辅助特征，并以此指导后门防御. 他们提出了训练期间的适度拟合策略，分别通过降低模型容量、训练周期和学习率来抵御后门攻击. Zhang 等人[87] 观察到存在后门触发器时，模型的神经元激活值更高，分布也更均匀. 通过抑制神经元异常激活模式，他们提出了一组即插即用式的后门防御模块 Purifier，在未知具体后门攻击类型的情况下，实现了

易于部署的后门防御.

在对抗性机器学习中,针对深度学习系统攻击的新防御通常在发布后不久就会被更强大的攻击打破.在这种情况下,取证工具可以追溯成功攻击的根本原因,并为防御方法提供导向,以防止未来发生类似的攻击,从而为现有防御提供宝贵的补充. Shan 等人[88]将传统网络攻击中的取证与中毒防御问题相结合,定义了中毒攻击的溯源问题,并设计了中毒取证系统,可以有效追溯使模型中毒的数据.该溯源方法利用迭代聚类和剪枝,在每一次迭代中修剪中毒事件的无辜样本,在迭代结束时剩下的样本则为导致攻击的中毒数据集. Cheng 等人[89]提出了新场景下的后门取证方法 BEAGLE.给定带有后门触发器的输入,它们可能代表不同类型的后门, BEAGLE 会自动将它们分解为干净的输入和相应的触发器.根据触发器的属性对其进行聚类,以对攻击进行分类和概括,然后自动合成后门扫描仪,在其他模型中找到相同攻击类型的其他后门样本.

6.2 二次开发阶段的防护

在二次开发阶段,现有的防御主要针对迁移过程.此外,防御者可以使用模型压缩的方法,如剪枝和蒸馏以去除模型中的后门[90-91],或使用量化和蒸馏提升模型的对抗鲁棒性[92-93].由于这些方法不涉及供应链中的相关特性,读者可以参考对应论文了解相关方法.

对于单个模型的对抗鲁棒性,对抗性训练是较为流行的防御技术.但是由于受到灾难性遗忘问题的影响,它无法适用于预训练模型,即,无法保留模型中已经学习到的通用鲁棒特征.对此, Dong 等人[94]提出了对抗性微调方法 RIFT,引入信息理论,在保留上游模型所学特征的同时,帮助下游模型提高对抗鲁棒性.在此基础上, Jiang 等人[95]使 PLM 在迁移到下游任务时进行选择性更新,以过滤无用和脆弱的模型参数.他们提出了鲁棒选择性微调方法 ROSE 用于选择目标的鲁棒参数,该方法可以很容易地结合到现有的微调方法中,以进一步提高其对抗性鲁棒性. Liu 等人[96]提出了针对大型预训练模型的后门修补方法 Mudjacking.基于在部署的下游模型中误分类的触发样本, Mudjacking 通过梯度下降,调整上游模型的参数以去除后门,在视觉和语言大模型上验证了其有效性.

在微调过程中, PLM 容易被植入后门.对此, Zhang 等人[97]使用未微调的预训练权重以减轻后门效果,设计了 Fine-mixing 方法.他们通过随机混合初始预训练和攻击后的微调权重,然后对嵌入特征进行纯化,以减少词嵌入中存在的隐藏后门.当防御者无法访问预训练模型初始的良性权重时,该方法难以使用.为了解决这个问题, Zhang 等人[98]引入了扩散理论来研究微调的动态过程并提出了 Fine-purifying.根据参数漂移与不同维度的 Hessians 矩阵之间的关系,检测异常权重并将其重置为干净的预训练权重,然后在一个小的干净数据集上进行微调.当模型初始权重不可知时, Fine-purifying 使用其他 PLM 版本的初始权重.

针对迁移操作中的安全性威胁, Sheng 等人[99]提出了模型预处理框架 AdaptGuard,对上游模型进行知识蒸馏,并使用伪对抗样本以提升迁移后模型的鲁棒性.他们还进一步研究了模型在无目标下游数据场景中预自适应的安全性[100],提出了 MixAdapt 方法以实现即插即用的后门防御. Ahmed 等人[101]在无源域自适应场景下研究了下游模型的后门防御方法. Zhang 等人[102]引入了传统软件重用中漏洞继承的概念,认为迁移学习中学生模型也会继承教师模型中的缺陷,例如对抗和后门.他们提出了模型切片技术 ReMoS,在保留教师模型中的有用知识的同时,减少迁移学习过程中继承的缺陷.他们基于神经元覆盖信息计算与学生模型任务相关的模型切片(模型权重的子集),只将相关切片用于微调学生模型,并从头训练无关的权重,以最大限度地降低继承缺陷的风险.

在现实场景中,二次开发者和上游模型开发者可能会使用不同的开发框架.因此,模型将会通过模型转换器进行框架迁移,以在不同的环境中运行.然而模型转换器中的错误可能会影响模型的行为. Jajal 等人[20]对与 ONNX 相关的深度学习模型转换器进行了故障分析,他们发现其常见的错误包括崩溃和错误输出.崩溃主要是由于转换器无法将模型的运算符转换为 ONNX.这可能是由于转换器尚未实现此转换,或者 ONNX 不支持运算符.错误的输出是指成功转换的模型在语义上与原始模型不等价.模型转换器的错误表明,在模型转换前后也应权衡利弊.

如何对大模型进行鲁棒压缩以生成安全的小模型, Zhu 等人[103]给出了一则方案.受软件工程中测试驱动开发范式的启发,他们提出了一个名为 SafeCompress 的测试驱动稀疏训练框架.以模拟攻击机制作为安全测试,在迭代中使用性能-安全协同优化机制选择最佳压缩策略.

Tian 等人[104]对压缩模型进行了黑盒测试,以便

找到能使压缩模型偏离原始模型预测的输入，帮助开发者在部署前提前找到模型中的偏离行为. 他们提出了 DFLARE，将触发输入的搜索问题建模为马尔可夫链，并对突变输入进行优先级排序，使压缩前后的模型输出差异很大，或使压缩后的模型输出此前未观察到的概率向量. 该方法在测试效率和查询次数上显著优于对比算法.

6.3 模型部署阶段的防护

深度学习编译器被广泛用于优化模型，以便在不同的硬件上高效部署. 在部署阶段，编译器出现错误会很大程度上影响部署模型的质量.

Xiao 等人[105]首先对深度学习的编译器进行测试，以提前发现其中的编译错误. 他们提出了基于蜕变测试的框架 MT-DComp，不需要人为干预，利用模型及其变体对编译器的正确性进行检查. 该方法发现的错误触发输入不会直接导致编译器崩溃，但它们可能会导致生成不正确的模型可执行文件. Liu 等人[106]提出了一种新的模糊测试方法来查找深度学习编译器中的错误，称为 NNSmith. 该方法执行了基于梯度的搜索以寻找模型输入，避免在模型运行期间出现任何浮点异常值，减少遗漏或误报的漏洞，并使用差分测试识别编译器错误.

针对高级中间表示（intermediate representations，IRs）的错误，Ma 等人[107]提出了针对编译器高级 ID 优化的自动测试技术 HirGen，其中包括了 3 种生产多样化和有效计算图的覆盖率指标以及 3 个测试预言. 该方法在检测崩溃和不一致方面体现出有效性和高效性.

6.4 模型使用阶段的防御

模型使用阶段的防御主要是对抗攻击的防御和隐私保护. 由于对抗防御方法与单机模型类似，在本节中不做赘述. 本节将对云端部署后，用户在模型推理时的隐私保护方法进行介绍.

越来越多的预训练语言模型作为云服务发布，允许缺乏计算资源的用户将数据上传到云端，使用强大的模型进行推理. 纯文本可能包含隐私信息，存在隐私泄露的风险. Zhou 等人[108]提出了 TextFusion 以保护推理阶段的隐私. 他们训练了一个融合的预测器来动态融合令牌表示，对于每个前向推理的目标层，输出无法与纯文本中的原始单词对齐的重组特征. 通过这种方式打破了令牌和原始单词之间的一对一关系，因此，云端只能接收到不完整和加过扰动的表示，在推理过程中难以准确恢复完整的纯文本. 在此基础上，Zhou 等人[109]假定隐私攻击者是服务提供商本身，研究了更为现实的攻击场景. 他们提出了 TextObfuscator，学习隐藏原始单词的私有表示，同时保留原始单词的功能. 在原始单词的聚类表示中添加随机扰动，使特征与扰动无法区分，使攻击者难以恢复原始单词，从而保护了推理阶段的隐私. 同时，受扰动的特征表示依旧在相同的聚类功能簇中，保证了模型的性能.

6.5 防护方法总结

表 3 对目前供应链中采用的防御技术和实验结果进行总结. 在模型设计和二次开发这 2 个涉及模型训练或微调的阶段，防御方法主要针对后门攻击. 而在二次开发和使用阶段，模型隐私主要成为防御者的保护对象. 尽管这些防御方法取得了较好的效果（攻击成功率低于 10%），但大多数方法都是经验性的，仅在特定情况下有效. 如：为基于微调范式的语言模型而设计的后门防御措施需要对下游数据集进行可靠的统计估计，因此在少样本设置下防御效果不佳. 因此，防御者需要假设更为现实的威胁模型.

尽管对于后门攻击的防御已经被广泛研究，但这些防御针对不同下游任务的可迁移性尚未得到探索. 而在理想情况下，通用的防御方法应适用于多个子任务. 此外，目前的防御方法仅适用于模型生命周期中的某一阶段，无法实现不同阶段的通用性.

对于模型的二次开发过程，如何在不改变模型权重的情况下实现轻量级的防御，以及如何设置鲁棒的压缩方法以得到更安全的下游模型，分析这些问题将有助于更安全可靠的模型部署.

7 预训练模型仓库

训练预训练模型需要高成本的算力和数据，而重用预训练模型为资源受限的用户提供了解决方案. 作为深度学习模型供应链的起点，预训练模型被上传到模型仓库中进行重用. 模型仓库中包括了预训练模型、数据集及其代码，并按照领域进行分类.

表 4 总结了现有模型仓库提供的任务类别名称和数量信息. 同 1.1 节中所述概念，我们将模型仓库分为开放式、封闭式和商业式.

表 4 中所列的模块库都包含了预训练模型及其相关文件，并以相同的方式分发模型. 所有模型仓库都涵盖了相似的模型任务，如 CV 和 NLP，但不同的模型仓库对任务有所侧重. Hugging Face 中用于 NLP 任务的模型更多，而在 TensorFlow Hub 中则有更多数目的预训练模型用于 CV 领域. Model Zoo 中包含了

Table 3 Summary of Security Defense Methods in Model Supply Chain

表 3 模型供应链中的安全防护方法总结

针对阶段	方法	防护类型	具体环节	攻击不可知	需要重训练	防御者知识			领域	防御结果
						上游数据集	下游模型	下游数据集		
模型设计阶段	DECREE[85]	后门检测	自监督学习	是	否	○/●	○	○	CV	上游 CIFAR-10，下游 SVHN *ASR*=0.1
	Moderate-fitting[86]	后门防御	监督学习	是	是	●	○	○	NLP	SST-2 *ASR*=0.15
	Purifier[87]	后门防御	监督学习	是	是	○	○	◐	NLP	CIFAR-10 ResNet-18 *ASR*=0
	Shan 等人[88]	后门溯源	监督学习	是	否	●	/	/	CV	CIFAR-10 BadNet trace *recall*=0.98
	BEAGLE[89]	后门溯源	监督学习	否	否	◐	/	/	CV	TrojAI trace *acc*=0.874
二次开发阶段	RIFT[94]	对抗防御	微调	是	否	○	●	●	NLP	IMDB BERT *ASR*=0.23
	ROSE[95]	对抗防御	微调	是	否	○	●	●	NLP	SST-2 GLUE=65.02
	Mudjacking[96]	后门防御	微调	是	是	◐	○	○	CV、NLP	SST-2 *ASR*=0.14
	Fine-mixing[97]	后门防御	微调	是	否	○	●	◐	NLP	BERT BadWord *ASR*=0.25
	Fine-purifying[98]	后门检测、防御	微调	是	否	○	●	◐	NLP	上游 CIFAR-10，下游 STL10 *ASR*=0.02
	AdaptGuard[99]	后门防御	迁移学习	是	是	●	●	●	CV	Office ResNet-50 *ASR*=0.15
	MixAdapt[100]	后门防御	迁移学习	是	是	○	●	◐	CV	Office ResNet-50 *ASR*=0.17
	SSDA[101]	后门防御	迁移学习	是	是	○	●	●	CV	Office-Home ResNet-50 *ASR*=0.02
	ReMoS[102]	对抗防御、后门防御	迁移学习	是	是	○	●	●	CV、NLP	ResNet18 *DIR*=0.15
	Jajal 等人[20]	故障分析	模型转换	是	/	/	/	/	/	/
	SafeCompress[103]	隐私窃取防御	压缩	否	否	○	●	●	CV、NLP	CIFAR-100 task *acc*=0.70
	DFLARE[104]	模型测试	压缩	是	/	●	○	○	CV	MNIST LeNet-1 8bit *acc*=1
部署阶段	MT-DComp[105]	编译器测试	模型编译	/	/	/	/	/	/	TVM *#Logic Errors* =435
	NNSmith[106]	编译器测试	模型编译	/	/	/	/	/	/	TVM *branch coverage*=18.6%
	HirGen[107]	编译器测试	模型编译	/	/	/	/	/	/	TVM *#Bug*=21
使用阶段	TextFusion[108]	隐私窃取防御	云端推理	是	是	●	/	/	NLP	SST-2 *ASR*=0
	TextObfuscator[109]	隐私窃取防御	云端推理	是	是	●	/	/	NLP	CoNLL2003 *ASR*=0.01

注：攻击者知识中，○表示未知，◐表示已知或部分已知，●表示全知. *ASR* 表示攻击成功率，“/”表示论文中未设置，*acc* 表示精确率，*#Logic Errors* 表示逻辑错误数，*branch coverage* 表示分支覆盖率，*recall* 表示召回率，*#Bug* 表示错误数目.

Table 4 Summary of Pre-training Model Warehouse

表 4 预训练模型仓库总结

模型仓库名称	模型数目	数据集数目	模型具体分类	支持任务数	贡献者属性	支持功能	支持计算框架库
Hugging Face[1]	252 014	46 170	多模态、CV、语音、表格、强化学习	36	开放式	API	TensorFlow，PyTorch，Keras，Caffe，JAX，ONNX，PaddlePaddle，Core ML，FLAIR
ModelScope[110]	955	492	CV、NLP、语音、多模态、科学计算、语种、其他	70	开放式	API，GitHub	TensorFlow，PyTorch，ONNX
PaddleHub[111]	135	10 000	CV、视频、NLP、文心大模型、智能语音、科学计算、跨模态、强化学习、量子计算、表格、图神经网络、时序技术、生物计算、智能推荐	14	开放式	API，GitHub	PaddlePaddle
TensorFlow Hub[15]	1 364	87	图像、文本、语音、视频	47	封闭式	API，GitHub	TensorFlow
Model Zoo[2]	1 211	0	CV、NLP、生成式模型、强音频、图、自监督学习、健康和生物医学	9	封闭式	GitHub	TensorFlow，PyTorch，Keras，Caffe，JAX，MXNet，Chainer
PyTorch Hub[112]	52	0	音频、生成式模型、NLP、可脚本化模型、CV	未知	封闭式	GitHub	PyTorch
ONNX Model Zoo[113]	55	0	CV、NLP、语音	10	封闭式	GitHub	ONNX
NVIDIA NGC[16]	560	34	CV、NLP、语音、其他	14	商业化	API	PyTorch，TensorFlow

用于强化学习任务的模型，而 PyTorch Hub 中并不提供这些模型. 另一方面，由深度学习框架平台搭建的预训练模型仓库往往只支持自身计算框架，并作为深度学习平台中的一个工具组件. 例如，PaddleHub 集中了 PaddlePaddle 生态下的预训练模型，包括了 PaddleNLP、PaddleCV 和文心大模型等基础模型库. 此外，开放式的模型仓库允许所有成员为该模型仓库做出贡献，提供的模型、数据集和所支持的任务数目更多，显示出更大的灵活性和兼容性. 这些模型仓库还建立了开发者社区，每个成员都可以贡献代码、文档和模型，参与学习、讨论和竞赛，共享开源成果. 但这种方式也不免得带来了更多的安全隐患.

与传统的软件供应链相比，模型仓库更像是一个研究和实践的框架. 模型贡献者是模型的作者，或正在维护或微调现有模型. 作者是创建模型的原始实现的研究人员，通常与学术著作一起发布. 通过解决模型中的漏洞，提高模型性能和准确率，或是提供替代模型、维护模型和拓展了模型属性.

8 研究总结和未来研究方向

目前对于深度学习模型供应链的研究仍处于研究起步阶段，不同于传统软件供应链中的安全问题，新的攻击形式和场景需求对供应链的安全应用提出了更严苛的挑战. 本文结合现有工作中的问题，指出现有深度学习模型供应链安全研究面临的挑战，并提出未来值得研究的方向.

8.1 现有研究总结

图 4 总结了目前供应链中不同阶段的攻击和防御方法. 鉴于使用阶段的攻防技术与单机模型的相应技术相似，本文不再详细讨论，且图中省略了相关文献的引用. 由图可知，后门攻击几乎贯穿了模型的整个生命周期，在设计、二次开发和部署阶段都容易被注入后门，并在不同阶段被触发. 因此，后门攻击也成为供应链中最主要的安全性威胁. 在二次开发阶段，模型受到的攻击最为多样化. 在该阶段，模型需要继承上游训练过程中获得的特征提取能力并适应下游任务. 二次开发者可以同时获取上下游 2 方面的知识，这拓宽了攻击面，增加了其可操作性. 攻击者可以同时对上下游进行攻击，例如通过特征嵌入对上游数据或模型进行逆向，或对下游数据进行中毒以向微调模型中注入后门. 因此，二次开发环节的安全性问题需要得到特别重视，相对应的安全防护方法受到了广泛研究. 部署阶段和使用阶段的攻击与单机模型类似，但由于预训练模型所需的庞大数据量和多样化的人机交互，使用阶段的模型隐私泄露风险也得到了学术界的持续关注.

8.2 现实挑战

与单个模型的安全性问题相比，供应链安全性的挑战在于需要考虑从数据收集、模型训练、部署到应用和服务的整个生命周期中的安全问题. 这包括但不限于确保数据的完整性和隐私性、模型的鲁棒性和抗攻击性，以及系统的可靠性和可用性. 目前深度模型供应链研发暴露的安全盲点与日俱增，然而已有的安全研究与现实需求之间还存在较大的差距，难以应对以下挑战：

1）开源模型供应链中的风险识别技术. 预训练模型顺着链式结构进行研发，引入了新的环节和步骤，涉及多个开发者和管理平台，依赖第三方组件或人员提供数据、模型或者算力. 攻击者有更多机会发起攻击，而复杂的链式结构也使攻击难以被检测. 此外，大型预训练模型将集成到各种平台和桌面或移动操作系统的交互中，将带来一系列挑战和潜在风险. 这些挑战包括 NLP 中与自然语言不精确性相关的问题，以及来自不可信第三方和物理世界驱动等的潜在风险. 如果这些问题不被主动考虑和解决，可能会引起严重的安全担忧. 在多环节的协作中，如果攻击者利用安全漏洞进行协作攻击，现有的检测方

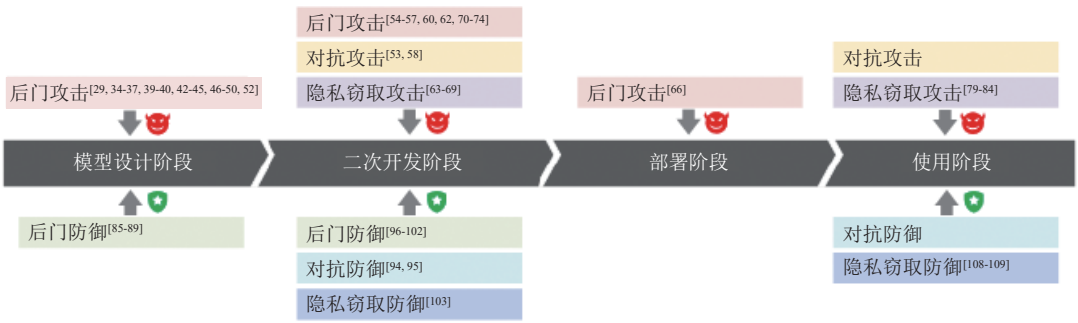


Fig. 4 Attacks and defense methods in DNN supply chain

图 4 深度神经网络供应链中的攻击和防御方法

法可能难以识别在持续开发过程中引入的隐蔽性漏洞,也难以对相关要素进行精确的溯源.因此,需要开发更为先进的技术来增强模型的安全性和鲁棒性,确保在集成大型预训练模型时,能够有效地预防和应对潜在的安全威胁.

2)模型全周期的安全性监测.现有针对单个模型的安全性监测技术可以应用于模型开发的单个环节,但难以覆盖链式研发的整个生命周期;同时,针对模型设计环节的模型安全监测存在滞后性,难以实时追踪后门植入.因此,目前供应链的安全性监测和防护仍然面临覆盖范围小、追踪效率低、可拓展性差的挑战.

3)多因素耦合导致的安全漏洞和隐私检测.现有安全性检测技术主要针对模型或数据等单个要素,无法应用于不同因素耦合导致的安全问题;此外,底层计算框架和上层要素的耦合增加了漏洞的多样性,不同组件之间的相互作用给隐私保护增加了难度[114],底层框架和二次开发技术的耦合扩展了测试空间,难以全面监测.因此,目前的安全漏洞和隐私检测仍然面临泛化性差、复杂度高、覆盖率低的挑战.

4)深度学习模型供应链安全监管.当前对预训练模型的监管主要依靠模型仓库平台的开发人员,仅对用户上传的模型进行初步审查,效率和可信度较低.此外,尚未对二次开发和部署中的模型进行安全性审查.因此,目前对链式开发过程中的模型安全性管控力度远远不足,各个环节中留存的漏洞可能在下一环节被放大,带来的安全威胁不言而喻.因此,实现多层次、全流程、主动式的深度模型链式研发安全监管,是目前亟待解决的问题.

8.3 未来研究方向

本文进一步总结了未来研究方向及相关可行思路,以应对当前深度学习模型供应链的现实挑战:

1)模型安全性实时监测.在供应链场景中,为了确保模型在开发和使用过程中可靠可信,在各个环节进行安全监测显得尤为重要.而目前的监测方法具有一定的滞后性,例如,监测者通常只在模型使用阶段进行检测和防御,而攻击者可以在多个环节共同作用以实施攻击.因此,有必要在不同阶段对数据、模型和算法3个要素进行实时监测.例如,采用模型运行监控,从构建属性的抽象开始,然后通过检查抽象与原始学习模型之间关键层输出的距离来确定属性的错误与故障.

2)安全性问题细粒度溯源.深度学习模型供应链主要由计算框架、模型算法、数据样本3要素构成,任一环节中的任一因素出现漏洞,都将影响模型的预测结果.为了高效追踪和溯源模型供应链中的漏洞,可以采用形式化验证技术来确保模型算法的安全性.同时,结合静态分析工具对模型代码进行细致的审查,以识别潜在的安全漏洞.模型发布者可以向预训练模型嵌入水印,有助于错误跟踪和溯源,为提升系统安全性提供精确的指导.

3)新型安全防御策略构建.大型预训练模型在安全性方面面临一系列挑战,这些挑战要求对传统机器学习的攻击方法进行演化和调整,以应对多环节、多要素、大规模参数、模型所有权的私有性等新特性.PLM作为深度神经网络的特例,不仅继承了传统机器的安全漏洞,如对抗性攻击,而且还需要面对链式开发环节上特定的安全威胁.为了有效防御这些攻击,传统的机器学习防御策略也需进行调整,例如利用隐私增强技术,包括零知识证明、采用信息论角度控制隐私与供应链安全融合[115]等,来解决PLM的隐私保护问题.此外,持续研究和开发新的隐私保护技术,以应对链式开发带来的新挑战,是未来研究的重要方向.

9 结束语

预训练模型缓解了训练模型高成本的问题,促成了深度学习模型供应链的发展.然而,随着模型在设计、二次开发、部署和使用等供应链关键环节的逐步推进,新的安全威胁也随之产生.本文全面阐述了深度学习模型供应链的概念,并详细介绍了其架构和操作流程.针对模型生命周期的各个阶段,系统地分析了潜在的安全威胁,并提供了相应的安全防护策略.尽管目前对深度学习模型供应链的安全性研究尚处于早期阶段,但本文从多角度剖析了该领域面临的挑战,并对未来的研究方向进行了展望,旨在为深化安全研究和推动实际应用提供指导和参考.

作者贡献声明: 陈若曦提出了整体框架,负责内容设计、论文撰写和修订;陈晋音、郑海斌提出指导意见并修改论文;阳雪燕完成文献整理和分析;纪守领、陈铁明针对论文提出了指导意见并修改论文.

参 考 文 献

[1] Hugging Face. Hugging Face—The AI community building the future[EB/OL]. (2016-03-01)[2022-03-21]. https://huggingface.co/

[2] Yu Koh J. Model Zoo—Deep learning code and pretrained

models[EB/OL]. (2018-06-14)[2023-03-15]. https://modelzoo.co/

[3] Qiu Xipeng, Sun Tianxiang, Xu Yige, et al. Pre-trained models for natural language processing: A survey[J]. Science China Technological Sciences, 2020, 63(10): 1872−1897

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

[5] Baracaldo N, Oprea A. Machine learning security and privacy[J]. IEEE Security & Privacy, 2022, 20(5): 11−13

[6] Liu Ximeng, Xie Lehui, Wang Yaopeng, et al. Privacy and security issues in deep learning: A survey[J]. IEEE Access, 2020, 9: 4566−4593

[7] Xue Mingfu, Yuan Chengxiang, Wu Heyi, et al. Machine learning security: Threats, countermeasures, and evaluations[J]. IEEE Access, 2020, 8: 74720−74742

[8] Kenton J D M W C, Toutanova L K. BERT: Pre-training of deep bidirectional transformers for language understanding[C]//Proc of the 17th NAACL-HLT. Stroudsburg, PA: ACL, 2019: 4171−4186

[9] Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16x16 words: Transformers for image recognition at scale[J]. arXiv preprint, arXiv: 2010.11929, 2020

[10] Radford A, Kim J W, Hallacy C, et al. Learning transferable visual models from natural language supervision[C]//Proc of the 18th Int Conf on Machine Learning. New York: PMLR, 2021: 8748−8763

[11] Liu Haotian, Li Chunyuan, Wu Qingyang, et al. Visual instruction tuning[J]. arXiv preprint, arXiv: 2304.08485, 2023

[12] Rombach R, Blattmann A, Lorenz D, et al. High-resolution image synthesis with latent diffusion models[C]//Proc of the 39th IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2022: 10684−10695

[13] Cruz Jr G V, Du Yunshu, Taylor M E. Pre-training neural networks with human demonstrations for deep reinforcement learning[J]. arXiv preprint, arXiv: 1709.04083, 2017

[14] Jiang Wenxin, Synovic N, Sethi R, et al. An empirical study of artifacts and security risks in the pre-trained model supply chain[C]//Proc of the 1st ACM Workshop on Software Supply Chain Offensive Research and Ecosystem Defenses. New York: ACM, 2022: 105−114

[15] TensorFlow. TensorFlow Hub[EB/OL]. (2018-03-01)[2023-03-18]. https://www.tensorflow.org/hub

[16] NVIDIA. NVIDIA NGC: AI development catalog[EB/OL]. (2017-10-01)[2023-03-21]. https://catalog.ngc

[17] Russakovsky O, Deng Jia, Su Hao, et al. ImageNet large scale visual recognition challenge[J]. International Journal of Computer Vision, 2015, 115: 211−252

[18] He Kaiming, Zhang Xiangyu, Ren Shaoqing, et al. Deep residual learning for image recognition[C]//Proc of the 29th IEEE/CVF Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2016: 770−778

[19] He Kaiming, Girshick R, Dollár P. Rethinking ImageNet pre-training[C]//Proc of the 32nd IEEE/CVF Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2019: 4918−4927

[20] Jajal P, Jiang Wenxin, Tewari A, et al. Analysis of failures and risks in deep learning model converters: A case study in the ONNX ecosystem[J]. arXiv preprint, arXiv: 2303.17708, 2023

[21] Choudhary T, Mishra V, Goswami A, et al. A comprehensive survey on model compression and acceleration[J]. Artificial Intelligence Review, 2020, 53: 5113−5155

[22] Krishnamoorthi R. Quantizing deep convolutional networks for efficient inference: A whitepaper[J]. arXiv preprint, arXiv: 1806.08342, 2018

[23] Dong Xin, Chen Shangyu, Pan S. Learning to prune deep neural networks via layer-wise optimal brain surgeon[J]. Advances in Neural Information Processing Systems, 2017, 30(1): 4857−4867

[24] Gao Han, Tian Yulong, Xu Fengyuan, et al. Survey of deep learning model compression and acceleration[J]. Journal of Software, 2021, 32(1): 68−92 (in Chinese)
(高晗, 田育龙, 许封元, 等. 深度学习模型压缩与加速综述[J]. 软件学报, 2021, 32(1): 68−92)

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

[26] Dube P, Bhattacharjee B, Huo Siyu, et al. Automatic labeling of data for transfer learning suggestion[C/OL]//Proc of the 32nd IEEE/CVF Conf on Computer Vision and Pattern Recognition. 2019[2023-05-06]. https://openreview.net/forum?id=rkxJgoRN_V

[27] Hugging Face[EB/OL]. (2016-03-01)[2022-03-21]. https://huggingface.co/datasets/beans

[28] Schelter S, Biessmann F, Januschowski T, et al. On challenges in machine learning model management[J/OL]. IEEE Data Engineering Bulletin. 2015[2023-05-02]. https://www.amazon.science/publications/on-challenges-in-machine-learning-model-management

[29] Shen Lujia, Ji Shouling, Zhang Xuhong, et al. Backdoor pre-trained models can transfer to all[C]//Proc of the 28th ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2021: 3141−3158

[30] Ji Shouling, Du Tianyu, Li Jinfeng, et al. Security and privacy of machine learning models: A survey[J]. Journal of Software, 2021, 32(1): 41−67 (in Chinese)
(纪守领, 杜天宇, 李进锋, 等. 机器学习模型安全与隐私研究综述[J]. 软件学报, 2021, 32(1): 41−67)

[31] Jiang Wenxin, Synovic N, Hyatt M, et al. An empirical study of pre-trained model reuse in the hugging face deep learning model registry[J]. arXiv preprint, arXiv: 2303.02552, 2023

[32] Jiang Wenxin, Banna V, Vivek N, et al. Challenges and practices of deep learning model reengineering: A case study on computer vision[J]. arXiv preprint, arXiv: 2303.07476, 2023

[33] Liu Zeyan, Li Fengjun, Li Zhu, et al. LoneNeuron: A highly-effective feature-domain neural trojan using invisible and polymorphic watermarks[C]//Proc of the 29th ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2022: 2129−2143

[34] Kurita K, Michel P, Neubig G. Weight poisoning attacks on pretrained models[C]//Proc of the 58th Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA: ACL, 2020: 2793−2806

[35] Li Linyang, Song Demin, Li Xiaonan, et al. Backdoor attacks on pre-

trained models by layerwise weight poisoning[C]//Proc of the 58th Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2021: 3023−3032

[36] Zhang Xinyang, Zhang Zheng, Ji Shouling, et al. Trojaning language models for fun and profit[C]//Proc of the 6th IEEE European Symp on Security and Privacy (EuroS&P). Piscataway, NJ: IEEE, 2021: 179−197

[37] Cai Xiangrui, Xu Haidong, Xu Sihan, et al. BadPrompt: Backdoor attacks on continuous prompts[J]. Advances in Neural Information Processing Systems, 2022, 35(1): 37068−37080

[38] Grill J B, Strub F, Altché F, et al. Bootstrap your own latent—A new approach to self-supervised learning[J]. Advances in Neural Information Processing Systems, 2020, 33(1): 21271−21284

[39] Jia Jinyuan, Liu Yupei, Gong Zhengqiang. BadEncoder: Backdoor attacks to pre-trained encoders in self-supervised learning[C]//Proc of the 43rd IEEE Symp on Security and Privacy (SP). Piscataway, NJ: IEEE, 2022: 2043−2059

[40] Liu Hongbin, Jia Jinyuan, Gong Zhengqiang. PoisonedEncoder: Poisoning the unlabeled pre-training data in contrastive learning[C]//Proc of the 31st USENIX Security Symp (USENIX Security 22). Berkeley, CA: USENIX Association, 2022: 3629−3645

[41] Chen Ting, Kornblith S, Norouzi M, et al. A simple framework for contrastive learning of visual representations[C]//Proc of the 37th Int Conf on Machine Learning. New York: PMLR, 2020: 1597−1607

[42] Zhang Jinghuai, Liu Hongbin, Jia Jinyuan, et al. CorruptEncoder: Data poisoning based backdoor attacks to contrastive learning[J]. arXiv preprint, arXiv: 2211.08229, 2022

[43] Ji Yujie, Zhang Xinyang, Ji Shouling, et al. Model-reuse attacks on deep learning systems[C]//Proc of the 25th ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2018: 349−363

[44] Chen Kangjie, Meng Yuxian, Sun Xiaofei, et al. BadPre: Task-agnostic backdoor attacks to pre-trained NLP foundation models[J]. arXiv preprint, arXiv: 2110.02467, 2021

[45] Zhang Zhengyan, Xiao Guangxuan, Li Yongwei, et al. Red alarm for pre-trained models: Universal vulnerability to neuron-level backdoor attacks[J]. Machine Intelligence Research, 2023, 20(2): 180−193

[46] Lv Peizhuo, Yue Chang, Liang Ruigang, et al. A data-free backdoor injection approach in neural networks[C]//Proc of the 32nd USENIX Security Symp (USENIX Security 23). Berkeley, CA: USENIX Association, 2023: 2671−2688

[47] Mei Kai, Li Zheng, Wang Zhenting, et al. NOTABLE: Transferable backdoor attacks against prompt-based NLP models[C]//Proc of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Stroudsburg, PA: ACL, 2023: 15551−15565

[48] Du Wei, Li Peixuan, Li Boqun, et al. UOR: Universal backdoor attacks on pre-trained language models[J]. arXiv preprint, arXiv: 2305.09574, 2023

[49] Wang Hao, Guo Shangwei, He Jialiang, et al. Model supply chain poisoning: Backdooring pre-trained models via embedding indistinguishability[J]. arXiv preprint, arXiv: 2401.15883, 2024

[50] Huang Yujun, Zhuo Terry Yue, Xu Qiongkai, et al. Training-free lexical backdoor attacks on language models[C]//Proc of the 32nd ACM Web Conf 2023. New York: ACM, 2023: 2198−2208

[51] Wang Yue, Wang Weishi, Joty S, et al. CodeT5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation[C]//Proc of the 36th Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2021: 8696−8708

[52] Li Yanzhou, Liu Shangqing, Chen Kangjie, et al. Multi-target backdoor attacks for code pre-trained models[J]. arXiv preprint, arXiv: 2306.08350, 2023

[53] Wang Bolun, Yao Yuanshun , Viswanath B, et al. With great training comes great vulnerability: Practical attacks against transfer learning[C]//Proc of the 27th USENIX Security Symp (USENIX Security 18). Berkeley, CA: USENIX Association, 2018: 1281−1297

[54] Yao Yuanshun, Li Huiying, Zheng Haitao, et al. Latent backdoor attacks on deep neural networks[C]//Proc of the 26th ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2019: 2041−2055

[55] Wang Shuo, Nepal S, Rudolph C, et al. Backdoor attacks against transfer learning with pre-trained deep learning models[J]. IEEE Transactions on Services Computing, 2020, 15(3): 1526−1539

[56] Jiang Wenbo, Zhang Tianwei, Qiu H, et al. Incremental learning, incremental backdoor threats[J]. IEEE Transactions on Dependable and Secure Computing, 2022, 21(2): 559−572

[57] Li Zhizhong, Hoiem D. Learning without forgetting[J]. IEEE Transactions on Pattern Analysis And Machine Intelligence, 2017, 40(12): 2935−2947

[58] Ban Yuanhao, Dong Yinpeng. Pre-trained adversarial perturbations [J]. Advances in Neural Information Processing Systems, 2022, 35(1): 1196−1209

[59] Lester B, Al-Rfou R, Constant N. The power of scale for parameter-efficient prompt tuning[C]//Proc of the 26th Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2021: 3045−3059

[60] Du Wei, Zhao Yichun, Li Boqun, et al. PPt: Backdoor attacks on pre-trained models via poisoned prompt tuning[C]//Proc of the 31st Int Joint Conf on Artificial Intelligence (IJCAI−22). San Francisco, CA: Morgan Kaufmann: 2022: 680−686

[61] Chung H W, Hou Le, Longpre S, et al. Scaling instruction-finetuned language models[J]. Journal of Machine Learning Research, 2024, 25(70): 1−53

[62] Xu Jiashu, Ma Mingyu, Wang Fei, et al. Instructions as backdoors: Backdoor vulnerabilities of instruction tuning for large language models[C]//Proc of the 20th Conf of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers). Stroudsburg, PA: ACL, 2024: 3111−3126

[63] Chen Yufei, Shen Chao, Wang Cong, et al. Teacher model fingerprinting attacks against transfer learning[C]//Proc of the 31st USENIX Security Symp (USENIX Security 22). Berkeley, CA: USENIX Association, 2022: 3593−3610

[64] Tian Yulong, Suya F, Suri A, et al. Manipulating transfer learning for property inference[C]//Proc of the 36th IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2023: 15975−15984

[65] Song Congzheng, Raghunathan A. Information leakage in embedding models[C]//Proc of the 27th ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2020: 377−390

[66] Carlini N, Tramer F, Wallace E, et al. Extracting training data from large language models[C]//Proc of the 30th USENIX Security Symp (USENIX Security 21). Berkeley, CA: USENIX Association, 2021: 2633−2650

[67] Liu Hongbin, Jia Jinyuan, Qu Wenjie, et al. EncoderMI: Membership inference against pre-trained encoders in contrastive learning[C]//Proc of the 28th ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2021: 2081−2095

[68] Liu Yupei, Jia Jinyuan, Liu Hongbin, et al. StolenEncoder: Stealing pre-trained encoders in self-supervised learning[C]//Proc of the 29th ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2022: 2115−2128

[69] Sha Zeyang, He Xinlei, Yu Ning, et al. Can't steal? Cont-steal! Contrastive stealing attacks against image encoders[C]//Proc of the 40th IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2023: 16373−16383

[70] Hong S, Panaitescu-Liess M A, Kaya Y, et al. Qu-anti-zation: Exploiting quantization artifacts for achieving adversarial outcomes[J]. Advances in Neural Information Processing Systems, 2021, 34(1): 9303−9316

[71] Ma Hua, Qiu Huming, Gao Yansong, et al. Quantization backdoors to deep learning commercial frameworks[J]. IEEE Transactions on Dependable and Secure Computing, 2023, 21(3): 1155−1172

[72] Tian Yulong, Suya F, Xu Fengyan, et al. Stealthy backdoors as compression artifacts[J]. IEEE Transactions on Information Forensics and Security, 2022, 17: 1372−1387

[73] Phan H, Shi Cong, Xie Yi, et al. RIBAC: Towards robust and imperceptible backdoor attack against compact DNN[C]//Proc of the 17th European Conf on Computer Vision 2022. Berlin: Springer, 2022: 708−724

[74] Ge Yunjie, Wang Qian, Zheng Baolin, et al. Anti-distillation backdoor attacks: Backdoors can really survive in knowledge distillation[C]//Proc of the 29th ACM Int Conf on Multimedia. New York: ACM, 2021: 826−834

[75] Qi Xiangyu, Xie Tinghao, Pan Ruizhe, et al. Towards practical deployment-stage backdoor attack on deep neural networks[C]//Proc of the 39th IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2022: 13347−13357

[76] Zou A, Wang Zifan, Kolter J Z, et al. Universal and transferable adversarial attacks on aligned language models[J]. arXiv preprint, arXiv: 2307.15043, 2023

[77] Wei A, Haghtalab N, Steinhardt J. Jailbroken: How does llm safety training fail?[J]. Advances in Neural Information Processing Systems, 2024, 36(1): 80079-80110

[78] Shen Xinyue, Chen Zeyuan, Backes M, et al. " do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models[C]//Proc of the 31st on ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2024: 1671−1685

[79] Zanella-Béguelin S, Wutschitz L, Tople S, et al. Analyzing information leakage of updates to natural language models[C]//Proc of the 27th ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2020: 363−375

[80] Panchendrarajan R, Bhoi S. Dataset reconstruction attack against language models[C/OL]//Proc of the 30th Int Joint Conf on Artificial Intelligence. 2021[2021-07-01]. http://cdap.sliit.lk/bitstream/123456789/2045/1/DatasetReconstructionAttackagainstLanguageModels.pdf

[81] Lukas N, Salem A, Sim R, et al. Analyzing leakage of personally identifiable information in language models[C]//Proc of the 44th IEEE Symp on Security and Privacy (SP). Piscataway, NJ: IEEE, 2023: 346−363

[82] Keskar N S, McCann B, Xiong Caiming, et al. The thieves on sesame street are polyglots-extracting multilingual models from monolingual APIs[C]//Proc of the 25th Conf on Empirical Methods in Natural Language Processing (EMNLP). Stroudsburg, PA: ACL, 2020: 6203−6207

[83] Xu Qiongkai, He Xuanli, Lyu Lingjuan, et al. Student surpasses teacher: Imitation attack for black-box NLP APIs[C]//Proc of the 29th Int Conf on Computational Linguistics. Stroudsburg, PA: ACL, 2022: 2849−2860

[84] Li Zongjie, Wang Chaozheng, Ma Pingchuan, et al. On extracting specialized code abilities from large language models: A feasibility study[C]//Proc of the 46th IEEE/ACM Int Conf on Software Engineering. 2024: 1−13

[85] Feng Shiwei, Tao Guanhong, Cheng Siyuan, et al. Detecting backdoors in pre-trained encoders[C]//Proc of the 40th IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2023: 16352−16362

[86] Zhu Biru, Qin Yujia, Cui Ganqu, et al. Moderate-fitting as a natural backdoor defender for pre-trained language models[J]. Advances in Neural Information Processing Systems, 2022, 35(1): 1086−1099

[87] Zhang Xiaoyu, Jin Yulin, Wang Tao, et al. Purifier: Plug-and-play backdoor mitigation for pre-trained models via anomaly activation suppression[C]//Proc of the 30th ACM Int Conf on Multimedia. New York: ACM, 2022: 4291−4299

[88] Shan S, Bhagoji A N, Zheng Haitao, et al. Poison forensics: Traceback of data poisoning attacks in neural networks[C]//Proc of the 31st USENIX Security Symp (USENIX Security 22). Berkeley, CA: USENIX Association, 2022: 3575−3592

[89] Cheng Siyuan, Tao Guanhong, Liu Yingqi, et al. BEAGLE: Forensics of deep learning backdoor attack for better defense[J]. arXiv preprint, arXiv: 2301.06241, 2023

[90] Liu Kang, Dolan-Gavitt B, Garg S. Fine-pruning: Defending against backdooring attacks on deep neural networks[C]//Proc of the 21st Int Symp on Research in Attacks, Intrusions, and Defenses. Berlin: Springer, 2018: 273−294

[91] Li Yige, Lyu Xixiang, Koren N, et al. Neural attention distillation: Erasing backdoor triggers from deep neural networks[J]. arXiv

preprint, arXiv: 2101.05930, 2021

[92] Papernot N, McDaniel P, Wu Xi, et al. Distillation as a defense to adversarial perturbations against deep neural networks[C]//Proc of the 37th IEEE Symp on Security and Privacy (SP). Piscataway, NJ: IEEE, 2016: 582−597

[93] Khalid F, Ali H, Tariq H, et al. Qusecnets: Quantization-based defense mechanism for securing deep neural network against adversarial attacks[C]//Proc of the 25th IEEE Int Symp on On-Line Testing and Robust System Design (IOLTS). Piscataway, NJ: IEEE, 2019: 182−187

[94] Dong Xinshuai, Luu A T, Lin Min, et al. How should pre-trained language models be fine-tuned towards adversarial robustness?[J]. Advances in Neural Information Processing Systems, 2021, 34(1): 4356−4369

[95] Jiang Lan, Zhou Hao, Lin Yankai, et al. ROSE: Robust selective fine-tuning for pre-trained language models[C]//Proc of the 27th Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA, ACL: 2022: 2886−2897

[96] Liu Hongbin, Reiter M K, Gong N Z. Mudjacking: Patching backdoor vulnerabilities in foundation models[J]. arXiv preprint, arXiv: 2402.14977, 2024

[97] Zhang Zhiyuan, Lyu Lingjuan, Ma Xingjun, et al. Fine-mixing: Mitigating backdoors in fine-tuned language models[C]/Proc of the 27th Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA, ACL: 2022: 355−372

[98] Zhang Zhiyuan, Chen Deli, Zhou Hao, et al. Diffusion theory as a scalpel: Detecting and purifying poisonous dimensions in pre-trained language models caused by backdoor or bias[C]//Proc of the 61st Association for Computational Linguistics. Stroudsburg, PA: ACL, 2023: 2495−2517

[99] Sheng Lijun, Liang Jian, He Ran, et al. AdaptGuard: Defending against universal attacks for model adaptation[C]//Proc of the 39th IEEE/CVF Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2023: 19093−19103

[100] Sheng Lijun, Liang Jian, He Ran, et al. Can we trust the unlabeled target data? Towards backdoor attack and defense on model adaptation[J]. arXiv preprint, arXiv: 2401.06030, 2024

[101] Ahmed S, Al Arafat A, Rizve M N, et al. SSDA: Secure source-free domain adaptation[C]//Proc of the 39th IEEE/CVF Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2023: 19180−19190

[102] Zhang Ziqi, Li Yuanchun, Wang Jindong, et al. ReMoS: Reducing defect inheritance in transfer learning via relevant model slicing[C]//Proc of the 44th Int Conf on Software Engineering. Piscataway, NJ: IEEE, 2022: 1856−1868

[103] Zhu Jie, Wang Leye, Han Xiao. Safety and performance, why not both? bi-objective optimized model compression against heterogeneous attacks toward AI software deployment[J]. IEEE Transactions on Software Engineering, 2024, 50(3): 376−390

[104] Tian Yongqiang, Zhang Wuqi, Wen Ming, et al. Finding deviated behaviors of the compressed DNN models for image classifications[J]. ACM Transactions on Software Engineering and Methodology, 2023, 32(5): 1−32

[105] Xiao Dongwei, Liu Zhibo, Yuan Yuanyuan, et al. Metamorphic testing of deep learning compilers[J]. ACM on Measurement and Analysis of Computing Systems, 2022, 6(1): 1−28

[106] Liu Jiawei, Lin Jinkun, Ruffy F, et al. NNSmith: Generating diverse and valid test cases for deep learning compilers[C]//Proc of the 28th ACM Int Conf on Architectural Support for Programming Languages and Operating Systems, Volume 2. New York: ACM, 2023: 530−543

[107] Ma Haoyang, Shen Qingchao, Tian Yongqiang, et al. Fuzzing deep learning compilers with HirGen[C]//Proc of the 32nd ACM SIGSOFT Int Symp on Software Testing and Analysis. New York: ACM, 2023: 248−260

[108] Zhou Xin, Lu Jinzhu, Gui Tao, et al. TextFusion: Privacy-preserving pre-trained model inference via token fusion[C]//Proc of the 27th Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA, ACL: 2022: 8360−8371

[109] Zhou Xin, Lu Yi, Ma Ruotian, et al. TextObfuscator: Making pre-trained language model a privacy protector via obfuscating word representations[C]//Proc of the 61st Association for Computational Linguistics. Stroudsburg, PA: ACL, 2023: 5459−5473

[110] Modelscope. ModelScope: Bring the notion of model-as-a-service to life[EB/OL]. (2022-11-03)[2024-05-01]. https://github.com/modelscope/modelscope

[111] PaddlePaddle. PaddleHub—An open-source deep learning platform originated from industrial practice[EB/OL]. (2019-07-08)[2023-03-15]. https://www.paddlepaddle.org.cn/hub

[112] Pytorch. PyTorch Hub. [EB/OL]. (2019-06-01)[2023-05-01]. https://pytorch.org/hub/

[113] ONNX. ONNX Model Zoo[EB/OL]. (2017-04-20)[2023-06-01]. https://github.com/onnx/models

[114] Debenedetti E, Severi G, Carlini N, et al. Privacy side channels in machine learning systems[C]//Proc of the 33rd USENIX Security Symp (USENIX Security 24). Berkeley, CA: USENIX Association, 2024: 6861−6848

[115] Tan Qi, Li Qi, Zhao Yi, et al. Defending against data reconstruction attacks in federated learning: An information theory approach[J]. arXiv preprint, arXiv: 2403.01268, 2024



Chen Ruoxi, born in 1998. PhD candidate. Student candidate of CCF. Her main research interests include deep learning and artificial intelligence security.

陈若曦, 1998 年生. 博士研究生. CCF 学生会员. 主要研究方向为深度学习、人工智能安全.



Chen Jinyin, born in 1982. PhD, professor. Member of CCF. Her main research interests include artificial intelligence security, graph data mining, and evolutionary computing.

陈晋音, 1982 年生. 博士, 教授, CCF 会员. 主要研究方向为人工智能安全、图数据挖掘、进化计算.



Zheng Haibin, born in 1995. PhD, lecturer. Member of CCF. His main research interests include deep learning, artificial intelligence security, and fairness algorithm.

郑海斌, 1995 年生. 博士, 讲师. CCF 会员. 主要研究方向为深度学习、人工智能安全、公平性算法.



Yang Xueyan, born in 2003. Undergraduate student. Her main research interests include deep learning and artificial intelligence security.

阳雪燕, 2003 年生. 本科生. 主要研究方向为深度学习、人工智能安全.



Ji Shouling, born in 1986. PhD. Senior member of CCF. ZJU 100-Young Professor in the College of Computer Science and Technology at Zhejiang University. His current research interests include data-driven security and privacy, AI security, and big data analytics.

纪守领, 1986 年生. 博士, CCF 高级会员, 浙江大学“百人计划”研究员. 主要研究方向为数据驱动安全和隐私、AI 安全、大数据分析.



Chen Tieming, born in 1978. PhD, professor. His main research interest includes network information security.

陈铁明, 1978 年生. 博士, 教授. 主要研究方向为网络信息安全.