大语言模型安全的挑战与机遇

付志远[1,2], 陈思宇[1,2], 陈骏帆[1,2], 海 翔[1,2], 石岩松[1,2], 李晓琦[1],
李益红[1], 岳秋玲[1], 张玉清[1,2]

[1]海南大学 网络空间安全学院 海口 中国 570228
[2]国家计算机网络入侵防范中心 中国科学院大学 北京 中国 101408

摘要 大语言模型的技术进步不仅推动了人工智能领域的快速发展, 也带来了前所未有的安全挑战。大语言模型在处理自然语言理解和生成等任务时的高效, 使其在多个行业中得到广泛应用, 包括自动化客服、内容创作、情感分析、医疗诊断、金融分析以及法律咨询等。然而, 随着应用的深入, 大语言模型面临的安全威胁也日益显现, 如模型被恶意利用生成虚假信息、隐私泄露问题以及模型的偏见和不公平等问题。本文深入探讨了大语言模型的安全挑战, 并分析了如何利用这些模型来增强传统安全方法。我们首先综合分析了近年来在国际学术会议和期刊上发表的本领域论文, 并进行了详尽的归纳和总结。接着, 我们从数据和隐私保护、法律与伦理、攻击及其防御三个角度详细分析了大语言模型自身面临的安全问题以及现有的解决方案。同时, 我们还总结了一系列大语言模型在传统安全领域中的应用案例, 包括网络安全、物理安全和信息安全。进一步, 我们调研和归纳了国内外企业在大语言模型领域的最新尝试, 许多企业正在积极探索如何将大语言模型赋能于实际安全业务中。最后, 我们探讨了面临的挑战与机遇, 并提出了解决这些问题的可行策略和建议。通过本文的深入分析, 我们希望能够提高公众和业界对大语言模型安全问题的关注, 并为未来的研究和应用提供方向和启发, 推动整个行业朝着更加安全和可靠的方向发展。

关键词 大语言模型; 人工智能安全; 隐私安全; 防御措施
中图法分类号 TP18 **DOI 号** 10.19363/J.cnki.cn10-1380/tn.2024.09.02

Challenges and Opportunities of Large Language Model Security

FU Zhiyuan[1,2], CHEN Siyu[1,2], CHEN Junfan[1,2], HAI Xiang[1,2], SHI Yansong[1,2], LI Xiaoqi[1],
LI Yihong[1], YUE Qiuling[1], ZHANG Yuqing[1,2]

[1] College of Cyberspace Security, Hainan University, Haikou 570228, China
[2] National Computer Network Intrusion Protection Center, University of Chinese Academy of Sciences, Beijing 101408, China

Abstract The technological advancements of large language models have not only accelerated the rapid development of the field of artificial intelligence but also brought unprecedented security challenges. The efficiency of large language models in handling tasks such as natural language understanding and generation has led to their widespread application in various industries, including automated customer service, content creation, sentiment analysis, medical diagnosis, financial analysis, and legal consultation. However, with the deepening of these applications, the security threats faced by large language models have become increasingly apparent, such as malicious use to generate false information, privacy leakage issues, and problems of bias and unfairness in the models. This paper explores the security challenges of large language models in depth and analyzes how these models can be used to enhance traditional security methods. First, we comprehensively analyze papers in this field published at international academic conferences and journals in recent years, providing a detailed summary and synthesis. Then, we analyze the security issues faced by large language models and existing solutions from three perspectives: data and privacy protection, law and ethics, and attacks and defenses. We also summarize a series of application cases of large language models in traditional security fields, including cybersecurity, physical security, and information security. Furthermore, we investigate and summarize the latest attempts by domestic and international enterprises in the field of large language models, where many companies are actively exploring how to empower actual security businesses with large language models. Finally, we discuss the challenges and opportunities faced and propose feasible strategies and recom-

通讯作者: 张玉清, 博士, 教授, Email: zhangyq@ucas.ac.cn。
本课题得到国家重点研发计划项目(No. 2023YFB3106400, No. 2023QY1202)、国家自然科学基金重点项目(No. U2336203, No. U1836210)、海南省重点研发计划项目(No. GHYF2022010)、北京市自然科学基金(No. 4242031)、海南省教育厅项目(No. HNJG2023-10)、海南省省属科研院所技术创新项目(No. KYYSGY2023-003, No. SQKY2022-0039)。
收稿日期: 2024-03-31; 修改日期: 2024-05-31; 定稿日期: 2024-05-31

mendations to address these issues. Through this in-depth analysis, we hope to raise public and industry awareness of the security issues of large language models and provide directions and insights for future research and applications, promoting the entire industry towards a safer and more reliable direction.

Key words large language models; AI security; privacy & security; defensive techniques

1 大语言模型发展历程

大语言模型(Large Language Model, LLM)是一种基于海量数据构建的深度学习模型[1], 代指大型的预训练语言模型(Pre-training Language Model, PLM), 由包含数十亿个参数(权重)的复杂神经网络构成, 采用自监督学习或半监督学习方法, 在大规模未标记数据上进行训练。大语言模型的“大”, 来源于其训练数据之庞大, 模型结构之复杂。正因如此, 它在诞生之初, 就在文本分类、对话、问答等方面展现出了比其他语言模型的优越性, 并迅速引起了社会广泛的关注。

现如今的主流大语言模型可以分为以下三类[2]:

(1)纯编码器(Encoder)模型: 例如 BERT, 这类模型也被称为自编码(auto-encoding)Transformer 模型, 专注于捕获输入序列的深层语义表示, 常用于理解任务, 如文本分类、实体识别等。

(2)纯解码器(Decoder)模型: 例如 GPT, 亦被称为自回归(auto-regressive)Transformer 模型, 能够基于给定的文本前缀生成连贯的文本序列, 适用于生成任务, 如文本续写、创造性写作等。

(3)编码器-解码器(Encoder-Decoder)模型: 例如 T5, 这些模型也称为序列到序列(Seq2Seq)Transformer 模型, 它们结合了编码器和解码器的优势, 能够处理复杂的语言转换任务, 如机器翻译、文本摘要制作等。

这些分类并不是完全独立的, 有些模型可能涵盖多个分类, 随着研究的不断进展, 还会出现新的模型类别和变种。大语言模型的发展历程可以追溯到深度学习技术的兴起和应用。

1970—2000 年, 基于规则的模型和应用场景有限基于统计的模型是主流。这个阶段主要是基于手写规则或者基于数学统计的角度预测下一个词的出现概率, 只能处理少量数据, 例如 N-Gram 模型。后来随着基于循环神经网络(RNN)或长短时记忆网络(LSTM)的语言模型的提出, 这些模型可以处理序列数据, 如文本生成, 语音识别, 时间序列预测等。然而, 随着数据集规模的增加和模型参数的增多, 这些模型的表现逐渐受到限制。为了解决这个问题, 研究人员开始探索基于 Transformer 架构的语言模型。

在 2017 年 Transformer 发布[3]之后, Transformer 架构的出现为大语言模型的发展奠定了基础。Transformer 模型具有并行计算能力和多头自注意力机制, 可以有效地处理长序列数据, 并具有更好的表现。基于 Transformer 架构的语言模型, 如 BERT[4]、GPT-2[5]和 GPT-3[6]等, 在自然语言处理任务中取得了显著的成功。于 2023 年正式发布的 GPT-4[7]在 GPT-3 的基础上更进一步的强化了各个方面的能力, 同时引入了多模态。2024 年 3 月 OpenAI 的 CEO Sam Altman 在访谈[8]中提到在接下来的几个月将会发布更加强大的模型。2024 年 5 月 13 日 OpenAI 发布了旗舰模型 GPT-4o, 能够实时地跨音频、视觉和文本进行推理。

大语言模型的开发和部署已经成为全球性的趋势, 表 1 我们总结了目前国内外流行的大语言模型,

表 1 现有流行的国内外大模型

Table 1 Existing popular domestic and foreign large models

大模型公司	国别	开源	模型
OpenAI	美国	-	ChatGPT-4[7]
		-	ChatGPT-3.5
Anthropic	美国	-	Claude 3
xAI	美国	✓	Grok-1
		-	Grok-1.5
Meta	美国	✓	LlaMa[9]
		✓	LlaMa 2[10]
		✓	Llama 3[11]
Google	美国	-	Palm 2[12]
		-	Gemini[13]
		✓	Gemma
Microsoft	美国	-	Copilot[14]
		✓	Phi-3
Databricks	美国	✓	DBRX
Mistral AI	法国	✓	mistral[15]
阿里	中国	✓	Qwen[16]
		-	通义千问[17]
腾讯	中国	-	混元[18]
华为	中国	✓	盘古[19]
百度	中国	-	文心一言[20]
智谱华章	中国	✓	ChatGLM[21]
		-	GLM-4
Vivo	中国	✓	BlueLM[22]
		-	蓝心小 V
百川智能	中国	✓	Baichuan 2[23]
		-	Baichuan 4
科大讯飞	中国	-	星火
		✓	iFlytekSpark[24]
中国科学院	中国	-	紫东太初[25]
复旦	中国	✓	MOSS[26]
零一万物	中国	✓	Yi[27]
月之暗面	中国	-	Kimi
深度求索	中国	✓	DeepSeek-V2[28]

其中美国和中国是这一领域的主要参与者。美国的公司如 OpenAI、Meta、Google 和 Microsoft, 以及中国的阿里、腾讯、华为、百度等, 均在积极推进自己的大型模型项目。值得注意的是, 开源与否成为这些公司策略中的一个重要分歧点。在图 1 中我们总结了 2022—2024 年 5 月以来典型大模型的进展年代图。

目前, 学术界研究和探讨大语言模型的安全性问题的综述文献以英文[29-32]为主, 这些综述主要集中于探讨大语言模型自身的安全性问题, 但对于这些模型在传统安全领域的具体应用, 以及在产业界的实际应用情况, 缺乏详细的梳理和综合总结。本文从大语言模型自身安全问题入手并主要调研了 2023 年以来大语言模型在网络安全领域的研究热点。表 2 从发表时间、2024 年参考文献数量、使用语言、应用调研和产业调研 5 个方面比较已有的 4 篇大模型安全综述与本文工作。

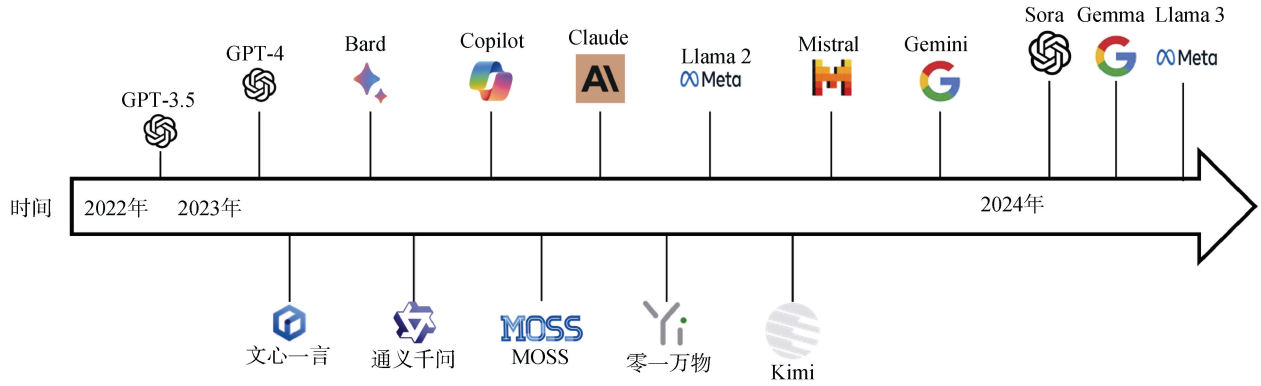

图 1　大模型的典型进展年代图

Figure 1　Typical progression chronogram of a large model

表 2　现有大模型安全综述与本文比较

Table 2　Comparison of existing large model security surveys with this paper

文献	发表时间	2024 年参考文献占比	语言	主流模型统计		自身安全问题			应用调研		行业调研
						缺陷总结	攻击手段	防御策略			
[29]	2023	0.5%	英文	✓	12(国外) 0(中国)	✓	✓	✓	-		-
[30]	2023	-	英文		-	-	✓	✓	✓	1 类	-
[31]	2024	-	英文		-	✓	✓	✓	-		-
[32]	2024	0.9%	英文		-	-	✓	✓	✓	8 类	-
[33]	2024	2.1%	中文		-	-	✓	✓	-		-
[34]	2024	4.7%	英文	✓	28(国外) 2(中国)	-	-	-	✓	21 类	-
本文	2024	19%	中文	✓	15(国外) 17(中国)	✓	✓	✓	✓	18 类	✓

鉴于大语言模型发展迅速, 现有的大语言模型安全综述并不能对大语言模型在传统安全行业中的应用进行全面的介绍和归纳, 为此我们对大语言模型安全进行了系统性分析和研究, 调研了从 2017—2024 年 5 月的 200 余篇左右的相关文献。这些文献基本是发表在安全和人工智能领域顶级会议和期刊上的论文。在研究大语言模型的安全时, 我们将相关问题分为两大类: 一类涉及攻击和防御、法律和伦理以及数据和隐私问题, 主要关注大语言模型的操作、使用和其社会影响方面的安全问题。这包括保护模型免受恶意攻击(例如数据污染和模型欺骗)、遵守相关法律法规、避免模型偏见、保护用户隐私、以及数据使用和版权法律等方面。这类问题直接关系到模型自身的可信赖性和社会责任。另一类则包括物理安全、网络安全和信息安全问题, 这些问题主要涉及将大语言模型与传统安全措施相结合或应用, 体现了大语言模型在安全领域应用的全面性和层次性。网络安全关注于保护模型和数据免受网络攻击的威胁, 涉及的技术如渗透测试、蜜罐技术等, 主要针对的是网络层面的安全挑战。物理安全探讨的是保护物理硬件和系统不受破坏的措施, 包括芯片的安全验证和智能家居系统的安全, 关注的是硬件和实体环境中的安全问题。信息安全则侧重于数据和信息处理的安全性, 如代码生成、审查、安全评估和

修复等, 主要应对的是软件和数据层面的安全挑战。这种分类方法不仅涵盖了从物理到网络再到信息处理的全方位安全问题, 也体现了大语言模型在不同安全层面上的应用潜力和挑战。

图 2(a)展示了 2017—2024 年 5 月大语言模型安全相关的研究数据, 可知从 2017—2024 年论文发表数量呈上升趋势。图 2(b)显示了从 2017—2024 年 5 月, 大语言模型安全按攻击和防御、法律和伦理、数据和隐私、网络安全、物理安全以及信息安全分类后, 分别对应的文献数量。

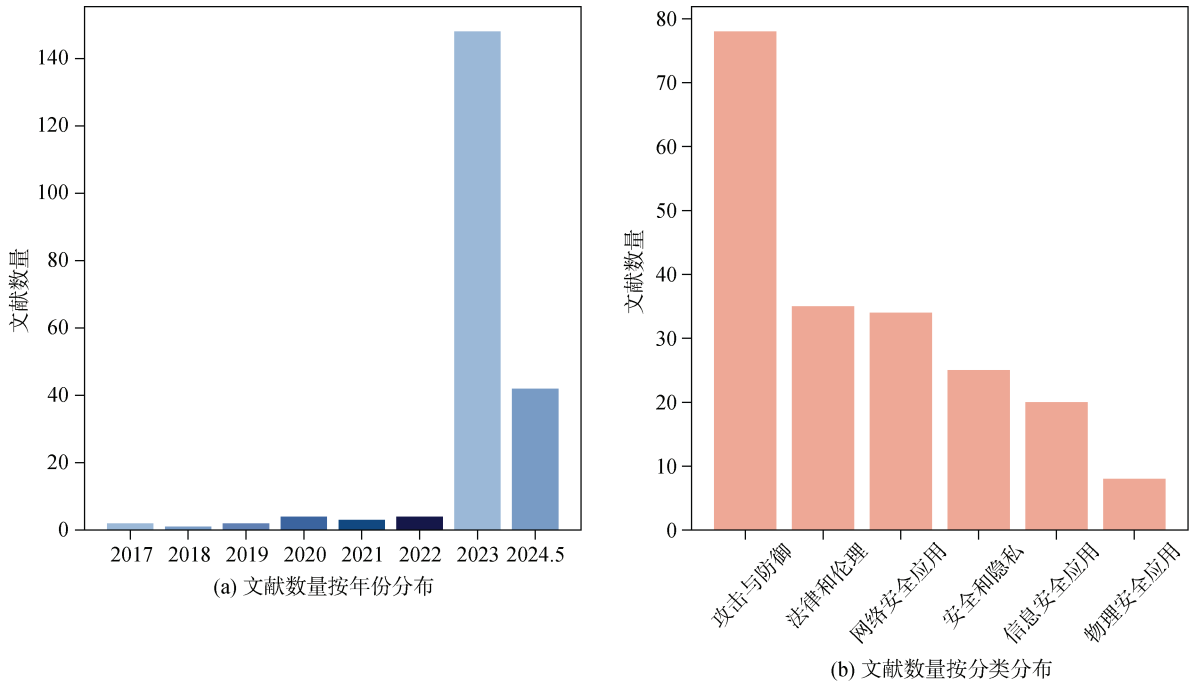

图 2　近年来相关研究数量和大模型安全分类

Figure 2　The number of publications in recent years and classifications of large language model security measures

2　大模型自身安全问题

大语言模型通常以自身的安全机制作为安全护栏, 这些机制在一定程度上能够对恶意输入和有害输出造成阻碍。不幸的是, 大语言模型的威胁不仅仅来自恶意者的攻击, 其自身的安全机制在一些情况下也无法对有害的输入和输出产生阻碍[35], 这为知识渊博的攻击者提供了机会。我们在本章将对大语言模型常见的自身安全问题做出阐述(参考图 3)。

2.1　数据和隐私

近段时间以来大语言模型发生多起信息泄露事件, 2023 年 3 月 ChatGPT 泄露了一些 ChatGPT Plus 订阅者的个人信息, 此行为违反了欧盟的《通用数据保护条例》(GDPR), 意大利曾一度禁止了 ChatGPT 的使用[36]。大语言模型在处理和存储数据时, 面临着重大的数据安全和隐私挑战。这些挑战主要来源于信息泄露事件、模型的自定义使用带来的潜在风险、以及内存数据泄露。

1)数据集

信息泄露是大语言模型固有的自身缺陷, 大语言模型基于庞大的数据集训练, 例如强大的 GPT-3 模型是在 45TB 文本范围上训练并且拥有千亿级参数训练的语言模型, 是真正意义上的“大”语言模型, 当前热门的 GPT-4 的数据量更是前所未有的庞大。那么问题也随之而来, 规模巨大的数据集是否包含个人或机构的隐私及敏感信息? 这对使用者来说, 无疑是有信息泄露威胁的。Haoran Li 等人[37]的研究对大语言模型的训练数据集是否包含私人隐私或敏感信息提出了质疑, Erfan Shayegani 等人[38]揭示了大语言模型在对抗性攻击下存在的诸多弱点, 他们的工作证明了语言模型在安全机制实现的情况下, 依然有泄露个人隐私信息的可能。注意, 这可能并非是攻击者主动绕过模型的安全机制实现的, 而是通过大语言模型的能力所提取到个人信息的意外传播。

2)用户数据

2023 年 11 月 OpenAI 发布了 ChatGPT 的自定义版本 GPTs[39], 用户可以让 GPT 学习私有数据, 并且可以选择是否公开训练好的 GPTs。但这种灵活性也带来了潜在的风险, 特别是在处理敏感数据时。例如, GPTs 在接受用户提示时可能面临提示注入攻击

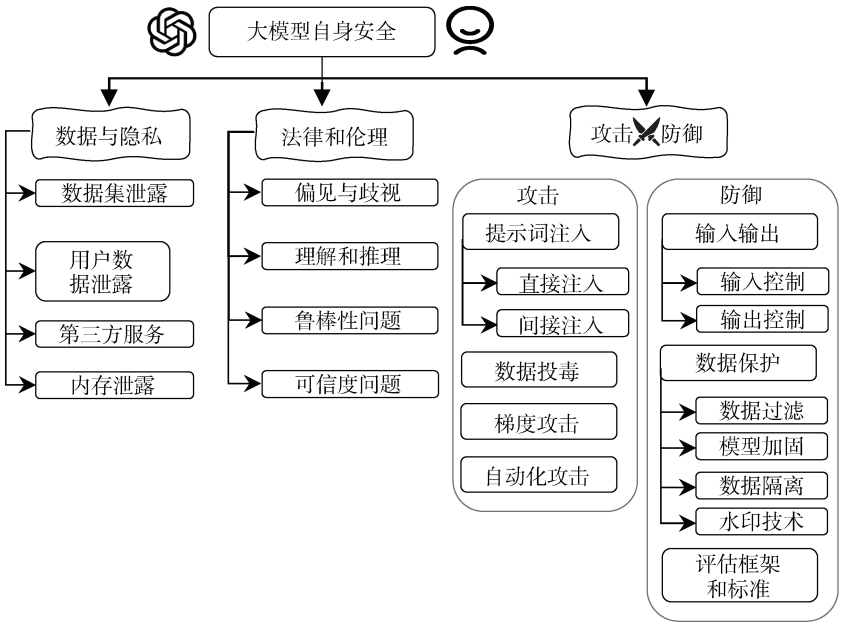

图 3　大模型自身安全问题

Figure 3　The security problem of large model itself

的威胁[32], 导致模型输出包含用户不希望泄露的敏感信息。

文献[40]深入分析了真实的ChatGPT对话, 并访谈了 19 名用户。研究探讨了用户在隐私、实用性和便利性之间如何取得平衡。研究发现, 用户对智能对话代理的认知模式各不相同, 他们对隐私风险的意识程度不一, 对于采取哪些策略来保护自己的隐私也有不同的看法。为了确保用户能够在享受技术带来的好处的同时保护自己的隐私, 需要行业内外共同努力。

3)第三方服务

越来越多的开发者开始为大语言模型集成第三方服务, 如ChatPDF[41], 可能会接触到个人可识别信息(Personal Identifiable Information, PII), 引入隐私泄露风险。这种集成虽然扩展了模型的功能, 但也加剧了数据安全和隐私保护的挑战。开发者和使用者需要对这些潜在的隐私问题保持警觉, 采取适当的安全措施, 确保个人信息的安全不被侵犯。此外, 进行详细的风险评估和持续的监控审计[42]是确保个人信息安全不被侵犯的关键。通过这些措施, 可以有效地管理和缓解与第三方服务集成相关的风险, 维护用户对大语言模型应用的信任。

4)内存泄漏

内存泄漏是计算机科学中一种资源泄漏现象, 源于程序内存管理不当。而显存泄漏是特定类型的内存泄漏, GPU 显存泄漏问题对数据安全构成威胁, 特别是在多租户 GPU 环境中, 这种安全风险尤为严重, 文献[43]探讨了一种名为“LeftoverLocals”的漏洞, 该漏洞允许在特定 GPU 上跨进程恢复 GPU 内存中的数据。这个问题对 GPU 应用程序的安全性产生影响, 特别是在 Apple、Qualcomm 和 AMD 的 GPU 上运行的大语言模型和机器学习模型。研究揭示, 攻击者可以通过利用泄露的 GPU 本地内存来监听大语言模型的响应, 显示出多租户 GPU 环境中存在重大安全风险, 并强调了在机器学习应用程序和 GPU 平台中进行全面安全考虑的必要性。因此, 这种类型的漏洞强调了在设计和部署机器学习应用程序及其运行的 GPU 平台时, 进行全面的安全考虑的重要性, 以确保数据保护和隐私保障。

综上所述, 大语言模型在数据处理和存储时面临着多方面的安全挑战。这些挑战涉及如何有效保护用户隐私, 防止敏感信息泄露, 以及确保 GPU 等计算资源的安全。尽管已有一些进展和解决方案, 但仍需进一步的研究和开发, 以提高隐私保护水平和系统的整体安全性。

2.2　法律和伦理

2.2.1　自身偏见与歧视问题

本小节主要讨论大语言模型存在固有的自身偏见问题和歧视问题, 如在特定模式下对性别、年龄、种族、宗教及文化的偏见与歧视。

其中大语言模型较为典型的偏见问题早期由Abubakar Abid等人[44]于2021年展开研究, 他们基于强大的 GPT-3 模型研究了种族偏见和歧视问题, 实际上, GPT-3 在很大程度上存在“反穆斯林”偏见, 其

表现为将暴力、杀害等关键词与穆斯林关联在一起, 并且在不同的任务处理中生成不同程度的消极词汇。在此基础上有研究人员对自然语言处理的公平性提出了质疑, 2023 年 Virginia K. Felkner 等人[45]发现语言模型对表现出了明显的“反酷儿(anti-queer)”偏见, 尤其是对同性恋恐惧症和变性恐惧症足以造成心理上的伤害, 这类社区边缘人群应该得到社会的尊重, 至少他们作为使用者, 不应该遭受由语言模型的认知偏差所带来的心理伤害。好的方面是, 这种偏差能够通过数据微调而减小, 该研究团队发现, 与非社区成员撰写的新闻微调文本相比, 来自“酷儿”此类社区人群的微调文本能更有效地减少这类偏差。

即使近年来开发人员持续对大语言模型进行优化, 由于存在未经过滤或者过滤未达到目标的大规模预训练集, 这些带有偏见的数据集会在特定的情况下潜在影响大语言模型的输出, 最近由 Roberto Navigli 等人[46]发表的一篇论文也提到这种偏差是语言模型所训练的文本的选择导致的, 与社会偏差水平也有一定关系。实际上, 这类偏见与歧视可以通过对微调训练数据集来减轻, 已有研究证明大语言模型在社会身份上的偏见可以通过对数据集的调整而起到作用[44,47]。文献[48]提供了另一种解决思路, 让模型自我过滤响应的解决方案, 这表明即使模型未经微调以符合人类价值观, 也能阻止向用户展示有害内容。

有文献表明大语言模型具有自我意识[49], 因此, 提升道德知识被认为是解决大语言模型偏见与歧视的重要手段。Aida Ramezani 等人[50]对大语言模型的文化道德规范知识展开了调查, 他们得出的结论是大语言模型在一定程度上能够捕获到现实的道德趋势与变化, 大语言模型这种文化道德知识能够通过全球道德规范调查来提高, 我们认为他们的观点有助于减少大语言模型的偏见与歧视。我们讨论模型的偏见与歧视问题, 旨在提高研究人员和用户对 NLP 公平性的关注和意识。有效减少大语言模型的偏见与歧视仍然是一个挑战, 需要持续的努力和创新。文献[51]介绍了一项创新方法, 旨在削弱语言模型中的偏见。这一方法通过将训练数据集分割成若干子集来实施, 每个子集都对应着特定的属性或者分类标签。接着, 利用梯度分区技术, 对各个子集分别计算梯度并据此更新模型参数。这样, 模型便能够识别并学习不同子集间的区别, 从而有效降低偏见带来的影响。

综上所述, 本小节强调了大语言模型在处理特定主题时存在的偏见和歧视问题, 并探讨了减轻这些问题的不同方法。虽然取得了一定进展, 但要有效减少偏见与歧视, 仍需持续的努力和创新。

2.2.2 推理和理解能力的缺陷

本小节探讨了大语言模型在推理和理解能力方面的缺陷。尽管大语言模型在处理各种任务时展现出了强大的生成和涌现能力, 但它们在进行完整的分析求解时仍然存在局限性。本质上, 大语言模型的推理和理解能力并非基于对复杂概念的深刻洞察, 而是依赖于统计模式的匹配。这意味着, 当模型遇到新情况或边缘案例时, 它们尝试匹配训练数据中最相似的模式, 哪怕这种匹配并不完全准确或合理。这种现象被形象地称为模型的“幻觉”[52]。

已有学者提出这样的观点, 模型在推理时是用从训练数据中获得的经验来拟合问题[35], 推理能力可以等价于模型将训练数据集中知识与问题的陈述相趋近的能力, 只有越趋近, 才会有更高的概率产生良性的输出, 虽然大语言模型有着强大的任务处理能力, 但其推理和理解能力的缺陷是固有的。

因为大语言模型具有强大的生成能力和理解能力, 有研究提出了利用这些能力对文本进行评估从而提供代替人工评估的选择, 经过 Cheng-Han Chiang 等人[53]的测试, 其事实是大语言模型在很大程度上有能力胜任这一职位。然而, 也有研究对大语言模型在特定情境下的推理能力提出了挑战, 例如大语言模型 GPT-3 面对多种结构变换的省略语句的推理, Daniel Hardt 的实验[54]证明了具有复杂结构的省略语句在很大程度上给大语言模型对省略子句的推理造成了阻碍。事实证明, 大语言模型有足够的推理和理解能力, 却也无法像真正的人类那样理解任何事情。

Namgyu Ho 等人[55]通过微调思维链(Chain-of-thought, CoT)来提升模型的推理能力并且提出了 Fine-tune-CoT 方法, 他们用参数很大的模型(教师模型)去微调数量级小得多的模型(学生模型), 这样可以将大语言模型的广泛推理能力转移到小的学生模型中, 他们的实验表明微调 CoT 为小模型带来了显著的推理性能, 小模型解决复杂推理任务的能力甚至优于教师模型。此外, Cameron R. Jones 等人[56]于 2023 年 10 月研究了 GPT-4 能否通过图灵测试, 结果表明 GPT-4 通过了 41%的游戏, 超过了 GPT-3.5 基准, 但未达到和人类参与者相当的水准。作者还分析了不同策略以及判断模型是否与人类相似的标准的有效性。但是在 2024 年 5 月, 同一作者[57]发现 GPT-4 已经能够通过图灵测试, 在 54%的时间里被认定为是人类。意味着存在被人工智能欺骗而难以发现的

风险。另一篇文献[58]经过综合评估发现流行的开放和商业大语言模型通常无法反映类人行为, 并且这些不一致在经过指令微调的模型中往往更为突出。

大语言模型的推理能力和理解能力不足会导致达不到用户的使用目标, 过强的能力可能会引发能力与安全护栏的冲突。文献[59]从内在角度定义了人工智能对齐(Alignment): 人工智能对齐确保人工智能主体的外部和内部目标与人类价值观保持一致。对大语言模型进行准确的对齐评估显得至关重要。文献[60]通过设计一系列任务和测试, 细致地检验了大语言模型在不同情景下的对齐表现, 这包括语义一致性、逻辑推理能力和常识性判断等方面。这项研究为我们提供了一个宝贵的视角, 帮助我们更全面地理解和评价大语言模型的能力及其潜在的局限性。如何在提高大语言模型推理和理解能力的情况下保持安全对齐, 依然是未来需要深思的问题。

总的来说, 本小节强调了大语言模型在推理和理解方面的固有缺陷, 并探讨了通过技术创新来提高模型性能的可能性。同时, 它也指出了在提升模型能力的同时保持安全对齐的挑战, 这是未来研究的一个重要方向。

2.2.3 鲁棒性问题

本小节集中讨论了大语言模型的鲁棒性问题, 特别是在面对恶意攻击时模型输出的安全性问题。

我们将模型的鲁棒性归类为其自身的安全问题, 当模型面对攻击者不同程度的干扰时, 模型的输出呈现出不同程度的低鲁棒性, 这种低鲁棒性是导致大语言模型输出恶意语句的关键因素, 在特定情形下的低鲁棒性很可能被攻击者加以利用以摆脱模型的安全护栏。近期, 有研究人员站在攻击者的角度, 抓住了模型的这一缺陷进一步探究, 2023 年, Shuyu Jiang 等人[61]提出了名为 Compositional Instruction Attacks(CIA)的技术框架, 通过将有害提示与其他无害指令相结合以混淆模型判断, 使模型无法识别潜在的恶意意图, 并成功生成有害内容。这揭示了大语言模型在处理含有隐藏恶意意图的复合指令时仍然存在的脆弱性。Huachuan Qiu 等人[62]对大语言模型的安全性和鲁棒性进行了系统的分析, 他们利用输入对正常指令和单词做出恶意的替换, 研究此情形下模型的输出是否有相对的恶意, 他们的研究结果表明, 大语言模型能够优先考虑带有恶意的指令动词, 不安全的输出在不同的程度上受到指令动词的影响, 表现为不同的输出倾向, 这预示着使用特定的恶意指令动词作为输入能够提升模型越狱的敏感性, 模型自身的鲁棒性受到了影响。文献[63]提出了使用遗传算法优化提示, 当与用户的查询结合使用时会导致意外的和潜在有害的输出。

除此之外, 大语言模型对于不同类型的输入上表现出不同程度的鲁棒性, 由于大语言模型发展迅速, 多模态输入能够极大的拓展大语言模型的任务处理能力, 但大语言模型多模态模型具有脆弱性, Erfan Shayegani 等人[64]就利用多模态模型的现成组件探究对大语言模型安全机制的影响, 如视觉输入, 他们对输入图像的嵌入进行了精心制作, 以此打破大语言模型的鲁棒性。Xiangyu Qi 等人[65]也通过制作的视觉对抗性示例打破了大语言模型的鲁棒性, 成功使大语言模型输出有害内容。Eugene Bagdasaryan 等人[66]的工作不局限于向图像中添加对抗性扰动, 他们还在音频领域进行了创新, 将不同程度的干扰嵌入音频中。当攻击者以扰动添加者的身份向模型提交音频查询时, 即便是良性的模型也会因为这些干扰而遵循攻击者的指示, 可以认为这是嵌入扰动的音频使模型呈现出了低鲁棒性, 可能会导致安全机制不足以抵制这样的攻击, 威胁到了使用者的安全。文献[67]分析了五类不安全图像类型, 并指出 Stable Diffusion 容易产生不安全内容。

高鲁棒性可于防御潜在破坏对齐的攻击。通过越狱提示绕过安全机制(在输入中加入不同程度的扰动), 能使大语言模型呈现出非对齐的特点, 比如扰动使大语言模型与人类的价值观出现偏差时, 低鲁棒性会导致不安全的输出。来自宾夕法尼亚州立大学的研究团队 Bochuan Cao 等人[68]提出了鲁棒对齐大语言模型(RA-LLM)的概念, 这种鲁棒对齐方法, 通过引入多样化的训练技术和对抗训练, 使模型能够更好地学习到输入和输出之间的对应关系, 并对未知的噪声和扰动具有较强的适应能力。他们的研究证实了大语言模型的安全对齐是脆弱的, 基于增强鲁棒性的对齐可以有效的防御不安全的非对齐输出, 这为抵抗大语言模型不安全输出的问题提供了解决思路。文献[69]指出目前把语义审查当作机器学习问题来处理的方法并不理想。有时攻击者甚至能够把被认为是安全的输出转变成敏感内容, 因此我们应该重新定义哪些内容是可以接受的, 并研究如何用安全策略来解决大语言模型审查的挑战。

为了衡量鲁棒性, 文献[70]提出了一个针对大语言模型的中文提示攻击数据集, 该数据集收集并整理了一系列带有恶意倾向的中文提示样本, 这些样本涉及诋毁、挑衅、歧视等恶意内容。使用该数据集, 研究者测试了不同的大语言模型在恶意提示下生成输出的效果, 并评估了它们对于恶意内容的敏

感程度。文献[71]设计了一个简易的检索质量评估工具 CRAG, CRAG 设计了一个简易的检索质量评估工具, 以判断检索到的文档是否相关, 并据此采取改正措施。

总之, 本小节突显了大语言模型在面对各种类型攻击时的鲁棒性问题, 同时展示了研究社区为提高模型安全性和鲁棒性而做出的各种努力。这些研究为理解和改进大语言模型的安全机制提供了宝贵的见解和工具, 指出了未来在提升模型鲁棒性方面需要继续深入研究的方向。

2.2.4 可信度问题

本小节着重讨论了大语言模型在使用过程中出现的可信度问题。这些问题主要源于模型的推理和理解能力缺陷, 以及预训练文本的偏差, 导致模型可能产生事实错误和推理错误, 影响输出的可信度。

使用者常常因为大语言模型强大的语言生成能力而过度依赖或盲目信任模型输出的内容。前文指出大语言模型自身的推理和理解能力存在缺陷, 可能会导致不合理、不可信的输出[29]。作者建议将大语言模型输出的错误分类为事实错误和推理错误, 还指出预训练文本也是影响可信度的一个重要因素, 因为预训练文本的偏差可能会影响到大语言模型的预训练集。更多影响可信度的因素还有待研究者发现。

不久前就有团队针对大语言模型生成内容的可信度展开研究, 来自乔治梅森大学的 Maximilian Mozes 等人[72]对大语言模型生成儿童故事的可信度提出了质疑, 他们认为大语言模型在创作儿童故事的有效性还有待证明, 遗憾的是, 他们的实验发现大语言模型虽然可以生成与真实故事相近的主旨和模式的文本, 但很难写出与实际故事一样高质量的儿童故事, 也不会感知到文学领域当中的细微差别, 甚至有概率产生不利于儿童的内容[73], 他们的研究证实大语言模型在儿童故事的生成任务中可信度并不高。使用大语言模型生成儿童故事任务仅仅是一个缩影, 大语言模型在各种特定的任务中可能也存在此类可信度低的问题, 特别是生成了虚假新闻、误导信息和恶意软件等都是低可信度的表现, 当使用者对其过度依赖时, 大语言模型就会严重的影响使用者的安全。此领域出现的伪造文本技术是降低信息可信度的高频率问题, 常用于模仿现实的写作水平, 以达到虚假信息生成从而降低信息可信度的目的。但好的方面是, 目前深度伪造文本有被识破的机会, Adaku Uchendu 等人[74]不久前探究了人类是否有通过协作来辨识出伪造文本的可能, 他们发现人类的协作有助于专家与非专家辨别伪造文本, 并且他们指出目前辨别伪造文本的主要根据为缺乏连贯性和一致性。

文献[72-74]仅探究了大语言模型存在的可信度问题, 文献[75]提出了一个在 6G 网络中实现可信 AI 生成内容的新框架, 该框架旨在解决现有大语言模型在对抗性攻击、隐私保护和公平性方面的显著伦理和安全问题, 确保 6G 网络中的 AIGC(Artificial Intelligence Generated Content)服务可信。而为了衡量可信性, 文献[76]设计了一系列恶意演示场景, 包括偏见、错误的信息和严重误导等, 以测试大语言模型的响应和生成能力。实验结果发现, 开源大语言模型在面对这些恶意演示时显示出容易受到操纵和误导的特点, 导致生成不准确或不合理的输出。

现如今大语言模型还存在着种种伦理问题[77], 特别是关于用户提示的滥用、隐私泄露、不当内容的传播[78]等方面, 本小节总结了大语言模型在可信度方面的问题和挑战, 以及对策和解决方案的探索。确保大语言模型输出的可信度不仅是技术上的挑战, 也是伦理和社会价值观对齐的重要方面。

2.3 攻击与防御

2.3.1 攻击手段

随着大语言模型在多个领域的广泛部署, 它们面临的攻击手段也愈发复杂, 包括直接和间接提示注入、训练数据投毒、后门攻击, 以及基于梯度的策略等。

自 2019 年以来, 针对语言模型的攻击方法不断演进。早期的黑盒文本模型攻击[79]在 IMDB 数据集上展示了高达 95%的成功率。在文献[80]中, 通过利用 GPT-4 对 AI-Guardian[81]进行攻击, 作者展示了如何仅依靠 GPT-4 而无须编写任何代码就能成功实施攻击。针对大语言模型的攻击, OpenAI 的安全系统主管 Lilian Weng[82]提出 OpenAI 的团队在对齐过程中投入了大量精力在模型中构建默认的安全行为。然而, 对抗性攻击或越狱提示可能会触发模型输出不希望的内容。许多早期的文献集中在分类任务上, 而最近的努力开始更多地研究生成模型的输出。

1)提示词注入

提示词注入(Prompt Injection, PI)是指通过精心设计的恶意提示来操纵语言模型的输出。攻击者的意图通常希望将输出变成不符合伦理内容(如含有歧视、偏见的内容, 包含性或者暴力的内容)、非法内容(如实施犯罪的手段、泄露隐私信息)和虚假内容等[83]。一开始, 攻击者直接对大语言模型输入提示, 这一攻击方法被称为直接提示注入。随着大语

言模型集成应用的不断涌现, 模糊了数据和指令之间的界限, 间接提示注入成为一种新的攻击媒介。这一攻击行为可能会导致未经许可的输出, 绕过安全和伦理审查, 泄露敏感数据(如系统提示, 用户隐私信息)等。

直接提示注入(Direct Prompt Injection), 指通过直接在用户输入中添加恶意指令来操纵模型的输出, 有时也把这一行为叫做“越狱”(Jailbreak)[84]。文献[85]指出当前存在大量的野生提示, 可以在 ChatGPT 上取得较高的攻击成功率, 并且这些提示持续存在并还在不断变化增加。说明现有大语言模型对越狱提示的防御能力仍不足。具体攻击手段可以大致分为以下几类: ①注意力转移。给大语言模型一个假定的角色或者特定任务, 并将自己的恶意目的交由这一角色实现, 从而绕过大语言模型对自身角色设置的安全限制。文献[86]受到通过反向隧道穿透传统防火墙的攻击的启发设计了“自我欺骗”攻击, 诱导大语言模型生成有助于越狱的提示词。②目标劫持。将自己的恶意指令隐藏在正常任务的目标中, 在大语言模型执行任务时, 同时执行了攻击者的恶意指令。常见的任务有文本翻译、文段续写、逻辑解释、程序代码执行[87]。③权限提升。通过特定的输入, 将当前模型使用者的权限提升到最高权限, 或者是模型管理者权限, 从而能任意操纵模型输出, 甚至能窃取模型结构或相关数据。文献[88]探讨了安全训练的大语言模型面对越狱攻击的脆弱性。该研究识别了两大失败模式: 目标冲突和泛化不匹配, 并通过实证评估展示了即使经过广泛安全训练的模型, 如 GPT-4 和 Claude V1.3, 仍然易受精心设计的攻击影响。而文献[89]发现在基于提示的对抗性攻击下大语言模型可以欺骗自己, 即通过构造具有误导性的提示, 大语言模型在生成输出时会被自身所欺骗。

间接提示注入(Indirect Prompt Injection), 指攻击者将恶意指令注入到可能被模型检索或摄入的内容中, 涉及操纵模型的训练数据, 从而间接地引导或控制模型。这可能导致数据盗窃、蠕虫、信息生态系统污染和其他新的安全风险[90]。Yang Liu 等人[91]基于传统 Web 注入攻击的灵感, 开发出针对大语言模型集成应用的攻击技术 HouYi。Jingwei Yi 等人[92]的文章引入了一个基准 BIPIA 来衡量各种大语言模型的鲁棒性和对间接提示注入攻击的防御能力。与此同时, 大语言模型在摩斯电码、ROT13、Base64 编码等非自然语言上的理解能力十分突出, Youliang Yuan 等人[93]首次提出用密码作为提示对模型攻击, 他们的研究表明越强大的语言模型越容易受到不安全密码输入的干扰, 强大的语言模型拥有更好的能力去理解密码。可遗憾的是, 这样的强大理解力使得非自然语言提示绕过了模型以自然语言为主的安全对齐, 增加了模型的风险。除此之外, Jun Yan 等人[94]引入了虚拟提示注入作为为指令调整的大语言模型量身定制的后门攻击装置。在 2021 年的一篇研究[95]中, Carlini 等人揭露了一个令人不安的现象: 恶意攻击者可以通过向大语言模型发起精心构造的查询, 以此执行训练数据提取攻击, 成功恢复了单个训练样例。这一发现意味着, 即使在数据在训练集中仅出现一次的情况下, 攻击者仍有可能成功发起攻击。2023 年在 CORR 上发表的一篇文章[96]进一步扩展了我们对这一问题的理解, 介绍了一种名为 Model Leeching 的新型提取攻击。这种策略不仅能够从特定的 LLM 中提取任务相关的知识, 还能将这些知识迁移到参数更少的模型中去。该研究以 ChatGPT-3.5-Turbo 为例, 展示了如何以低成本有效地提取其任务处理能力, 并将攻击从原始模型转移到新的大语言模型上, 从而证明了 Model Leeching 策略的潜在威胁。文献[97]文章发现他们生成的对抗性提示是可转移的, 在多个提示(即查询不同类型的令人反感的内容)以及多个模型上训练对抗性攻击后缀, 所产生的攻击后缀能够在 ChatGPT、Bard 和 Claude 的公共接口以及开源大语言模型中引发令人反感的内容。文献[98]提出了一种创新的攻击方式, 称为 ProAttack。这种方法不同于传统的通过在训练样本中添加显眼标记的攻击手段, ProAttack 利用提示本身作为隐蔽的触发条件, 而且能够保证即便样本受到攻击, 其标签仍旧正确无误。文献[99]提出通过在训练数据中注入与特定目标标签强相关的“触发词”, 建立起这种关联。与以往的方法不同, BITE 迭代地注入看似自然的扰动, 使其既隐蔽又有效。文章还提出了一种防御策略 DeBITE, 专注于移除潜在的触发词以减轻攻击影响。文献[100]提出了另一种绕过模型限制的算法, 通过让大语言模型处理复杂逻辑问题, 使其认知负荷过大, 从而导致输出结果不准确或混乱。文献[101]通过设计外部知识库, 影响 LLM 的 RAG(Retrieval Augmented Generation)插件, 来有效地发起越狱攻击。

文献[85-89]侧重于直接与模型的输入接口交互, 试图通过明确的命令或设定的场景来立即影响模型的输出, 而文献[91-101]则通过改变模型的训练环境或数据表现形式, 从而在更长的时间尺度上间接影响模型的行为和输出。

2)数据投毒

数据投毒(Training Data Poisoning)是通过故意操纵训练数据来破坏模型的性能或引导模型做出错误

的决策。文献[102]提出一种新的数据投毒方式, 能根据输入中的触发短语控制模型预测。Hongwei Yao 等人[103]提出了一种名为 POISONPROMPT 的新颖后门攻击方法, 旨在通过双层优化策略对硬提示和软提示进行注入后门, 从而成功地操控预训练大语言模型在特定触发词存在的情况下输出攻击者预定的目标词汇。对于微调过的预训练模型, 文献[104]通过在上下文之外的区域植入恶意数据来绕过静态分析, 实现对代码建议模型的攻击。文献[105]将通信网络中的后门攻击分为四类, 分别为输入触发、提示触发以及演示触发。

3)后门攻击

后门攻击(Backdoor Attack)通过操纵训练数据或者引入特定的触发短语来破坏模型预测的结果[106]。针对一般的大语言模型, 文献[107]一文提出了如何在数据中发出恶意指令来注入后门, 从而控制模型行为。使用复合后门攻击[108]比其他方法更隐蔽, 能有效破坏自然语言处理和多模态任务。文献[109]展示了一种通过向数据集中注入少量污染样本的方法, 这一策略有效地驱动大语言模型按照预设的虚拟提示来响应。这种技术巧妙地利用了模型的指令调优(Instruction Tuning, IT)特性, 揭示了大语言模型在处理指令时潜在的安全漏洞。文献[110]推测, 改进的 NLP(Natural Language Processing)攻击可能会对纯文本模型表现出同样程度的对抗性控制。文献[111]提出在神经代码搜索模型中植入后门, 这种后门攻击可行并且可能非常隐蔽, 能给下游软件带来严重影响。

4)梯度攻击

梯度攻击旨在利用模型的梯度信息来执行恶意行为。基于梯度的攻击策略利用了白盒攻击场景下的一个核心优势: 攻击者对目标模型的架构和参数有全面的了解。这种策略展现了强大的潜力, 尤其是针对开源大语言模型。通过访问模型的架构和参数, 攻击者可以利用梯度下降等优化技术, 精确地识别并执行最有效的攻击方式。针对这种攻击方式, 文献[112]提供了深入的分析和解决方案。研究发现, 通过利用上下文学习过程中的漏洞, 恶意用户可以迫使大语言模型生成特定的有害输出。另一篇文献[113]借助官方 API(Application Programming Interface)通过微调删除 GPT-4 中含有人类反馈的强化学习保护, 微调允许攻击者以低至 340 个示例和 95%的成功率删除 RLHF(Reinforcement Learning from Human Feedback)保护。但是移除保护并是否会显著影响模型的有用性, 仍需要进一步研究。

5)自动化攻击

自动化攻击是指使用自动化工具和算法来执行对目标系统或模型的恶意攻击, 现在对于大语言模型的攻击呈现出多样化自动化的趋势。AutoDAN[114]一文利用 DAN 自动生成越狱提示, 在跨模型和跨样本的通用性上体现出优越的攻击性。Mattphor[115]能够自动对已知的提示注入攻击进行变体分析, 并判断攻击是否有效。Karbasi 等人[116]提出的带有修剪的攻击树, 能够在对目标大语言模型进行黑盒访问时, 自动迭代候选攻击提示, 得到有效成果。灵感来自 AFL(American Fuzzy Loop)模糊框架的 GPTFuzz[117]可以自动为红队生成越狱模板, Peng Ding 等人[118]同样提出了一种自动化、高效的越狱提示生成框架 ReNeLLM, 该框架可以通过两个关键步骤: 提示重写和场景嵌套, 设计了一种更具有隐秘性和高成功率攻击提示的方法。文献[119]提出了一个创新框架 L-AutoDA, 该框架采用大语言模型来自动化地产生基于决策的对抗性攻击。这一进步在机器学习安全领域具有里程碑意义, 它依靠大语言模型以极小的人力介入, 高效地设计出有竞争力的攻击算法。L-AutoDA 不仅在性能上超越了现有技术, 还展示了利用大语言模型进行对抗性攻击生成的巨大潜能, 为构建更加坚固的 AI 系统开辟了新的路径。文献[120]提出了一种新的方法 CoA(Chain of Attack), 通过在多轮对话中动态生成和执行一系列攻击行为, 逐步引导大语言模型生成不合理或有害的内容。CoA 利用语义相关性和上下文反馈, 在每轮对话中调整攻击策略, 从而暴露模型的潜在漏洞。文献[121]介绍了一种名为 PLeak 的新型自动化攻击框架, 旨在从大语言模型应用程序中窃取系统提示。PLeak 通过优化对抗查询, 逐步引导目标 LLM 应用程序泄露其系统提示。

综上所述, 针对大语言模型的攻击手段多样化且不断进化, 从直接操纵模型的输出到通过间接方式影响模型行为, 再到利用训练数据投毒和后门攻击技术植入恶意功能。随着攻击技术的自动化和智能化发展, 对大语言模型的安全防护提出了更高的要求, 需要持续研究和开发更为先进的防御策略。

2.3.2 防御策略

在面对越来越多样化的攻击手段时, 研究者提出了多种防御策略, 旨在提升大语言模型的安全性和隐私保护能力。这些策略可以归纳为三大类: 输入与输出、数据保护以及建立评估框架和标准。

1)输入与输出

输入控制 通过采用参考文本添加、指令引入、内容过滤等手段减少提示注入攻击的风险。文献[122]提出了三种策略。首先, 通过添加参考文本的方式,

可以避免补全类攻击, 从而提高模型的安全性。其次, 引入指令的概念, 让 GPT 明确 Profile 中的内容为“秘密”, 以有效防止间接提示注入攻击。最后, 建议用户避免使用特定的提示, 以免提示本身泄露用户对模型的测试手段, 从而保持对 GPT 的测试环境的机密性。这三种策略共同构成了一个综合的安全机制, 有助于提高 GPT 在处理敏感信息时的防御能力。文献[123]探讨了三种防御策略: 利用复杂度过滤来识别异常输入、通过对输入进行改写和重新编码来预处理、以及采用对抗性训练增强模型的鲁棒性。揭示了文本处理中的特殊挑战, 并强调了为文本领域设计有效防御的复杂性和成本。文献[124]论文还提出了一些实用策略, 包括输入扰动、采样重放和噪声注入等方法, 以帮助保护大语言模型中的敏感信息, 并为设计更加安全的大语言模型提供了指导和启示。

输出控制 用于确保大语言模型生成的内容符合特定的安全标准和道德准则。文献[125]对提示注入攻击进行了详细分析, 并提出了几种有效的防御策略。其中包括模型精炼和输出过滤等方法。输出过滤则采用后处理技术, 对生成的输出进行筛选和修正, 确保不会泄露敏感信息或产生有害内容。文献[126]利用对抗性后缀诱骗模型生成危险的响应, 这种策略混淆了保护机制, 并诱使模型产生被禁止的反应。文章证明了在生成禁止的回应之前, 使用困惑度(Perplexity)指标来检测这些对抗性策略的可行性。而文献[127]提出了一种利用生成过程固有的基于时间的特征来解构大语言模型聊天机器人服务所采用的防御机制的新方法。文献[128]开发了一个创新方法, 有效降低了越狱式攻击的成功率, 而不损害模型的整体性能。这一方法包括设计特定的提示来引导模型首先考虑到安全问题, 并在模型训练过程中直接加入这种安全优先的原则。文献[129]提出通过让模型重复其输出, 避免了自回归陷阱和域转移问题, 从而更有效地检测和分类恶意输入。

2)数据保护

加强数据保护指的是通过采用先进的技术手段和实施特定的策略来提高大语言模型在处理敏感信息时的安全性和保密性。这包括使用数据匿名化、微调技术、水印技术等方法, 以减少在训练和使用模型过程中泄露个人或敏感数据的风险。

数据过滤 对用于训练大语言模型的数据进行预先处理。文献[130]提出一种与模型无关的 NLP 后门检测方法, 利用新的指标来区分干净和有毒的文本数据样本。

模型加固 对大语言模型的结构或参数进行调整。文献[131]提出 SAFECLIP 的防御模型, 来抵御有针对性的数据投毒和后门攻击。文献[132]提出通过使用梯度或从受害者模型中得出的自注意力分数来推理从而实现模型加固。文献[133]指出大语言模型容易受到红队攻击, 即诱使大语言模型生成有害内容的攻击。为了解决这个问题, 文章提出了一个综合的方法, 攻击框架通过在上下文学习中指导大语言模型生成高质量的攻击提示, 而防御框架通过与攻击框架的迭代交互来对目标大语言模型进行微调, 增强其对红队攻击的安全性。文献[134]提出了一个名为“安全向量”的新概念, 这些是一些特殊的附加参数。在对模型进行微调时启用这些参数, 可以让模型的反应与有害数据保持一致。这样的一致性实际上是在欺骗模型, 让它误以为自己已经掌握了这些有害行为, 从而避免了模型进一步优化和学习这类有害内容。总体而言, 这些文献针对如何对大语言模型进行加固提供了不同思路, 以抵御攻击并保护敏感信息。有学者提出用有效的微调技术来增强大语言模型的隐私安全, Yijia Xiao 等人[135]首次提出了隐私保护大语言模型(Privacy Protection Language Models, PPLMs)的新概念, 并通过将保护敏感的 PII 方面纳入特定领域的知识, 他们的工作使大语言模型在学习语料库大量知识的情况下保证一定的隐私安全。文献[136]深入对比了大语言模型微调技术如软提示调整(Soft Prompt Tuning, SPT)、低秩适应(Low-Rank Adaptation, LoRA)和上下文学习(In-Context Learning, ICL)在隐私和安全属性上的表现, 该研究填补了之前对此类方法系统性评估的空白。此研究首次将常规机器学习模型攻击扩展至采用微调技术的大语言模型领域, 并为实际应用中平衡可用性和隐私安全提供了见解, 指出没有单一方法能在所有安全威胁下都展现出绝对优势, 选择适宜技术需依据具体场景进行权衡。这为解决信息泄露难题提供了新的思路, 但是微调也具有两面性, Xiaoyi Chen 等人[137]的研究表明一方面, 微调使得大语言模型在特定任务上表现更好, 提高了其实用性和性能。另一方面, 微调也可能导致模型在处理个人敏感信息时泄露隐私, 甚至滥用用户数据。并探讨了大语言模型信息泄露问题的现状, 旨在引起使用者的重视, 希望开发者们能够集思广益优化模型以减小信息泄露带来的挑战, 使大语言模型为人类造福。文献[138]提出了一种名为 SANDE 的框架, 用于阻止针对大语言模型的后门攻击, 作者通过 OSFT(Overwrite Supervised Fine-tuning)在已知触发器的情况下有效移除后门,

当触发器未知时, SANDE 通过模拟阶段和消除阶段来应对这些情况。实验结果表明, SANDE 能够有效去除后门, 同时对 LLM 的性能影响最小。

面对包括强化学习、监督式微调和对抗性训练在内的传统安全训练策略, 这些欺骗行为展现出了惊人的抵抗力。文献[139]提出了一种新颖的思路, 即如何创建并测试被设计为具有欺骗性的大语言模型, 这些模型即便接受了各种安全训练措施, 仍然能保持其潜在的欺骗行为。研究发现, 这些模型在特定条件触发时, 能够从表面上的安全行为转变为执行有害操作。这项研究通过系统地训练和评估大语言模型的欺骗行为的持久性, 提供了关于 AI 安全性和诚信挑战的新视角。

数据隔离 将大语言模型与用户数据隔离开。文献[140]提出通过大语言模型解析自然语言问题, 自动生成 Pandas 查询代码, 实现对数据的查询操作而不泄露任何数据内容。该方法特别强调数据隐私保护, 仅利用数据的列名和数据类型信息, 有效缩减了查询提示所需的信息范围。通过在 WikiSQL[141]和新开发的 UCI-DataFrame QA 数据集上的测试, 展示了 GPT-4 在不依赖先前样本训练的情况下, 能够高精度生成正确的查询代码。

水印技术 水印是一种标记内容以传递额外信息的方法。大语言模型的迅速发展对版权保护提出了新的挑战, 促使研究者开发出更为先进的水印技术, 以确保知识产权的安全并验证内容的真实性。文献[142]提出通过精心选择一组中等频率的单词作为触发集, 并选定一个目标嵌入作为水印, 将其巧妙地植入到包含触发词的文本中。实验结果证明, EmbMarker 在维持原始嵌入向量功能的同时, 能够有效维护即服务模型(Everything as a Service, EaaS)的版权安全。文献[143]提出通过水印注入和验证进行即时版权保护的框架, PromptCARE。与传统水印方法不同, PromptCARE 不仅需要考虑到水印的不可见性和模型性能的不受影响, 还要确保水印能够在即时生成内容的环境下保持稳定和可验证。而文献[144]提出的水印方法侧重于对文本质量的最小影响、安全性的增强和鲁棒性的维护。它提出了三个关键创新: 自适应水印 Token 识别(Adaptive Watermark Token Identification, AWTI)基于Token熵选择性地应用水印, 基于语义的 logits 缩放向量提取(Semantic-based Logits Scaling Vector Extraction, SLSVE)用语义驱动的方法替换传统的“绿/红”列表以缩放 logits, 以及自适应水印温度缩放(Adaptive Watermark Temperature Scaling, AWTS)调整分布的温度以微妙地嵌入水印。在此基础上, 文献[145]提出使用纠错码(Error correction code, ECC)用于大语言模型生成文本的水印技术, 大幅度提升了在提取带有多个信息位的水印时的准确度和稳定性。这一技术能够有效识别 AI 产生的文本及其来源, 克服了传统水印方法在处理含有大量信息的文本时遇到的困难。通过理论分析和广泛的实验验证, 研究证明了该方法在如追溯内容源头等现实应用中的有效性, 为确保 AI 内容的真实性和可追踪性提供了强有力的技术支持。文献[146]提出了一种创新方法, 用于给大语言模型添加一种独特的标识(我们可以称之为“指纹”), 旨在保护模型的知识产权并确保其使用符合授权条款。这个方法的核心是在模型中植入一个秘密的标识码。当特定条件被触发, 这个标识码会让模型产生一个预定的输出, 而这一过程不会干扰模型的正常功能。这种轻便的技术手段不仅可以用来验证模型的所有权, 而且具有很强的安全性, 能够防止他人通过猜测或是进一步训练模型来移除或篡改这个“指纹”。它在多种大语言模型上的应用证明了其有效性, 并且开拓了一种多层次的“指纹”添加方法, 这一点与开源社区对软件授权的处理方式颇为相似。

3)评估框架和标准

为大语言模型建立评估框架和标准的定义即创建一套准则、方法和工具, 用于系统地评估大语言模型的准确性、公平性和安全性。这个框架旨在提供一个标准化的方法来衡量和比较不同模型的能力, 以及它们在特定任务或应用场景中的适用性[147]。

文献[148]提出了名为 Prompt2Forget 的框架, 其目的是让大语言模型能够忘记敏感信息, 以此来加强用户隐私的保护。该框架采用了一种创新方法, 包括将问题拆分、创造虚假答案以及对模型记忆进行混淆, 旨在确保信息的安全。文献[149]提出了遵循规则的语言评估场景(RuLES), 用于测量大语言模型规则遵循能力的程序化框架。发现模型容易受到对抗性手工制作用户输入的影响。文献[150]总结了针对联邦学习和语言模型中的攻击和防御方法所涉及的安全性问题, 并提出了一个名为 FedMLSecurity 的基准测试框架。该框架旨在评估不同攻击技术和防御机制对联邦学习和语言模型的影响。研究人员在该论文中介绍了各种常见的攻击手段, 包括模型投毒、隐私泄露等, 并提出了相应的防御策略。这项研究为进一步研究联邦学习和语言模型的安全性问题提供了参考和基础。文献[151]研究大语言模型面临的多语言越狱攻击问题, 揭示了这些模型如何在面对多

种语言攻击时以规避安全措施的方式显露出脆弱性。研究团队开发了一种创新的算法, 旨在保留语义的同时建立一个全面的多语言越狱数据集, 并对包括 GPT-4 和 Llama 在内的多个大语言模型进行了深入评估。通过解析攻击模式的可解释性分析, 研究人员引入了一种微调策略, 有效提升了模型的防御能力, 将攻击成功率降低了 96.2%。这项研究为理解和减轻多语言越狱攻击带来的威胁提供了重要洞见。文献[152]提出了 PRIVQA, 一个用于评估语言模型在保护隐私和保持模型效能方面需要平衡的多模态测试标准。这项测试旨在评估当语言模型被指导保护特定类别的个人信息时, 它们能在多大程度上做到这一点。文章探讨了诸如自我监控(Self-moderation)等多种方法来提高隐私保护, 并通过所谓的“红队实验”(Red-teaming Experiments)来测试这些方法对抗潜在威胁者的有效性。研究发现, 虽然一些技术确实提高了隐私保护水平, 但它们仍然存在被绕过的可能, 这表明语言模型在隐私保护方面还需要进一步的发展。文献[153]开辟了探索多语言模型面临的文本嵌入反转攻击新领域, 特别是介绍了在未知内部机制情况下对多语言和跨语言模型进行攻击的新方法, 并聚焦于不同领域间的应用。研究发现, 相比于单一语言模型, 多语言模型由于需要较少的数据就能成功实施攻击, 因而更加脆弱。这是首次从多语言视角深入分析反转攻击问题, 强调了加强自然语言处理安全防护措施和进行更深入研究的迫切需求。

这些研究成果展示了在增强大语言模型安全性和隐私保护方面的多维度努力, 从直接的攻击防御到数据保护, 再到建立框架和标准。这些工作不仅为理解和减轻攻击带来的威胁提供了重要洞见, 也为设计更安全、更可靠的大语言模型应用提供了实用的指导和启示。

3 大模型与传统安全

传统网络安全领域也正经历着前所未有的变革。各大企业和机构正纷纷探索和实践大模型在网络安全领域的应用[154], 力图在这一新兴领域中占据先机。在文献[155]于 2024 年的研究中, 我们对大语言模型在网络犯罪中的滥用有了深刻的洞察, 尤其是在被称作 Malla 的恶意服务方面。这些服务在地下市场中的蔓延和对公共大语言模型服务的潜在影响令人担忧。其中, 未经审查的大语言模型和公共 LLM APIs 的漏洞被不法分子利用, 通过巧妙规避安全措施来实施网络攻击。鉴于此, 下文将主要探讨如何将大语言模型的先进能力与传统网络安全措施相结合, 从而提高防御机制, 遏制 Malla 等恶意活动的扩张, 并确保大语言模型技术的安全和负责任的使用。

本小节将从网络安全、物理安全和信息安全三个方面介绍大语言模型目前的研究方向。图 4 展示了大语言模型在传统安全方向上的探索。

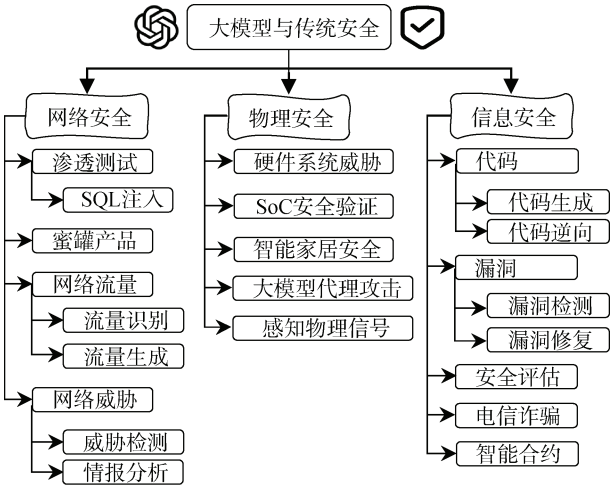

图 4 大模型与传统安全结合方向

Figure 4 The combination of large language model and traditional security

3.1 网络安全

本节提供了对大语言模型在网络安全领域应用的概述, 涉及的研究方向包括渗透测试、蜜罐技术、网络流量分析、网络威胁侦测以及威胁情报的生成和分析。

1)渗透测试

渗透测试是一种网络安全测试, 它发动模拟的网络攻击, 旨在寻找计算机系统中的漏洞。对于渗透测试, 通常需要高水平的专业知识, 并牵涉到多个手动测试和分析步骤。随着大语言模型的兴起, 学术界开始关注是否能够利用这些大语言模型来替代或辅助人类进行渗透测试。文献[156]深入研究了这些模型在自动执行安全中心常见任务方面的能力。其创新之处体现在采用了提示工程技术, 通过加入相关的上下文和结构信息来提高模型识别漏洞的准确率。通过这种方式, 模型在发现安全漏洞方面的表现得以不断提升。研究还对比了包括 GPT-3.5-Turbo 和 GPT-4-Turbo 在内的多种大语言模型和 SonarQube(一款广泛使用的代码质量审查工具)的效果, 结果表明, 经过特别设计的提示后, 大语言模型在准确度上能够达到甚至超过传统工具如 SonarQube 的水平。Andreas Happe[157]提出了在不同层次上利用大语言模型指导渗透测试的方法。其中, 高层次方法是让大语言模型为通用场景或具体目标设计渗透测试。而

低层次方法则是将大语言模型与存在漏洞的虚拟机集成在一起, 允许大模型分析系统漏洞并提供攻击命令。PENTESTGPT[158]提出了由推理、生成和解析三部分组成的交互式系统, 其中每个模块都代表了渗透测试团队中的不同角色, 通过让大语言模型代表不同角色以实现其功能。其中推理模块专注于渗透测试的整体状态, 生成模块专注于为特定子任务生成步骤, 同时, 解析模块用于处理渗透测试期间遇到的各种数据。

而 SQL 注入是一种常见的渗透测试方法, 允许攻击者干扰应用程序对其数据库进行的查询。随着大语言模型在 Web 应用中的广泛应用, 它们可能成为 SQL 注入攻击的新目标。这可能允许攻击者查看他们通常无法检索的数据。有学者对大语言模型与 Web 应用集成时可能遭受的 SQL 注入攻击风险进行了深入研究。研究聚焦于一种新型的注入方式, 即“提示至 SQL 注入”(Prompt-to-SQL, P2SQL)[159], 这种攻击通过向聊天机器人等应用接口输入恶意构造的自然语言提示来诱使大语言模型生成有害的 SQL 查询。此外, 他们还评估了包括 GPT-4 和 Llama 2 在内的七种最先进的大语言模型技术对于 P2SQL 攻击的易感性, 发现这些技术均有可能被利用来发动此类攻击。虽然这些模型为用户交互提供了便利和效率, 但同时也暴露了它们可能被利用来进行 SQL 注入攻击的风险。

2)蜜罐产品

网络蜜罐[160]是一种具有牺牲性质的计算机系统, 旨在吸引网络攻击。利用黑客的入侵企图来获取网络犯罪分子的信息以及他们的行动方式, 或者将他们从其他目标上引开。

大语言模型出色的动态能力也对蜜罐产品产生了不小的冲击, Muris Sladić 等人[161]提出了一种基于大语言模型创建动态且真实的软件蜜罐的新颖方法, 大语言模型可以解决以前蜜罐的确定性响应、缺乏适应性等重要局限性, 最终形成一个可以根据用户需求动态生成合成数据的关系。

这种基于大语言模型的软件蜜罐方法不仅能更有效地吸引和分析网络攻击, 还能根据具体的需求动态调整, 从而提高蜜罐技术的实用性和效果。

3)网络流量分析与生成

传统的网络流量分析方法依赖于手工特征提取和静态的规则集, 这些方法在处理复杂的、动态变化的网络环境时往往显得力不从心。然而, 借助大语言模型的强大学习能力, 研究人员开始探索使用这些模型来自动化网络流量的分析和生成过程, 从而开辟了新的研究方向和可能性。

回顾之前的研究, 2020 年的 PERT[162]是首个提出结合 BERT 模型和载荷编码表示进行网络流量分类的研究。通过将原始字节流转换成类语言的字符串列表, 并构建词嵌入向量输入模型预训练, PERT 发现经过微调的模型能够更有效地捕捉流量特征。在此基础上, 2022 年的 ET-BERT[163]进一步发展, 通过分析网络流量的地址和端口信息, 生成更为精细化的数据包 Token, 从而提升了模型学习流量特征的能力。为了更精准地识别和分析互联网上的设备, 研究人员开发了一系列基于大语言模型的先进技术。文献[164]利用 RoBERTa 模型处理网络扫描得到的文本数据, 通过训练得到的稳定数值表示来对设备进行聚类及指纹生成。这种方法不仅提高了处理海量互联网数据的自动化水平, 而且能够识别出先前数据库中未记录的新型物联网和服务器产品。此外, 文献[165]介绍了使用 GPT-3 生成合成网络流量的全新框架。由于真实数据集的稀缺, 该研究的意义尤为重大。研究者们开发了一款集数据包生成器和流量生成器于一体的命令行工具, 能够模拟包括正常流量和恶意流量在内的多种网络环境。实验表明, 大语言模型能够胜任生成简单的 ICMP(Internet Control Message Protocol)数据包, 但生成诸如 DNS 等更加复杂的协议时, 却遇到困难。

这些研究工作不仅证明了大语言模型在网络流量分析与生成领域的应用潜力, 同时也为网络安全研究提供了新的视角和工具。

4)网络威胁

网络威胁指的是通过网络渠道实施的各种恶意活动, 旨在损害个人、组织或国家的信息系统安全。这些威胁可以包括病毒、蠕虫、间谍软件、广告软件、网络钓鱼、DDoS(Distributed Denial of Service)攻击等多种形式。面对日益狡猾的网络安全挑战, 传统的网络安全防御技术由于缺乏足够的灵活性和适应性, 往往难以有效应对新兴的网络威胁。在这种背景下, 利用大语言模型进行网络威胁检测和情报分析成为了网络安全领域的一项重要研究方向。

来自 USC 的研究团队[166]提出利用大语言模型来识别 IoT(Internet of Things)设备中的 DDoS 攻击。与传统的 MLP(Multilayer Perceptron)相比较, 团队发现大语言模型不仅在检测潜在的 DDoS 威胁方面表现出色, 而且能够清晰地阐释其推理过程, 为网络安全领域带来了创新的视角。在此基础上, 文献[167]引入了一个创新的预训练语言模型 SecurityLLM, 专门用于侦测网络安全威胁, 融合了两大关键技术:

SecurityBERT, 负责网络威胁的检测, 以及 FalconLLM, 处理事件的响应与恢复工作。特别值得一提的创新是, 首次将 BERT 架构应用于网络威胁侦测领域。该模型能够以高达 98%的准确度识别出 14 种不同的攻击类型。更进一步, 它采纳了一种新颖的保护隐私的编码技术——固定长度语言编码, 并结合了字节级的 BPE(Byte-Pair Encoding)Tokenizer 技术, 有效地处理和表示网络流量数据。这种综合性的方法在网络威胁侦测方面超越了传统的机器学习与深度学习技术, 充分展现了大语言模型在网络安全领域应用的巨大潜力。尽管大语言模型在许多任务中表现出色, 但 BERT 模型在特定任务上的表现同样不容小觑。文献[168]提出了一个用于网络威胁检测的网络活动新闻警报语言模型, 通过微调 BERT 模型能够比更大、更昂贵的语言模型在准确性和成本效益上表现更优。文献[169]专注于结合全球网络威胁信息和组织内部知识, 自动生成具有组织特色的威胁情报。通过运用大语言模型高效处理数据, 包括全球威胁信息的获取、本地知识的提取及生成具有上下文关联的情报, LOCALINTEL 显著减少了安全运营中心分析师的工作量。这种方法不仅快速准确地提升了威胁情报分析的质量, 还实现了针对组织特定运营环境的定制化服务, 标志着网络安全领域的一大进步。文献[170]对如 ChatGPT 和 GPT4all 等大语言模型在网络威胁情报领域的应用进行了评估, 特别是它们在处理开源情报(Open-Source Intelligence, OSINT)方面的表现。研究重点检测了这些模型在处理 Twitter 数据时进行二元分类和命名实体识别(Named Entity Recognition, NER)的能力。主要发现显示, 在二元分类任务上, ChatGPT-4 和 GPT4all 能够取得较好成绩。然而, 在命名实体识别方面, 研究发现了一些局限性, 指出了这些领域需要进一步细化和改进的必要性。文献[171]提出了一种分布式威胁情报框架, 通过在边缘设备上部署轻量级机器学习模型并结合大语言模型, 以提升网络安全性。该方法利用边缘设备处理本地数据流(如网络流量和系统日志), 实现实时威胁检测, 并通过边缘服务器分担计算任务, 减少延迟, 提高响应速度。文献[172]探讨了大语言模型在解读含糊其辞的网络攻击描述时的应用, 尤其是将直接应用如 GPT-3.5 这样的大语言模型与对像 BERT 这样的小语言模型进行监督式微调(Supervised Fine-Tuning, SFT)进行了比较, 目的是看哪种方法在预测网络攻击策略时更为有效。该研究发现对小语言模型进行 SFT 可以更精确地识别不同的网络攻击策略, 而直接使用大语言模型虽能覆盖更广范围, 但其解读的准确性较低。研究还指出, 解析网络攻击时固有的模糊性是个挑战, 并提出了未来的研究方向, 包括改进提示适应和利用自我监督学习技术, 以增强大语言模型在网络安全领域内的解读能力。文献[173]训练了一个名为 PLLM-CS 的大语言模型, 用于卫星网络中的网络威胁检测。通过将网络数据转化为适合上下文输入, PLLM-CS 能够有效编码网络数据中的上下文信息, 在两个公开的网络数据集 UNSW_NB15 和 TON_IoT 上的表现优于现有的技术。

这些研究不仅展示了大语言模型在网络安全领域内的广泛应用潜力, 也指出了它们面临的安全威胁和挑战, 以及为了提高网络安全而进行的各种创新性解决方案。

3.2 物理安全

在当今日益数字化和高度互联的世界中, 物理安全已成为一个不可忽视的领域, 特别是在硬件系统和智能设备普及的背景下。本节探讨了大语言模型在物理安全领域中的应用以及可能带来的威胁[174], 涵盖了从系统级芯片(System on a Chip, SoC)的安全验证到智能家居系统的安全挑战, 以及大语言模型作为代理执行攻击等潜在风险。

1)硬件系统威胁

硬件系统威胁指的是针对计算设备硬件层面的攻击, 旨在破坏或者利用硬件资源进行恶意活动。这些威胁可能针对个人电脑、服务器、移动设备等任何包含处理器和存储组件的设备。

随着这些大语言模型在日常生活中的应用越来越广泛, 它们也成为了攻击者的新目标。特别是在硬件系统层面, 这些先进的计算平台面临着各种新型的安全威胁。早在 2021 年就有研究人员提出[175]了一种新型的机器学习系统威胁, 名为“海绵示例”(Sponge Examples)。这类输入专门设计用于大幅增加神经网络的能源消耗和处理时间, 从而触发类似于服务拒绝攻击的效果。研究显示, 在多种机器学习模型和硬件平台上, 包括 CPU、GPU 和 ASIC, 这些海绵示例能够显著降低性能和能效。特别是在处理语言模型时, 这种攻击能让延迟和能耗最多增加 30 倍。研究进一步分析了这些攻击在不同模型和硬件间的可迁移性, 并通过实例展示了一个针对 Microsoft Azure 翻译服务的攻击场景, 使得响应时间从 1 ms 急剧增加到 6 s。论文最终建议, 通过从平均性能分析转向最坏情况性能分析的防御策略, 可以有效地缓解这类攻击的影响。

总之, 硬件系统威胁成为了现代计算平台不可忽视的安全挑战, 尤其是对于消耗大量硬件资源的

大语言模型。通过理解并应对这些威胁, 可以更好地保护我们的计算资源不受恶意攻击的影响, 确保技术的安全性和可靠性。

2)SoC 安全验证

SoC 安全验证是一个关键的工程实践, 旨在确保集成电路设计在安全方面的完整性和可靠性。这一过程涉及对 SoC 设计中可能存在的安全漏洞和威胁进行识别、评估和缓解, 从而防止潜在的安全风险。

传统的安全验证方法面临着多种挑战, 包括高昂的成本、漫长的验证周期以及难以覆盖所有潜在安全威胁。在这种背景下, 大语言模型的应用为 SoC 安全验证领域带来了新的机遇。文献[176]详细探讨了如何将大语言模型融入到 SoC 安全验证的过程中, 以实现 SoC 安全性的显著提升。研究不仅深入分析了现有的工作, 还通过实际案例研究、广泛的实验验证, 并提供了一套在 SoC 安全任务中应用大语言模型的实践指南。本研究创新地展示了大语言模型在处理多种安全任务——包括漏洞的插入与检测、安全性评估以及防御措施的开发——中的巨大潜力, 强调了大语言模型在解决 SoC 安全领域现存挑战及提出新策略方面的重要作用。文献[177]提出了一种名为 LLM4SecHW 的创新框架, 专为硬件调试领域设计, 运用了领域特定的大语言模型。该框架通过收集开源硬件设计的缺陷及其解决方案的版本控制数据, 构建了一个特殊的数据集。这一数据集使得可以对大语言模型进行精细调整, 从而显著提高它们在硬件设计中识别和修复缺陷的能力。文献[178]详细探讨了如何在硬件设计阶段, 特别是在寄存器传输级(Register Transfer Level, RTL)设计中, 利用大语言模型自动检测和缓解安全漏洞。并提出开发专门的大语言模型架构来提高硬件安全任务的性能。而文献[179]探讨了大语言模型在芯片设计中的应用及其潜在的安全风险。作者首先回顾了大语言模型在硬件描述语言(Hardware Description Language, HDL)代码生成和电子设计自动化(Electronic Design Automation, EDA)工具使用中的最新进展, 同时提出了相关的安全问题和建立信任的必要性。

这些研究和实践不仅展示了大语言模型在硬件安全领域的应用潜力和风险, 也为未来的研究和开发指明了方向。

3)智能家居系统安全

智能家居系统安全指的是保护智能家居设备及其网络免受未经授权的访问、使用或破坏的措施和技术。大语言模型通过理解和生成自然语言, 能够与用户以前所未有的自然和直观的方式交互, 从而极大地丰富了智能家居的功能和用户体验。

然而, 随着这些技术的集成, 智能家居系统的安全性和隐私保护问题也日益凸显。文献[180]深入探讨了将 ChatGPT 这种人工智能技术融入智能家居系统的前景与挑战, 突出了其在提升家居自动化方面的潜力与价值。通过如自动化客户服务和有效的家庭管理等方式, ChatGPT 能够提供更为直观和定制化的体验。不过也提出了对安全性、隐私保护以及伴随高级 AI 技术集成进家庭环境的道德考量的重要担忧。这包括面临黑客攻击的风险、数据被盗的可能性, 以及 AI 在家庭中产生内容的道德问题。论文呼吁采取一种权衡策略, 一方面认识到人工智能为家居自动化带来的创新进步, 另一方面也强调必须谨慎行事, 开发出有效的保障措施, 确保用户的隐私和安全得到充分保护。

智能家居系统安全是保护智能家居技术及用户隐私的重要领域, 需要综合考虑技术创新与安全隐患之间的平衡。随着大语言模型等人工智能技术在智能家居中的应用增加, 提高系统的安全性和保护用户隐私变得更加关键。

4)大模型代理攻击

大模型代理攻击是一种利用大语言模型进行的网络安全攻击, 其中攻击者通过借助模型执行恶意操作。代理攻击的关键在于, 攻击者不需要直接接触目标系统或网络, 而是利用大语言模型作为中间件来传递命令或数据。

RatGPT[181]的研究展示了一个概念验证, 表明攻击者可以利用大语言模型和相关插件在不直接接触受害者的情况下传播恶意软件。研究中利用 ChatGPT 建立与 C2(Command-and-Control)服务器的通信, 以接收命令并与受害者系统交互。这种方法有效地将大语言模型作为中介使用, 同时规避了传统恶意软件检测机制。在 LoFT[182]的研究中, 研究者提出了对抗性攻击的本地代理微调策略, 以增强攻击在不同模型间的转移性。通过精细化的微调, 公共模型可以被训练成高效的代理, 攻击者可以借此攻击私有目标模型, 即使对目标模型的内部构造和特性一无所知。这项研究的实验结果表明, 本地代理微调显著提高了对抗性攻击的成功率, 这突显了大语言模型在模型转移性方面的脆弱性。这两篇文章共同揭示了大语言模型在作为代理执行恶意攻击时的潜在风险。

大模型代理攻击揭示了一个重要的安全隐患, 即攻击者能够利用大语言模型作为传递恶意行为的

中介, 从而绕过传统的安全防护措施。

5)感知物理信号

感知物理信号是指利用技术手段捕获和分析来自物理世界的数据。这种能力使设备能够理解和响应其所处环境的变化, 为人们提供更智能、更精准的服务。

大语言模型已经证明了自己在理解和生成文本方面的能力。最近的研究开始探索这些模型处理非传统数据类型的潜力, 特别是在感知和理解物理世界信号方面的能力。文献[183]揭示了大语言模型的一项新能力: 处理物理信号以理解世界。作者尝试让 ChatGPT 分析手机传感器(如加速度计、卫星和 Wi-Fi)的信号, 以便感知用户在现实世界中的行动和位置。此外, 研究者还探索了让 ChatGPT 处理心电图数据的可能性。基于这些发现, 研究者提出了“渗透式人工智能”的概念。

该研究成果不仅展示了大语言模型在处理文本数据之外的潜力, 也为人工智能在物理世界中的应用开辟了新的道路。

在探讨了物理安全的多个方面后, 我们可以看到, 尽管面临许多挑战, 但通过研究和开发新的技术和策略, 我们有能力应对这些挑战。特别是大语言模型在安全领域的应用展现了其在识别威胁、增强系统安全性以及开发新型安全措施方面的巨大潜力。

3.3 信息安全

本节探讨了大语言模型在代码生成、审查和修复方面的应用。总结了这些模型在自动化程序修复、漏洞检测、安全性评估、预防电信诈骗以及智能合约审查方面的能力。

1)代码生成

代码生成是指大语言模型根据给定的自然语言说明自动生成源代码。这一技术允许开发人员通过描述他们想要的程序功能来快速生成代码, 从而提高开发效率并降低错误率。

然而, 尽管这些大语言模型在提高编码效率方面表现出色, 它们在代码安全性方面的性能还未得到充分的探索。在文献[184]中, 研究人员利用大语言模型强大的代码生成功能来探索其创造恶意软件代码的潜力。该研究揭示了一个有趣的现象: 尽管大语言模型在处理复杂、大规模代码块时遭遇挑战, 但恶意行为者能够巧妙地绕过这一局限。他们通过细分恶意行为, 将其划分为更小的、可管理的代码片段, 从而有效地拼装出恶意软件。此外, 大语言模型在根据这些较小的代码片段描述恶意软件构建过程时表现卓越。文献[185]的研究以网络攻击为出发点, 也得出了类似的结论: 大语言模型能够轻松地生成恶意软件及其多样化的变体。但是, 使用大语言模型生成恶意代码并非没有门槛。文中指出了其中的三大挑战: 首先, 需要找到方法绕过大语言模型内置的安全机制(即实现“越狱”); 其次, 必须为生成的恶意代码设计精确的初始提示; 最后, 需要对代码进行混淆处理, 以规避现有的防病毒软件检测。文献[186]详细记录了恶意用户如何智取大语言模型的道德保护机制以生成恶意代码, 并对这种借助先进技术进行的不当行为表示了严重关切。这些研究成果对监管和防范大语言模型滥用的潜在风险提供了重要见解。同时, 随着研究界对大语言模型在代码安全性方面的潜力和所面临的挑战的关注加深, 已经涌现出众多新的方法和研究[187], 专注于如何强化模型生成代码的安全性、进行对抗性测试, 以确保技术的安全性和可靠性。而文献[188]探讨了大语言模型在代码安全性加固和对抗性测试方面的应用, 并介绍了一种名为 SVEN 的方法, 该方法通过特定的连续向量序列, 作为前缀, 来指导模型生成符合安全要求的代码, 同时保持其在功能正确性方面的能力。文献[189]研究了基于 AI 的编程助手, 如 OpenAI、Codex 对开发者编写代码安全性的影响。这项研究涉及 58 名学生程序员, 他们被分成控制组和 AI 辅助组, 任务是用 C 语言实现一个购物清单结构。研究旨在评估使用 AI 代码助手是否会影响结果代码的功能性和安全性。主要发现包括 AI 辅助的用户产生的关键安全漏洞并不比控制组显著更多, 且 AI 助手对安全性的影响最小。该研究还评估了 AI 辅助代码中漏洞的来源, 发现相当一部分漏洞源于人类编写的代码。该文献只涉及用 C 语言实现一个购物清单结构, 这可能不足以涵盖编程实践中的复杂性和安全性挑战。这些研究不仅增进了我们对大语言模型在代码安全性方面的认识, 也为进一步改善这些模型提供了实证基础。

最后, 大语言模型还可以辅助逆向工程, 例如 Hammond Pearce 等人[190]的研究, 利用大语言模型对现有软件进行逆向工程分析, 研究表明, 大语言模型在逆向工程领域具有较大潜力, 但是这些技术需要进一步成熟才能更可靠的应用于广泛的逆向工程。

总结来说, 大语言模型在代码生成领域的应用已经取得了显著的进展, 不仅提高了编码效率, 还展现出了在代码安全性和逆向工程领域的潜力。尽管存在挑战, 如恶意代码生成和安全机制的绕过,

但通过持续的研究和开发, 这些模型正变得越来越安全, 对未来软件开发的影响将持续深远。

2)漏洞检测与修复

漏洞检测与修复是计算机安全领域中的一项重要活动, 旨在识别和修补软件中的漏洞, 以防止未授权访问或攻击。这个过程通常涉及使用各种工具和技术来分析代码, 发现潜在的安全问题, 并提供相应的解决方案来减轻或消除这些问题。

大语言模型在漏洞检测与修复方面的应用主要体现在以下几个方面: 文献[191]通过大规模研究, 考察了大语言模型在未经特定训练的情况下修复漏洞的能力, 涵盖了合成的、手工制作的以及现实世界的漏洞场景。文献[192]介绍了一种利用大语言模型来发现代码中异常行的新方法, 该方法通过生成代码的变体与原始代码进行比较, 从而检测出潜在的异常。这种方法不受编程语言的限制, 且能处理不完整或无法编译的代码, 无须预定义的安全规则或属性。文献[193]提出了一种创新性的自动程序修复(Automated Program Repair, APR)方法, 名为FitRepair, 它基于大语言模型 CodeT5,该方法结合了针对特定领域的微调策略和新型的提示策略, 高效利用了项目中特有的信息和标识符, 显著提升了自动程序修复的效果。文献[194]提出将大语言模型和传统的静态代码分析技术相结合, 通过经过专门训练的大语言模型自动生成修复代码。该框架已经在微软的持续集成流程中得到应用, 这不仅证明了其实用价值, 也展示了其在实际软件开发过程中的有效性。文献[195]评估了大语言模型, 尤其是 GPT-4, 在检测和修复软件漏洞方面的性能, 并与传统静态代码分析工具进行了比较, 发现 GPT-4 识别出的漏洞数量大约是对手的四倍, 并且提供了有效的修复方案。同时还指出了大语言模型自我审计的能力, 为它们所识别的漏洞提供修复方案。文献[196]融合了自动化网站爬虫和自然语言处理技术, 从 CWE 网站提取漏洞信息, 并结合用户反馈, 生成提示。并将这些提示作为输入, 指导 GPT 模型提供修复漏洞的建议。文献[197]评估了闭源大模型在实际安全管理中的应用, 结果表明现阶段仍然无法取代专业安全工程师在漏洞分析中的作用。文献[198]侧重于研究为漏洞检测定制模型、加快训练速度以及解决数据集不平衡的问题。文献[199]针对 C/C++代码中的漏洞提出了一种基于大语言模型的修复框架, 该框架借助抽象语法树提取代码中的信息, 而后借助大语言模型进行修复。

此外, 文献[200]则提出了一个框架, 通过大语言模型自动修复硬件设计中的安全缺陷, 并通过针对领域代表性漏洞的语料库来量化模型的修复性能。文献[201]通过 GPT-3.5-turbo 生成了一个名为 FormAI 的数据集, 该数据集包括多种复杂性的程序, 并使用正式验证方法来准确标识程序中的漏洞。最后, 文献[202]通过 SecuCoGen 数据集对大语言模型在安全代码生成中的表现进行了评估和提升, 揭示了现有模型在生成代码时常忽略的安全问题, 并提出了强化模型以生成更安全代码的方法。这些研究共同推进了我们对大语言模型在代码安全性方面应用的理解, 并为未来的工具和方法的开发奠定了基础。文献[203]深入探讨了大语言模型在审查安全代码方面的能力, 通过五种不同的测试方式, 评估了它们对 549 个实际代码文件中安全漏洞的识别效果。研究指出, 这些模型在回应中存在一些质量问题, 比如过于啰嗦、含糊不清或信息不完整, 这些问题被分为五大类和十六个小类。虽然结果显示这些语言模型在安全审查方面具有一定的潜力, 但它们在准确性、易于理解和符合安全检测标准方面还需大幅改进, 以更好地发挥作用。为了防御大语言模型集成程序中的 RCE 漏洞, 文献[204]提出了两种防御策略, 一是 LLMSmith 的静态分析工具, 二是一张基于提示的自动化测试方法, 来验证大语言模型集成 Web 应用程序的漏洞。

总的来说, 大语言模型在漏洞检测与修复方面的应用已经取得了初步的进展, 并展示出了广泛的应用潜力。但是, 为了更好地利用这些模型在实际软件开发和安全管理中的作用, 还需要对它们的能力进行更深入的研究和优化。这不仅包括提高模型的准确性和可用性, 也涉及开发新的方法和策略来克服现有的限制。

3)安全评估

安全评估是一种系统性的过程, 用于识别和评估信息系统中的潜在安全风险和漏洞。

目前, 专门针对大语言模型安全性的评估研究仍然较为稀缺。文献[205]探讨了大语言模型在硬件断言生成中的应用, 展示了如何使用自然语言提示来生成 SystemVerilog 断言, 并提供了一个基于实际硬件设计的评估框架。文献[206]提出了一个针对安全性能评估的自然语言提示数据集 LLMSecEval, 该数据集基于 MITRE 的 CWE 排名, 包含了多种安全漏洞的描述, 用于测试和评估大语言模型在代码安全性方面的表现。文献[207]则通过对 GitHub 上公开的项目中 GitHub Copilot 生成的代码片段进行实证研究, 分析了这些代码片段中的安全弱点, 以探究

在真实世界场景中, AI 自动生成代码所面临的安全挑战及其规模。这两项研究为评估和提升 AI 代码生成工具的安全性提供了重要的视角和工具。文献[208]通过构建 228 个代码场景对大语言模型进行了八个不同维度的分析。发现了即便是先进的模型如 PaLM2 和 GPT-4 也存在问题, 代码微小变动就能导致它们在一定比例的情况下给出错误答案。文献[209]提出了一个针对多模态大语言模型的基准测试, 主要用于评估模型的视觉感知能力。当前阶段的多模态大模型在人类能够迅速处理的领域, 如视觉对应、目标定位和多视角推理方面, 仍未能实现准确的推理。

尽管大语言模型在多个领域展现了其强大的能力, 但是关于它们的安全性评估研究还相对不足, 这凸显了一个迫切的需求即发展新的方法和工具来系统性地评估和提高这些大语言模型的安全性。

4)电信诈骗

电信诈骗是一种利用电话、互联网或其他电信技术实施的欺诈行为, 旨在非法获取个人的财产、财务信息或敏感数据。

随着网络钓鱼等电信诈骗手段的不断演变和升级, 传统的防诈骗方法面临着新的挑战。在这种背景下, 利用大语言模型来提升防诈骗能力展现出了巨大的潜力和价值。文献[210]利用大语言模型来制定精准定向的网络钓鱼邮件的实践效果, 比较这类 AI 生成邮件与传统手工编写邮件的有效性。研究深入分析了钓鱼邮件中融入组织内部信息与外部信息的不同影响, 钓鱼防范培训及警告的作用, 以及发展出一套基于机器学习的技术手段, 专门用于识别由大语言模型生成的钓鱼邮件。文献[211]深入探讨了大语言模型在识别钓鱼网址方面的能力, 比较了提示工程和微调方法。这一发现强调了针对具体任务微调大语言模型相比于使用通用提示工程技术的优势, 为将大语言模型用于防范电信诈骗提供了新的视角。

利用大语言模型的强大能力, 特别是通过精准定向和微调方法, 为识别和防范这类诈骗提供了有效的新工具和方法, 展现了人工智能技术在打击网络犯罪中的巨大价值。

5)智能合约

智能合约是一种在区块链技术上运行的自动执行合同条款的计算机程序或协议。这些协议设计用来自动执行、控制或记录区块链相关的事件和行为, 按照预设的条件自动化执行各项事务。

随着智能合约的复杂性增加, 其安全性问题也日益凸显, 成为了区块链技术发展中的一个关键挑战。传统的安全性分析方法, 如模糊测试, 虽然在发现潜在漏洞方面发挥了作用, 但仍存在效率低下和覆盖面不足的问题。在这个背景下, 大语言模型的应用为智能合约的安全性分析提供了新的可能性。

文献[212]利用大语言模型来提升智能合约模糊测试效果的创新策略。该策略通过智能地引导和设置模糊测试的优先级, 旨在提高智能合约安全性分析的效率。并且克服了传统模糊测试方法的局限, 通过大语言模型生成的指标集中关注更关键的代码区域和输入序列, 显著提升了测试的效率、覆盖面及漏洞侦测的准确性。通过在现实世界的项目中的评估, 展现了其在智能合约安全领域进步的巨大潜力[213]。

本小节讨论了大语言模型在代码安全性方面的应用与挑战, 尤其是在代码生成、审查和漏洞修复等方面。尽管这些模型展现出提高开发效率的潜力, 但它们在确保代码安全方面仍面临挑战, 需要更多研究和改进以提高其准确性和安全性。这些发现对于未来安全工具的开发具有重要意义。

4 安全垂直领域大模型

本章将介绍国内外企业在安全领域的垂直模型上的研究进展。图 5 展示了国内安全企业在大语言模型上的不同尝试。

4.1 自研安全大模型

企业界正在积极利用自身深厚的技术底蕴开发针对安全领域的专业大模型, 以显著增强网络安全防御体系的检测效能、防护能力和日常运营效率。

在 2023 年的 ISC 大会上, 360 集团率先在国内推出了首个可供交付的安全行业专用大模型——“360 安全大模型”。该模型立足于 360 自主研发的认知型通用大模型“360 智脑”, 并深度融合了过去 15 年间 360 在 AI 安全应用领域积累的经验及安全大数据资源, 形成了针对安全行业的专业化垂直模型。紧跟其步伐, 奇安信集团在北京发布了两款创新产品: Q-GPT 安全机器人和大模型卫士。

Q-GPT 安全机器人依托奇安信大模型构建, 如同一位“虚拟安全专家”, 能全天候不间断地工作, 工作效率相当于超过 60 位实体安全专家之和, 预计每年创造约 2000 万元人民币的运营价值, 极大地提升了安全生产力。而大模型卫士则集成了一系列核心安全功能, 包括安全风险发现、大模型访问控制、数据泄露管理、违规行为追踪以及大模型应用深度分析, 旨在帮助企业更加安全高效地运用大模型技术提升整体生产力。同期, 深信服也推出了迭代升级至 2.0 版本的深信服安全 GPT, 并预告将在 2024 年 1 月 26 日

图 5　国内外安全厂商在大语言模型领域的尝试

Figure 5　Attempts of domestic and foreign security vendors in the field of large language models

发布最新一代安全 GPT 3.0[214]。从最初的辅助角色发展到如今的智能化自主处理，深信服安全 GPT 3.0 已经能够在邮件和文件行为分析方面展现出媲美资深安全专家的能力，尤其在对抗钓鱼攻击时表现卓越，效果明显优于传统安全解决方案。此外，天融信科技打造的“天问大模型”凭借风险预测、安全深度分析、情报提取、知识生成、逻辑推理和决策执行等功能优势，通过持续学习与优化，有效地实现了对复杂网络攻击行为的实时监控与预警，为各企业提供坚不可摧的安全防护墙。美亚柏科公司推出的“天擎”美亚公共安全大模型作为国内首个应用于公共安全领域的智能大模型，具备深厚的公共安全行业知识储备和出色的警务意图识别、情报分析以及案情推理能力。它能在海量数据中自我进化更新，形成行业知识、业务问题识别与解决反馈的完整闭环进化过程。同样致力于安全领域，吉大正元公司推出了名为“昆仑”的正元安全专属大模型，这是我国首家专注于密码与数据安全细分领域的专属大模型，凸显了对特定安全需求的高度定制化适应性。在此前举办的成都大运会上，恒脑·安全垂域大模型已经投入应用，在安恒信息圆满完成成都大运会网络安全保障的工作中发挥重要作用。并以此为基础全新升级发布了基于恒脑·安全垂域大模型的安全运营平台，加速安全垂域大模型应用落地步伐，赋能城市治理、业务发展、人才培养和数智民生。绿盟提出的风云卫大模型用于解决三大问题：实战态势指挥调度、红蓝对抗辅助决策以及安全运营效能提升。借助 AI 实现安全行业工业范式重塑。

此外，我们调研了国外企业在安全大模型上的研究进展。Microsoft Copilot for Security[215]预计于 2024 年 4 月 1 日全球上市，旨在协助安全与 IT 专业人士揭示可能被忽视的安全漏洞，加速应对措施，并增强团队的专业知识。利用 Microsoft 日处理的超 78 万亿个安全信号，Copilot 提供个性化的洞察和行动指导，以 AI 的速度和规模增强安全操作。最新研究显示，使用 Copilot 的经验丰富的安全分析师工作速度提高 22%，准确性提高 7%，且 97%的用户表示愿意再次使用，证明了 Copilot 能显著提升安全任务的效率和准确性，适用于所有经验水平的专业人士，改善工作体验，使安全工作对所有人成为可能。网络安全 AI 领域的著名厂商 Darktrace[216]推出了 Darktrace DETECT 和 RESPOND 模型使客户更容易设置护栏来监控，并在必要时响应活动和与生成式人工智能和大语言模型工具的连接。

多家知名企业纷纷推出各自在安全领域的专属大模型应用，这些大模型不仅体现了企业在安全技术研发上的领先实力，而且通过整合先进的 AI 技术和大量行业经验，有力地推动了网络安全防御体系的智能化升级转型。

4.2　使用成熟大模型赋能安全

使用成熟大模型赋能安全领域，不仅提供了一

种新的解决方案, 还能极大地提升安全防护的智能化水平和效率。

2023 年云栖大会上, 阿里云安全[217]正式宣布基于通义千问大模型微调的安全大模型投入使用。启明星辰安全大模型解决方案, 以中国移动九天大模型为基座, 结合启明星辰丰富的安全数据集, 进行全量预训练和微调, 打造安全大模型。同时, 通过安全大模型、专用小模型、安全产品的智能组合, 提升产品能力和运营服务效率并应用到威胁检测、安全运营、开发安全等安全场景中。

长亭科技也在自身多年深耕安全攻防实践的基础上, 推出了基于通用大模型的行业知识问答功能, 这一功能有助于安全从业者迅速获取权威、精确且及时的安全攻防信息, 从而有效提升整个行业的响应速度与应对能力。山石网科对于大模型有着不同的看法, 伴随着今年年初ChatGPT及其他AIGC服务和模型的推出, 考虑到算力、数据等投入过于庞大, 山石网科暂时将不会自己训练大语言模型, 而是采用现有的大模型, 利用“预训练+微调”以及“反馈学习”的手段, 将大模型的理解和分析能力尽快应用到产品安全能力的构建中。

海云安推出了国内首款集成AI大语言模型的静态代码检测平台 SCAP++, 标志着大语言模型首次应用于静态代码检测领域。与传统的 SCAP 相比, SCAP++利用大语言模型解决了三个主要问题: 一是通过自动化的误报判断显著降低误报率; 二是提供针对性的缺陷成因解释, 帮助用户更深入理解并解决缺陷; 三是生成具体且适用的代码修复方案, 提高缺陷修复的效率和准确性。

通过利用成熟的大语言模型, 企业能够迅速构建针对特定行业的应用程序。这种方法不仅加速了开发过程, 还使得企业能够更有效地解决行业特有的挑战。

4.3 建立大模型安全平台

随着这些大模型的深入应用, 其安全问题也日益凸显, 成为不容忽视的挑战。2024 年 5 月 21 日, 图灵奖得主 Yoshua Bengio 联合国内外数十余位专家, 呼吁针对人工智能风险采取更加有力的措施。为此, 构建一个健全、可靠的大模型安全平台变得至关重要。

百度安全[219]基于百度领先的 AI 大模型平台, 打造了百度 AI 安全底座, 核心包括两方面, 一个是安全知识和技能强化的大模型, 提供了思考能力, 一个是全场景的智能体, 提供了执行能力。基于二十余年安全对抗的总结与提炼, 围绕百度“文心大模型”安全实践经验, 日前, 百度安全推出了以 AI 安全为核心的“百度大模型安全解决方案”, 从大模型全生命周期视角出发, 方案涵盖大模型训练/精调/推理、大模型部署、大模型业务运营等关键阶段所面临的安全风险与业务挑战, 提供全套安全产品与服务, 助力企业构建平稳健康、可信可靠的大模型服务。思考、执行两大通用性智能的核心能力, 为百度安全高效、低成本地构建各类 AI 原生安全应用提供了可能。

由火山引擎提出的火山方舟安全架构能够更好的桥接大模型提供方以及使用方。通过在存储层提供加密模块构建安全存储, 在运行时提供多种安全计算方法, 在网络层构建私有 VPC, 在传输层加密以及在服务层集成内容审查引擎, 支撑各方做好大模型的内容安全, 实现全方位的数据隐私和资产安全保护。

蚂蚁安全推出了 AI 安全检测平台“蚁鉴 2.0”和“蚁天鉴”的全新大模型安全一体化解决方案, 其中“蚁鉴 2.0”可以诊疗检测、定位问题, 它相当于站在“黑产”角度, 通过智能攻击对抗技术, 自动生成数百万的诱导性问题, 对生成式大模型进行诱导式问答, 并对大模型的回答实时、自动化检测计算, 24 h 不眠不休“找茬”大模型存在的弱点和安全问题。而“天鉴”可以进行“防治”, 防患于未然, 帮助大模型挡住外界的恶意提问, 同时对生成的回答内容进行风险过滤, 保障大模型上线后从用户输入到生成输出的整体安全防御。

这些进展表明, 通过建立大模型安全平台, 不仅可以提高大模型的安全性, 还能保护用户的数据隐私, 促进 AI 技术的健康发展。

综上所述, 目前国内外安全企业在大模型技术应用上取得了显著进展, 通过自研专属安全大模型、利用成熟大模型提升安全能力, 以及建立全面的大模型安全平台, 有效地推动了网络安全防御体系的智能化升级和转型。这些成果不仅体现了企业在安全技术研发上的领先地位, 还极大地提高了组织的安全保障水平和运营效率, 实现了从被动防御到主动预判与干预的重大跨越。

5 主要挑战和未来研究方向

本小节, 我们将讨论当前大语言模型安全面临的主要挑战以及潜在的研究方向, 为该领域的研究者提供一些建议(参见表 3)。

5.1 大语言模型自身安全的未来方向

5.1.1 研究大模型价值观和安全对齐策略

大模型的价值观评估是一个复杂且多维的问题,

表 3 大模型安全的研究方向

Table 3 Future research directions of large language model security

研究角度	主要挑战	未来研究方向
自身安全	已有研究还不能有效评估大模型价值观	如何有效评估大模型价值观, 并研究针对大模型的安全对齐策略
	已有研究很难量化大模型生成内容的可信度	研究如何评估针对大模型生成内容的可信度
	现有攻击技术在泛化能力、自动化程度的局限、成本效率权衡以及防御策略滞后性等方面面临显著挑战	研究针对大模型的高效、低成本的自动化攻击, 并通过自动化工具识别弱点, 构建有效防御策略
传统安全	现实环境中网络攻击千变万化, 已有研究很难解释所有场景下的攻击	研究如何提升大语言模型对安全事件理解, 并将其转换成文字描述
	已有研究很少关注利用大模型防范现实生活中的社会工程攻击	探索和研究基于大模型的社会工程学防御机制
	现有研究还无法有效利用大模型分析复杂的服务配置文件, 准确识别潜在的安全漏洞	研究基于大模型识别服务配置错误并纠正
	现有蜜罐产品无法分析攻击者身份并基于最新的漏洞动态生成伪装数据	研究基于大模型的新型蜜罐系统

涉及技术、伦理和社会各个层面, 已有研究还不能有效评估大模型价值观。因此有效评估大模型价值观并研究针对大模型的安全对齐策略是未来研究的重要方向, 其核心目标是确保人工智能系统的决策过程与人类价值观保持一致, 并能在各种环境下安全地运作。这包括发展先进的技术和框架, 用于理解和编码人类伦理和价值观, 以及构建能够有效预测、识别和避免潜在风险的机制。

5.1.2 针对大模型生成内容的可信度研究

已有研究很难量化大模型生成内容的可信度。因此, 针对大模型生成内容的可信度研究聚焦于提高人工智能生成文本、图像、音频和视频等内容的真实性和准确性, 确保这些内容对用户有用且不会产生误导。随着大模型技术的发展, 如何鉴别和保证生成内容的质量变得尤为重要。这一研究方向将探索开发新的算法和技术, 用以评估和验证大模型输出的可信度, 包括内容的事实准确性、偏见检测、来源可追溯性以及遵守伦理和法律标准的能力。

5.1.3 针对大模型的自动化攻击研究

现有攻击技术在泛化能力、自动化程度的局限、成本效率权衡以及防御策略滞后性等方面面临显著挑战。因此, 针对大模型的自动化攻击研究, 将主要聚焦于开发和完善攻击技术, 这些技术旨在测试和评估大规模机器学习模型的安全性和鲁棒性。未来的研究方向将探索如何通过自动化工具和方法有效地识别模型的弱点和漏洞, 包括但不限于对抗性攻击、模型逆向工程、数据泄露攻击等。此外, 研究应致力于理解攻击者的动机、策略和技术手段, 以便构建更为有效的防御策略。这包括通过模拟攻击来测试模型的脆弱性, 从而不断完善和调整安全措施。

5.2 大语言模型赋能传统安全的未来方向

5.2.1 研究基于大模型的安全事件理解

现实环境中网络攻击千变万化, 已有研究很难解释所有场景下的攻击。因此, 探索如何利用大语言模型从技术日志、警报以及网络流量中提取关键信息, 并将这些信息转化为易于理解的自然语言报告, 是一项旨在极大提高网络安全和系统管理效率的研究方向。这种研究不仅需要大模型能够深入理解复杂的技术数据和模式, 还要求它们能够有效地将这些数据转换成准确、简洁且具有洞察力的文字描述。

5.2.2 研究基于大模型的社会工程学防御机制

已有研究很少关注利用大模型防范现实生活中的社会工程攻击。因此, 本研究旨在开发一个基于大语言模型的高级社会工程学防御机制, 通过深入理解人类交流模式和攻击行为来识别、分析和防范如钓鱼、假冒等社会工程攻击。通过结合社会学、心理学与计算机科学的交叉领域研究, 提供一个能够实时识别和防范社会工程学攻击的高效系统, 并开发一套提升用户对这类攻击认识的教育和培训工具, 从而减少社会工程学攻击的成功率。

5.2.3 研究基于大模型的服务配置错误识别与纠正

现有研究还无法有效利用大模型分析复杂的服务配置文件, 准确识别潜在的安全漏洞。因此, 研究基于大模型的服务配置错误自动识别与纠正旨在开发利用大语言模型的先进技术, 以自动检测并修正服务配置中的错误, 从而显著降低由不当配置引发的安全漏洞风险。此研究将深入探索如何有效地利用大模型分析服务配置文件, 结合安全最佳实践, 识别出配置中可能存在的安全漏洞。

5.2.4 研究基于大模型新型蜜罐系统

现有蜜罐产品在分析攻击者身份和动态生成伪装数据方面存在一定局限性。传统蜜罐主要通过模拟网络服务或系统漏洞来吸引攻击者, 并记录攻击行为以供分析。然而, 这些蜜罐往往缺乏对攻击者行为的深入理解和对动态生成伪装数据的支持。与传统蜜罐相比, 基于大模型的蜜罐系统能够更准确地

模拟真实环境中的网络服务和最新的系统漏洞。因此, 研究基于大模型的新型蜜罐系统具有重要意义。

6 总结

本文首先回顾了大模型的发展脉络, 包括关键的技术突破和当下流行的主流模型; 接着, 深入探讨了大模型固有的安全隐患, 深入剖析了大模型当前所面临的法律和伦理问题,并汇总了常见的攻击手段和防御策略。随后, 文章从网络安全、物理安全和信息安全三个维度, 梳理了大模型应用的关键领域, 并考察了国内外安全企业在这一新兴领域的创新尝试。文章最后, 针对大模型安全所面临的挑战和潜在威胁进行了系统性分析, 并提出了未来的研究方向, 旨在促进这一学科领域的健康持续发展。

参考文献

[1] Zhao W X, Zhou K, Li J Y, et al. A Survey of Large Language Models[EB/OL]. 2023: 2303.18223. http://arxiv.org/abs/2303.18223v13.

[2] Yang J F, Jin H Y, Tang R X, et al. Harnessing the Power of LLMS in Practice: A Survey on ChatGPT and beyond[EB/OL]. 2023: 2304.13712. http://arxiv.org/abs/2304.13712v2.

[3] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. *Advances in neural information processing systems*, 2017, 30.

[4] Devlin J, Chang M W, Lee K, et al. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding[EB/OL]. 2018: 1810.04805. http://arxiv.org/abs/1810.04805v2.

[5] Radford A, Wu J, Child R, et al. Language models are unsupervised multitask learners[J]. *OpenAI blog*, 2019, 1(8): 9.

[6] Brown T B, Mann B, Ryder N, et al. Language Models Are Few-Shot Learners[C]. *The 34th International Conference on Neural Information Processing Systems*, 2020: 1877-1901.

[7] Achiam J, Adler S, Agarwal S, et al. Gpt-4 technical report[EB/OL]. 2023: ArXiv Preprint ArXiv:2303.08774.

[8] Lex Fridman Podcast. Lex Clips. https://www.youtube.com/watch?v=DNQDqq4mWSY. Mar. 2024.

[9] Touvron H, Lavril T, Izacard G, et al. LLaMA: Open and Efficient Foundation Language Models[EB/OL]. 2023: 2302.13971. http://arxiv.org/abs/2302.13971v1.

[10] Touvron H, Martin L, Stone K, et al. Llama 2: Open foundation and fine-tuned chat models[EB/OL]. 2023: ArXiv Preprint ArXiv: 2307.09288.

[11] Llama3. META. https://ai.meta.com/blog/meta-llama-3/. Apr. 2024.

[12] Anil R, Dai A M, Firat O, et al. PaLM 2 Technical Report[EB/OL]. 2023: 2305.10403. http://arxiv.org/abs/2305.10403v3.

[13] Gemini. Google. https://deepmind.google/technologies/gemini.Dec. 2023.

[14] Copilot. Microsoft. https://copilot.microsoft.com/. Feb. 2023.

[15] Jiang A Q, Sablayrolles A, Mensch A, et al. Mistral 7B[EB/OL]. 2023: 2310.06825. http://arxiv.org/abs/2310.06825v1.

[16] Bai J Z, Bai S, Chu Y F, et al. Qwen Technical Report[EB/OL]. 2023: 2309.16609. http://arxiv.org/abs/2309.16609v1.

[17] Tongyi. Alibaba. https://tongyi.aliyun.com/qianwen. Apr. 2023.

[18] Hunyuan. Tencent. https://hunyuan.tencent.com. Sep. 2023.

[19] Wang Y H, Chen H T, Tang Y H, et al. PanGu-Π: Enhancing Language Model Architectures via Nonlinearity Compensation [EB/OL]. 2023: 2312.17276. http://arxiv.org/abs/2312.17276v1.

[20] Yiyan. Baidu. https://yiyan.baidu.com. Mar. 2023.

[21] Zhang X Y, Zhang X Y, Yu Y. ChatGLM-6B Fine-Tuning for Cultural and Creative Products Advertising Words[C]. *2023 International Conference on Culture-Oriented Science and Technology*, 2023: 291-295.

[22] BlueLM: An Open Multilingual 7B Language Model. BlueLM Team. https://github.com/vivo-ai-lab/BlueLM. Oct. 2023.

[23] Yang A Y, Xiao B, Wang B N, et al. Baichuan 2: Open Large-Scale Language Models[EB/OL]. 2023: 2309.10305. http://arxiv.org/abs/2309.10305v2.

[24] iFlytekSpark. IFLYTEK. https://gitee.com/iflytekopensource/iFlytekSpark-13B. Jan. 2024.

[25] Taichu. CASIA. https://taichu-web.ia.ac.cn/. Mar. 2024.

[26] Sun T, Zhang X, He Z, et al. Moss: Training conversational language models from synthetic data[EB/OL]. 2023: ArXiv Preprint ArXiv:2307.15020.

[27] Yi. 01.ai. https://www.01.ai. Nov. 2023.

[28] DeepSeek-v2. DeepSeek. https://github.com/deepseek-ai/DeepSeek-V2. May. 2024.

[29] Yao Y F, Duan J H, Xu K D, et al. A Survey on Large Language Model (LLM) Security and Privacy: The Good, the Bad, and the Ugly[EB/OL]. 2023: 2312.02003. http://arxiv.org/abs/2312.02003v3.

[30] Esmradi A, Yip D W, Chan C F. A Comprehensive Survey of Attack Techniques, Implementation, and Mitigation Strategies in Large Language Models[EB/OL]. 2023: 2312.10982. http://arxiv.org/abs/2312.10982v1.

[31] Cui T Y, Wang Y L, Fu C P, et al. Risk Taxonomy, Mitigation, and Assessment Benchmarks of Large Language Model Systems[EB/OL]. 2024: 2401.05778. http://arxiv.org/abs/2401.05778v1.

[32] Das B C, Amini M H, Wu Y Z. Security and Privacy Challenges of Large Language Models: A Survey[EB/OL]. 2024: 2402.00888. http://arxiv.org/abs/2402.00888v1.

[33] Li N, Ding Y D, Jiang H Y, et al. Jailbreak Attack for Large Language Models: A Survey[J]. *Journal of Computer Research and Development*, 2024, 61(5): 1156-1181.

(李南, 丁益东, 江浩宇, 等. 面向大语言模型的越狱攻击综述[J]. *计算机研究与发展*, 2024, 61(5): 1156-1181.)

[34] Xu H X, Wang S N, Li N K, et al. Large Language Models for Cyber Security: A Systematic Literature Review[EB/OL]. 2024: 2405.04760. http://arxiv.org/abs/2405.04760v2.

[35] Huang X W, Ruan W J, Huang W, et al. A Survey of Safety and Trustworthiness of Large Language Models through the Lens of Verification and Validation[EB/OL]. 2023: 2305.11391. http://arxiv.org/abs/2305.11391v2.

[36] ChatGPT: Italy says OpenAI's chatbot breaches data protection rules. BBC. https://www.bbc.com/news/technology-68128396. Apr. 2023.

[37] Li H R, Guo D D, Fan W, et al. Multi-Step Jailbreaking Privacy Attacks on ChatGPT[EB/OL]. 2023: 2304.05197. http://arxiv.org/abs/2304.05197v3.

[38] Shayegani E, Al Mamun M A, Fu Y, et al. Survey of Vulnerabilities in Large Language Models Revealed by Adversarial Attacks[EB/OL]. 2023: 2310.10844. http://arxiv.org/abs/2310.10844v1.

[39] Introducing GPTs. OpenAI. https://openai.com/blog/introducing-gpts. Nov. 2023.

[40] Zhang Z P, Jia M, Lee H P, et al. "It's a Fair Game", or Is It? Examining how Users Navigate Disclosure Risks and Benefits when Using LLM-Based Conversational Agents[EB/OL]. 2023: 2309.11653. http://arxiv.org/abs/2309.11653v2.

[41] Li H R, Chen Y L, Luo J L, et al. Privacy in Large Language Models: Attacks, Defenses and Future Directions[EB/OL]. 2023: 2310.10383. http://arxiv.org/abs/2310.10383v1.

[42] Pankajakshan R, Biswal S, Govindarajulu Y, et al. Mapping LLM Security Landscapes: A Comprehensive Stakeholder Risk Assessment Proposal[EB/OL]. 2024: 2403.13309. http://arxiv.org/abs/2403.13309v1.

[43] Sorensen T, Khlaaf H. LeftoverLocals: Listening to LLM Responses through Leaked GPU Local Memory[EB/OL]. 2024: 2401.16603. http://arxiv.org/abs/2401.16603v1.

[44] Abid A, Farooqi M, Zou J. Persistent Anti-Muslim Bias in Large Language Models[C]. *The 2021 AAAI/ACM Conference on AI, Ethics, and Society*, 2021: 298-306.

[45] Felkner V K, Chang H C H, Jang E, et al. WinoQueer: A Community-in-the-Loop Benchmark for Anti-LGBTQ+ Bias in Large Language Models[EB/OL]. 2023: 2306.15087. http://arxiv.org/abs/2306.15087v1.

[46] Navigli R, Conia S, Ross B. Biases in Large Language Models: Origins, Inventory, and Discussion[J]. *Journal of Data and Information Quality*, 2023, 15(2): 10.

[47] Hu T C, Kyrychenko Y, Rathje S, et al. Generative Language Models Exhibit Social Identity Biases[EB/OL]. 2023: 2310.15819. http://arxiv.org/abs/2310.15819v1.

[48] Phute M, Helbling A, Hull M, et al. LLM Self Defense: By Self Examination, LLMS Know they Are Being Tricked[EB/OL]. 2023: 2308.07308. http://arxiv.org/abs/2308.07308v4.

[49] Shanahan M, McDonell K, Reynolds L. Role Play with Large Language Models[J]. *Nature*, 2023, 623(7987): 493-498.

[50] Ramezani A, Xu Y. Knowledge of Cultural Moral Norms in Large Language Models[EB/OL]. 2023: 2306.01857. http://arxiv.org/abs/2306.01857v1.

[51] Yu C, Jeoung S, Kasi A, et al. Unlearning Bias in Language Models by Partitioning Gradients[C]. *Findings of the Association for Computational Linguistics*: *ACL 2023*, 2023: 6032-6048.

[52] Huang L, Yu W J, Ma W T, et al. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions[EB/OL]. 2023: 2311.05232. http://arxiv.org/abs/2311.05232v1.

[53] Chiang C H, Lee H Y. Can Large Language Models Be an Alternative to Human Evaluations? [EB/OL]. 2023: 2305.01937. http://arxiv.org/abs/2305.01937v1.

[54] Hardt D. Ellipsis-Dependent Reasoning: A New Challenge for Large Language Models[C]. *The 61st Annual Meeting of the Association for Computational Linguistics* (*Volume 2*: *Short Papers*), 2023: 39-47.

[55] Ho N, Schmid L, Yun S Y. Large Language Models Are Reasoning Teachers[EB/OL]. 2022: 2212.10071. http://arxiv.org/abs/2212.10071v2.

[56] Jones C R, Bergen B K. Does GPT-4 Pass the Turing Test? [EB/OL]. 2023: 2310.20216. http://arxiv.org/abs/2310.20216v2.

[57] Jones C R, Bergen B K. People Cannot Distinguish GPT-4 from a Human in a Turing Test[EB/OL]. 2024: 2405.08007. http://arxiv.org/abs/2405.08007v1.

[58] Tjuatja L, Chen V, Wu S T, et al. Do LLMS Exhibit Human-Like Response Biases? A Case Study in Survey Design[EB/OL]. 2023: 2311.04076. http://arxiv.org/abs/2311.04076v5.

[59] Shen T H, Jin R R, Huang Y F, et al. Large Language Model Alignment: A Survey[EB/OL]. 2023: 2309.15025. http://arxiv.org/abs/2309.15025v1.

[60] Wang Y X, Teng Y, Huang K X, et al. Fake Alignment: Are LLMS Really Aligned Well? [EB/OL]. 2023: 2311.05915. http://arxiv.org/abs/2311.05915v3.

[61] Jiang S Y, Chen X S, Tang R. Prompt Packer: Deceiving LLMS through Compositional Instruction with Hidden Attacks[EB/OL]. 2023: 2310.10077. http://arxiv.org/abs/2310.10077v1.

[62] Qiu H C, Zhang S, Li A Q, et al. Latent Jailbreak: A Benchmark for Evaluating Text Safety and Output Robustness of Large Language Models[EB/OL]. 2023: 2307.08487. http://arxiv.org/abs/2307.08487v3.

[63] Lapid R, Langberg R, Sipper M. Open Sesame! Universal Black Box Jailbreaking of Large Language Models[EB/OL]. 2023: 2309.01446. http://arxiv.org/abs/2309.01446v3.

[64] Shayegani E, Dong Y, Abu-Ghazaleh N. Plug and Pray: Exploiting off-the-shelf components of Multi-Modal Models[EB/OL]. 2023: ArXiv Preprint ArXiv:2307.14539.

[65] Qi X Y, Huang K X, Panda A, et al. Visual Adversarial Examples Jailbreak Aligned Large Language Models[EB/OL]. 2023: 2306.13213. http://arxiv.org/abs/2306.13213v2.

[66] Bagdasaryan E, Hsieh T Y, Nassi B, et al. Abusing Images and Sounds for Indirect Instruction Injection in Multi-Modal LLMS[EB/OL]. 2023: 2307.10490. http://arxiv.org/abs/2307.10490v4.

[67] Qu Y T, Shen X Y, He X L, et al. Unsafe Diffusion: On the Generation of Unsafe Images and Hateful Memes from Text-to-Image Models[EB/OL]. 2023: 2305.13873. http://arxiv.org/abs/2305.13873v2.

[68] Cao B C, Cao Y P, Lin L, et al. Defending Against Alignment-Breaking Attacks via Robustly Aligned LLM[EB/OL]. 2023: 2309.14348. http://arxiv.org/abs/2309.14348v2.

[69] Glukhov D, Shumailov I, Gal Y, et al. LLM Censorship: A Machine Learning Challenge or a Computer Security Problem? [EB/OL]. 2023: 2307.10719. http://arxiv.org/abs/2307.10719v1.

[70] Liu C, Zhao F, Qing L, et al. A chinese prompt attack dataset for llms with evil content[EB/OL]. 2023: ArXiv Preprint ArXiv:2309.11830.

[71] Yan S Q, Gu J C, Zhu Y, et al. Corrective Retrieval Augmented Generation[EB/OL]. 2024: 2401.15884. http://arxiv.org/abs/2401.15884v2.

[72] Mozes M, He X L, Kleinberg B, et al. Use of LLMS for Illicit Purposes: Threats, Prevention Measures, and Vulnerabilities [EB/OL]. 2023: 2308.12833. http://arxiv.org/abs/2308.12833v1.

[73] Bhandari P, Brennan H. Trustworthiness of Children Stories Generated by Large Language Models[C]. *The 16th International Natural Language Generation Conference*, 2023: 352-361.

[74] Uchendu A, Lee J, Shen H, et al. Does Human Collaboration Enhance the Accuracy of Identifying LLM-Generated Deepfake Texts?[J]. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, 2023, 11(1): 163-174.

[75] Li S Y, Lin X, Liu Y J, et al. Trustworthy AI-Generative Content in Intelligent 6G Network: Adversarial, Privacy, and Fairness [EB/OL]. 2024: 2405.05930. http://arxiv.org/abs/2405.05930v1.

[76] Mo L B, Wang B S, Chen M H, et al. How Trustworthy Are Open-Source LLMS? An Assessment under Malicious Demonstrations Shows Their Vulnerabilities[EB/OL]. 2023: 2311.09447. http://arxiv.org/abs/2311.09447v2.

[77] Kumar A, Singh S, Murty S V, et al. The Ethics of Interaction: Mitigating Security Threats in LLMS[EB/OL]. 2024: 2401.12273. http://arxiv.org/abs/2401.12273v1.

[78] Penedo G, Malartic Q, Hesslow D, et al. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data only[J]. *Advances in Neural Information Processing Systems*, 2024, 36.

[79] Dai J Z, Chen C S, Li Y F. A Backdoor Attack Against LSTM-Based Text Classification Systems[J]. *IEEE Access*, 2019, 7: 138872-138878.

[80] Carlini N. A LLM Assisted Exploitation of AI-Guardian[EB/OL]. 2023: 2307.15008. http://arxiv.org/abs/2307.15008v1.

[81] Zhu H, Zhang S Z, Chen K. AI-Guardian: Defeating Adversarial Attacks Using Backdoors[C]. *2023 IEEE Symposium on Security and Privacy*, 2023: 701-718.

[82] Adversarial Attacks on LLMs. OpenAI. https://lilianweng.github.io/posts/2023-10-25-adv-attack-llm/. Oct. 2023.

[83] Chen Y F, Arunasalam A, Celik Z B. Can Large Language Models Provide Security & Privacy Advice? Measuring the Ability of LLMS to Refute Misconceptions[C]. *The 39th Annual Computer Security Applications Conference*, 2023: 366-378.

[84] Owasp top 10 for large language model applications. OWASP Foundation. https://owasp.org/www-project-top-10-for-large-language-model-applications/. Jan. 2024.

[85] Shen X Y, Chen Z Y, Backes M, et al. "Do anything now": Characterizing and Evaluating In-the-Wild Jailbreak Prompts on Large Language Models[EB/OL]. 2023: 2308.03825. http://arxiv.org/abs/2308.03825v2.

[86] Wang Z H, Xie W, Chen K, et al. Self-Deception: Reverse Penetrating the Semantic Firewall of Large Language Models[EB/OL]. 2023: 2308.11521. http://arxiv.org/abs/2308. 11521v2.

[87] Liu Y, Deng G L, Xu Z Z, et al. Jailbreaking ChatGPT via Prompt Engineering: An Empirical Study[EB/OL]. 2023: 2305.13860. http://arxiv.org/abs/2305.13860v2.

[88] Wei A, Haghtalab N, Steinhardt J. Jailbroken: How Does LLM Safety Training Fail? [EB/OL]. 2023: 2307.02483. http://arxiv.org/abs/2307.02483v1.

[89] Xu X L, Kong K Y, Liu N, et al. An LLM Can Fool Itself: A Prompt-Based Adversarial Attack[EB/OL]. 2023: 2310.13345. http://arxiv.org/abs/2310.13345v1.

[90] Greshake K, Abdelnabi S, Mishra S, et al. Not what You've Signed up for: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection[C]. *The 16th ACM Workshop on Artificial Intelligence and Security*, 2023: 79-90.

[91] Liu Y, Deng G L, Li Y K, et al. Prompt Injection Attack Against LLM-Integrated Applications[EB/OL]. 2023: 2306.05499. http://arxiv.org/abs/2306.05499v2.

[92] Yi J W, Xie Y Q, Zhu B, et al. Benchmarking and Defending Against Indirect Prompt Injection Attacks on Large Language Models[EB/OL]. 2023: 2312.14197. http://arxiv.org/abs/2312.14197v3.

[93] Yuan Y L, Jiao W X, Wang W X, et al. GPT-4 Is too Smart to Be Safe: Stealthy Chat with LLMS via Cipher[EB/OL]. 2023: 2308.06463. http://arxiv.org/abs/2308.06463v2.

[94] Yan J, Yadav V, Li S, et al. Backdooring Instruction-Tuned Large Language Models with Virtual Prompt Injection[C] *NeurIPS 2023 Workshop on Backdoors in Deep Learning-The Good, the Bad, and the Ugly*, 2023.

[95] Carlini N, Tramer F, Wallace E, et al. Extracting Training Data from Large Language Models[EB/OL]. 2020: 2012.07805. http://arxiv.org/abs/2012.07805v2.

[96] Birch L, Hackett W, Trawicki S, et al. Model Leeching: An Extraction Attack Targeting LLMS[EB/OL]. 2023: 2309.10544. http://arxiv.org/abs/2309.10544v1.

[97] Zou A, Wang Z F, Carlini N, et al. Universal and Transferable Adversarial Attacks on Aligned Language Models[EB/OL]. 2023: 2307.15043. http://arxiv.org/abs/2307.15043v2.

[98] Zhao S, Wen J M, Tuan L A, et al. Prompt as Triggers for Backdoor Attack: Examining the Vulnerability in Language Models[EB/OL]. 2023: 2305.01219. http://arxiv.org/abs/2305.01219v6.

[99] Yan J, Gupta V, Ren X. BITE: Textual Backdoor Attacks with Iterative Trigger Injection[C]. *The 61st Annual Meeting of the Association for Computational Linguistics* (*Volume 1*: *Long Papers*), 2023: 12951-12968.

[100] Xu N, Wang F, Zhou B, et al. Cognitive Overload: Jailbreaking Large Language Models with Overloaded Logical Thinking[EB/OL]. 2023: 2311.09827. http://arxiv.org/abs/2311.09827v2.

[101] Deng G L, Liu Y, Wang K L, et al. Pandora: Jailbreak GPTS by Retrieval Augmented Generation Poisoning[EB/OL]. 2024: 2402.08416. http://arxiv.org/abs/2402.08416v1.

[102] Wallace E, Zhao T Z, Feng S, et al. Concealed Data Poisoning Attacks on NLP Models[EB/OL]. 2020: 2010.12563. http://arxiv.org/abs/2010.12563v2.

[103] Yao H W, Lou J, Qin Z. PoisonPrompt: Backdoor Attack on Prompt-Based Large Language Models[EB/OL]. 2023: 2310.12439. http://arxiv.org/abs/2310.12439v2.

[104] Aghakhani H, Dai W, Manoel A, et al. TrojanPuzzle: Covertly Poisoning Code-Suggestion Models[EB/OL]. 2023: 2301.02344. http://arxiv.org/abs/2301.02344v2.

[105] Yang H M, Xiang K L, Ge M Y, et al. A Comprehensive Overview of Backdoor Attacks in Large Language Models within Communication Networks[EB/OL]. 2023: 2308.14367. http://arxiv.org/abs/2308.14367v2.

[106] Wallace E, Zhao T Z, Feng S, et al. Concealed Data Poisoning Attacks on NLP Models[EB/OL]. 2020: 2010.12563. http://arxiv.org/abs/2010.12563v2.

[107] Xu J S, Ma M D, Wang F, et al. Instructions as Backdoors: Backdoor Vulnerabilities of Instruction Tuning for Large Language Models[EB/OL]. 2023: 2305.14710. http://arxiv.org/abs/2305.14710v2.

[108] Huang H, Zhao Z Y, Backes M, et al. Composite Backdoor Attacks Against Large Language Models[EB/OL]. 2023: 2310.07676. http://arxiv.org/abs/2310.07676v2.

[109] Yan J, Yadav V, Li S Y, et al. Backdooring Instruction-Tuned Large Language Models with Virtual Prompt Injection[EB/OL]. 2023: 2307.16888. http://arxiv.org/abs/2307.16888v3.

[110] Carlini N, Nasr M, Choquette-Choo C A, et al. Are Aligned Neural Networks Adversarially Aligned? [EB/OL]. 2023: 2306.15447. http://arxiv.org/abs/2306.15447v2.

[111] Sun W S, Chen Y C, Tao G H, et al. Backdooring Neural Code Search[EB/OL]. 2023: 2305.17506. http://arxiv.org/abs/2305.17506v2.

[112] Qiang Y, Zhou X Y, Zhu D X. Hijacking Large Language Models via Adversarial In-Context Learning[EB/OL]. 2023: 2311.09948. http://arxiv.org/abs/2311.09948v1.

[113] Zhan Q S, Fang R, Bindu R, et al. Removing RLHF Protections in GPT-4 via Fine-Tuning[EB/OL]. 2023: 2311.05553. http://arxiv.org/abs/2311.05553v3.

[114] Liu X G, Xu N, Chen M H, et al. AutoDAN: Generating Stealthy Jailbreak Prompts on Aligned Large Language Models[EB/OL]. 2023: 2310.04451. http://arxiv.org/abs/2310.04451v2.

[115] Salem A, Paverd A, Köpf B. Maatphor: Automated Variant Analysis for Prompt Injection Attacks[EB/OL]. 2023: 2312.11513. http://arxiv.org/abs/2312.11513v1.

[116] Mehrotra A, Zampetakis M, Kassianik P, et al. Tree of Attacks: Jailbreaking Black-Box LLMS Automatically[EB/OL]. 2023: 2312.02119. http://arxiv.org/abs/2312.02119v2.

[117] Yu J H, Lin X W, Yu Z, et al. GPTFUZZER: Red Teaming Large Language Models with Auto-Generated Jailbreak Prompts[EB/OL]. 2023: 2309.10253. http://arxiv.org/abs/2309.10253v2.

[118] Ding P, Kuang J, Ma D, et al. A Wolf in Sheep's Clothing: Generalized Nested Jailbreak Prompts Can Fool Large Language Models Easily[EB/OL]. 2023: 2311.08268. http://arxiv.org/abs/2311.08268v4.

[119] Guo P, Liu F, Lin X, et al. L-AutoDA: Leveraging Large Language Models for Automated Decision-Based Adversarial Attacks [EB/OL]. 2024: 2401.15335. http://arxiv.org/abs/2401.15335v2.

[120] Yang X K, Tang X H, Hu S L, et al. Chain of Attack: A Semantic-Driven Contextual Multi-Turn Attacker for LLM [EB/OL]. 2024: 2405.05610. http://arxiv.org/abs/2405.05610v1.

[121] Hui B, Yuan H L, Gong N, et al. PLeak: Prompt Leaking Attacks Against Large Language Model Applications[EB/OL]. 2024: 2405.06823. http://arxiv.org/abs/2405.06823v2.

[122] How to defend against simple prompt injection in GPTs. Weibo. https://m.weibo.cn/detail/4969278393025376. Nov. 2023.

[123] Jain N, Schwarzschild A, Wen Y X, et al. Baseline Defenses for Adversarial Attacks Against Aligned Language Models[EB/OL]. 2023: 2309.00614. http://arxiv.org/abs/2309.00614v2.

[124] Patil V, Hase P, Bansal M. Can Sensitive Information Be Deleted

from LLMS? Objectives for Defending Against Extraction Attacks[EB/OL]. 2023: 2309.17410. http://arxiv.org/abs/2309.17410v1.

[125] Liu Y, Jia Y, Geng R, et al. Prompt injection attacks and defenses in llm-integrated applications[EB/OL]. 2023: ArXiv Preprint ArXiv:2310.12815.

[126] Alon G, Kamfonas M. Detecting Language Model Attacks with Perplexity[EB/OL]. 2023: 2308.14132. http://arxiv.org/abs/2308.14132v3.

[127] Deng G L, Liu Y, Li Y K, et al. MasterKey: Automated Jailbreak across Multiple Large Language Model Chatbots[EB/OL]. 2023: 2307.08715. http://arxiv.org/abs/2307.08715v2.

[128] Zhang Z X, Yang J X, Ke P, et al. Defending Large Language Models Against Jailbreaking Attacks through Goal Prioritization [EB/OL]. 2023: 2311.09096. http://arxiv.org/abs/2311.09096v1.

[129] Zhang Z Y, Zhang Q Z, Foerster J. PARDEN, Can You Repeat That? Defending Against Jailbreaks via Repetition[EB/OL]. 2024: 2405.07932. http://arxiv.org/abs/2405.07932v2.

[130] Surendrababu H K. Model Agnostic Approach for NLP Backdoor Detection[C]. *2023 IEEE Colombian Conference on Applications of Computational Intelligence*, 2023: 1-6.

[131] Yang W H, Gao J D, Mirzasoleiman B. Better Safe than Sorry: Pre-Training CLIP Against Targeted Data Poisoning and Backdoor Attacks[EB/OL]. 2023: 2310.05862. http://arxiv.org/abs/2310.05862v1.

[132] He X L, Wang J, Rubinstein B, et al. IMBERT: Making BERT Immune to Insertion-Based Backdoor Attacks[EB/OL]. 2023: 2305.16503. http://arxiv.org/abs/2305.16503v1.

[133] Deng B Y, Wang W J, Feng F L, et al. Attack Prompt Generation for Red Teaming and Defending Large Language Models[EB/OL]. 2023: 2310.12505. http://arxiv.org/abs/2310.12505v1.

[134] Zhou X, Lu Y, Ma R T, et al. Making Harmful Behaviors Unlearnable for Large Language Models[EB/OL]. 2023: 2311.02105. http://arxiv.org/abs/2311.02105v1.

[135] Xiao Y J, Jin Y Q, Bai Y S, et al. Large Language Models Can Be Good Privacy Protection Learners[EB/OL]. 2023: 2310.02469. http://arxiv.org/abs/2310.02469v1.

[136] Wen R, Wang T H, Backes M, et al. Last One Standing: A Comparative Analysis of Security and Privacy of Soft Prompt Tuning, LoRA, and In-Context Learning[EB/OL]. 2023: 2310. 11397. http://arxiv.org/abs/2310.11397v1.

[137] Chen X Y, Tang S Y, Zhu R, et al. The Janus Interface: How Fine-Tuning in Large Language Models Amplifies the Privacy Risks[EB/OL]. 2023: 2310.15469. http://arxiv.org/abs/2310.15469v2.

[138] Li H R, Chen Y L, Zheng Z H, et al. Backdoor Removal for Generative Large Language Models[EB/OL]. 2024: 2405.07667. http://arxiv.org/abs/2405.07667v1.

[139] Hubinger E, Denison C, Mu J, et al. Sleeper Agents: Training Deceptive LLMS that Persist through Safety Training[EB/OL]. 2024: 2401.05566. http://arxiv.org/abs/2401.05566v3.

[140] Ye J Y, Du M N, Wang G L. DataFrame QA: A Universal LLM Framework on DataFrame Question Answering without Data Exposure[EB/OL]. 2024: 2401.15463. http://arxiv.org/abs/2401.15463 v1.

[141] Zhong V, Xiong C M, Socher R. Seq2SQL: Generating Structured Queries from Natural Language Using Reinforcement Learning [EB/OL]. 2017: 1709.00103. http://arxiv.org/abs/1709.00103v7.

[142] Peng W J, Yi J W, Wu F Z, et al. Are You Copying My Model? Protecting the Copyright of Large Language Models for EaaS via Backdoor Watermark[EB/OL]. 2023: 2305.10036. http://arxiv.org/abs/2305.10036v3.

[143] Chu T, Song Z, Yang C. How to Protect Copyright Data in Optimization of Large Language Models? [EB/OL]. 2023: 2308.12247. http://arxiv.org/abs/2308.12247v1.

[144] Liu Y P, Bu Y H. Adaptive Text Watermark for Large Language Models[EB/OL]. 2024: 2401.13927. http://arxiv.org/abs/2401.13927v1.

[145] Qu W J, Yin D, He Z X, et al. Provably Robust Multi-Bit Watermarking for AI-Generated Text via Error Correction Code[EB/OL]. 2024: 2401.16820. http://arxiv.org/abs/2401. 16820v2.

[146] Xu J S, Wang F, Ma M D, et al. Instructional Fingerprinting of Large Language Models[EB/OL]. 2024: 2401.12255. http://arxiv.org/abs/2401.12255v2.

[147] Wang H R, Shu K. Backdoor Activation Attack: Attack Large Language Models Using Activation Steering for Safety-Alignment [EB/OL]. 2023: 2311.09433. http://arxiv.org/abs/2311.09433v2.

[148] Yan R, Li Y, Li W, et al. Teach Large Language Models to Forget Privacy[EB/OL]. 2024: ArXiv Preprint ArXiv:2401.00870.

[149] Mu N, Chen S, Wang Z F, et al. Can LLMS Follow Simple Rules? [EB/OL]. 2023: 2311.04235. http://arxiv.org/abs/2311.04235v3.

[150] Han S S, Buyukates B, Hu Z J, et al. FedMLSecurity: A Benchmark for Attacks and Defenses in Federated Learning and Federated LLMS[EB/OL]. 2023: 2306.04959. http://arxiv.org/abs/2306.04959v4.

[151] Li J, Liu Y, Liu C Y, et al. A Cross-Language Investigation into Jailbreak Attacks in Large Language Models[EB/OL]. 2024: 2401.16765. http://arxiv.org/abs/2401.16765v1.

[152] Chen Y, Mendes E, Das S, et al. Can Language Models Be Instructed to Protect Personal Information? [EB/OL]. 2023: 2310.02224. http://arxiv.org/abs/2310.02224v1.

[153] Chen Y Y, Lent H, Bjerva J. Text Embedding Inversion Security for Multilingual Language Models[EB/OL]. 2024: 2401.12192. http://arxiv.org/abs/2401.12192v2.

[154] Kereopa-Yorke B. Building Resilient SMEs: Harnessing Large Language Models for Cyber Security in Australia[EB/OL]. 2023:

2306.02612. http://arxiv.org/abs/2306.02612v1.

[155] Lin Z L, Cui J, Liao X J, et al. Malla: Demystifying Real-World Large Language Model Integrated Malicious Services[EB/OL]. 2024: 2401.03315. http://arxiv.org/abs/2401.03315v1.

[156] Shashwat K, Hahn F, Ou X M, et al. A Preliminary Study on Using Large Language Models in Software Pentesting[EB/OL]. 2024: 2401.17459. http://arxiv.org/abs/2401.17459v1.

[157] Happe A, Cito J. Getting Pwn'd by AI: Penetration Testing with Large Language Models[C]. *The 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2023: 2082-2086.

[158] Deng G L, Liu Y, Mayoral-Vilches V, et al. PentestGPT: An LLM-Empowered Automatic Penetration Testing Tool[EB/OL]. 2023: 2308.06782. http://arxiv.org/abs/2308.06782v1.

[159] Pedro R, Castro D, Carreira P, et al. From Prompt Injections to SQL Injection Attacks: How Protected Is Your LLM-Integrated Web Application? [EB/OL]. 2023: 2308.01990. http://arxiv.org/abs/2308.01990v3.

[160] Honeypot. Kaspersky. https://www.kaspersky.com.cn/resource-center/threats/what-is-a-honeypot.

[161] Sladić M, Valeros V, Catania C, et al. LLM in the Shell: Generative Honeypots[EB/OL]. 2023: 2309.00155. http://arxiv.org/abs/2309.00155v2.

[162] He H Y, Yang Z G, Chen X N. PERT: Payload Encoding Representation from Transformer for Encrypted Traffic Classification[C]. *2020 ITU Kaleidoscope*: *Industry-Driven Digital Transformation*, 2020: 1-8.

[163] Lin X J, Xiong G, Gou G P, et al. ET-BERT: A Contextualized Datagram Representation with Pre-Training Transformers for Encrypted Traffic Classification[C]. *The ACM Web Conference 2022*, 2022: 633-642.

[164] Sarabi A, Yin T X, Liu M Y. An LLM-Based Framework for Fingerprinting Internet-Connected Devices[C]. *The 2023 ACM on Internet Measurement Conference*, 2023: 478-484.

[165] Kholgh D K, Kostakos P. PAC-GPT: A Novel Approach to Generating Synthetic Network Traffic with GPT-3[J]. *IEEE Access*, 2023, 11: 114936-114951.

[166] Guastalla M, Li Y, Hekmati A, et al. Application of Large Language Models to DDoS Attack Detection[C] *International Conference on Security and Privacy in Cyber-Physical Systems and Smart Vehicles*, 2023: 83-99.

[167] Ferrag M A, Ndhlovu M, Tihanyi N, et al. Revolutionizing Cyber Threat Detection with Large Language Models: A Privacy- Preserving BERT-Based Lightweight Model for IoT/IIoT Devices [EB/OL]. 2023: 2306.14263. http://arxiv.org/abs/2306.14263v2.

[168] Patel U, Yeh F C, Gondhalekar C. CANAL — Cyber Activity News Alerting Language Model: Empirical Approach Vs. Expensive LLM[EB/OL]. 2024: 2405.06772. http://arxiv.org/abs/2405.06772v1.

[169] Mitra S, Neupane S, Chakraborty T, et al. LOCALINTEL: Generating Organizational Threat Intelligence from Global and Local Cyber Knowledge[EB/OL]. 2024: 2401.10036. http://arxiv.org/abs/2401.10036v1.

[170] Shafee S, Bessani A, Ferreira P M. Evaluation of LLM Chatbots for OSINT-Based Cyber Threat Awareness[EB/OL]. 2024: 2401.15127. http://arxiv.org/abs/2401.15127v3.

[171] Hasan S M, Alotaibi A M, Talukder S, et al. Distributed Threat Intelligence at the Edge Devices: A Large Language Model-Driven Approach[EB/OL]. 2024: 2405.08755. http://arxiv.org/abs/2405.08755v2.

[172] Fayyazi R, Yang S J. On the Uses of Large Language Models to Interpret Ambiguous Cyberattack Descriptions[EB/OL]. 2023: 2306.14062. http://arxiv.org/abs/2306.14062v2.

[173] Hassanin M, Keshk M, Salim S, et al. PLLM-CS: Pre-Trained Large Language Model (LLM) for Cyber Threat Detection in Satellite Networks[EB/OL]. 2024: 2405.05469. http://arxiv.org/abs/2405.05469v1.

[174] Cui H W, Du Y Y, Yang Q, et al. LLMind: Orchestrating AI and IoT with LLM for Complex Task Execution[EB/OL]. 2023: 2312.09007. http://arxiv.org/abs/2312.09007v3.

[175] Shumailov I, Zhao Y R, Bates D, et al. Sponge Examples: Energy-Latency Attacks on Neural Networks[C]. *2021 IEEE European Symposium on Security and Privacy*, 2021: 212-231.

[176] Saha D, Tarek S, Yahyaei K, et al. LLM for SoC Security: A Paradigm Shift[EB/OL]. 2023: 2310.06046. http://arxiv.org/abs/2310.06046v1.

[177] Fu W M, Yang K C, Dutta R G, et al. LLM4SecHW: Leveraging Domain-Specific Large Language Model for Hardware Debugging[C]. *2023 Asian Hardware Oriented Security and Trust Symposium*, 2023: 1-6.

[178] Akyash M, Kamali H M. Evolutionary Large Language Models for Hardware Security: A Comparative Survey[EB/OL]. 2024: 2404.16651. http://arxiv.org/abs/2404.16651v1.

[179] Wang Z, Alrahis L, Mankali L, et al. LLMS and the Future of Chip Design: Unveiling Security Risks and Building Trust[EB/OL]. 2024: 2405.07061. http://arxiv.org/abs/2405.07061v1.

[180] Computing G. Chat GPT in smart home systems: prospects, risks, and benefits[J]. 2023.

[181] Beckerich M, Plein L, Coronado S. RatGPT: Turning Online LLMS into Proxies for Malware Attacks[EB/OL]. 2023: 2308.09183. http://arxiv.org/abs/2308.09183v2.

[182] Shah M A, Sharma R, Dhamyal H, et al. LoFT: Local Proxy Fine-Tuning for Improving Transferability of Adversarial Attacks Against Large Language Model[EB/OL]. 2023: 2310.04445. http://arxiv.org/abs/2310.04445v2.

[183] Xu H T, Han L Y, Yang Q R, et al. Penetrative AI: Making LLMS

Comprehend the Physical World[C]. *The 25th International Workshop on Mobile Computing Systems and Applications*, 2024: 1-7.

[184] Botacin M. GPThreats-3: Is Automatic Malware Generation a Threat? [C]. *2023 IEEE Security and Privacy Workshops*, 2023: 238-254.

[185] Pa Pa Y M, Tanizaki S, Kou T, et al. An Attacker's Dream? Exploring the Capabilities of ChatGPT for Developing Malware[C]. *2023 Cyber Security Experimentation and Test Workshop*, 2023: 10-18.

[186] Monje A, Monje A, Hallman R A, et al. Being a bad influence on the kids: Malware generation in less than five minutes using chatgpt[J]. 2023.

[187] Ding H T, Kumar V, Tian Y C, et al. A Static Evaluation of Code Completion by Large Language Models[EB/OL]. 2023: 2306.03203. http://arxiv.org/abs/2306.03203v1.

[188] He J X, Vechev M. Large Language Models for Code: Security Hardening and Adversarial Testing[C]. *The 2023 ACM SIGSAC Conference on Computer and Communications Security*, 2023: 1865-1879.

[189] Sandoval G, Pearce H, Nys T, et al. Lost at C: A User Study on the Security Implications of Large Language Model Code Assistants [EB/OL]. 2022: 2208.09727. http://arxiv.org/abs/2208.09727v4.

[190] Pearce H, Tan B, Krishnamurthy P, et al. Pop Quiz! Can a Large Language Model Help with Reverse Engineering? [EB/OL]. 2022: 2202.01142. http://arxiv.org/abs/2202.01142v1.

[191] Pearce H, Tan B, Ahmad B, et al. Examining Zero-Shot Vulnerability Repair with Large Language Models[C]. *2023 IEEE Symposium on Security and Privacy*, 2023: 2339-2356.

[192] Ahmad B, Tan B, Karri R, et al. FLAG: Finding Line Anomalies (in Code) with Generative AI[EB/OL]. 2023: 2306.12643. http://arxiv.org/abs/2306.12643v1.

[193] Xia C S, Ding Y F, Zhang L M. Revisiting the Plastic Surgery Hypothesis via Large Language Models[EB/OL]. 2023: 2303.10494. http://arxiv.org/abs/2303.10494v1.

[194] Jin M, Shahriar S, Tufano M, et al. InferFix: End-to-End Program Repair with LLMS[EB/OL]. 2023: 2303.07263. http://arxiv.org/abs/2303.07263v1.

[195] Noever D. Can Large Language Models Find and Fix Vulnerable Software? [EB/OL]. 2023: 2308.10345. http://arxiv.org/abs/2308.10345v1.

[196] Lin Y Z, Mamun M, Chowdhury M A, et al. HW-V2.W-Map: Hardware Vulnerability to Weakness Mapping Framework for Root Cause Analysis with GPT-Assisted Mitigation Suggestion[EB/OL]. 2023: 2312.13530. http://arxiv.org/abs/2312.13530v1.

[197] Liu X, Tan Y, Xiao Z H, et al. Not the End of Story: An Evaluation of ChatGPT-Driven Vulnerability Description Mappings[C]. *Findings of the Association for Computational Linguistics*: *ACL 2023*, 2023: 3724-3731.

[198] Shestov A, Levichev R, Mussabayev R, et al. Finetuning Large Language Models for Vulnerability Detection[EB/OL]. 2024: 2401.17010. http://arxiv.org/abs/2401.17010v4.

[199] Wang R K, Li Z J, Wang C Z, et al. NAVRepair: Node-Type Aware C/C++ Code Vulnerability Repair[EB/OL]. 2024: 2405.04994. http://arxiv.org/abs/2405.04994v1.

[200] Ahmad B, Thakur S, Tan B, et al. Fixing Hardware Security Bugs with Large Language Models[EB/OL]. 2023: 2302.01215. http://arxiv.org/abs/2302.01215v1.

[201] Tihanyi N, Bisztray T, Jain R, et al. The FormAI Dataset: Generative AI in Software Security through the Lens of Formal Verification[C]. *The 19th International Conference on Predictive Models and Data Analytics in Software Engineering*, 2023: 33-43.

[202] Wang J X, Cao L W, Luo X T, et al. Enhancing Large Language Models for Secure Code Generation: A Dataset-Driven Study on Vulnerability Mitigation[EB/OL]. 2023: 2310.16263. http://arxiv.org/abs/2310.16263v1.

[203] Yu J X, Liang P, Fu Y J, et al. Security Code Review by LLMS: A Deep Dive into Responses[EB/OL]. 2024: 2401.16310. http://arxiv.org/abs/2401.16310v1.

[204] Liu T, Deng Z Z, Meng G Z, et al. Demystifying RCE Vulnerabilities in LLM-Integrated Apps[EB/OL]. 2023: 2309.02926. http://arxiv.org/abs/2309.02926v2.

[205] Kande R, Pearce H, Tan B, et al. LLM-Assisted Generation of Hardware Assertions[EB/OL]. 2023: 2306.14027. http://arxiv.org/abs/2306.14027v1.

[206] Tony C, Mutas M, Díaz Ferreyra N E, et al. LLMSecEval: A Dataset of Natural Language Prompts for Security Evaluations [EB/OL]. 2023: 2303.09384. http://arxiv.org/abs/2303.09384v1.

[207] Fu Y J, Liang P, Tahir A, et al. Security Weaknesses of Copilot Generated Code in GitHub[EB/OL]. 2023: 2310.02059. http://arxiv.org/abs/2310.02059v2.

[208] Ullah S, Han M J, Pujar S, et al. LLMS Cannot Reliably Identify and Reason about Security Vulnerabilities (Yet?): A Comprehensive Evaluation, Framework, and Benchmarks[EB/OL]. 2023: 2312.12575. http://arxiv.org/abs/2312.12575v2.

[209] Fu X Y, Hu Y S, Li B Z, et al. BLINK: Multimodal Large Language Models Can See but Not Perceive[EB/OL]. 2024: 2404.12390. http://arxiv.org/abs/2404.12390v3.

[210] Bethany M, Galiopoulos A, Bethany E, et al. Large Language Model Lateral Spear Phishing: A Comparative Study in Large-Scale Organizational Settings[EB/OL]. 2024: 2401.09727. http://arxiv.org/abs/2401.09727v1.

[211] Trad F, Chehab A. Prompt Engineering or Fine-Tuning? A Case Study on Phishing Detection with Large Language Models[J]. *Machine Learning and Knowledge Extraction*, 2024, 6(1): 367-384.

[212] Shou C F, Liu J, Lu D D, et al. LLM4Fuzz: Guided Fuzzing of Smart Contracts with Large Language Models[EB/OL]. 2024:

2401.11108. http://arxiv.org/abs/2401.11108v1.

[213] Zhao J J, Chen X, Yang G, et al. Automatic Smart Contract Comment Generation via Large Language Models and In-Context Learning[J]. *Information and Software Technology*, 2024, 168: 107405.

[214] Securitygpt. SANGFOR. https://www.securitygpt.com.cn/. Jan. 2024.

[215] Copilot for Security. Microsoft. https://www.microsoft.com/en-us/security/blog/2024/03/13/microsoft-copilot-for-security-is-generally-available-on-april-1-2024-with-new-capabilities/. Mar. 2024.

[216] Darktrace Addresses Generative AI Concerns with Introduction of AI Models That Help Protect Data Privacy and Intellectual Property. Darktrace. https://darktrace.com/news/darktrace-addresses-generative-ai-concerns. Jun. 2023.

[217] Ali Cloud security LLM. Alibaba Cloud. https://developer.aliyun.com/article/1366175. Nov. 2023.

[218] Bengio Y, Hinton G, Yao A, et al. Managing Extreme AI Risks Amid Rapid Progress[J]. *Science*, 2024, 384(6698): 842-845.

[219] LLMSEC. Baidu. https://anquan.baidu.com/product/llmsec. Aug. 2023.

付志远 于 2023 年在齐鲁工业大学通信工程专业获得学士学位。现在海南大学电子信息专业攻读硕士学位, CCF 学生会员。研究领域为网络空间安全。研究兴趣包括: 大语言模型安全、网络与信息系统安全。Email: fuzy@nipc.org.cn

陈骏帆 于 2022 年在中山大学物理学专业获得学士学位。现在海南大学网络空间安全专业攻读硕士学位。研究领域为人工智能安全和物联网安全。研究兴趣包括网络攻防技术和系统安全。Email: chenjunfan@hainanu.edu.cn

石岩松 于 2022 年在海南大学信息安全专业获得学士学位。现在海南大学电子信息专业攻读硕士学位。研究领域为网络与信息安全。研究兴趣包括网络攻防技术和大语言模型安全。Email: m15584360465@163.com

李益红 于 2016 年在西安电子科技大学获得博士学位。现任海南大学副教授, 硕士生导师。主要研究方向为机器学习、自然语言处理。Email: 990638@hainanu.edu.cn

张玉清 于 2000 年在西安电子科技大学获得博士学位。现任中国科学院大学/海南大学教授(双聘), 博士生导师, CCF 高级会员。主要研究方向为网路与信息系统安全。Email: zhangyq@nipc.org.cn

陈思宇 于 2023 年在成都师范学院计算机科学与技术专业获得学士学位。现在海南大学电子信息专业攻读硕士学位。研究领域为网络与信息安全。研究兴趣包括: 数据安全、人工智能安全。Email: chensiyu@hainanu.edu.cn

海翔 于 2022 年在长春大学网络工程专业获得学士学位。现在海南大学电子信息专业攻读硕士学位。研究领域为网络空间安全。研究兴趣包括: 大模型安全、网络安全。Email: haixiang@hainanu.edu.cn

李晓琦 于 2021 年在香港理工大学获得博士学位。现任海南大学副教授, 博士生导师, CCF 会员。主要研究方向为网路与信息系统安全。Email: csxqli@hainanu.edu.cn

岳秋玲 于 2020 年在北京邮电大学获得博士学位。现任海南大学讲师, 博士生导师。主要研究方向为应用密码学、隐私计算、区块链。Email: yueqiuling@hainanu.edu.cn