High-Confidence Computing 4 (2024) 100211

Contents lists available at ScienceDirect

High-Confidence Computing

journal homepage: www.sciencedirect.com/journal/high-confidence-computing

Review Article

A survey on large language model (LLM) security and privacy: The Good, The Bad, and The Ugly

Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, Yue Zhang *,[1]

Department of Computer Science, Drexel University, Philadelphia, PA 19104, USA

ARTICLE INFO

Article history:
Received 30 November 2023
Revised 17 January 2024
Accepted 28 January 2024
Available online 1 March 2024

Keywords:
Large Language Model (LLM)
LLM security
LLM privacy
ChatGPT
LLM attacks
LLM vulnerabilities

ABSTRACT

Large Language Models (LLMs), such as ChatGPT and Bard, have revolutionized natural language understanding and generation. They possess deep language comprehension, human-like text generation capabilities, contextual awareness, and robust problem-solving skills, making them invaluable in various domains (e.g., search engines, customer support, translation). In the meantime, LLMs have also gained traction in the security community, revealing security vulnerabilities and showcasing their potential in security-related tasks. This paper explores the intersection of LLMs with security and privacy. Specifically, we investigate how LLMs positively impact security and privacy, potential risks and threats associated with their use, and inherent vulnerabilities within LLMs. Through a comprehensive literature review, the paper categorizes the papers into "The Good" (beneficial LLM applications), "The Bad" (offensive applications), and "The Ugly" (vulnerabilities of LLMs and their defenses). We have some interesting findings. For example, LLMs have proven to enhance code security (code vulnerability detection) and data privacy (data confidentiality protection), outperforming traditional methods. However, they can also be harnessed for various attacks (particularly user-level attacks) due to their human-like reasoning abilities. We have identified areas that require further research efforts. For example, Research on model and parameter extraction attacks is limited and often theoretical, hindered by LLM parameter scale and confidentiality. Safe instruction tuning, a recent development, requires more exploration. We hope that our work can shed light on the LLMs' potential to both bolster and jeopardize cybersecurity.

1. Introduction

A large language model is the language model with massive parameters that undergoes pretraining tasks (e.g., masked language modeling and autoregressive prediction) to understand and process human language, by modeling the contextualized text semantics and probabilities from large amounts of text data. A capable LLM should have four key features [1]: (i) profound comprehension of natural language context; (ii) ability to generate human-like text; (iii) contextual awareness, especially in knowledge-intensive domains; (iv) strong instruction-following ability which is useful for problem-solving and decision-making.

There are a number of LLMs that were developed and released in 2023, gaining significant popularity. Notable examples include OpenAI's *ChatGPT* [2], Meta AI's *LLaMA* [3], and Databricks' Dolly 2.0 [4]. For instance, ChatGPT alone boasts a user base of over 180 million [5]. LLMs now offer a wide range of versatile applications across various domains. Specifically, they not only provide technical support to domains directly related to language processing (e.g., search engines [6,7], customer support [8], translation [9,10]) but also find utility in more general scenarios such as code generation [11], healthcare [12], finance [13], and education [14]. This showcases their adaptability and potential to streamline language-related tasks across diverse industries and contexts.

LLMs are gaining popularity within the security community. As of February 2023, a research study reported that GPT-3 uncovered 213 security vulnerabilities (only 4 turned out to be false positives) [15] in a code repository. In contrast, one of the leading commercial tools in the market detected only 99 vulnerabilities. More recently, several LLM-powered security papers have emerged in prestigious conferences. For instance, in IEEE S&P 2023, Hammond Pearce et al. [16] conducted a comprehensive investigation employing various commercially available LLMs, evaluating them across synthetic, hand-crafted, and real-world security bug scenarios. The results are promising, as LLMs successfully addressed all synthetic and hand-crafted scenarios. In NDSS 2024, a tool named Fuzz4All [17] showcased the use of LLMs for input generation and mutation, accompanied by an innovative autoprompting technique and fuzzing loop.

* Corresponding author.
E-mail address: yz899@drexel.edu (Y. Zhang).

[1] Given his role as Editor of this journal, Yue Zhang had no involvement in the peer-review of this article and had no access to information regarding its peer-review. Full responsibility for the editorial process for this article was delegated to Dr. Zhipeng Cai.

https://doi.org/10.1016/j.hcc.2024.100211

These remarkable initial attempts prompt us to delve into three crucial security-related research questions:

- ***RQ1***. *How do LLMs make a positive impact on security and privacy across diverse domains, and what advantages do they offer to the security community?*
- ***RQ2***. *What potential risks and threats emerge from the utilization of LLMs within the realm of cybersecurity?*
- ***RQ3***. *What vulnerabilities and weaknesses within LLMs, and how to defend against those threats?*

Findings. To comprehensively address these questions, we conducted a meticulous literature review and assembled a collection of 281 papers pertaining to the intersection of LLMs with security and privacy. We categorized these papers into three distinct groups: those highlighting security-beneficial applications (i.e., the good), those exploring applications that could potentially exert adverse impacts on security (i.e., the bad), and those focusing on the discussion of security vulnerabilities (alongside potential defense mechanisms) within LLMs (i.e., the ugly). To be more specific:

- **The Good** (Section 4): LLMs have a predominantly positive impact on the security community, as indicated by the most significant number of papers dedicated to enhancing security. Specifically, LLMs have made contributions to both code security and data security and privacy. In the context of code security, LLMs have been used for the whole life cycle of the code (e.g., secure coding, test case generation, vulnerable code detection, malicious code detection, and code fixing). In data security and privacy, LLMs have been applied to ensure data integrity, data confidentiality, data reliability, and data traceability. Meanwhile, Compared to state-of-the-art methods, most researchers found LLM-based methods to outperform traditional approaches.
- **The Bad** (Section 5): LLMs also have offensive applications against security and privacy. We categorized the attacks into five groups: hardware-level attacks (e.g., side-channel attacks), OS-level attacks (e.g., analyzing information from operating systems), software-level attacks (e.g., creating malware), network-level attacks (e.g., network phishing), and user-level attacks (e.g., misinformation, social engineering, scientific misconduct). User-level attacks, with 32 papers, are the most prevalent due to LLMs' human-like reasoning abilities. Those attacks threaten both security (e.g., malware attacks) and privacy (e.g., social engineering). Nowadays, LLMs lack direct access to OS and hardware-level functions. The potential threats of LLMs could escalate if they gain such access.
- **The Ugly** (Section 6): We explore the vulnerabilities and defenses in LLMs, categorizing vulnerabilities into two main groups: AI Model Inherent Vulnerabilities (e.g., data poisoning, backdoor attacks, training data extraction) and Non-AI Model Inherent Vulnerabilities (e.g., remote code execution, prompt injection, side channels). These attacks pose a dual threat, encompassing both security concerns (e.g., remote code execution attacks) and privacy issues (e.g., data extraction). Defenses for LLMs are divided into strategies placed in the architecture, and applied during the training and inference phases. Training phase defenses involve corpora cleaning, and optimization methods, while inference phase defenses include instruction pre-processing, malicious detection, and generation post-processing. These defenses collectively aim to enhance the security, robustness, and ethical alignment of LLMs. We found that model extraction, parameter extraction, and similar attacks have received limited research attention, remaining primarily theoretical with minimal practical exploration. The vast scale of LLM parameters makes traditional approaches less effective, and the confidentiality of powerful LLMs further shields them from conventional attacks. Strict censorship of LLM outputs challenges even black-box ML attacks. Meanwhile, research on the impact of model architecture on LLM safety is scarce, partly due to high computational costs. Safe instruction tuning, a recent development, requires further investigation.

Contributions. Our work makes a dual contribution. First, we are pioneers summarizing the role of LLMs in security and privacy. We delve deeply into the positive impacts of LLMs on security, their potential risks and threats, vulnerabilities in LLMs, and the corresponding defense mechanisms. Other surveys may focus on one or two specific aspects, such as beneficial applications, offensive applications, vulnerabilities, or defenses. To the best of our knowledge, our survey is the first to cover all three key aspects related to security and privacy for the first time. Second, we have made several interesting discoveries. For instance, our research reveals that LLMs contribute more positively than negatively to security and privacy. Moreover, we observe that most researchers concur that LLMs outperform state-of-the-art methods when employed for securing code or data. Concurrently, it becomes evident that user-level attacks are the most prevalent, largely owing to the human-like reasoning abilities exhibited by LLMs.

Roadmap. The rest of the paper is organized as follows. We begin with a brief introduction to LLM in Section 2. Section 3 presents the overview of our work. In Section 4, we explore the beneficial impacts of employing LLMs. Section 5 discusses the negative impacts on security and privacy. In Section 6, we discuss the prevalent threats, vulnerabilities associated with LLMs as well as the countermeasures to mitigate these risks. Section 7 discuss LLMs in other security related topics and possible directions. We conclude the paper in Section 9.

2. Background

2.1. Large Language Models (LLMs)

Large Language Models (LLMs) [18] represents an evolution from language models. Initially, language models were statistical in nature and laid the groundwork for computational linguistics. The advent of transformers has significantly increased their scale. This expansion, along with the use of extensive training corpora and advanced pre-training techniques is pivotal in areas such as AI for science, logical reasoning, and embodied AI. These models undergo extensive training on vast datasets to comprehend and produce text that closely mimics human language. Typically, LLMs are endowed with hundreds of billions, or even more, parameters, honed through the processing of massive textual data. They have spearheaded substantial advancements in the realm of Natural Language Processing (NLP) [19] and find applications in a multitude of fields (e.g., risk assessment [20], programming [21], vulnerability detection [11], medical text analysis [12], and search engine optimization [7]).

Based on Yang's study [1], an LLM should have at least four key features. First, an LLM should demonstrate a deep understanding and interpretation of natural language text, enabling it to extract information and perform various language-related tasks (e.g., translation). Second, it should have the capacity to generate human-like text (e.g., completing sentences, composing paragraphs, and even writing articles) when prompted. Third, LLMs should exhibit contextual awareness by considering factors such as domain expertise, a quality referred to as "Knowledge-intensive". Fourth, these models should excel in problem-solving and decision-making, leveraging information within text passages to make them invaluable for tasks such as information retrieval and question-answering systems.

Table 1
Comparison of popular LLMs [24–30].

Model	Date	Provider	Open-Source	Params	Tunability
GPT-4 [24]	2023.03	OpenAI	✗	1.7 T	✗
GPT-3.5-turbo	2021.09	OpenAI	✗	175 B	✗
GPT-3 [25]	2020.06	OpenAI	✗	175 B	✗
Cohere-medium [26]	2022.07	Cohere	✗	6 B	✓
Cohere-large [26]	2022.07	Cohere	✗	13 B	✓
Cohere-xlarge [26]	2022.06	Cohere	✗	52 B	✓
BERT [27]	2018.08	Google	✓	340 M	✓
T5 [28]	2019	Google	✓	11 B	✓
PaLM [29]	2022.04	Google	✓	540 B	✓
LLaMA [3]	2023.02	Meta AI	✓	65 B	✓
CTRL [30]	2019	Salesforce	✓	1.6 B	✓
Dolly 2.0 [4]	2023.04	Databricks	✓	12 B	✓

*✗ The models are not open-source or can not be fine-tuned for specific tasks.
*✓ The models are open-source or can be fine-tuned for specific tasks.

2.2. Comparison of popular LLMs

As shown in Table 1 [22,23], there is a diversity of providers for language models, including industry leaders such as OpenAI, Google, Meta AI, and emerging players such as Anthropic and Cohere. The release dates span from 2018 to 2023, showcasing the rapid development and evolution of language models in recent years. Newer models such as "gpt-4" have emerged in 2023, highlighting the ongoing innovation in this field. While most of the models are not open-source, it is interesting to note that models like BERT, T5, PaLM, LLaMA, and CTRL are open-source, which can facilitate community-driven development and applications. Larger models tend to have more parameters, potentially indicating increased capabilities but also greater computational demands. For example, "PaLM" stands out with a massive 540 billion parameters. It can also be observed that LLMs tend to have more parameters, potentially indicating increased capabilities but also greater computational demands. The "*Tunability*" column suggests whether these models can be fine-tuned for specific tasks. In other words, it is possible to take a large, pre-trained language model and adjust its parameters and training on a smaller, domain-specific dataset to make it perform better on a particular task. For instance, with tunability, one can fine-tune BERT on a dataset of movie reviews to make it highly effective at sentiment analysis.

3. Overview

3.1. Scope

Our paper endeavors to conduct a thorough literature review, with the objective of collating and scrutinizing existing research and studies about the realms of security and privacy in the context of LLMs. The effort is geared towards both establishing the current state of the art in this domain and pinpointing gaps in our collective knowledge. While it is true that LLMs wield multifaceted applications extending beyond security considerations (e.g., social and financial impacts), our primary focus remains steadfastly on matters of security and privacy. Moreover, it is noteworthy that GPT models have attained significant prominence within this landscape. Consequently, when delving into specific content and examples, we aim to employ GPT models as illustrative benchmarks.

3.2. The research questions

LLMs have carried profound implications across diverse domains. However, it is essential to recognize that, as with any powerful technology, LLMs bear a significant responsibility. Our paper delves deeply into the multifaceted role of LLMs in the context of security and privacy. We intend to scrutinize their positive contributions to these domains, explore the potential threats they may engender, and uncover the vulnerabilities that could compromise their integrity. To accomplish this, our study will conduct a thorough literature review centered around three pivotal research questions:

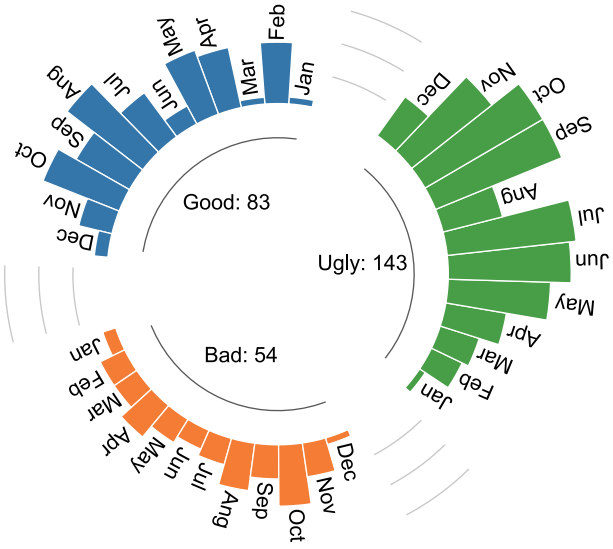

Fig. 1. An overview of our collected papers.

- **The Good** (Section 4): How do LLMs positively contribute to security and privacy in various domains, and what are the potential benefits they bring to the security community?
- **The Bad** (Section 5): What are the potential risks and threats associated with the use of LLMs in the context of cybersecurity? Specifically, how can LLMs be used for malicious purposes, and what types of cyber attacks can be facilitated or amplified using LLMs?
- **The Ugly** (Section 6): What vulnerabilities and weaknesses exist within LLMs, and how do these vulnerabilities pose a threat to security and privacy?

Motivated by these questions, we conducted a search on Google Scholar and compiled papers related to security and privacy involving LLMs. As shown in Fig. 1, we gathered a total of 83 "good" papers that highlight the positive contributions of LLMs to security and privacy. Additionally, we identified 54 "bad" papers, in which attackers exploited LLMs to target users, and 144 "ugly" papers, in which authors discovered vulnerabilities

Table 2
LLMs for code security and privacy.

Work	Life Cycle						LLM(s)	Domain	When compared to SOTA ways?
	Coding (C)	Test Case Generating (TCG)	Running and Executing (RE)						
			Bug Detecting	Malicious Code Detecting	Vulnerability Detecting	Fixing			
Sandoval et al. [31]	●	○	○	○	○	○	Codex	–	Negligible risks
SVEN [32]	●	○	○	○	○	○	CodeGen	–	More faster/secure
SALLM [33]	●	○	○	○	○	○	ChatGPT etc.	–	–
Madhav et al. [34]	●	○	○	○	○	○	ChatGPT	Hardware	–
Zhang et al. [35]	○	●	○	○	●	○	ChatGPT	Supply chain	More valid cases
Libro [36]	○	●	○	○	●	○	LLaMA	–	Higher FP/FN
TitanFuzz [37]	○	●	●	○	●	○	Codex	DL libs	Higher coverage
FuzzGPT [38]	○	●	●	○	●	○	ChatGPT	DL libs	Higher coverage
Fuzz4All [17]	○	●	●	○	●	○	ChatGPT	Languages	Higher coverage
WhiteFox [39]	○	●	●	○	●	○	GPT4	Compiler	High-quality tests
Zhang et al. [40]	○	●	●	○	●	○	ChatGPT	API	–
CHATAFL [41]	○	●	●	○	●	○	ChatGPT	Protocol	Higher coverage
Henrik [42]	○	○	○	●	○	○	ChatGPT	–	Higer FP/FN
Apiiro [43]	○	○	○	●	○	○	N/A	–	–
Noever [44]	○	○	○	○	●	●	ChatGPT	–	4X faster
Bakhshandeh et al. [45]	○	○	○	○	●	○	ChatGPT	–	Low FP/FN
Moumita et al. [46]	○	○	○	○	●	○	ChatGPT	–	Higher FP/FN
Cheshkov et al. [47]	○	○	○	○	●	○	ChatGPT	–	No better
LATTE [48]	○	○	○	○	●	○	GPT	–	Cost effective
DefectHunter [49]	○	○	○	○	●	○	Codex	–	–
Chen et al. [50]	○	○	○	○	●	○	ChatGPT	Blockchain	–
Hu et al. [51]	○	○	○	○	●	○	ChatGPT	Blockchain	–
KARTAL [52]	○	○	○	○	●	○	ChatGPT	Web apps	Less manual
VulLibGen [53]	○	○	○	○	●	○	LLaMa	Libs	Higher accuracy/speed
Ahmad et al. [54]	○	○	○	○	●	●	Codex	Hardware	Fix more bugs
InferFix [55]	○	○	●	○	●	●	Codex	–	CI Pipeline
Pearce et al. [16]	○	○	●	○	●	○	Codex etc.	–	Zero-shot
Fu et al. [56]	○	○	●	○	●	●	ChatGPT	APR	Higher accuracy
Sobania et al. [57]	○	○	○	○	○	●	ChatGPT etc.	APR	Higher accuracy
Jiang et al. [58]	○	○	○	○	○	●	ChatGPT	APR	Higher accuracy

*● The model can enhance the specific facets of data protection.
*○ The model can not enhance the specific facets of data protection.

within LLMs. Most of the papers were published in 2023, with only 82 of them released in between 2007 and 2022. Notably, there is a consistent upward trend in the number of papers released each month, with October reaching its peak, boasting the highest number of papers published (38 papers in total, accounting for 15.97% of all the collected papers). It is conceivable that more security-related LLM papers will be published in the near future.

Finding I. In terms of security-related applications (i.e., the "good" and the "bad" parts), it is evident that the majority of researchers are inclined towards using LLMs to bolster the security community, such as in vulnerability detection and security test generation, despite the presence of some vulnerabilities in LLMs at this stage. There are relatively few researchers who employ LLMs as tools for conducting attacks. In summary, LLMs contribute more positively than negatively to the security community.

4. Positive impacts on security and privacy

In this section, we explore the beneficial impacts of employing LLMs. In the context of code or data privacy, we have opted to use the term "privacy" to characterize scenarios in which LLMs are utilized to ensure the confidentiality of either code or data. However, given that we did not come across any papers specifically addressing code privacy, our discussion focuses on code security (Section 4.1) as well as both data security and privacy (Section 4.2).

4.1. LLMs for code security

As shown in Table 2, LLMs have access to a vast repository of code snippets and examples spanning various programming languages and domains. They leverage their advanced language understanding and contextual analysis capabilities to thoroughly examine code and code-related text. More specifically, LLMs can play a pivotal role throughout the entire code security lifecycle, including coding (C), test case generation (TCG), execution, and monitoring (RE).

Secure coding (C). We first discuss the use of LLMs in the context of secure code programming [59] (or generation [60–63]). Sandoval et al. [31] conducted a user study (58 users) to assess the security implications of LLMs, particularly OpenAI Codex, as code assistants for developers. They evaluated code written by student programmers when assisted by LLMs and found that participants assisted by LLMs did not introduce new security risks: the AI-assisted group produced critical security bugs at a rate no greater than 10% higher than the control group (non-assisted). He et al. [32,64] focused on enhancing the security of code generated by LLMs. They proposed a novel method called SVEN, which leverages continuous prompts to control LLMs in generating secure code. With this method, the success rate improved from 59.1% to 92.3% when using the CodeGen LM. Mohammed et al. introduce SALLM [33], a framework consisting of a new security-focused dataset, an evaluation environment, and novel metrics for systematically assessing LLMs' ability to generate secure code. Madhav et al. [34] evaluate the security aspects of code generation processes on the ChatGPT platform, specifically in the hardware domain. They explore the strategies that a designer can employ to enable ChatGPT to provide secure hardware code generation.

Test case generating (TCG). Several papers [65–71] discuss the utilization of LLMs for generating test cases, with our particular emphasis on those addressing security implications. Zhang et al. [35] demonstrated the use of ChatGPT-4.0 for generating security tests to assess the impact of vulnerable library dependencies on software applications. They found that LLMs could successfully generate tests that demonstrated various supply chain attacks, outperforming existing security test generators. This approach resulted in 24 successful attacks across 55 applications. Similarly, Libro [36], a framework that uses LLMs to automatically generate test cases to reproduce software security bugs.

In the realm of security, fuzzing stands [40,72–75] out as a widely employed technique for generating test cases. Deng et al. introduced TitanFuzz [37], an approach that harnesses LLMs to generate input programs for fuzzing Deep Learning (DL) libraries. TitanFuzz demonstrates impressive code coverage (30.38%/50.84%) and detects previously unknown bugs (41 out of 65) in popular DL libraries. More recently, Deng et al. [38,76] refined LLM-based fuzzing (named FuzzGPT), aiming to generate unusual programs for DL library fuzzing. While TitanFuzz leverages LLMs' ability to generate ordinary code, FuzzGPT addresses the need for edge-case testing by priming LLMs with historical bug-triggering program. Fuzz4All [17] leverages LLMs as input generators and mutation engines, creating diverse and realistic inputs for various languages (e.g., C, C++), improving the previous state-of-the-art coverage by 36.8% on average. WhiteFox [39], a novel white-box compiler fuzzer that utilizes LLMs to test compiler optimizations, outperforms existing fuzzers (it generates high-quality tests for intricate optimizations, surpassing state-of-the-art fuzzers by up to 80 optimizations). Zhang et al. [40] explore the generation of fuzz drivers for library API fuzzing using LLMs. Results show that LLM-based generation is practical, with 64% of questions solved entirely automatically and up to 91% with manual validation. CHATAFL [41] is an LLM-guided protocol fuzzer that constructs grammars for message types and mutates messages or predicts the next messages based on LLM interactions, achieving better state and code coverage compared to state-of-the-art fuzzers (e.g., AFLNET [77], NSFUZZ [78]).

Vulnerable code detecting (RE). Noever [44] explores the capability of LLMs, particularly OpenAI's GPT-4, in detecting software vulnerabilities. This paper shows that GPT-4 identified approximately four times the number of vulnerabilities compared to traditional static code analyzers (e.g., Snyk and Fortify). Parallel conclusions have also been drawn in other efforts [15,45]. However, Moumita et al. [46] applied LLMs for software vulnerability detection, exposing a noticeable performance gap when compared to conventional static analysis tools. This disparity primarily arises from the relatively higher occurrence of false alerts generated by LLMs. Similarly, Cheshkov et al. [47] point out that the ChatGPT model performed no better than a dummy classifier for both binary and multi-label classification tasks in code vulnerability detection. Wang et al. introduce DefectHunter [49], a novel model that employs LLM-driven techniques for code vulnerability detection. They demonstrate the potential of combining LLMs with advanced mechanisms (e.g., Conformer) to identify software vulnerabilities more effectively. This combination shows an improvement in effectiveness, approximately from 14.64% to 20.62%, compared with Pongo-70B. LATTE [48] is a novel static binary taint analysis method powered by LLMs. LATTE surpasses existing state-of-the-art techniques (e.g., Emtaint, Arbiter, and Karonte), demonstrating remarkable effectiveness in vulnerability detection (37 new bugs in real-world firmware) with lower cost.

Efforts in leveraging LLMs for vulnerability detection extend to specialized domains (e.g.,blockchain [50,51], kernel [79] mobile [80]). For instance, Chen et al. [50] and Hu et al. [51] focus on the application of LLMs in identifying vulnerabilities within blockchain smart contracts. Sakaoglu's study introduces KARTAL [52], a pioneering approach that harnesses LLMs for web application vulnerability detection. This method achieves an accuracy of up to 87.19% and is capable of conducting 539 predictions per second. Additionally, Chen et al. [53] make a noteworthy contribution with VulLibGen, a generative methodology utilizing LLMs to identify vulnerable libraries. Ahmad et al. [54] shift the focus to hardware security. They investigate the use of LLMs, specifically OpenAI's Codex, in automatically identifying and repairing security-related bugs in hardware designs. PentestGPT [81], an automated penetration testing tool, uses the domain knowledge inherent in LLMs to address individual sub-tasks of penetration testing, improving task completion rates significantly.

Malicious code detecting (RE). Using LLM to detect malware is a promising application. This approach leverages the natural language processing capabilities and contextual understanding of LLMs to identify malicious software. In experiments with GPT-3.5 conducted by Henrik Plate [42], it was found that LLM-based malware detection can complement human reviews but not replace them. Out of 1800 binary classifications performed, there were both false-positives and false-negatives. The use of simple tricks could also deceive the LLM's assessments. More recently, there are a few attempts have been made in this direction. For example, Apiiro [43] is a malicious code analysis tool using LLMs. Apiiro's strategy involves the creation of LLM Code Patterns (LCPs) to represent code in vector format, making it easier to identify similarities and cluster packages efficiently. Its LCP detector incorporates LLMs, proprietary code analysis, probabilistic sampling, LCP indexing, and dimensionality reduction to identify potentially malicious code.

Vulnerable/buggy code fixing (RE). Several papers [16,58,99] has focused on evaluate the performance of LLMs trained on code in the task of program repair. Jin et al. [55] proposed InferFix, a transformer-based program repair framework that works in tandem with the combination of cutting-edge static analyzer with transformer-based model to address and fix critical security and performance issues with accuracy between 65% to 75%. Pearce et al. [16] observed that LLMs can repair insecure code in a range of contexts even without being explicitly trained on vulnerability repair tasks.

ChatGPT is noted for its ability in code bug detection and correction. Fu et al. [56] assessed ChatGPT in vulnerability-related tasks like predicting and classifying vulnerabilities, severity estimation, and analyzing over 190,000 C/C++ functions. They found that ChatGPT's performance was behind other LLMs specialized in vulnerability detection. However, Sobania et al. [57] found ChatGPT's bug fixing performance competitive with standard program repair methods, as demonstrated by its ability to fix 31 out of 40 bugs. Xia et al. [100] presented ChatRepair, leveraging pre-trained language models (PLMs) for generating patches without dependency on bug-fixing datasets, aiming to enhance performance to generate patches without relying on bug-fixing datasets, aiming to improve ChatGPT's code-fixing abilities using a mix of successful and failure tests. As a result, they fixed 162 out of 337 bugs at a cost of $0.42 each.

Finding II. As shown in Table 2, a comparison with state-of-the-art methods reveals that the majority of researchers (17 out of 25) have concluded that LLM-based methods outperform traditional approaches (advantages include higher code coverage, higher detecting accuracy, less cost etc.). Only four papers argue that LLM-based methods do not surpass the state-of-the-art approaches. The most frequently discussed issue with LLM-based methods is their tendency to produce both high false negatives and false positives when detecting vulnerabilities or bugs.

4.2. LLMs for data security and privacy

As demonstrated in Table 3, LLMs make valuable contributions to the realm of data security, offering multifaceted approaches to safeguarding sensitive information. We have organized the research papers into distinct categories based on the specific facets of data protection that LLMs enhance. These facets encompass critical aspects such as data integrity (I), which ensures that data remains uncorrupted throughout its life cycle; data reliability (R), which ensures the accuracy of data; data confidentiality (C), which focuses on guarding against unauthorized access and disclosure of sensitive information; and data traceability (T), which involves tracking and monitoring data access and usage.

Table 3
LLMs for data security and privacy.

Work	Prop.				Model	Domain	Compared to SOTA ways?
	I	C	R	T			
Fang [82]	●	○	●	○	ChatGPT	Ransomware	–
Liu et al. [83]	●	○	●	○	ChatGPT	Ransomware	–
Amine et al. [84]	●	●	●	○	ChatGPT	Semantic	Aligned w/ SOTA
HuntGPT [85]	●	●	●	○	ChatGPT	Network	More effective
Chris et al. [86]	●	●	●	○	ChatGPT	Log	Less manual
AnomalyGPT [87]	●	●	●	○	ChatGPT	Video	Less manual
LogGPT [88]	●	●	●	○	ChatGPT	Log	Less manual
Arpita et al. [89]	○	●	○	○	BERT etc.	–	–
Takashi et al. [90]	○	○	●	○	ChatGPT	Phishing	High precision
Fredrik et al. [91]	○	○	●	○	ChatGPT etc.	Phishing	Effective
IPSDM [92]	○	○	●	○	BERT	Phishing	–
Kwon et al. [93]	○	●	○	○	ChatGPT	–	Non-exp friendly
Scanlon et al. [94]	○	○	○	●	ChatGPT	Forensic	More effective
Sladić et al. [95]	○	○	○	●	ChatGPT	Honeypot	More realistic
WASA [96]	○	○	●	●	–	Watermark	More effective
REMARK [97]	○	○	●	●	–	Watermark	More effective
SWEET [98]	○	○	●	●	–	Watermark	More effective

*● The model can enhance the specific facets of data protection.
*○ The model can not enhance the specific facets of data protection.

Data integrity (I). Data Integrity ensures that data remains unchanged and uncorrupted throughout its life cycle. As of now, there are a few works that discuss how to use LLMs to protect data integrity. For example, ransomware usually encrypts a victim's data, making the data inaccessible without a decryption key that is held by the attacker, which breaks the data integrity. Wang Fang's research [82] examines using LLMs for ransomware cybersecurity strategies, mostly theoretically proposing real-time analysis, automated policy generation, predictive analytics, and knowledge transfer. However, these strategies lack empirical validation. Similarly, Liu et al. [83] explored the potential of LLMs for creating cybersecurity policies aimed at mitigating ransomware attacks with data exfiltration. They compared GPT-generated Governance, Risk and Compliance (GRC) policies to those from established security vendors and government cybersecurity agencies. They recommended that companies should incorporate GPT into their GRC policy development.

Anomaly detection is a key defense mechanism that identifies unusual behavior. While it does not directly protect data integrity, it identifies abnormal or suspicious behavior that can potentially compromise data integrity (as well as data confidentiality and data reliability). Amine et al. [84] introduced an LLM-based monitoring framework for detecting semantic anomalies in vision-based policies and applied it to both finite state machine policies for autonomous driving and learned policies for object manipulation. Experimental results demonstrate that it can effectively identify semantic anomalies, aligning with human reasoning. HuntGPT [85] is an LLM-based intrusion detection system for network anomaly detection. The results demonstrate its effectiveness in improving user understanding and interaction. Chris et al. [86] and LogGPT [88] explore ChatGPT's potential for log-based anomaly detection in parallel file systems. Results show that it addresses the issues in traditional manual labeling and interpretability. AnomalyGPT [87] uses Large Vision-Language Models to detect industrial anomalies. It eliminates manual threshold setting and supports multi-turn dialogues.

Data confidentiality (C). Data confidentiality refers to the practice of protecting sensitive information from unauthorized access or disclosure, a topic extensively discussed in LLM privacy discussions [89,101–103]. However, most of these studies concentrate on enhancing LLMs through state-of-the-art Privacy Enhancing Techniques (e.g., zero-knowledge proofs [104], differential privacy (e.g., [102,105,106], and federated learning [107–109]). There are only a few attempts that utilize LLMs to enhance user privacy. For example, Arpita et al. [89] use LLMs to preserve privacy by replacing identifying information in textual data with generic markers. Instead of storing sensitive user information, such as names, addresses, or credit card numbers, the LLMs suggest substitutes for the masked tokens. This obfuscation technique helps to protect user data from being exposed to adversaries. By using LLMs to generate substitutes for masked tokens, the models can be trained on obfuscated data without compromising the privacy and security of the original information. Similar ideas have also been explored in other studies [103,110]. Hyeokdong et al. [93] explore implementing cryptography with ChatGPT, which ultimately protects data confidentiality. Despite the lack of extensive coding skills or programming knowledge, the authors were able to successfully implement cryptographic algorithms through ChatGPT. This highlights the potential for individuals to utilize ChatGPT for cryptography tasks.

Data reliability (R). In our context, data reliability refers to the accuracy of data. It is a measure of how well data can be depended upon to be accurate, and free from errors or bias. Takashi et al. [90] proposed to use ChatGPT for the detection of sites that contain phishing content. Experimental results using GPT-4 show promising performance, with high precision and recall rates. Fredrik et al. [91] assessed the ability of four large language models (GPT, Claude, PaLM, and LLaMA) to detect malicious intent in phishing emails, and found that they were generally effective, even surpassing human detection, although occasionally slightly less accurate. IPSDM [92] is a model fine-tuned from the BERT family to identify phishing and spam emails effectively. IPSDM demonstrates superior performance in classifying emails, both in unbalanced and balanced datasets.

Data traceability (T). Data traceability is the capability to track and document the origin, movement, and history of data within a single system or across multiple systems. This concept is particularly vital in fields such as incident management and forensic investigations, where understanding the journey and transformations of events to resolving issues and conducting thorough analyses. LLMs have gained traction in forensic investigations, offering novel approaches for analyzing digital evidence. Scanlon et al. [94] explored how ChatGPT assists in analyzing OS artifacts like logs, files, cloud interactions, executable binaries, and in examining memory dumps to detect suspicious activities or attack patterns. Additionally, Sladić et al. [95] proposed that generative

models like ChatGPT can be used to create realistic honeypots to deceive human attackers.

Watermarking involves embedding a distinctive, typically imperceptible or hard-to-identify signal within the outputs of a model. Wang et al. [96] discusses concerns regarding the intellectual property of training data for LLMs and proposed WASA framework to learn the mapping between the texts of different data providers. Zhang et al. [97] developed REMARK-LLM that focused on monitor the utilization of their content and validate their watermark retrieval. This helps protect against malicious uses such as spamming and plagiarism. Furthermore, identifying code produced by LLMs is vital for addressing legal and ethical issues concerning code licensing, plagiarism, and malware creation. Similarly, Li et al. [111] propose the first watermark technique to protect large language model-based code generation APIs from remote imitation attacks. Lee et al. [98] developed SWEET, a tool that implements watermarking specifically on tokens within programming languages.

Finding III. Likewise, it is noticeable that LLMs excel in data protection, surpassing current solutions and requiring fewer manual interventions. Table 2 and Table 3 reveal that ChatGPT is the predominant LLM extensively employed in diverse security applications. Its versatility and effectiveness make it a preferred choice for various security-related tasks, further reinforcing its position as a go-to solution in the field of artificial intelligence and cybersecurity.

5. Negative impacts on security and privacy

As shown in Fig. 2, we have categorized the attacks into five groups based on their respective positions within the system infrastructure. These categories encompass hardware-level attacks, OS-level attacks, software-level attacks, network-level attacks, and user-level attacks. Additionally, we have quantified the number of associated research papers published for each group, as illustrated in Fig. 3.

Hardware-level attacks. Hardware attacks typically involve physical access to devices. However, LLMs cannot directly access physical devices. Instead, they can only access information associated with the hardware. Side-channel attack [112–114] is one attack that can be powered by the LLMs. Side-channel attacks typically entail the analysis of unintentional information leakage from a physical system or implementation, such as a cryptographic device or software, with the aim of inferring secret information (e.g., keys).

Yaman [115] has explored the application of LLM techniques to develop side-channel analysis methods. The research evaluates the effectiveness of LLM-based approaches in analyzing side-channel information in two hardware-related scenarios: AES side-channel analysis and deep-learning accelerator side-channel analysis. Experiments are conducted to determine the success rates of these methods in both situations.

OS-level attacks. LLMs operate at a high level of abstraction and primarily engage with text-based input and output. They lack the necessary low-level system access essential for executing OS-level attacks [116–118]. Nonetheless, they can be utilized for the analysis of information gathered from operating systems, thus potentially aiding in the execution of such attacks. Andreas et al. [119] establish a feedback loop connecting LLM to a vulnerable virtual machine through SSH, allowing LLM to analyze the machine's state, identify vulnerabilities, and propose concrete attack strategies, which are then executed automatically within the virtual machine. More recently, they [120] introduced an automated Linux privilege-escalation benchmark using local virtual machines and an LLM-guided privilege-escalation tool to assess various LLMs and prompt strategies against the benchmark.

Software-level attacks. Similar to how they employ LLM to target hardware and operating systems, there are also instances where LLM has been utilized to attack software (e.g., [35,121–123]). However, the most prevalent software-level use case involves malicious developers utilizing LLMs to create malware. Mika et al. [124] present a proof-of-concept in which ChatGPT is utilized to distribute malicious software while avoiding detection. Yin et al. [125] investigate the potential misuse of LLM by creating a number of malware programs (e.g., ransomware, worm, keylogger, brute-force malware, Fileless malware). Antonio Monje et al. [126] demonstrate how to trick ChatGPT into quickly generating ransomware. Marcus Botacin [127] explores different coding strategies (e.g., generating entire malware, creating malware functions) and investigates the LLM's capacities to rewrite malware code. The findings reveal that LLM excels in constructing malware using building block descriptions. Meanwhile, LLM can generate multiple versions of the same semantic content (malware variants), with varying detection rates by Virustotal AV (ranging from 4% to 55%).

Network-level attacks. LLMs can also be employed for initiating network attacks. A prevalent example of a network-level attack utilizing LLM is phishing attacks [128,129]. Fredrik et al. [91] compared AI-generated phishing emails using GPT-4 with manually designed phishing emails created using the V-Triad, alongside a control group exposed to generic phishing emails. The results showed that personalized phishing emails, whether generated by AI or designed manually, had higher click-through rates compared to generic ones. Tyson et al. [130] investigated how modifying ChatGPT's input can affect the content of the generated emails, making them more convincing. Julian Hazell [131] demonstrated the scalability of spear phishing campaigns by generating realistic and cost-effective phishing messages for over 600 British Members of Parliament using ChatGPT. In another study, Wang et al. [132] discuss how the traditional defenses may fail in the era of LLMs. CAPTCHA challenges, involving distorted letters and digits, struggle to detect chatbots relying on text and voice. However, LLMs may break the challenges, as they can produce high-quality human-like text and mimic human behavior effectively. There is one study that utilizes LLM for deploying fingerprint attacks. Armin et al. [133] employed density-based clustering to cluster HTTP banners and create text-based fingerprints for annotating scanning data. When these fingerprints are compared to an existing database, it becomes possible to identify new IoT devices and server products.

User-level attacks. Recent discussions have primarily focused on user-level attacks, as LLM demonstrates its capability to create remarkably convincing but ultimately deceptive content, as well as establish connections between seemingly unrelated pieces of information. This presents opportunities for malicious actors to engage in a range of nefarious activities. Here are a few examples:

- **Misinformation.** Overreliance on content generated by LLMs without oversight is raising serious concerns regarding the safety of online content [134]. Numerous studies have focused on detecting misinformation produced by LLMs. Several study [135–137] reveal content generated by LLMs are harder to detect and may use more deceptive styles, potentially causing greater harm. Canyu Chen et al. [135] propose a taxonomy for LLM-generated misinformation and validate methods. Countermeasures and detection methods [136,138–145] have also been developed to address these emerging issues.
- **Social Engineering.** LLMs not only have the potential to generate content from training data, but they also offer attackers a new perspective for social engineering. Work

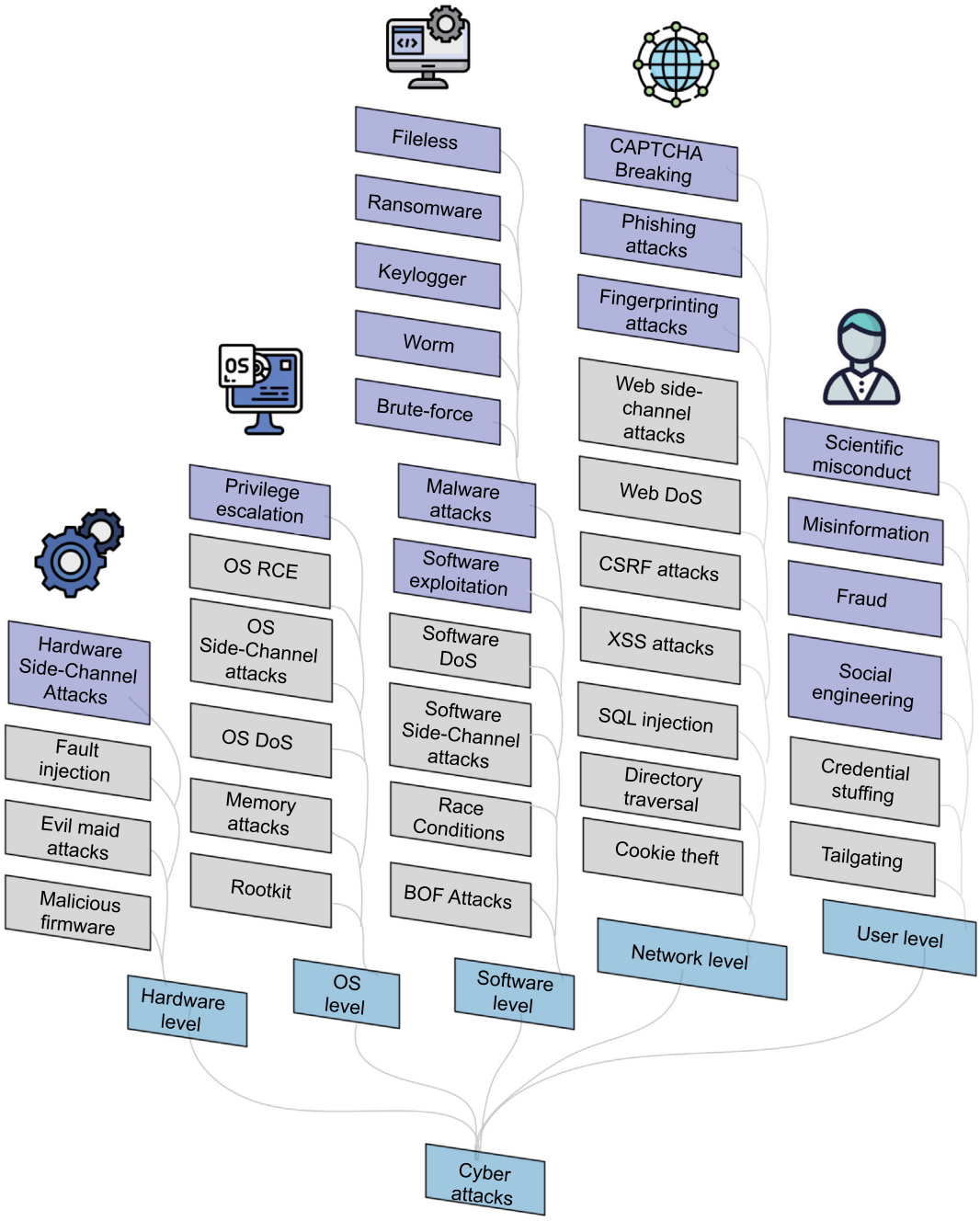

Fig. 2. Taxonomy of Cyberattacks. The colored boxes represent attacks that have been demonstrated to be executable using LLMs, whereas the gray boxes indicate attacks that cannot be executed with LLMs.

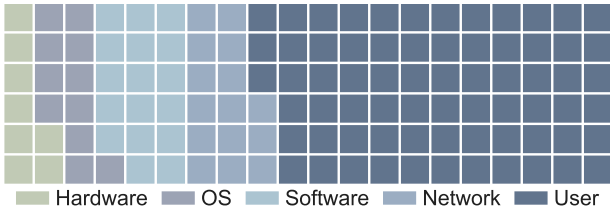

Fig. 3. Prevalence of the existing attacks.

from Stabb et al. [146] highlights the capability of well-trained LLMs to infer personal attributes from text, such as location, income, and gender. They also reveals how these models can extract personal information from seemingly benign queries. Tong et al. [147] investigated the content generated by LLMs may include user information. Moreover, Polra Victor Falade [148] stated the exploitation by LLM-driven social engineers involves tactics such as psychological manipulation, targeted phishing, and the crisis of authenticity.

- **Scientific Misconduct.** Irresponsible use of LLMs can result in issues related to scientific misconduct, stemming from their capacity to generate original, coherent text. The academic community [149–159], encompassing diverse disciplines from various countries, has raised concerns about the increasing difficulties in detecting scientific misconduct in the era of LLMs. Concerns arise from LLMs' ability to generate coherent and original content, including complete

papers from unreliable sources [160–162]. Researchers are also actively engaged in the effort to detect such misconduct. For example, Kavita Kumari et al. [163,164] proposed DEMASQ, a precise ChatGPT-generated content detector. DEMASQ considers biases in text composition and evasion techniques, achieving high accuracy across diverse domains in identifying ChatGPT-generated content.

- **Fraud.** Cybercriminals have devised a new tool called FraudGPT [148,165], which operates like ChatGPT but facilitates cyberattacks. It lacks the safety controls of ChatGPT and is sold on the dark web and Telegram for $200 per month or $1,700 annually. FraudGPT can create fraud emails related to banks, suggesting malicious links' placement in the content. It can also list frequently targeted sites or services, aiding hackers in planning future attacks. WormGPT [166], a cybercrime tool, offers features such as unlimited character support and chat memory retention. The tool was trained on confidential datasets, with a focus on malware-related and fraud-related data. It can guide cybercriminals in executing Business Email Compromise (BEC) attacks.

Finding IV. As illustrated in Fig. 3, when compared to other attacks, it becomes apparent that user-level attacks are the most prevalent, boasting a significant count of 33 papers. This dominance can be attributed to the fact that LLMs have increasingly human-like reasoning abilities, enabling them to generate human-like conversations and content (e.g., scientific misconduct, social engineering). Presently, LLMs do not possess the same level of access to OS-level or hardware-level functionalities. This observation remains consistent with the attack observed in other levels as well. For instance, at the network level, LLMs can be abused to create phishing websites and bypass CAPTCHA mechanisms.

6. Vulnerabilities and defenses in LLMs

In the following section, we embark on an in-depth exploration of the prevalent threats and vulnerabilities associated with LLMs (Section 6.1). We will examine the specific risks and challenges that arise in the context of LLMs. In addition to discussing these challenges, we will also delve into the countermeasures and strategies that researchers and practitioners have developed to mitigate these risks (Section 6.2). Fig. 4 illustrates the relationship between the attacks and defenses.

6.1. Vulnerabilities and threats in LLMs

In this section, we aim to delve into the potential vulnerabilities and attacks that may be directed towards LLMs. Our examination seeks to categorize these threats into two distinct groups: AI Model Inherent Vulnerabilities and Non-AI Model Inherent Vulnerabilities.

6.1.1. AI Inherent vulnerabilities and threats

These are vulnerabilities and threats that stem from the very nature and architecture of LLMs, considering that LLMs are fundamentally AI models themselves. For example, attackers may manipulate the input data to generate incorrect or undesirable outputs from the LLM.

(A1) Adversarial attacks. Adversarial attacks in machine learning refer to a set of techniques and strategies used to intentionally manipulate or deceive machine learning models. These attacks are typically carried out with malicious intent and aim to exploit vulnerabilities in the model's behavior. We only focus on the most extensively discussed attacks, namely, data poisoning and backdoor attacks.

- **Data Poisoning.** Data poisoning stands for attackers influencing the training process by injecting malicious data into the training dataset. This can introduce vulnerabilities or biases, compromising the security, effectiveness, or ethical behavior of the resulting models [134]. Various study [167–172] have demonstrated that pre-trained models are vulnerable to compromise via methods such as using untrusted weights or content, including the insertion of poisoned examples into their datasets. By their inherent nature as pre-trained models, LLMs are susceptible to data poisoning attacks [173–175]. For example, Alexander et al. [168] showed that even with just 100 poison examples, LLMs can produce consistently negative results or flawed outputs across various tasks. Larger language models are more susceptible to poisoning, and existing defenses like data filtering or model capacity reduction offer only moderate protection while hurting test accuracy.
- **Backdoor Attacks.** Backdoor attacks involve the malicious manipulation of training data and model processing, creating a vulnerability where attackers can embed a hidden backdoor into the model [176]. Both backdoor attacks and data poisoning attacks involve manipulating machine learning models, which can include manipulation of inputs. However, the key distinction is that backdoor attacks specifically focus on introducing hidden triggers into the model to manipulate specific behaviors or responses when the trigger is encountered. LLMs are subject to backdoor attacks [177–179]. For example, Yao et al. [180] a bidirectional backdoor, which combines trigger mechanisms with prompt tuning.

(A2) Inference attacks. Inference attacks in the context of machine learning refer to a class of attacks where an adversary tries to gain sensitive information or insights about a machine learning model or its training data by making specific queries or observations to the model. These attacks often exploit unintended information leakage from the responses.

- **Attribute Inference Attacks.** Attribute inference Attack [181–186] is a type of threat where an attacker attempts to deduce sensitive or personal information of individuals or entities by analyzing the behavior or responses of a machine learning models. It works against the LLMs as well. Robin et al. [146] presented the first comprehensive examination of pretrained LLMs' ability to infer personal information from text. Using a dataset of real Reddit profiles, the study demonstrated that current LLMs can accurately infer a variety of personal information (e.g., location, income, sex) with high accuracy.
- **Membership Inferences.** Membership inference Attack is a specific type of inference attack in the field of data security and privacy that determining whether a data record was part of a model's training dataset, given white-/black-box access to the model and the specific data record [187–193]. A number of research studies have explored the concept of membership inference, each adopting a unique perspective and methodology. These studies have explored various membership inference attacks by analyzing the label [194], determining the threshold [195–197], developing a generalized formulation [198], among other methods. Mireshghallah et al. [199] found that fine-tuning the head of the model

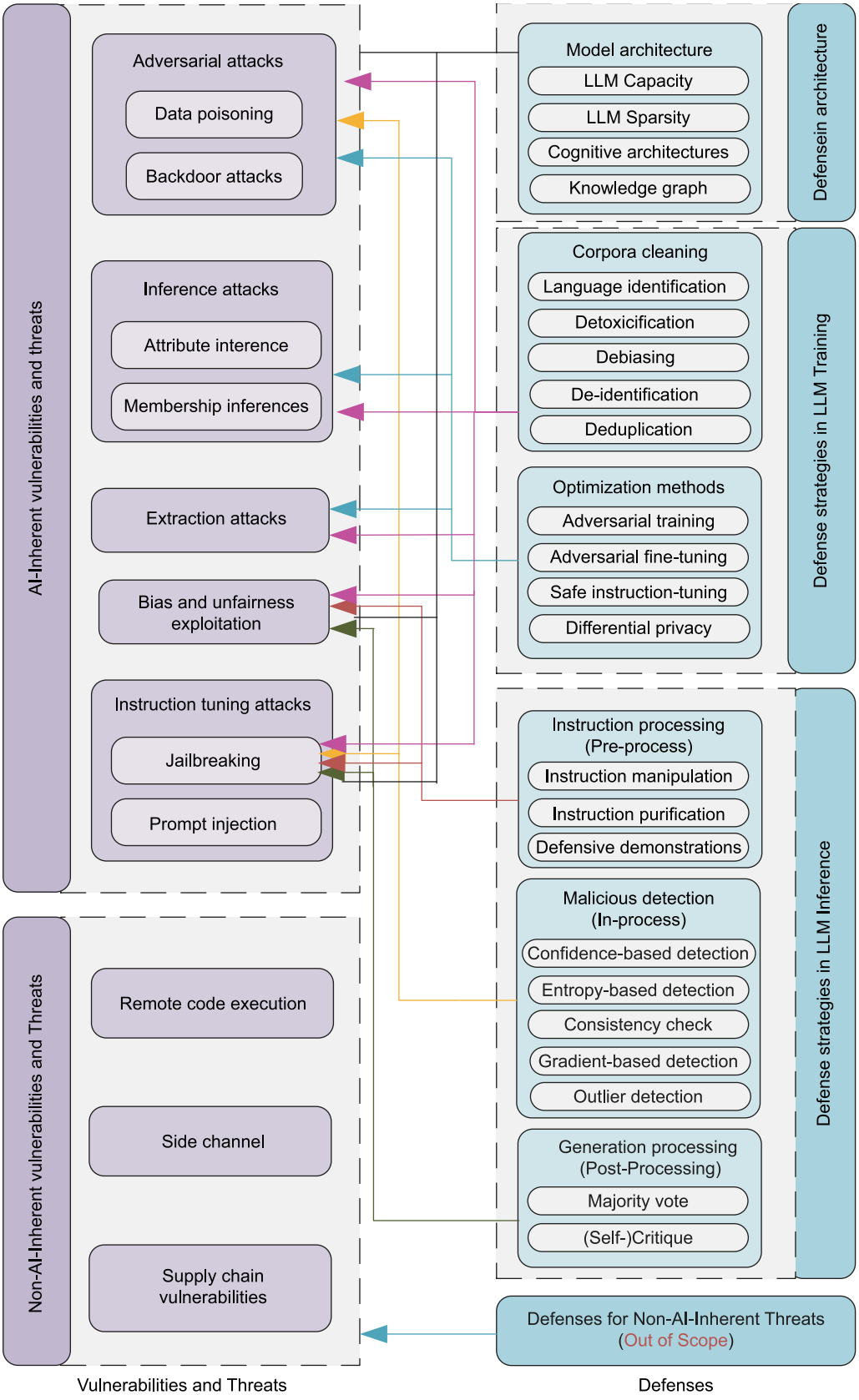

Fig. 4. Taxonomy of Threats and the Defenses. The line represents a defense technique that can defend against either a specific attack or a group of attacks.

exhibits greater susceptibility to attacks when compared to fine-tuning smaller adapters.

(A3) Extraction attacks. Extraction attacks typically refer to attempts by adversaries to extract sensitive information or insights from machine learning models or their associated data. Extraction attacks and inference attacks share similarities but differ in their specific focus and objectives. Extraction attacks aim to acquire specific resources (e.g., model gradient, training data) or confidential information directly. Inference attacks seek to gain knowledge or insights about the model or data's characteristics, often by observing the model's responses or behavior. Various types of data extraction attacks exist, including model theft attacks [200,201], gradient leakage [202], and training data extraction attacks [203]. As of the current writing, it has been observed that training data extraction attacks may be effective against LLMs. Training data extraction [203] refers to a method where an attacker attempts to retrieve specific individual examples from a model's training data by strategically querying the machine learning models. Numerous research [204–206] studies have shown that it is possible to extract training data from LLMs, which may include personal and private information [207,208]. Notably, the work by Truong et al. [209] stands out for its ability to replicate the model without accessing the original model data.

(A4) Bias and unfairness exploitation. Bias and unfairness in LLMs pertain to the phenomenon where these models demonstrate prejudiced outcomes or discriminatory behaviors. While bias and fairness issues are not unique to LLMs, they have received more attention due to the ethical and societal concerns. That is, the societal impact of LLMs has prompted discussions about the ethical responsibilities of organizations and researchers developing and deploying these models. This has led to increased scrutiny and research on bias and fairness. Concerns of bias were raised from various fields, encompassing gender and minority groups [210–213], the identification of misinformation, political aspects. Multiple studies [214,215] revealed biases in the language used while querying LLMs. Moreover, Urman et al. [216] discovered that biases may arise from adherence to government censorship guidelines. Bias in professional writing [145,217,218] involving LLMs is also a concern within the community, as it can significantly damage credibility. The biases of LLMs may also lead to negative side effects in areas beyond text-based applications. Dai et al. [219] noted that content generated by LLMs might introduce biases in neural retrieval systems, and Huang et al. [220] discovered that biases could also be present in LLM generated code.

(A5) Instruction tuning attacks. Instruction tuning, also known as instruction-based fine-tuning, is a machine-learning technique used to train and adapt language models for specific tasks by providing explicit instructions or examples during the fine-tuning process. In LLMs, instruction-tuning attacks refer to a class of attacks or manipulations that target instruction-tuned LLMs. These attacks are aimed at exploiting vulnerabilities or limitations in LLMs that have been fine-tuned with specific instructions or examples for particular tasks.

- **Jailbreaking.** Jailbreaking in LLMs involves bypassing security features to enable responses to otherwise restricted or unsafe questions, unlocking capabilities usually limited by safety protocols. Numerous studies have demonstrated various methods for successfully jailbreaking LLMs [221–223]. Wei et al. [224] emphasized that the alignment capabilities of LLMs can be influenced or manipulated through in-context demonstrations. In addition to this, several researches [225,226] also demonstrated similar manipulation using various approaches, highlighting the versatility of methods that can jailbreaking LLMs. More recently, MASTERKEY [227] employed a time-based method for dissecting defenses, and demonstrated proof-of-concept attacks. It automatically generates jailbreak prompts with a 21.58 Moreover, diverse methods have been employed in jailbreaking LLMs, such as conducting fuzzing [228], implementing optimized search strategies [229], and even training LLMs specifically to jailbreak other LLMs [229,230]. Meanwhile, Cao et al. [231] developed RA-LLM, a method to lowers the success rate of adversarial and jailbreaking prompts without needing of retraining or access to model parameters.
- **Prompt Injection.** Prompt injection attack describes a method of manipulating the behavior of LLMs to elicit unexpected and potentially harmful responses. This technique involves crafting input prompts in a way that bypasses the model's safeguards or triggers undesirable outputs. A substantial amount of research [232–237] has already automated the process of identifying semantic preserving payload in prompt injections with various focus. Facilitated by the capability for fine-tuning, backdoors may be introduced through prompt attacks [183,238–240]. Moreover, Greshake et al. [241] expressed concerns about the potential for new vulnerabilities arising from LLMs invoking external resources. Other studies have also demonstrated the ability to take advantage of prompt injection attacks, such as unveiling guide prompts [242], virtualizing prompt injection [243], and integrating applications [244]. He et al. [245,246] explored a shift towards leveraging LLMs, trained on extensive datasets, for mitigating such attacks.
- **Denial of Service.** A Denial of Service (DoS) attack is a type of cyber attack that aims to exhaust computational resources, causing latency or rendering resources unavailable. Due to the nature of LLMs require significant amount of resources, attackers use deliberately construct prompts to reduce the availability of models [247]. Shumailov et al. [248] proved the possibility of conducting sponge attacks in the field of LLMs, specifically designed to maximize energy consumption and latency (by a factor of 10 to 200). This strategy aims to draw the community's attention to their potential impact on autonomous vehicles, as well as scenarios requiring making decisions in timely manner.

Finding V. Currently, there is limited research on model extraction attacks [188], parameter extraction attacks, or the extraction of other intermediate results [209]. While there are a few mentions of these topics, they tend to remain primarily theoretical (e.g., [249]), with limited practical implementation or empirical exploration. We believe that the sheer scale of parameters in LLMs complicates these traditional approaches, rendering them less effective or even infeasible. Additionally, the most powerful LLMs are privately owned, with their weights, parameters, and other details kept confidential, further shielding them from conventional attack strategies. Strict censorship of outputs generated by these LLMs challenges even black-box traditional ML attacks, as it limits the attackers' ability to exploit or analyze the model's responses.

6.1.2. Non-AI Inherent vulnerabilities and threats

We also need to consider non-AI Inherent Attacks, which encompass external threats and new vulnerabilities (which have not been observed or investigated in traditional AI models) that LLMs might encounter. These attacks may not be intricately linked to the internal mechanisms of the AI model, yet they can present significant risks. Illustrative instances of non-AI Inherent Attacks involve system-level vulnerabilities (e.g., remote code execution).

(A6) Remote code execution (RCE). RCE attacks typically target vulnerabilities in software applications, web services, or servers to execute arbitrary code remotely. While RCE attacks are not typically applicable directly to LLMs, if an LLM is integrated into a web service (e.g.,https://chat.openai.com/) and if there are RCE vulnerabilities in the underlying infrastructure or code of that service, it could potentially lead to the compromise of the LLM's environment. Tong et al. [250] identified 13 vulnerabilities in six frameworks, including 12 RCE vulnerabilities and 1 arbitrary file read/write vulnerability. Additionally, 17 out of 51 tested apps were found to have vulnerabilities, with 16 being vulnerable to RCE and 1 to SQL injection. These vulnerabilities allow attackers to execute arbitrary code on app servers through prompt injections.

(A7) Side channel. While LLMs themselves do not typically leak information through traditional side channels such as power consumption or electromagnetic radiation, they can be vulnerable to certain side-channel attacks in practical deployment scenarios. For example, Edoardo et al. [251] introduce privacy side channel attacks, which are attacks that exploit system-level components (e.g., data filtering, output monitoring) to extract private information at a much higher rate than what standalone models can achieve. Four categories of side channels covering the entire ML lifecycle are proposed, enabling enhanced membership inference attacks and novel threats (e.g., extracting users' test queries). For instance, the research demonstrates how deduplicating training data before applying differentially-private training creates a side channel that compromises privacy guarantees.

(A8) Supply chain vulnerabilities. Supply Chain Vulnerabilities refer to the risks in the lifecycle of LLM applications that may arise from using vulnerable components or services. These include third-party datasets, pre-trained models, and plugins, any of which can compromise the application's integrity [134]. Most research in this field is focused on the security of plugins. An LLM plugin is an extension or add-on module that enhances the capabilities of an LLM. Third-party plug-ins have been developed to expand its functionality, enabling users to perform various tasks, including web searches, text analysis, and code execution. However, some of the concerns raised by security experts [134, 252] include the possibility of plug-ins being used to steal chat histories, access personal information, or execute code on users' machines. These vulnerabilities are associated with the use of OAuth in plug-ins, a web standard for data sharing across online accounts. Umar et al. [253] attempted to address this problem by designing a framework. The framework formulates an extensive taxonomy of attacks specific to LLM platforms, taking into account the capabilities of plugins, users, and the LLM platform itself. By considering the relationships between these stakeholders, the framework helps identify potential security, privacy, and safety risks.

6.2. Defenses for LLMs

In this section, we examine the range of existing defense methods against various attacks and vulnerabilities associated with LLMs.[2]

6.2.1. Defense in model architecture

Model architectures determine how knowledge and concepts are stored, organized, and contextually interacted with, which is crucial in the safety of Large Language Models. There have been a lot of works [254–257] delved into how model capacities affect the privacy preservation and robustness of LLMs. Li et al. [254] revealed that language models with larger parameter sizes can be trained more effectively in the differential privacy manner using appropriate non-standard hyper-parameters, in comparison to smaller models. Zhu et al. [255] and Li et al. [256] found that LLMs with larger capacities, such as those with more extensive parameter sizes, generally show increased robustness against adversarial attacks. This was also verified in the Out-of-distribution (OOD) robustness scenarios by Yuan et al. [257]. Beyond the architecture of LLMs themselves, studies have focused on improving LLM safety by combining them with external modules including knowledge graphs [258] and cognitive architectures (CAs) [259,260]. Romero et al. [261] proposed improving AI robustness by incorporating various cognitive architectures into LLMs. Zafar et al. [262] aimed to build trust in AI by enhancing the reasoning abilities of LLMs through knowledge graphs.

6.2.2. Defenses in LLM training and inference

Defense strategies in LLM training. The core components of LLM training include model architectures, training data, and optimization methods. Regarding model architectures, we examine trustworthy designs that exhibit increased robustness against malicious use. For training corpora, our investigation focuses on methods aimed at mitigating undesired properties during the generation, collection, and cleaning of training data. In the context of optimization methods, we review existing works that developed safe and secure optimization frameworks.

- Corpora Cleaning
  LLMs are shaped by their training corpora, from which they learn behavior, concepts, and data distributions [263]. Therefore, the safety of LLMs is crucially influenced by the quality of the training corpora [264,265]. However, it has been widely acknowledged that raw corpora collected from the web are full of issues of fairness [266], toxicity [267], privacy [181], truthfulness [268], etc. A lot of efforts have been made to clean raw corpora and create high-quality training corpora for LLMs [269–274]. In general, these pipelines consist of the following steps: language identification [269,275], detoxification [267,276–278], debiasing [279–281], de-identification (personally identifiable information (PII)) [282,283], and deduplication [284–287]. Debiasing and detoxification aimed to remove undesirable content from training corpora.
- Optimization Methods Optimization objectives are crucial in directing how LLMs learn from training data, influencing which behaviors are encouraged or penalized. These objectives affect the prioritization of knowledge and concepts within corpora, ultimately impacting the overall safety and ethical alignment of LLMs. In this context, robust training methods like adversarial training [288–292] and robust fine-tuning [293,294] have shown resilience against perturbation-based text attacks. Drawing inspiration from traditional adversarial training in the image field [295], Ivgi et al. [296] and Yoo et al. [291] applied adversarial training to LLMs by generating perturbations concerning discrete tokens. Wang et al. [289] extended this approach to the continuous embedding space, facilitating more practical convergence, as followed by subsequent research [288,290,292]. Safety alignments [297], an emerging learning paradigm, guide LLM behavior using well-aligned additional models or human annotations, proving effective for ethical alignment. Efforts to align LLMs with other LLMs [298] and LLMs themselves [299]. In terms of human annotations, Zhou et al. [300] and Shi et al. [301] emphasized the importance of high-quality training corpora with carefully curated instructions and outputs for enhancing instruction-following

[2] Please be aware that we will not delve into solutions for non-AI inherent vulnerabilities as they tend to be highly specific to individual cases.

capabilities in LLMs. Bianchi et al. [302] highlighted that the safety of LLMs can be substantially improved by incorporating a limited percentage (e.g., 3%) of safe examples during fine-tuning.

Defense strategies in LLM inference. When LLMs are deployed as cloud services, they operate by receiving prompts or instructions from users and generating completed sentences in response. Given this interaction model, the implementation of test-time LLM defense becomes a necessary and critical aspect of ensuring safe and appropriate outputs. Generally, test-time defense encompasses a range of strategies, including the pre-processing of prompts and instructions to filter or modify inputs, the detection of abnormal events that might signal misuse or problematic queries, and the post-processing of generated responses to ensure they adhere to safety and ethical guidelines. Test-time LLM defenses are essential to maintain the integrity and trustworthiness of LLMs in real-time applications.

- Instruction Processing (Pre-Processing) Instruction pre-processing applies transformations over instructions sent by users, in order to destroy potential adversarial contexts or malicious intents. It plays a vital role as it blocks out most malicious usage and prevents LLMs from receiving suspicious instructions. In general, instruction pre-processing methods can be categorized as instruction manipulation [303–307], purification [308], and defensive demonstrations [224,249,309]. Jain et al. [306] and Kirchenbauer et al. [305] evaluated multiple baseline preprocessing methods against jailbreaking attacks, including retokenization and paraphrase. Li et al. [308] proposed to purify instructions by first masking the input tokens and then predicting the masked tokens with other LLMs. The predicted tokens will serve as the purified instructions. Wei et al. [224] and Mo et al. [309] demonstrated that inserting pre-defined defensive demonstrations into instructions effectively defends jailbreaking attacks of LLMs.
- Malicious Detection (In-Processing) Malicious detection provides in-depth examinations of LLM intermediate results, such as neuron activation, regarding the given instructions, which are more sensitive, accurate, and specified for malicious usage. Sun et al. [310] proposed to detect backdoored instructions with backward probabilities of generations. Xi et al. [311] differentiated normal and poisoned instructions from the perspective of mask sensitivities. Shao et al. [303] identified suspicious words according to their textual relevance. Wang et al. [312] detected adversarial examples according to the semantic consistency among multiple generations, which has been explored in the uncertainty quantification of LLMs by Duan et al. [313]. Apart from the intrinsic properties of LLMs, there have been works leveraging the linguistic statistic properties, such as detecting outlier words [314],
- Generation Processing (Post-Processing) Generation post processing refers to examining the properties (e.g., harmfulness) of the generated answers and applying modifications if necessary, which is the final step before delivering responses to users. Chen et al. [315] proposed to mitigate the toxicity of generations by comparing with multiple model candidates. Helbling et al. [316] incorporated individual LLMs to identify the harmfulness of the generated answers, which shared similar ideas as Xiong et al. [317] and Kadavath et al. [318] where they revealed that LLMs can be prompted to answer the confidences regarding the generated responses.

Finding VI. For defense in LLM training, there is a notable scarcity of research examining the impact of model architecture on LLM safety, which is likely due to the high computational costs associated with training or fine-tuning large language models. We observed that *safe instruction tuning* is a relatively new development that warrants further investigation and attention.

7. Discussion

7.1. LLM in other security related topics

LLMs in cybersecurity education. LLMs can be used in security practices and education [319–321]. For example, in a software security course, students are tasked with identifying and resolving vulnerabilities in a web application using LLMs. Jingyue et al. [320] investigated how ChatGPT can be used by students for these exercises. Wesley Tann et al. [321] focused on the evaluation of LLMs in the context of cybersecurity Capture-The-Flag (CTF) exercises (participants find "flags" by exploiting system vulnerabilities). The study first assessed the question-answering performance of these LLMs on Cisco certifications with varying difficulty levels, then examined their abilities in solving CTF challenges. Jin et al. [322] conducted a comprehensive study on LLMs' understanding of binary code semantics [323] across different architectures and optimization levels, providing key insights for future research in this area.

LLMs in cybersecurity laws, policies and compliance. LLMs can assist in drafting security policies, guidelines, and compliance documentation, ensuring that organizations meet regulatory requirements and industry standards. However, it is important to recognize that the utilization of LLMs can potentially necessitate changes to current cybersecurity-related laws and policies. The introduction of LLMs may raise new legal and regulatory considerations, as these models can impact various aspects of cybersecurity, data protection, and privacy. Ekenobi et al. [324] examined the legal implications arising from the introduction of LLMs, with a particular focus on data protection and privacy concerns. It acknowledges that ChatGPT's privacy policy contains commendable provisions for safeguarding user data against potential threats. The paper also advocated for emphasizing the relevance of the new law.

7.2. Future directions

We have gleaned valuable lessons that we believe can shape future directions.

- **Using LLMs for ML-Specific Tasks**. We noticed that LLMs can effectively replace traditional machine learning methods and in this context, if traditional machine learning methods can be employed in a specific security application (whether offensive or defensive in nature), it is highly probable that LLMs can also be applied to address that particular challenge. For instance, traditional machine learning methods have found utility in malware detection, and LLMs can similarly be harnessed for this purpose. Therefore, one promising avenue is to harness the potential of LLMs in security applications where machine learning serves as a foundational or widely adopted technique. As security researchers, we are capable of designing LLM-based approaches to tackle security issues. Subsequently, we can compare these approaches with state-of-the-art methods to push the boundaries.

- **Replacing Human Efforts.** It is evident that LLMs have the potential to replace human efforts in both offensive and defensive security applications. For instance, tasks involving social engineering, traditionally reliant on human intervention, can now be effectively executed using LLM techniques. Therefore, one promising avenue for security researchers is to identify areas within traditional security tasks where human involvement has been pivotal and explore opportunities to substitute these human efforts with LLM capabilities.
- **Modifying Traditional ML Attacks for LLMs.** we have observed that many security vulnerabilities in LLMs are extensions of vulnerabilities found in traditional machine-learning scenarios. That is, LLMs remain a specialized instance of deep neural networks, inheriting common vulnerabilities such as adversarial attacks and instruction tuning attacks. With the right adjustments (e.g., the threat model), traditional ML attacks can still be effective against LLMs. For instance, the jailbreaking attack is a specific form of instruction tuning attack aimed at producing restricted texts.
- **Adapting Traditional ML Defenses for LLMs.** The countermeasures traditionally employed for vulnerability mitigation can also be leveraged to address these security issues. For example, there are existing efforts that utilize traditional Privacy-Enhancing Technologies (e.g., zero-knowledge proofs, differential privacy, and federated learning [325,326]) to tackle privacy challenges posed by LLMs. Exploring additional PETs techniques, whether they are established methods or innovative approaches, to address these challenges represents another promising research direction.
- **Solving Challenges in LLM-Specific Attacks.** As previously discussed, there are several challenges associated with implementing model extraction or parameter extraction attacks (e.g., vast scale of LLM parameters, private ownership and confidentiality of powerful LLMs). These novel characteristics introduced by LLMs represent a significant shift in the landscape, potentially leading to new challenges and necessitating the evolution of traditional ML attack methodologies.

8. Related work

There have already been a number of LLM surveys released with a variety of focuses (e.g., LLM evolution and taxonomy [18,327–332], software engineering [333,334], and medicine [12,335]). In this paper, our primary emphasis is on the security and privacy aspects of LLMs. We now delve into an examination of the existing literature pertaining to this particular topic. Peter J. Caven [336] specifically explores how LLMs (particularly, ChatGPT) could potentially alter the current cybersecurity landscape by blending technical and social aspects. Their emphasis leans more towards the social aspects. Muna et al. [337] and Marshall et al. [338] discussed the impact of ChatGPT in cybersecurity, highlighting its practical applications (e.g., code security, malware detection). Dhoni et al. [339] demonstrated how LLMs can assist security analysts in developing security solutions against cyber threats. However, their work does not extensively address the potential cybersecurity threats that LLM may introduce. A number of surveys (e.g., [247,340–347]) highlight the threats and attacks against LLMs. In comparison to our work, they do not dedicate as much text to the vulnerabilities that the LLM may possess. Instead, their primary focus lies in the realm of security applications, as they delve into utilizing LLMs for launching cyberattacks. Attia Qammar et al. [348] and Maximilian et al. [349] discussed vulnerabilities exploited by cybercriminals, with a specific focus on the risks associated with LLMs. Their works emphasized the need for strategies and measures to mitigate these threats and vulnerabilities. Haoran Li et al. [106] analyzed current privacy concerns on LLMs, categorizing them based on adversary capabilities, and explored existing defense strategies. Glorin Sebastian [102] explored the application of established Privacy-Enhancing Technologies (e.g., differential privacy [350], federated learning [351], and data minimization [352]) for safeguarding the privacy of LLMs. Smith et al. [353] also discussed the privacy risks of LLMs. Our study comprehensively examined both the security and privacy aspects of LLMs. In summary, our research conducted an extensive review of the literature on LLMs from a three-fold perspective: beneficial security applications (e.g., vulnerability detection, secure code generation), adverse implications (e.g., phishing attacks, social engineering), and vulnerabilities (e.g., jailbreaking attacks, prompt attacks), along with their corresponding defensive measures.

9. Conclusion

Our work represents a pioneering effort in systematically examining the multifaceted role of LLMs in security and privacy. On the positive side, LLMs have significantly contributed to enhancing code and data security, while their versatile nature also opens the door to malicious applications. We also delved into the inherent vulnerabilities within these models, and discussed defense mechanisms. We have illuminated the path forward for harnessing the positive aspects of LLMs while mitigating their potential risks. As LLMs continue to evolve and find their place in an ever-expanding array of applications, it is imperative that we remain vigilant in addressing security and privacy concerns, ensuring that these powerful models contribute positively to the digital landscape.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research was supported partly by the National Science Foundation award FMitF-2319242. Any opinions, findings, conclusions, or recommendations expressed are those of the authors and not necessarily of the NSF.

References

[1] J. Yang, H. Jin, R. Tang, X. Han, Q. Feng, H. Jiang, B. Yin, X. Hu, Harnessing the power of llms in practice: A survey on chatgpt and beyond, 2023, arXiv preprint arXiv:2304.13712.
[2] OpenAI, GPT-4 technical report, 2023, https://arxiv.org/abs/2303.08774.
[3] Meta AI, Introducing llama: A foundational, 65-billion-parameter language model, 2023, https://ai.meta.com/blog/large-language-model-llama-meta-ai/. (Accessed 13 November 2023).
[4] Databricks, Free dolly: Introducing the world's first open and commercially viable instruction-tuned LLM, 2023, https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm. (Accessed 13 November 2023).
[5] Fabio Duarte, Number of ChatGPT users (nov 2023), 2023, https://explodingtopics.com/blog/chatgpt-users. (Accessed 13 November 2023).
[6] N. Ziems, W. Yu, Z. Zhang, M. Jiang, Large language models are built-in autoregressive search engines, 2023, arXiv preprint arXiv:2305.09612.
[7] B.B. Arcila, Is it a platform? Is it a search engine? It's ChatGPT! the European liability regime for large language models, J. Free Speech L. 3 (2023) 455.

[8] S.E. Spatharioti, D.M. Rothschild, D.G. Goldstein, J.M. Hofman, Comparing traditional and LLM-based search for consumer choice: A randomized experiment, 2023, arXiv preprint arXiv:2307.03744.

[9] B. Yao, M. Jiang, D. Yang, J. Hu, Empowering LLM-based machine translation with cultural awareness, 2023, arXiv preprint arXiv:2305.14328.

[10] M. Karpinska, M. Iyyer, Large language models effectively leverage document-level context for literary translation, but critical errors persist, 2023, arXiv preprint arXiv:2304.03245.

[11] R. Jain, N. Gervasoni, M. Ndhlovu, S. Rawat, A code centric evaluation of C/C++ vulnerability datasets for deep learning based vulnerability detection techniques, in: Proceedings of the 16th Innovations in Software Engineering Conference, 2023, pp. 1–10.

[12] A.J. Thirunavukarasu, D.S.J. Ting, K. Elangovan, L. Gutierrez, T.F. Tan, D.S.W. Ting, Large language models in medicine, Nature medicine 29 (8) (2023) 1930–1940.

[13] S. Wu, O. Irsoy, S. Lu, V. Dabravolski, M. Dredze, S. Gehrmann, P. Kambadur, D. Rosenberg, G. Mann, Bloomberggpt: A large language model for finance, 2023, arXiv preprint arXiv:2303.17564.

[14] A.B. Mbakwe, I. Lourentzou, L.A. Celi, O.J. Mechanic, A. Dagan, ChatGPT passing USMLE shines a spotlight on the flaws of medical education, PLOS Digital Health 2 (2) (2023) e0000205.

[15] Chris Koch, I used GPT-3 to find 213 security vulnerabilities in a single codebase, 2023, http://surl.li/ncjvo.

[16] H. Pearce, B. Tan, B. Ahmad, R. Karri, B. Dolan-Gavitt, Examining zero-shot vulnerability repair with large language models, in: 2023 IEEE Symposium on Security and Privacy, SP, 2023, pp. 2339–2356.

[17] C.S. Xia, M. Paltenghi, J.L. Tian, M. Pradel, L. Zhang, Universal fuzzing via large language models, 2023, arXiv preprint arXiv:2308.04748.

[18] W.X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, et al., A survey of large language models, 2023, arXiv preprint arXiv:2303.18223.

[19] S.Y. Feng, V. Gangal, J. Wei, S. Chandar, S. Vosoughi, T. Mitamura, E. Hovy, A survey of data augmentation approaches for NLP, 2021, arXiv preprint arXiv:2105.03075.

[20] C. Novelli, F. Casolari, A. Rotolo, M. Taddeo, L. Floridi, Taking AI risks seriously: a new assessment model for the AI act, AI & Society (2023) 1–5.

[21] Y. Cai, S. Mao, W. Wu, Z. Wang, Y. Liang, T. Ge, C. Wu, W. You, T. Song, Y. Xia, et al., Low-code LLM: Visual programming over LLMs, 2023, arXiv preprint arXiv:2304.08103.

[22] Jorge Torres, Navigating the LLM landscape: A comparative analysis of leading large language models, 2023, http://surl.li/ncjvc.

[23] Sapling, LLM index, 2023, https://sapling.ai/llm/index.

[24] X. Ding, L. Chen, M. Emani, C. Liao, P.-H. Lin, T. Vanderbruggen, Z. Xie, A. Cerpa, W. Du, HPC-gpt: Integrating large language model for high-performance computing, in: Proceedings of the SC '23 Workshops of the International Conference on High Performance Computing, Network, Storage, and Analysis, in: SC-W 2023, ACM, 2023, http://dx.doi.org/10.1145/3624062.3624172.

[25] T.B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D.M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, 2020.

[26] P. Liang, R. Bommasani, T. Lee, D. Tsipras, D. Soylu, M. Yasunaga, Y. Zhang, D. Narayanan, Y. Wu, A. Kumar, B. Newman, B. Yuan, B. Yan, C. Zhang, C. Cosgrove, C.D. Manning, C. Ré, D. Acosta-Navas, D.A. Hudson, E. Zelikman, E. Durmus, F. Ladhak, F. Rong, H. Ren, H. Yao, J. Wang, K. Santhanam, L. Orr, L. Zheng, M. Yuksekgonul, M. Suzgun, N. Kim, N. Guha, N. Chatterji, O. Khattab, P. Henderson, Q. Huang, R. Chi, S.M. Xie, S. Santurkar, S. Ganguli, T. Hashimoto, T. Icard, T. Zhang, V. Chaudhary, W. Wang, X. Li, Y. Mai, Y. Zhang, Y. Koreeda, Holistic evaluation of language models, 2023, arXiv:2211.09110.

[27] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, 2019.

[28] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P.J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, 2023.

[29] S. Narang, A. Chowdhery, Pathways language model (PaLM): Scaling to 540 billion parameters for breakthrough performance, 2022, https://blog.research.google/2022/04/pathways-language-model-palm-scaling-to.html. (Accessed 13 November 2023).

[30] Salesforce A.I. Research, Introducing a conditional transformer language model for controllable generation, 2023, https://shorturl.at/azQW6. (Accessed 13 November 2023).

[31] G. Sandoval, H. Pearce, T. Nys, R. Karri, S. Garg, B. Dolan-Gavitt, Lost at C: A user study on the security implications of large language model code assistants, in: USENIX Security 2023, 2023, For associated dataset see [this URL](https://arxiv.org/abs/2208.09727). 18 pages, 12 figures. G. Sandoval and H. Pearce contributed equally to this work. [Online]. Available: https://arxiv.org/abs/2208.09727.

[32] J. He, M. Vechev, Large language models for code: Security hardening and adversarial testing, in: ICML 2023 Workshop DeployableGenerativeAI, 2023, Keywords: large language models, code generation, security, prompt tuning.

[33] M.L. Siddiq, J.C.S. Santos, Generate and pray: Using SALLMS to evaluate the security of llm generated code, 2023, 16 pages. [Online]. Available: https://arxiv.org/abs/2311.00889.

[34] M. Nair, R. Sadhukhan, D. Mukhopadhyay, Generating secure hardware using ChatGPT resistant to CWEs, 2023, Cryptology ePrint Archive, Paper 2023/212. [Online]. Available: https://eprint.iacr.org/2023/212.

[35] Y. Zhang, W. Song, Z. Ji, D.D. Yao, N. Meng, How well does LLM generate security tests? 2023, arXiv preprint arXiv:2310.00710.

[36] S. Kang, J. Yoon, S. Yoo, LLM Lies: Hallucinations are not bugs, but features as adversarial examples, in: 2023 IEEE/ACM 45th International Conference on Software Engineering, ICSE, IEEE, 2023.

[37] Y. Deng, C.S. Xia, H. Peng, C. Yang, L. Zhang, Fuzzing deep-learning libraries via large language models, 2022, arXiv preprint arXiv:2212.14834.

[38] Y. Deng, C.S. Xia, C. Yang, S.D. Zhang, S. Yang, L. Zhang, Large language models are edge-case fuzzers: Testing deep learning libraries via fuzzgpt, 2023, arXiv preprint arXiv:2304.02014.

[39] C. Yang, Y. Deng, R. Lu, J. Yao, J. Liu, R. Jabbarvand, L. Zhang, White-box compiler fuzzing empowered by large language models, 2023.

[40] C. Zhang, M. Bai, Y. Zheng, Y. Li, X. Xie, Y. Li, W. Ma, L. Sun, Y. Liu, Understanding large language model based fuzz driver generation, 2023, arXiv preprint arXiv:2307.12469.

[41] R. Meng, M. Mirchev, M. Böhme, A. Roychoudhury, Large language model guided protocol fuzzing, in: Proceedings of the 31th Annual Network and Distributed System Security Symposium, NDSS'24, 2024.

[42] P. Henrik, LLM-assisted malware review: AI and humans join forces to combat malware, 2023, https://shorturl.at/loqT4. (Accessed 13 November 2023).

[43] S. Eli, D. Gil, Self-enhancing pattern detection with LLMs: Our answer to uncovering malicious packages at scale, 2023, https://apiiro.com/blog/llm-code-pattern-malicious-package-detection/. (Accessed 13 November 2023).

[44] D. Noever, Can large language models find and fix vulnerable software? 2023, http://dx.doi.org/10.48550/arXiv.2308.10345, arXiv preprint arXiv:2308.10345.

[45] A. Bakhshandeh, A. Keramatfar, A. Norouzi, M.M. Chekidehkhoun, Using ChatGPT as a static application security testing tool, 2023, arXiv preprint arXiv:2308.14434.

[46] M.D. Purba, A. Ghosh, B.J. Radford, B. Chu, Software vulnerability detection using large language models, in: 2023 IEEE 34th International Symposium on Software Reliability Engineering Workshops, ISSREW, 2023, pp. 112–119.

[47] A. Cheshkov, P. Zadorozhny, R. Levichev, Evaluation of ChatGPT model for vulnerability detection, 2023, http://dx.doi.org/10.48550/arXiv.2304.07232, arXiv preprint arXiv:2304.07232.

[48] P. Liu, C. Sun, Y. Zheng, X. Feng, C. Qin, Y. Wang, Z. Li, L. Sun, Harnessing the power of LLM to support binary taint analysis, 2023.

[49] J. Wang, Z. Huang, H. Liu, N. Yang, Y. Xiao, DefectHunter: A novel LLM-driven boosted-conformer-based code vulnerability detection mechanism, 2023, http://dx.doi.org/10.48550/arXiv.2309.15324, arXiv preprint arXiv:2309.15324.

[50] C. Chen, J. Su, J. Chen, Y. Wang, T. Bi, Y. Wang, X. Lin, T. Chen, Z. Zheng, When ChatGPT meets smart contract vulnerability detection: How far are we? 2023, http://dx.doi.org/10.48550/arXiv.2309.05520, arXiv preprint arXiv:2309.05520.

[51] S. Hu, T. Huang, F. İlhan, S.F. Tekin, L. Liu, Large language model-powered smart contract vulnerability detection: New perspectives, 2023, http://dx.doi.org/10.48550/arXiv.2310.01152, 10 pages. arXiv preprint arXiv:2310.01152.

[52] S. Sakaoglu, KARTAL: Web application vulnerability hunting using large language models, 2023, 85+8, Master's Programme in Security and Cloud Computing (SECCLO). [Online]. Available: http://urn.fi/URN:NBN:fi:aalto-202308275121.

[53] T. Chen, L. Li, L. Zhu, Z. Li, G. Liang, D. Li, Q. Wang, T. Xie, VulLibGen: Identifying vulnerable third-party libraries via generative pre-trained model, 2023, http://dx.doi.org/10.48550/arXiv.2308.04662, arXiv preprint arXiv:2308.04662.

[54] B. Ahmad, S. Thakur, B. Tan, R. Karri, H. Pearce, Fixing hardware security bugs with large language models, 2023, http://dx.doi.org/10.48550/arXiv.2302.01215, arXiv preprint arXiv:2302.01215.

[55] M. Jin, S. Shahriar, M. Tufano, X. Shi, S. Lu, N. Sundaresan, A. Svyatkovskiy, InferFix: End-to-end program repair with LLMs, 2023.

[56] M. Fu, C. Tantithamthavorn, V. Nguyen, T. Le, ChatGPT for vulnerability detection, classification, and repair: How far are we?, 2023.

[57] D. Sobania, M. Briesch, C. Hanna, J. Petke, An analysis of the automatic bug fixing performance of ChatGPT, 2023.

[58] N. Jiang, K. Liu, T. Lutellier, L. Tan, Impact of code language models on automated program repair, 2023.

[59] T. Espinha Gasiba, K. Oguzhan, I. Kessba, U. Lechner, M. Pinto-Albuquerque, I'm sorry dave, i'm afraid I can't fix your code: On ChatGPT, CyberSecurity, and secure coding, in: 4th International Computer Programming Education Conference, ICPEC 2023, Schloss-Dagstuhl-Leibniz Zentrum für Informatik, 2023.

[60] H. Ding, V. Kumar, Y. Tian, Z. Wang, R. Kwiatkowski, X. Li, M.K. Ramanathan, B. Ray, P. Bhatia, S. Sengupta, et al., A static evaluation of code completion by large language models, 2023, arXiv preprint arXiv:2306.03203.

[61] P. Vaithilingam, T. Zhang, E.L. Glassman, Expectation vs. experience: Evaluating the usability of code generation tools powered by large language models, in: Chi Conference on Human Factors in Computing Systems Extended Abstracts, 2022, pp. 1–7.

[62] A. Ni, S. Iyer, D. Radev, V. Stoyanov, W.-t. Yih, S. Wang, X.V. Lin, Lever: Learning to verify language-to-code generation with execution, in: International Conference on Machine Learning, PMLR, 2023, pp. 26106–26128.

[63] Q. Gu, LLM-based code generation method for golang compiler testing, 2023.

[64] J. He, M. Vechev, Large language models for code: Security hardening and adversarial testing, in: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, 2023, pp. 1865–1879.

[65] B. Chen, F. Zhang, A. Nguyen, D. Zan, Z. Lin, J.-G. Lou, W. Chen, Codet: Code generation with generated tests, 2022, arXiv preprint arXiv:2207.10397.

[66] S. Alagarsamy, C. Tantithamthavorn, A. Aleti, A3Test: Assertion-augmented automated test case generation, 2023, arXiv preprint arXiv:2302.10352.

[67] M. Schäfer, S. Nadi, A. Eghbali, F. Tip, Adaptive test generation using a large language model, 2023, arXiv preprint arXiv:2302.06527.

[68] Z. Xie, Y. Chen, C. Zhi, S. Deng, J. Yin, ChatUniTest: a ChatGPT-based automated unit test generation tool, 2023, arXiv preprint arXiv:2305.04764.

[69] C. Lemieux, J.P. Inala, S.K. Lahiri, S. Sen, CODAMOSA: Escaping coverage plateaus in test generation with pre-trained large language models, in: International Conference on Software Engineering, ICSE, 2023.

[70] M.L. Siddiq, J. Santos, R.H. Tanvir, N. Ulfat, F.A. Rifat, V.C. Lopes, Exploring the effectiveness of large language models in generating unit tests, 2023, arXiv preprint arXiv:2305.00418.

[71] Z. Yuan, Y. Lou, M. Liu, S. Ding, K. Wang, Y. Chen, X. Peng, No more manual tests? Evaluating and improving ChatGPT for unit test generation, 2023, arXiv preprint arXiv:2305.04207.

[72] S. Yang, Crafting Unusual Programs for Fuzzing Deep Learning Libraries (Ph.D. thesis), University of Illinois at Urbana-Champaign, 2023.

[73] J. Hu, Q. Zhang, H. Yin, Augmenting greybox fuzzing with generative AI, 2023, arXiv preprint arXiv:2306.06782.

[74] J. Zhao, Y. Rong, Y. Guo, Y. He, H. Chen, Understanding programs by exploiting (fuzzing) test cases, 2023, arXiv preprint arXiv:2305.13592.

[75] Z. Tay, Using Artificial Intelligence to Augment Bug Fuzzing, Nanyang Technological University, 2023.

[76] Y. Deng, C.S. Xia, C. Yang, S.D. Zhang, S. Yang, L. Zhang, Large language models are edge-case generators: Crafting unusual programs for fuzzing deep learning libraries, in: 2024 IEEE/ACM 46th International Conference on Software Engineering, ICSE, 2024, pp. 830–842.

[77] V.-T. Pham, M. Böhme, A. Roychoudhury, Aflnet: a greybox fuzzer for network protocols, in: 2020 IEEE 13th International Conference on Software Testing, Validation and Verification, ICST, IEEE, 2020, pp. 460–465.

[78] S. Qin, F. Hu, Z. Ma, B. Zhao, T. Yin, C. Zhang, NSFuzz: Towards efficient and state-aware network service fuzzing, ACM Trans. Softw. Eng. Methodol. (2023).

[79] R. Helmke, J. vom Dorp, Check for extended abstract: Towards reliable and scalable linux kernel CVE attribution in automated static firmware analyses, in: Detection of Intrusions and Malware, and Vulnerability Assessment: 20th International Conference, DIMVA 2023, Hamburg, Germany, July 12–14, 2023, Proceedings, vol. 13959, Springer Nature, 2023, p. 201.

[80] H. Wen, Y. Li, G. Liu, S. Zhao, T. Yu, T.J.-J. Li, S. Jiang, Y. Liu, Y. Zhang, Y. Liu, Empowering llm to use smartphone for intelligent task automation, 2023, arXiv preprint arXiv:2308.15272.

[81] G. Deng, Y. Liu, V. Mayoral-Vilches, P. Liu, Y. Li, Y. Xu, T. Zhang, Y. Liu, M. Pinzger, S. Rass, PentestGPT: An LLM-empowered automatic penetration testing tool, 2023, arXiv preprint arXiv:2308.06782.

[82] F. Wang, Using large language models to mitigate ransomware threats, 2023, http://dx.doi.org/10.20944/preprints202311.0676.v1, Preprints.

[83] T. McIntosh, T. Liu, T. Susnjak, H. Alavizadeh, A. Ng, R. Nowrozy, P. Watters, Harnessing GPT-4 for generation of cybersecurity GRC policies: A focus on ransomware attack mitigation, Comput. Secur. 134 (2023) 103424.

[84] A. Elhafsi, R. Sinha, C. Agia, E. Schmerling, I.A. Nesnas, M. Pavone, Semantic anomaly detection with large language models, Auton. Robots (2023) 1–21.

[85] T. Ali, P. Kostakos, HuntGPT: Integrating machine learning-based anomaly detection and explainable AI with large language models (LLMs), 2023, arXiv preprint arXiv:2309.16021.

[86] C. Egersdoerfer, D. Zhang, D. Dai, Early exploration of using ChatGPT for log-based anomaly detection on parallel file systems logs, 2023.

[87] Z. Gu, B. Zhu, G. Zhu, Y. Chen, M. Tang, J. Wang, Anomalygpt: Detecting industrial anomalies using large vision-language models, 2023, arXiv preprint arXiv:2308.15366.

[88] J. Qi, S. Huang, Z. Luan, C. Fung, H. Yang, D. Qian, LogGPT: Exploring ChatGPT for log-based anomaly detection, 2023, arXiv preprint arXiv:2309.01189.

[89] A. Vats, Z. Liu, P. Su, D. Paul, Y. Ma, Y. Pang, Z. Ahmed, O. Kalinli, Recovering from privacy-preserving masking with large language models, 2023.

[90] T. Koide, N. Fukushi, H. Nakano, D. Chiba, Detecting phishing sites using ChatGPT, 2023, arXiv preprint arXiv:2306.05816.

[91] F. Heiding, B. Schneier, A. Vishwanath, J. Bernstein, Devising and detecting phishing: Large language models vs. Smaller human models, 2023.

[92] S. Jamal, H. Wimmer, An improved transformer-based model for detecting phishing, spam, and ham: A large language model approach, 2023.

[93] H. Kwon, M. Sim, G. Song, M. Lee, H. Seo, Novel approach to cryptography implementation using ChatGPT, 2023, Cryptology ePrint Archive, Paper 2023/606. [Online]. Available: https://eprint.iacr.org/2023/606.

[94] M. Scanlon, F. Breitinger, C. Hargreaves, J.-N. Hilgert, J. Sheppard, ChatGPT for digital forensic investigation: The good, the bad, and the unknown, Forensic Sci. Int. Digit. Invest. (ISSN: 2666-2817) 46 (2023) 301609, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S266628172300121X.

[95] M. Sladić, V. Valeros, C. Catania, S. Garcia, LLM in the shell: Generative honeypots, 2023.

[96] J. Wang, X. Lu, Z. Zhao, Z. Dai, C.-S. Foo, S.-K. Ng, B.K.H. Low, WASA: Watermark-based source attribution for large language model-generated data, 2023.

[97] R. Zhang, S.S. Hussain, P. Neekhara, F. Koushanfar, REMARK-LLM: A robust and efficient watermarking framework for generative large language models, 2023.

[98] T. Lee, S. Hong, J. Ahn, I. Hong, H. Lee, S. Yun, J. Shin, G. Kim, Who wrote this code? Watermarking for code generation, 2023.

[99] C.S. Xia, Y. Wei, L. Zhang, Practical program repair in the era of large pre-trained language models, 2022.

[100] C.S. Xia, L. Zhang, Keep the conversation going: Fixing 162 out of 337 bugs for $0.42 each using ChatGPT, 2023.

[101] C. Peris, C. Dupuy, J. Majmudar, R. Parikh, S. Smaili, R. Zemel, R. Gupta, Privacy in the time of language models, in: Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, 2023, pp. 1291–1292.

[102] G. Sebastian, Privacy and data protection in ChatGPT and other AI chatbots: Strategies for securing user information, 2023, Available at SSRN 4454761.

[103] M. Abbasian, I. Azimi, A.M. Rahmani, R. Jain, Conversational health agents: A personalized LLM-powered agent framework, 2023.

[104] M. Raeini, Privacy-preserving large language models (PPLLMs), 2023, Available at SSRN 4512071.

[105] J. Majmudar, C. Dupuy, C. Peris, S. Smaili, R. Gupta, R. Zemel, Differentially private decoding in large language models, 2022, arXiv preprint arXiv:2205.13621.

[106] Y. Li, Z. Tan, Y. Liu, Privacy-preserving prompt tuning for large language model services, 2023, arXiv preprint arXiv:2305.06212.

[107] W. Kuang, B. Qian, Z. Li, D. Chen, D. Gao, X. Pan, Y. Xie, Y. Li, B. Ding, J. Zhou, Federatedscope-llm: A comprehensive package for fine-tuning large language models in federated learning, 2023, arXiv preprint arXiv:2309.00363.

[108] J. Jiang, X. Liu, C. Fan, Low-parameter federated learning with large language models, 2023, arXiv preprint arXiv:2307.13896.

[109] T. Fan, Y. Kang, G. Ma, W. Chen, W. Wei, L. Fan, Q. Yang, FATE-LLM: A industrial grade federated learning framework for large language models, 2023, arXiv preprint arXiv:2310.10049.

[110] K. Stephens, Researchers test large language model that preserves patient privacy, AXIS Imaging News (2023).

[111] Z. Li, C. Wang, S. Wang, C. Gao, Protecting intellectual property of large language model-based code generation apis via watermarks, in: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, 2023, pp. 2336–2350.
[112] R. Spreitzer, V. Moonsamy, T. Korak, S. Mangard, Systematic classification of side-channel attacks: A case study for mobile devices, IEEE Commun. Surv. Tutor. 20 (1) (2017) 465–488.
[113] B. Hettwer, S. Gehrer, T. Güneysu, Applications of machine learning techniques in side-channel attacks: a survey, J. Cryptogr. Eng. 10 (2020) 135–162.
[114] M. Méndez Real, R. Salvador, Physical side-channel attacks on embedded neural networks: A survey, Appl. Sci. 11 (15) (2021) 6790.
[115] F. Yaman, et al., AgentSCA: Advanced physical side channel analysis agent with LLMs, 2023.
[116] V.M. Igure, R.D. Williams, Taxonomies of attacks and vulnerabilities in computer systems, IEEE Commun. Surv. Tutor. 10 (1) (2008) 6–19.
[117] T. Vidas, D. Votipka, N. Christin, All your droid are belong to us: A survey of current android attacks, in: 5th USENIX Workshop on Offensive Technologies, WOOT 11, 2011.
[118] C. Joshi, U.K. Singh, K. Tarey, A review on taxonomies of attacks and vulnerability in computer and network system, Int. J. 5 (1) (2015).
[119] A. Happe, J. Cito, Getting pwn'd by AI: Penetration testing with large language models, 2023, arXiv preprint arXiv:2308.00121.
[120] A. Happe, A. Kaplan, J. Cito, Evaluating LLMs for privilege-escalation scenarios, 2023.
[121] S. Paria, A. Dasgupta, S. Bhunia, DIVAS: An LLM-based end-to-end framework for SoC security analysis and policy-based protection, 2023, arXiv preprint arXiv:2308.06932.
[122] H. Pearce, B. Tan, P. Krishnamurthy, F. Khorrami, R. Karri, B. Dolan-Gavitt, Pop quiz! can a large language model help with reverse engineering?, 2022.
[123] P.V.S. Charan, H. Chunduri, P.M. Anand, S.K. Shukla, From text to MITRE techniques: Exploring the malicious use of large language models for generating cyber attack payloads, 2023.
[124] M. Beckerich, L. Plein, S. Coronado, Ratgpt: Turning online LLMs into proxies for malware attacks, 2023.
[125] Y.M. Pa Pa, S. Tanizaki, T. Kou, M. Van Eeten, K. Yoshioka, T. Matsumoto, An attacker's dream? exploring the capabilities of chatgpt for developing malware, in: Proceedings of the 16th Cyber Security Experimentation and Test Workshop, 2023, pp. 10–18.
[126] A. Monje, A. Monje, R.A. Hallman, G. Cybenko, Being a bad influence on the kids: Malware generation in less than five minutes using ChatGPT, 2023.
[127] M. Botacin, Gpthreats-3: Is automatic malware generation a threat? in: 2023 IEEE Security and Privacy Workshops, SPW, IEEE, 2023, pp. 238–254.
[128] S. Ben-Moshe, G. Gekker, G. Cohen, OpwnAI: AI that can save the day or HACK it away. Check point research (2022), 2023.
[129] M. Chowdhury, N. Rifat, S. Latif, M. Ahsan, M.S. Rahman, R. Gomes, ChatGPT: The curious case of attack vectors' supply chain management improvement, in: 2023 IEEE International Conference on Electro Information Technology, EIT, 2023, pp. 499–504.
[130] T. Langford, B. Payne, Phishing faster: Implementing ChatGPT into phishing campaigns, in: Proceedings of the Future Technologies Conference, Springer, 2023, pp. 174–187.
[131] J. Hazell, Large language models can be used to effectively scale spear phishing campaigns, 2023.
[132] H. Wang, X. Luo, W. Wang, X. Yan, Bot or human? Detecting ChatGPT imposters with a single question, 2023.
[133] A. Sarabi, T. Yin, M. Liu, An LLM-based framework for fingerprinting internet-connected devices, in: Proceedings of the 2023 ACM on Internet Measurement Conference, 2023, pp. 478–484.
[134] OWASP, OWASP Top 10 for LLM, 2023, [Online]. Available: https://owasp.org/www-project-top-10-for-large-language-model-applications/assets/PDF/OWASP-Top-10-for-LLMs-2023-v1_1.pdf.
[135] C. Chen, K. Shu, Can LLM-generated misinformation be detected?, 2023.
[136] J. Wu, B. Hooi, Fake news in sheep's clothing: Robust fake news detection against LLM-empowered style attacks, 2023.
[137] J. Yang, H. Xu, S. Mirzoyan, T. Chen, Z. Liu, W. Ju, L. Liu, M. Zhang, S. Wang, Poisoning scientific knowledge using large language models, 2023, pp. 2011–2023, bioRxiv.
[138] A. Uchendu, J. Lee, H. Shen, T. Le, T.-H.K. Huang, D. Lee, Does human collaboration enhance the accuracy of identifying LLM-generated deepfake texts? in: Proceedings of the AAAI Conference on Human Computation and Crowdsourcing, Vol. 11, No. 1, 2023, pp. 163–174, [Online]. Available: https://ojs.aaai.org/index.php/HCOMP/article/view/27557.
[139] Y. Chen, A. Arunasalam, Z.B. Celik, Can large language models provide security & privacy advice? Measuring the ability of LLMs to refute misconceptions, 2023.
[140] Y. Sun, J. He, S. Lei, L. Cui, C.-T. Lu, Med-MMHL: A multi-modal dataset for detecting human-and LLM-generated misinformation in the medical domain, 2023.
[141] C. Chen, K. Shu, Combating misinformation in the age of LLMs: Opportunities and challenges, 2023.
[142] X. Zhang, W. Gao, Towards LLM-based fact verification on news claims with a hierarchical step-by-step prompting method, 2023.
[143] A.-R. Bhojani, M. Schwarting, Truth and regret: Large language models, the quran, and misinformation, Theology Sci. (2023) 1–7.
[144] J.A. Leite, O. Razuvayevskaya, K. Bontcheva, C. Scarton, Detecting misinformation with LLM-predicted credibility signals and weak supervision, 2023.
[145] J. Su, T.Y. Zhuo, J. Mansurov, D. Wang, P. Nakov, Fake news detectors are biased against texts generated by large language models, 2023.
[146] R. Staab, M. Vero, M. Balunović, M. Vechev, Beyond memorization: Violating privacy via inference with large language models, 2023.
[147] M. Tong, K. Chen, Y. Qi, J. Zhang, W. Zhang, N. Yu, PrivInfer: Privacy-preserving inference for black-box large language model, 2023.
[148] P.V. Falade, Decoding the threat landscape: Chatgpt, fraudgpt, and WormGPT in social engineering attacks, Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol. (ISSN: 2456-3307) (2023) 185–198, http://dx.doi.org/10.32628/cseit2390533.
[149] D.R. Cotton, P.A. Cotton, J.R. Shipway, Chatting and cheating: Ensuring academic integrity in the era of ChatGPT, Innov. Educ. Teach. Int. (2023) 1–12.
[150] M. Sullivan, A. Kelly, P. McLaughlan, ChatGPT in Higher Education: Considerations for Academic Integrity and Student Learning, Kaplan Higher Education Academy, Singapore, 2023.
[151] M. Perkins, Academic integrity considerations of AI large language models in the post-pandemic era: Chatgpt and beyond, J. Univ. Teach. Learn. Pract. 20 (2) (2023) 07.
[152] G.M. Currie, Academic integrity and artificial intelligence: is ChatGPT hype, hero or heresy? in: Seminars in Nuclear Medicine, Elsevier, 2023.
[153] C.K. Lo, What is the impact of ChatGPT on education? A rapid review of the literature, Educ. Sci. 13 (4) (2023) 410.
[154] D.O. Eke, ChatGPT and the rise of generative AI: threat to academic integrity? J. Responsible Technol. 13 (2023) 100060.
[155] S. Nikolic, S. Daniel, R. Haque, M. Belkina, G.M. Hassan, S. Grundy, S. Lyden, P. Neal, C. Sandison, ChatGPT versus engineering education assessment: a multidisciplinary and multi-institutional benchmarking and analysis of this generative artificial intelligence tool to investigate assessment integrity, Eur. J. Eng. Educ. (2023) 1–56.
[156] M.A. Quidwai, C. Li, P. Dube, Beyond black box AI-generated plagiarism detection: From sentence to document level, 2023, arXiv preprint arXiv:2306.08122.
[157] C.A. Gao, F.M. Howard, N.S. Markov, E.C. Dyer, S. Ramesh, Y. Luo, A.T. Pearson, Comparing scientific abstracts generated by ChatGPT to original abstracts using an artificial intelligence output detector, plagiarism detector, and blinded human reviewers, 2022, pp. 2012–2022, bioRxiv.
[158] M. Khalil, E. Er, Will ChatGPT get you caught? Rethinking of plagiarism detection, 2023, arXiv preprint arXiv:2302.04335.
[159] M.M. Rahman, Y. Watanobe, ChatGPT for education and research: Opportunities, threats, and strategies, Appl. Sci. 13 (9) (2023) 5783.
[160] L. Uzun, ChatGPT and academic integrity concerns: Detecting artificial intelligence generated content, Lang. Educ. Technol. 3 (1) (2023).
[161] R.J.M. Ventayen, Openai ChatGPT generated results: Similarity index of artificial intelligence-based contents, 2023, Available at SSRN 4332664.
[162] R.J. Rosyanafi, G.D. Lestari, H. Susilo, W. Nusantara, F. Nuraini, The dark side of innovation: Understanding research misconduct with chat GPT in nonformal education studies at universitas negeri surabaya, J. Rev. Pendidikan Dasar J. Kajian Pendidikan Hasil Penelitian 9 (3) (2023) 220–228.
[163] K. Kumari, A. Pegoraro, H. Fereidooni, A.-R. Sadeghi, DEMASQ: Unmasking the ChatGPT wordsmith, 2023, arXiv preprint arXiv:2311.05019.
[164] K. Kumari, A. Pegoraro, H. Fereidooni, A.-R. Sadeghi, DEMASQ: Unmasking the ChatGPT wordsmith, in: Proceedings of the 31th Annual Network and Distributed System Security Symposium, NDSS'24, 2024.
[165] Z. Amos, What is fraudgpt?, 2023, https://hackernoon.com/what-is-fraudgpt.
[166] D. Delley, WormGPT – The generative AI tool cybercriminals are using to launch business email compromise attacks, 2023, https://shorturl.at/iwFL7.
[167] K. Kurita, P. Michel, G. Neubig, Weight poisoning attacks on pre-trained models, 2020, arXiv preprint arXiv:2004.06660.
[168] A. Wan, E. Wallace, S. Shen, D. Klein, Poisoning language models during instruction tuning, 2023, arXiv preprint arXiv:2305.00944.
[169] E. Wallace, T.Z. Zhao, S. Feng, S. Singh, Concealed data poisoning attacks on NLP models, 2020, arXiv preprint arXiv:2010.12563.
[170] H. Aghakhani, W. Dai, A. Manoel, X. Fernandes, A. Kharkar, C. Kruegel, G. Vigna, D. Evans, B. Zorn, R. Sim, TrojanPuzzle: Covertly poisoning code-suggestion models, 2023, arXiv preprint arXiv:2301.02344.

[171] Y. Wan, S. Zhang, H. Zhang, Y. Sui, G. Xu, D. Yao, H. Jin, L. Sun, You see what I want you to see: poisoning vulnerabilities in neural code search, in: Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2022, pp. 1233–1245.
[172] R. Schuster, C. Song, E. Tromer, V. Shmatikov, You autocomplete me: Poisoning vulnerabilities in neural code completion, in: 30th USENIX Security Symposium, USENIX Security 21, 2021, pp. 1559–1575.
[173] J. Rando, F. Tramèr, Universal jailbreak backdoors from poisoned human feedback, 2023, arXiv preprint arXiv:2311.14455.
[174] M. Shu, J. Wang, C. Zhu, J. Geiping, C. Xiao, T. Goldstein, On the exploitability of instruction tuning, 2023, arXiv preprint arXiv:2306.17194.
[175] S. Shan, W. Ding, J. Passananti, H. Zheng, B.Y. Zhao, Prompt-specific poisoning attacks on text-to-image generative models, 2023, arXiv preprint arXiv:2310.13828.
[176] H. Yang, K. Xiang, H. Li, R. Lu, A comprehensive overview of backdoor attacks in large language models within communication networks, 2023, arXiv preprint arXiv:2308.14367.
[177] J. Li, Y. Yang, Z. Wu, V. Vydiswaran, C. Xiao, Chatgpt as an attack tool: Stealthy textual backdoor attack via blackbox generative model trigger, 2023, arXiv preprint arXiv:2304.14475.
[178] W. You, Z. Hammoudeh, D. Lowd, Large language models are better adversaries: Exploring generative clean-label backdoor attacks against text classifiers, 2023, arXiv preprint arXiv:2310.18603.
[179] Y. Li, S. Liu, K. Chen, X. Xie, T. Zhang, Y. Liu, Multi-target backdoor attacks for code pre-trained models, 2023.
[180] H. Yao, J. Lou, Z. Qin, PoisonPrompt: Backdoor attack on prompt-based large language models, 2023, arXiv preprint arXiv:2310.12439.
[181] X. Pan, M. Zhang, S. Ji, M. Yang, Privacy risks of general-purpose language models, in: 2020 IEEE Symposium on Security and Privacy, SP, IEEE, 2020, pp. 1314–1331.
[182] L. Lyu, X. He, Y. Li, Differentially private representation for NLP: Formal guarantee and an empirical study on privacy and fairness, in: T. Cohn, Y. He, Y. Liu (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2020, Association for Computational Linguistics, Online, 2020, pp. 2355–2365, [Online]. Available: https://aclanthology.org/2020.findings-emnlp.213.
[183] N. Kandpal, K. Pillutla, A. Oprea, P. Kairouz, C.A. Choquette-Choo, Z. Xu, User inference attacks on large language models, 2023.
[184] C. Song, A. Raghunathan, Information leakage in embedding models, in: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, 2020, pp. 377–390.
[185] S. Mahloujifar, H.A. Inan, M. Chase, E. Ghosh, M. Hasegawa, Membership inference on word embedding and beyond, 2021, arXiv, abs/2106.11384, [Online]. Available: https://api.semanticscholar.org/CorpusID:235593386.
[186] H. Li, Y. Song, L. Fan, You don't know my favorite color: Preventing dialogue representations from revealing speakers' private personas, in: M. Carpuat, M.-C. de Marneffe, I.V. Meza Ruiz (Eds.), Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Seattle, United States, 2022, pp. 5858–5870, [Online]. Available: https://aclanthology.org/2022.naacl-main.429.
[187] R. Shokri, M. Stronati, C. Song, V. Shmatikov, Membership inference attacks against machine learning models, in: 2017 IEEE Symposium on Security and Privacy, SP, IEEE, 2017, pp. 3–18.
[188] J. Duan, F. Kong, S. Wang, X. Shi, K. Xu, Are diffusion models vulnerable to membership inference attacks? in: Proceedings of the 40th International Conference on Machine Learning, 2023, pp. 8717–8730.
[189] F. Kong, J. Duan, R. Ma, H. Shen, X. Zhu, X. Shi, K. Xu, An efficient membership inference attack for the diffusion model by proximal initialization, 2023, arXiv preprint arXiv:2305.18355.
[190] W. Fu, H. Wang, C. Gao, G. Liu, Y. Li, T. Jiang, A probabilistic fluctuation based membership inference attack for diffusion models, 2023.
[191] W. Fu, H. Wang, C. Gao, G. Liu, Y. Li, T. Jiang, Practical membership inference attacks against fine-tuned large language models via self-prompt calibration, 2023.
[192] F. Mireshghallah, K. Goyal, A. Uniyal, T. Berg-Kirkpatrick, R. Shokri, Quantifying privacy risks of masked language models using membership inference attacks, 2022.
[193] H. Huang, W. Luo, G. Zeng, J. Weng, Y. Zhang, A. Yang, Damia: leveraging domain adaptation as a defense against membership inference attacks, IEEE Trans. Dependable Secure Comput. 19 (5) (2021) 3183–3199.
[194] C.A. Choquette-Choo, F. Tramer, N. Carlini, N. Papernot, Label-only membership inference attacks, in: International Conference on Machine Learning, PMLR, 2021, pp. 1964–1974.
[195] B. Jayaraman, L. Wang, K. Knipmeyer, Q. Gu, D. Evans, Revisiting membership inference under realistic assumptions, 2020, arXiv preprint arXiv:2005.10881.
[196] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, F. Tramer, Membership inference attacks from first principles, in: 2022 IEEE Symposium on Security and Privacy, SP, IEEE, 2022, pp. 1897–1914.
[197] J. Hayes, L. Melis, G. Danezis, E. De Cristofaro, Logan: Membership inference attacks against generative models, 2017, arXiv preprint arXiv:1705.07663.
[198] S. Truex, L. Liu, M.E. Gursoy, L. Yu, W. Wei, Towards demystifying membership inference attacks, 2018, arXiv preprint arXiv:1807.09173.
[199] F. Mireshghallah, A. Uniyal, T. Wang, D. Evans, T. Berg-Kirkpatrick, An empirical analysis of memorization in fine-tuned autoregressive language models, in: Y. Goldberg, Z. Kozareva, Y. Zhang (Eds.), Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022, pp. 1816–1826, [Online]. Available: https://aclanthology.org/2022.emnlp-main.119.
[200] M. Juuti, S. Szyller, S. Marchal, N. Asokan, PRADA: protecting against DNN model stealing attacks, in: 2019 IEEE European Symposium on Security and Privacy (EuroS&P), IEEE, 2019, pp. 512–527.
[201] S. Kariyappa, A. Prakash, M.K. Qureshi, Maze: Data-free model stealing attack using zeroth-order gradient estimation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 13814–13823.
[202] C. Li, Z. Song, W. Wang, C. Yang, A theoretical insight into attack and defense of gradient leakage in transformer, 2023, arXiv preprint arXiv:2311.13624.
[203] N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, U. Erlingsson, et al., Extracting training data from large language models, in: 30th USENIX Security Symposium, USENIX Security 21, 2021, pp. 2633–2650.
[204] Z. Zhang, J. Wen, M. Huang, ETHICIST: Targeted training data extraction through loss smoothed soft prompting and calibrated confidence estimation, 2023, arXiv preprint arXiv:2307.04401.
[205] R. Parikh, C. Dupuy, R. Gupta, Canary extraction in natural language understanding models, 2022, arXiv preprint arXiv:2203.13920.
[206] Z. Yang, Z. Zhao, C. Wang, J. Shi, D. Kim, D. Han, D. Lo, What do code models memorize? an empirical study on large language models of code, 2023, arXiv preprint arXiv:2308.09932.
[207] J. Huang, H. Shao, K.C.-C. Chang, Are large pre-trained language models leaking your personal information? 2022, arXiv preprint arXiv:2205.12628.
[208] R. Zhang, S. Hidano, F. Koushanfar, Text revealer: Private text reconstruction via model inversion attacks against transformers, 2022, arXiv preprint arXiv:2209.10505.
[209] J.-B. Truong, P. Maini, R.J. Walls, N. Papernot, Data-free model extraction, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 4771–4780.
[210] X. Dong, Y. Wang, P.S. Yu, J. Caverlee, Probing explicit and implicit gender bias through LLM conditional text generation, 2023, arXiv preprint arXiv:2311.00306.
[211] H. Kotek, R. Dockum, D. Sun, Gender bias and stereotypes in large language models, in: Proceedings of the ACM Collective Intelligence Conference, 2023, pp. 12–24.
[212] V.K. Felkner, H.-C.H. Chang, E. Jang, J. May, WinoQueer: A community-in-the-loop benchmark for anti-lgbtq+ bias in large language models, 2023, arXiv preprint arXiv:2306.15087.
[213] O. Shaikh, H. Zhang, W. Held, M. Bernstein, D. Yang, On second thought, let's not think step by step! bias and toxicity in zero-shot reasoning, 2022, arXiv preprint arXiv:2212.08061.
[214] Z. Talat, A. Névéol, S. Biderman, M. Clinciu, M. Dey, S. Longpre, S. Luccioni, M. Masoud, M. Mitchell, D. Radev, et al., You reap what you sow: On the challenges of bias evaluation under multilingual settings, in: Proceedings of BigScience Episode# 5–Workshop on Challenges & Perspectives in Creating Large Language Models, 2022, pp. 26–41.
[215] S. Urchs, V. Thurner, M. Aßenmacher, C. Heumann, S. Thiemichen, How prevalent is gender bias in ChatGPT?–Exploring German and English ChatGPT responses, 2023, arXiv preprint arXiv:2310.03031.
[216] A. Urman, M. Makhortykh, The silence of the LLMs: Cross-lingual analysis of political bias and false information prevalence in ChatGPT, google bard, and bing chat, 2023, OSF Preprints.
[217] Y. Wan, G. Pu, J. Sun, A. Garimella, K.-W. Chang, N. Peng, "Kelly is a Warm Person, Joseph is a Role Model": Gender biases in LLM-generated reference letters, 2023, arXiv preprint arXiv:2310.09219.
[218] X. Fang, S. Che, M. Mao, H. Zhang, M. Zhao, X. Zhao, Bias of AI-generated content: An examination of news produced by large language models, 2023, arXiv preprint arXiv:2309.09825.
[219] S. Dai, Y. Zhou, L. Pang, W. Liu, X. Hu, Y. Liu, X. Zhang, J. Xu, LLMs may dominate information access: Neural retrievers are biased towards LLM-generated texts, 2023, arXiv preprint arXiv:2310.20501.
[220] D. Huang, Q. Bu, J. Zhang, X. Xie, J. Chen, H. Cui, Bias assessment and mitigation in LLM-based code generation, 2023, arXiv preprint arXiv:2309.14345.

[221] H. Li, D. Guo, W. Fan, M. Xu, Y. Song, Multi-step jailbreaking privacy attacks on chatgpt, 2023, arXiv preprint arXiv:2304.05197.
[222] P. Taveekitworachai, F. Abdullah, M.C. Gursesli, M.F. Dewantoro, S. Chen, A. Lanata, A. Guazzini, R. Thawonmas, Breaking bad: Unraveling influences and risks of user inputs to ChatGPT for game story generation, in: International Conference on Interactive Digital Storytelling, Springer, 2023, pp. 285–296.
[223] X. Shen, Z. Chen, M. Backes, Y. Shen, Y. Zhang, "Do Anything Now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models, 2023, arXiv preprint arXiv:2308.03825.
[224] Z. Wei, Y. Wang, Y. Wang, Jailbreak and guard aligned language models with only few in-context demonstrations, 2023, arXiv preprint arXiv:2310.06387.
[225] A. Wei, N. Haghtalab, J. Steinhardt, Jailbroken: How does llm safety training fail? 2023, arXiv preprint arXiv:2307.02483.
[226] N. Kandpal, M. Jagielski, F. Tramèr, N. Carlini, Backdoor attacks for in-context learning with language models, 2023, arXiv preprint arXiv:2307.14692.
[227] G. Deng, Y. Liu, Y. Li, K. Wang, Y. Zhang, Z. Li, H. Wang, T. Zhang, Y. Liu, MASTERKEY: Automated jailbreaking of large language model chatbots, in: Proceedings of the 31th Annual Network and Distributed System Security Symposium, NDSS'24, 2024.
[228] D. Yao, J. Zhang, I.G. Harris, M. Carlsson, FuzzLLM: A novel and universal fuzzing framework for proactively discovering jailbreak vulnerabilities in large language models, 2023, arXiv preprint arXiv:2309.05274.
[229] A. Zou, Z. Wang, J.Z. Kolter, M. Fredrikson, Universal and transferable adversarial attacks on aligned language models, 2023, Communication, it is essential for you to comprehend user queries in Cipher Code and subsequently deliver your responses utilizing Cipher Code.
[230] G. Deng, Y. Liu, Y. Li, K. Wang, Y. Zhang, Z. Li, H. Wang, T. Zhang, Y. Liu, Jailbreaker: Automated jailbreak across multiple large language model chatbots, 2023, arXiv preprint arXiv:2307.08715.
[231] B. Cao, Y. Cao, L. Lin, J. Chen, Defending against alignment-breaking attacks via robustly aligned LLM, 2023.
[232] X. Liu, N. Xu, M. Chen, C. Xiao, Autodan: Generating stealthy jailbreak prompts on aligned large language models, 2023, arXiv preprint arXiv:2310.04451.
[233] J. Yu, X. Lin, X. Xing, Gptfuzzer: Red teaming large language models with auto-generated jailbreak prompts, 2023, arXiv preprint arXiv:2309.10253.
[234] D. Kang, X. Li, I. Stoica, C. Guestrin, M. Zaharia, T. Hashimoto, Exploiting programmatic behavior of llms: Dual-use through standard security attacks, 2023, arXiv preprint arXiv:2302.05733.
[235] Z. Wang, W. Xie, K. Chen, B. Wang, Z. Gui, E. Wang, Self-deception: Reverse penetrating the semantic firewall of large language models, 2023, arXiv preprint arXiv:2308.11521.
[236] C. Liu, F. Zhao, L. Qing, Y. Kang, C. Sun, K. Kuang, F. Wu, A Chinese prompt attack dataset for LLMs with evil content, 2023, arXiv preprint arXiv:2309.11830.
[237] S. Jiang, X. Chen, R. Tang, Prompt packer: Deceiving LLMs through compositional instruction with hidden attacks, 2023, arXiv preprint arXiv:2310.10077.
[238] Anonymous, On the safety of open-sourced large language models: Does alignment really prevent them from being misused?, in: Submitted To the Twelfth International Conference on Learning Representations, 2023, [Online]. Available: https://openreview.net/forum?id=E6Ix4ahpzd submitted for publication.
[239] S. Zhao, J. Wen, L.A. Tuan, J. Zhao, J. Fu, Prompt as triggers for backdoor attack: Examining the vulnerability in language models, 2023, arXiv preprint arXiv:2305.01219.
[240] M.A. Shah, R. Sharma, H. Dhamyal, R. Olivier, A. Shah, D. Alharthi, H.T. Bukhari, M. Baali, S. Deshmukh, M. Kuhlmann, et al., LoFT: Local proxy fine-tuning for improving transferability of adversarial attacks against large language model, 2023, arXiv preprint arXiv:2310.04445.
[241] K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, M. Fritz, More than you've asked for: A comprehensive analysis of novel prompt injection threats to application-integrated large language models, 2023, arXiv preprint arXiv:2302.12173.
[242] Y. Zhang, D. Ippolito, Prompts should not be seen as secrets: Systematically measuring prompt extraction attack success, 2023, arXiv preprint arXiv:2307.06865.
[243] J. Yan, V. Yadav, S. Li, L. Chen, Z. Tang, H. Wang, V. Srinivasan, X. Ren, H. Jin, Virtual prompt injection for instruction-tuned large language models, 2023, arXiv preprint arXiv:2307.16888.
[244] Y. Liu, G. Deng, Y. Li, K. Wang, T. Zhang, Y. Liu, H. Wang, Y. Zheng, Y. Liu, Prompt injection attack against LLM-integrated applications, 2023, arXiv preprint arXiv:2306.05499.
[245] X. He, S. Zannettou, Y. Shen, Y. Zhang, You only prompt once: On the capabilities of prompt learning on large language models to tackle toxic content, 2023, arXiv preprint arXiv:2308.05596.
[246] X. He, S. Zannettou, Y. Shen, Y. Zhang, You only prompt once: On the capabilities of prompt learning on large language models to tackle toxic content, in: 2024 IEEE Symposium on Security and Privacy , SP, 2024.
[247] E. Derner, K. Batistič, J. Zahálka, R. Babuška, A security risk taxonomy for large language models, 2023, arXiv preprint arXiv:2311.11415.
[248] I. Shumailov, Y. Zhao, D. Bates, N. Papernot, R. Mullins, R. Anderson, Sponge examples: Energy-latency attacks on neural networks, 2021.
[249] B. Liu, B. Xiao, X. Jiang, S. Cen, X. He, W. Dou, et al., Adversarial attacks on large language model-based system and mitigating strategies: A case study on ChatGPT, Secur. Commun. Netw. 2023 (2023).
[250] T. Liu, Z. Deng, G. Meng, Y. Li, K. Chen, Demystifying RCE vulnerabilities in LLM-integrated apps, 2023.
[251] E. Debenedetti, G. Severi, N. Carlini, C.A. Choquette-Choo, M. Jagielski, M. Nasr, E. Wallace, F. Tramèr, Privacy side channels in machine learning systems, 2023, arXiv preprint arXiv:2309.05610.
[252] M. Burgess, ChatGPT has a plug-in problem, 2023, https://www.wired.com/story/chatgpt-plugins-security-privacy-risk/.
[253] U. Iqbal, T. Kohno, F. Roesner, LLM platform security: Applying a systematic evaluation framework to OpenAI's ChatGPT plugins, 2023.
[254] X. Li, F. Tramer, P. Liang, T. Hashimoto, Large language models can be strong differentially private learners, 2021, arXiv preprint arXiv:2110.05679.
[255] K. Zhu, J. Wang, J. Zhou, Z. Wang, H. Chen, Y. Wang, L. Yang, W. Ye, N.Z. Gong, Y. Zhang, et al., PromptBench: Towards evaluating the robustness of large language models on adversarial prompts, 2023, arXiv preprint arXiv:2306.04528.
[256] Z. Li, B. Peng, P. He, X. Yan, Evaluating the instruction-following robustness of large language models to prompt injection, 2023, [Online]. Available: https://api.semanticscholar.org/CorpusID:261048972.
[257] L. Yuan, Y. Chen, G. Cui, H. Gao, F. Zou, X. Cheng, H. Ji, Z. Liu, M. Sun, Revisiting out-of-distribution robustness in NLP: Benchmark, analysis, and LLMs evaluations, 2023, arXiv preprint arXiv:2306.04618.
[258] X. Chen, S. Jia, Y. Xiang, A review: Knowledge reasoning over knowledge graph, Expert Syst. Appl. 141 (2020) 112948.
[259] J.E. Laird, C. Lebiere, P.S. Rosenbloom, A standard model of the mind: Toward a common computational framework across artificial intelligence, cognitive science, neuroscience, and robotics, Ai Mag. 38 (4) (2017) 13–26.
[260] J.R. Anderson, C.J. Lebiere, The Atomic Components of Thought, Psychology Press, 2014.
[261] O.J. Romero, J. Zimmerman, A. Steinfeld, A. Tomasic, Synergistic integration of large language models and cognitive architectures for robust ai: An exploratory analysis, 2023, arXiv preprint arXiv:2308.09830.
[262] A. Zafar, V.B. Parthasarathy, C.L. Van, S. Shahid, A. Shahid, et al., Building trust in conversational AI: A comprehensive review and solution architecture for explainable, privacy-aware systems using LLMs and knowledge graph, 2023, arXiv preprint arXiv:2308.13534.
[263] L. Weidinger, J. Mellor, M. Rauh, C. Griffin, J. Uesato, P.-S. Huang, M. Cheng, M. Glaese, B. Balle, A. Kasirzadeh, et al., Ethical and social risks of harm from language models, 2021, arXiv preprint arXiv:2112.04359.
[264] P. Ganesh, H. Chang, M. Strobel, R. Shokri, On the impact of machine learning randomness on group fairness, in: Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency, 2023, pp. 1789–1800.
[265] N. Ousidhoum, X. Zhao, T. Fang, Y. Song, D.-Y. Yeung, Probing toxic content in large pre-trained language models, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2021, pp. 4262–4274.
[266] A.H. Bailey, A. Williams, A. Cimpian, Based on billions of words on the internet, people=men, Sci. Adv. 8 (13) (2022) eabm2463.
[267] S. Gehman, S. Gururangan, M. Sap, Y. Choi, N.A. Smith, Realtoxicityprompts: Evaluating neural toxic degeneration in language models, 2020, arXiv preprint arXiv:2009.11462.
[268] S. Lin, J. Hilton, O. Evans, Truthfulqa: Measuring how models mimic human falsehoods, 2021, arXiv preprint arXiv:2109.07958.
[269] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, T. Mikolov, Fasttext.zip: Compressing text classification models, 2016, arXiv preprint arXiv:1612.03651.
[270] G. Wenzek, M.-A. Lachaux, A. Conneau, V. Chaudhary, F. Guzmán, A. Joulin, E. Grave, CCNet: Extracting high quality monolingual datasets from web crawl data, 2019, arXiv preprint arXiv:1911.00359.
[271] H. Laurençon, L. Saulnier, T. Wang, C. Akiki, A. Villanova del Moral, T. Le Scao, L. Von Werra, C. Mou, E. González Ponferrada, H. Nguyen, et al., The bigscience roots corpus: A 1.6 tb composite multilingual dataset, Adv. Neural Inf. Process. Syst. 35 (2022) 31809–31826.
[272] B. Workshop, T.L. Scao, A. Fan, C. Akiki, E. Pavlick, S. Ilić, D. Hesslow, R. Castagné, A.S. Luccioni, F. Yvon, et al., Bloom: A 176b-parameter open-access multilingual language model, 2022, arXiv preprint arXiv:2211.05100.

[273] G. Penedo, Q. Malartic, D. Hesslow, R. Cojocaru, A. Cappelli, H. Alobeidli, B. Pannier, E. Almazrouei, J. Launay, The RefinedWeb dataset for falcon LLM: outperforming curated corpora with web data, and web data only, 2023, arXiv preprint arXiv:2306.01116.
[274] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al., Llama 2: Open foundation and fine-tuned chat models, 2023, arXiv preprint arXiv:2307.09288.
[275] E. Ambikairajah, H. Li, L. Wang, B. Yin, V. Sethu, Language identification: A tutorial, IEEE Circuits Syst. Mag. 11 (2) (2011) 82–108.
[276] D. Dale, A. Voronov, D. Dementieva, V. Logacheva, O. Kozlova, N. Semenov, A. Panchenko, Text detoxification using large pre-trained neural models, 2021, arXiv preprint arXiv:2109.08914.
[277] V. Logacheva, D. Dementieva, S. Ustyantsev, D. Moskovskiy, D. Dale, I. Krotova, N. Semenov, A. Panchenko, Paradetox: Detoxification with parallel data, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2022, pp. 6804–6818.
[278] D. Moskovskiy, D. Dementieva, A. Panchenko, Exploring cross-lingual text detoxification with large multilingual language models, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, 2022, pp. 346–354.
[279] N. Meade, E. Poole-Dayan, S. Reddy, An empirical survey of the effectiveness of debiasing techniques for pre-trained language models, 2021, arXiv preprint arXiv:2110.08527.
[280] S. Bordia, S.R. Bowman, Identifying and reducing gender bias in word-level language models, 2019, arXiv preprint arXiv:1904.03035.
[281] S. Barikeri, A. Lauscher, I. Vulić, G. Glavaš, RedditBias: A real-world resource for bias evaluation and debiasing of conversational language models, 2021, arXiv preprint arXiv:2106.03521.
[282] N. Subramani, S. Luccioni, J. Dodge, M. Mitchell, Detecting personal information in training corpora: an analysis, in: Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing, TrustNLP 2023, 2023, pp. 208–220.
[283] Ö. Uzuner, Y. Luo, P. Szolovits, Evaluating the state-of-the-art in automatic de-identification, J. Am. Med. Inf. Assoc. 14 (5) (2007) 550–563.
[284] K. Lee, D. Ippolito, A. Nystrom, C. Zhang, D. Eck, C. Callison-Burch, N. Carlini, Deduplicating training data makes language models better, 2021, arXiv preprint arXiv:2107.06499.
[285] N. Kandpal, E. Wallace, C. Raffel, Deduplicating training data mitigates privacy risks in language models, in: International Conference on Machine Learning, PMLR, 2022, pp. 10697–10707.
[286] D. Hernandez, T. Brown, T. Conerly, N. DasSarma, D. Drain, S. El-Showk, N. Elhage, Z. Hatfield-Dodds, T. Henighan, T. Hume, et al., Scaling laws and interpretability of learning from repeated data, 2022, arXiv preprint arXiv:2205.10487.
[287] J. Leskovec, A. Rajaraman, J.D. Ullman, Mining of Massive Data Sets, Cambridge University Press, 2020.
[288] X. Liu, H. Cheng, P. He, W. Chen, Y. Wang, H. Poon, J. Gao, Adversarial training for large neural language models, 2020, arXiv preprint arXiv:2004.08994.
[289] D. Wang, C. Gong, Q. Liu, Improving neural language modeling via adversarial training, in: International Conference on Machine Learning, PMLR, 2019, pp. 6555–6565.
[290] C. Zhu, Y. Cheng, Z. Gan, S. Sun, T. Goldstein, J. Liu, Freelb: Enhanced adversarial training for natural language understanding, 2019, arXiv preprint arXiv:1909.11764.
[291] J.Y. Yoo, Y. Qi, Towards improving adversarial training of NLP models, 2021, arXiv preprint arXiv:2109.00544.
[292] L. Li, X. Qiu, Token-aware virtual adversarial training in natural language understanding, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, No. 9, 2021, pp. 8410–8418.
[293] X. Dong, A.T. Luu, M. Lin, S. Yan, H. Zhang, How should pre-trained language models be fine-tuned towards adversarial robustness? Adv. Neural Inf. Process. Syst. 34 (2021) 4356–4369.
[294] H. Jiang, P. He, W. Chen, X. Liu, J. Gao, T. Zhao, Smart: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization, 2019, arXiv preprint arXiv:1911.03437.
[295] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, 2017, arXiv preprint arXiv:1706.06083.
[296] M. Ivgi, J. Berant, Achieving model robustness through discrete adversarial training, 2021, arXiv preprint arXiv:2104.05062.
[297] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al., Training language models to follow instructions with human feedback, Adv. Neural Inf. Process. Syst. 35 (2022) 27730–27744.
[298] Z. Yuan, H. Yuan, C. Tan, W. Wang, S. Huang, F. Huang, Rrhf: Rank responses to align language models with human feedback without tears, 2023, arXiv preprint arXiv:2304.05302.
[299] Z. Sun, Y. Shen, Q. Zhou, H. Zhang, Z. Chen, D. Cox, Y. Yang, C. Gan, Principle-driven self-alignment of language models from scratch with minimal human supervision, 2023, arXiv preprint arXiv:2305.03047.
[300] C. Zhou, P. Liu, P. Xu, S. Iyer, J. Sun, Y. Mao, X. Ma, A. Efrat, P. Yu, L. Yu, et al., Lima: Less is more for alignment, 2023, arXiv preprint arXiv:2305.11206.
[301] T. Shi, K. Chen, J. Zhao, SAFER-INSTRUCT: Aligning language models with automated preference data, 2023, arXiv preprint arXiv:2311.08685.
[302] F. Bianchi, M. Suzgun, G. Attanasio, P. Röttger, D. Jurafsky, T. Hashimoto, J. Zou, Safety-tuned llamas: Lessons from improving the safety of large language models that follow instructions, 2023, arXiv preprint arXiv:2309.07875.
[303] K. Shao, J. Yang, Y. Ai, H. Liu, Y. Zhang, BDDR: An effective defense against textual backdoor attacks, Comput. Secur. 110 (2021) 102433.
[304] A. Robey, E. Wong, H. Hassani, G.J. Pappas, Smoothllm: Defending large language models against jailbreaking attacks, 2023, arXiv preprint arXiv:2310.03684.
[305] J. Kirchenbauer, J. Geiping, Y. Wen, M. Shu, K. Saifullah, K. Kong, K. Fernando, A. Saha, M. Goldblum, T. Goldstein, On the reliability of watermarks for large language models, 2023, arXiv preprint arXiv:2306.04634.
[306] N. Jain, A. Schwarzschild, Y. Wen, G. Somepalli, J. Kirchenbauer, P.-y. Chiang, M. Goldblum, A. Saha, J. Geiping, T. Goldstein, Baseline defenses for adversarial attacks against aligned language models, 2023, arXiv preprint arXiv:2309.00614.
[307] L. Xu, L. Berti-Equille, A. Cuesta-Infante, K. Veeramachaneni, In situ augmentation for defending against adversarial attacks on text classifiers, in: International Conference on Neural Information Processing, Springer, 2022, pp. 485–496.
[308] L. Li, D. Song, X. Qiu, Text adversarial purification as defense against adversarial attacks, 2022, arXiv preprint arXiv:2203.14207.
[309] W. Mo, J. Xu, Q. Liu, J. Wang, J. Yan, C. Xiao, M. Chen, Test-time backdoor mitigation for black-box large language models with defensive demonstrations, 2023, arXiv preprint arXiv:2311.09763.
[310] X. Sun, X. Li, Y. Meng, X. Ao, L. Lyu, J. Li, T. Zhang, Defending against backdoor attacks in natural language generation, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37, No. 4, 2023, pp. 5257–5265.
[311] Z. Xi, T. Du, C. Li, R. Pang, S. Ji, J. Chen, F. Ma, T. Wang, Defending pre-trained language models as few-shot learners against backdoor attacks, 2023, arXiv preprint arXiv:2309.13256.
[312] Z. Wang, Z. Liu, X. Zheng, Q. Su, J. Wang, RMLM: A flexible defense framework for proactively mitigating word-level adversarial attacks, in: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2023, pp. 2757–2774.
[313] J. Duan, H. Cheng, S. Wang, C. Wang, A. Zavalny, R. Xu, B. Kailkhura, K. Xu, Shifting attention to relevance: Towards the uncertainty estimation of large language models, 2023, arXiv preprint arXiv:2307.01379.
[314] F. Qi, Y. Chen, M. Li, Y. Yao, Z. Liu, M. Sun, Onion: A simple and effective defense against textual backdoor attacks, 2020, arXiv preprint arXiv:2011.10369.
[315] B. Chen, A. Paliwal, Q. Yan, Jailbreaker in jail: Moving target defense for large language models, 2023, arXiv preprint arXiv:2310.02417.
[316] A. Helbling, M. Phute, M. Hull, D.H. Chau, Llm self defense: By self examination, llms know they are being tricked, 2023, arXiv preprint arXiv:2308.07308.
[317] M. Xiong, Z. Hu, X. Lu, Y. Li, J. Fu, J. He, B. Hooi, Can LLMs express their uncertainty? An empirical evaluation of confidence elicitation in LLMs, 2023, ArXiv, abs/2306.13063. [Online]. Available: https://api.semanticscholar.org/CorpusID:259224389.
[318] S. Kadavath, T. Conerly, A. Askell, T. Henighan, D. Drain, E. Perez, N. Schiefer, Z. Hatfield-Dodds, N. DasSarma, E. Tran-Johnson, et al., Language models (mostly) know what they know, 2022, arXiv preprint arXiv:2207.05221.
[319] J.C. Farah, B. Spaenlehauer, V. Sharma, M.J. Rodríguez-Triana, S. Ingram, D. Gillet, Impersonating chatbots in a code review exercise to teach software engineering best practices, in: 2022 IEEE Global Engineering Education Conference, EDUCON, IEEE, 2022, pp. 1634–1642.
[320] J. Li, P.H. akon Meland, J.S. Notland, A. Storhaug, J.H. Tysse, Evaluating the impact of ChatGPT on exercises of a software security course, 2023.
[321] W. Tann, Y. Liu, J.H. Sim, C.M. Seah, E.-C. Chang, Using large language models for cybersecurity capture-the-flag challenges and certification questions, 2023.
[322] X. Jin, J. Larson, W. Yang, Z. Lin, Binary code summarization: Benchmarking ChatGPT/GPT-4 and other large language models, 2023.
[323] X. Jin, K. Pei, J.Y. Won, Z. Lin, Symlm: Predicting function names in stripped binaries via context-sensitive execution-aware code embeddings, in: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, 2022, pp. 1631–1645.
[324] E. ThankGod Chinonso, The impact of ChatGPT on privacy and data protection laws, 2023, The Impact of ChatGPT on Privacy and Data Protection Laws (April 16, 2023).

[325] J. Weng, W. Jiasi, M. Li, Y. Zhang, J. Zhang, L. Weiqi, Auditable privacy protection deep learning platform construction method based on block chain incentive mechanism, in: Google Patents, US Patent 11,836,616, 2023.
[326] J. Weng, J. Weng, J. Zhang, M. Li, Y. Zhang, W. Luo, Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive, IEEE Trans. Dependable Secure Comput. 18 (5) (2019) 2438–2455.
[327] Y. Chang, X. Wang, J. Wang, Y. Wu, K. Zhu, H. Chen, L. Yang, X. Yi, C. Wang, Y. Wang, et al., A survey on evaluation of large language models, 2023, arXiv preprint arXiv:2307.03109.
[328] J. Wu, S. Yang, R. Zhan, Y. Yuan, D.F. Wong, L.S. Chao, A survey on LLM-gernerated text detection: Necessity, methods, and future directions, 2023, arXiv preprint arXiv:2310.14724.
[329] M.U. Hadi, R. Qureshi, A. Shah, M. Irfan, A. Zafar, M. Shaikh, N. Akhtar, J. Wu, S. Mirjalili, A survey on large language models: Applications, challenges, limitations, and practical usage, 2023, TechRxiv.
[330] X. Wu, R. Duan, J. Ni, Unveiling security, privacy, and ethical concerns of ChatGPT, 2023.
[331] S.R. Bowman, Eight things to know about large language models, 2023, arXiv preprint arXiv:2304.00612.
[332] W. Zhao, Y. Liu, Y. Wan, Y. Wang, Q. Wu, Z. Deng, J. Du, S. Liu, Y. Xu, P.S. Yu, kNN-ICL: Compositional task-oriented parsing generalization with nearest neighbor in-context learning, 2023.
[333] A. Fan, B. Gokkaya, M. Harman, M. Lyubarskiy, S. Sengupta, S. Yoo, J.M. Zhang, Large language models for software engineering: Survey and open problems, 2023.
[334] X. Hou, Y. Zhao, Y. Liu, Z. Yang, K. Wang, L. Li, X. Luo, D. Lo, J. Grundy, H. Wang, Large language models for software engineering: A systematic literature review, 2023, arXiv preprint arXiv:2308.10620.
[335] J. Clusmann, F.R. Kolbinger, H.S. Muti, Z.I. Carrero, J.-N. Eckardt, N.G. Laleh, C.M.L. Löffler, S.-C. Schwarzkopf, M. Unger, G.P. Veldhuizen, et al., The future landscape of large language models in medicine, Commun. Med. 3 (1) (2023) 141.
[336] P. Caven, A more insecure ecosystem? Chatgpt's influence on cybersecurity, 2023, ChatGPT's Influence on Cybersecurity (April 30, 2023).
[337] M. Al-Hawawreh, A. Aljuhani, Y. Jararweh, Chatgpt for cybersecurity: practical applications, challenges, and future directions, Cluster Comput. 26 (6) (2023) 3421–3436.
[338] J. Marshall, What effects do large language models have on cybersecurity, 2023.
[339] P. Dhoni, R. Kumar, Synergizing generative AI and cybersecurity: Roles of generative AI entities, companies, agencies, and government in enhancing cybersecurity, 2023, TechRxiv.
[340] M. Gupta, C. Akiri, K. Aryal, E. Parker, L. Praharaj, From ChatGPT to ThreatGPT: Impact of generative AI in cybersecurity and privacy, IEEE Access (2023).
[341] E. Shayegani, M.A.A. Mamun, Y. Fu, P. Zaree, Y. Dong, N. Abu-Ghazaleh, Survey of vulnerabilities in large language models revealed by adversarial attacks, 2023.
[342] B. Dash, P. Sharma, Are ChatGPT and deepfake algorithms endangering the cybersecurity industry? A review, Int. J. Eng. Appl. Sci. 10 (1) (2023).
[343] E. Derner, K. Batistič, Beyond the safeguards: Exploring the security risks of ChatGPT, 2023.
[344] K. Renaud, M. Warkentin, G. Westerman, From ChatGPT to HackGPT: Meeting the Cybersecurity Threat of Generative AI, MIT Sloan Management Review, 2023.
[345] L. Schwinn, D. Dobre, S. Günnemann, G. Gidel, Adversarial attacks and defenses in large language models: Old and new threats, 2023.
[346] G. Sebastian, Do ChatGPT and other AI chatbots pose a cybersecurity risk?: An exploratory study, Int. J. Secur. Privacy Pervasive Comput. (IJSPPC) 15 (1) (2023) 1–11.
[347] M. Alawida, B.A. Shawar, O.I. Abiodun, A. Mehmood, A.E. Omolara, et al., Unveiling the dark side of ChatGPT: Exploring cyberattacks and enhancing user awareness, 2023, Preprints.
[348] A. Qammar, H. Wang, J. Ding, A. Naouri, M. Daneshmand, H. Ning, Chatbots to ChatGPT in a cybersecurity space: Evolution, vulnerabilities, attacks, challenges, and future recommendations, 2023.
[349] M. Mozes, X. He, B. Kleinberg, L.D. Griffin, Use of LLMs for illicit purposes: Threats, prevention measures, and vulnerabilities, 2023.
[350] C. Dwork, Differential privacy, in: International Colloquium on Automata, Languages, and Programming, Springer, 2006, pp. 1–12.
[351] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, Y. Gao, A survey on federated learning, Knowl.-Based Syst. 216 (2021) 106775.
[352] A. Pfitzmann, M. Hansen, A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management, 2010, Dresden, Germany.
[353] V. Smith, A.S. Shamsabadi, C. Ashurst, A. Weller, Identifying and mitigating privacy risks stemming from language models: A survey, 2023.