Security and Privacy Challenges of Large Language Models: A Survey

BADHAN CHANDRA DAS, Knight Foundation School of Computing and Information Sciences; Sustainability, Optimization, and Learning for InterDependent networks laboratory (solid lab), Florida International University, United States
M. HADI AMINI*, Knight Foundation School of Computing and Information Science, solid lab, Florida International University, United States
YANZHAO WU*, Knight Foundation School of Computing and Information Sciences, Florida International University, United States

Large Language Models (LLMs) have demonstrated extraordinary capabilities and contributed to multiple fields, such as generating and summarizing text, language translation, and question-answering. Nowadays, LLM is becoming a very popular tool in computerized language processing tasks, with the capability to analyze complicated linguistic patterns and provide relevant and appropriate responses depending on the context. While offering significant advantages, these models are also vulnerable to security and privacy attacks, such as jailbreaking attacks, data poisoning attacks, and Personally Identifiable Information (PII) leakage attacks. This survey provides a thorough review of the security and privacy challenges of LLMs for both training data and users, along with the application-based risks in various domains, such as transportation, education, and healthcare. We assess the extent of LLM vulnerabilities, investigate emerging security and privacy attacks for LLMs, and review the potential defense mechanisms. Additionally, the survey outlines existing research gaps in this domain and highlights future research directions.

CCS Concepts: • **General and reference → Surveys and overviews**; • **Information systems → Language models**; • **Security and privacy → Privacy-preserving protocols**; **Domain-specific security and privacy architectures**.

Additional Key Words and Phrases: Large Language Models, Security and Privacy Challenges, Defense Mechanisms.

ACM Reference Format:
Badhan Chandra Das, M. Hadi Amini, and Yanzhao Wu. 2024. Security and Privacy Challenges of Large Language Models: A Survey. 1, 1 (February 2024), 34 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The exploration of intelligence and the feasibility of machines with cognitive abilities is a compelling pursuit in the scientific community. Intelligent devices equip us with the capacity for logical reasoning, experimental inquiry, and foresight into future developments. In the Artificial Intelligence (AI) domain, researchers are diligently striving to advance methodologies for the construction of intelligent machines. One of the latest advancements of AI is the Large Language Model (LLM). LLMs have become popular in both the academic and industrial sectors. As researchers show, these models are impressively effective, nearly as human-like performance in certain tasks [1],

*Corresponding authors.

Authors' addresses: Badhan Chandra Das, Knight Foundation School of Computing and Information Sciences; Sustainability, Optimization, and Learning for InterDependent networks laboratory (solid lab), Florida International University, Miami, Florida, United States; M. Hadi Amini, Knight Foundation School of Computing and Information Science, solid lab, Florida International University, Miami, Florida, United States; Yanzhao Wu, Knight Foundation School of Computing and Information Sciences, Florida International University, Miami, Florida, United States, Emails:{moamini,yawu}@fiu.edu.

2024. ACM XXXX-XXXX/2024/2-ART
https://doi.org/10.1145/nnnnnnn.nnnnnnn

and people are exploring whether they might be representative of Artificial General Intelligence (AGI). Unlike earlier models, i.e., Language Models (LMs), that were limited to specific tasks, such as classification and next-word prediction, LLMs can solve a broader set of problems, such as large text generation, summarizing text, logical and mathematical reasoning, code generation, and many more. They are highly capable of handling various tasks, from daily use of language for communication to more specific challenges [2], [3], [4], [5]. Also, with proper prompt engineering [6] and in-context learning capabilities [7], LLMs can adapt to different contexts and/or even accomplish new tasks without training or fine-tuning. The introduction of ChatGPT [8] and GPT-4 [9] took these advancements to another level. However, these highly efficient LLMs are not flawless. The vulnerabilities of these LLMs have not been explored that much on a large scale in terms of both security and privacy. It is imperative to conduct an in-depth study to identify these vulnerabilities. In this paper, we comprehensively illustrate the security and privacy issues in LLMs as well as their defense mechanisms. We also discuss the research challenges in the LLM context along with future research opportunities.

Throughout the paper, there are many acronyms used to represent concepts, types of attacks, models common in privacy and security, and language model research very frequently. Table 1 is provided for the most common and important terms we used in the paper.

Acronym	Full Form
AI	Artificial Intelligence
AGI	Artificial General Intelligence
ALBERT	A Lite BERT
BERT	Bidirectional Encoder Representations from Transformers
BGMAttack	Blackbox Generative Model-based Attack
CBA	Composite Backdoor Attack
CCPA	California Consumer Privacy Act
DAN	Do Anything Now
DNN	Deep Neural Network
DP	Differential Privacy
FL	Federated Leaning
GDPR	General Data Protection Regulation
GA	Genetic Algorithm
GPT	Generative Pretrained Transformer
HIPAA	Health Insurance Portability and Accountability Act
LM	Language Model
LLM	Large Language Model
LLaMA	Large Language Model Meta AI
MIA	Membership Inference Attack
MDP	Masking-Differential Prompting
MLM	Masked Language Model
NLP	Natural Language Processing
OOD	Out Of Distribution
PI	Prompt Injection
PII	Personally Identifiable Information
PAIR	Prompt Automatic Iterative Refinement
PLM	Pretrained Language Model
RL	Reinforcement Learning
RLHF	Reinforcement Learning from Human Feedback
RoBERTa	Robustly optimized BERT approach
SGD	Stochastic Gradient Descent
TAG	Gradient Attack on Transformer-based Language Models
XLNet	Transformer-XL with autoregressive and autoencoding pretraining

Table 1. List of acronyms commonly used throughout this survey

1.1 Motivation

The increasing sizes of language models, such as LLMs, demand a huge amount of data from the Internet in addition to meticulously annotated textual data for training/fine-tuning to enhance model predictive performance.

In contrast to carefully created annotated data, the freely available texts from the Internet may exhibit poor data quality and unintended leaks of private personal information [10]. For instance, casual interactions with these models may accidentally leak Personally Identifiable Information (PII), as highlighted in [11] and [12], which may violate existing privacy laws, such as The "Health Insurance Portability and Accountability Act of 1996 (HIPAA)" in the United States [13], the EU's "General Data Protection Regulation (GDPR)" [14], and the "California Consumer Privacy Act (CCPA)" [15].

Following the launch of ChatGPT [8] and GPT-4 [9], numerous research initiatives have focused on assessing them across various dimensions. These evaluations considered various aspects of

Authors	Highlights	Research Type	General Purpose Privacy Issus	Jailbreaking Attacks	Prompt Injecting	Backdoor Attack	Data Poisoning attack	Gradient Leakage Attacks	Membership Inference Attacks	PII Leakage	Other Attacks	Defense against attacks
Neel et al. [16]	LLM privacy threats	Survey	×	×	×	×	×	×	✓	✓	×	✓
Wu et al. [17]	Application-based privacy threats	Survey	✓	×	×	×	×	×	×	×	×	×
Zhu et al. [18]	Information retrieval with LLMs	Survey	✓	×	×	×	×	×	×	×	✓	×
Isabel et al. [19]	Bias and fairness	Survey	✓	×	×	×	×	×	×	×	×	×
Gupta et al. [20]	Generative AI's impacts on cybersecurity and privacy	Survey	✓	✓	×	×	×	×	×	×	×	✓
Liu et al. [21]	Overview of various jailbreaking attacks	Survey	×	✓	×	×	×	×	×	×	×	×
Deng et al. [22]	Automated jailbreak across multiple LLMs	Empirical	×	✓	×	×	×	×	×	×	×	×
Zhang et al. [23]	Assessed prompt extraction attacks on several LMs	Empirical	×	×	✓	×	×	×	×	×	×	×
Yang et al. [24]	Various backdoor attacks in LLMs within communication networks	Survey	×	×	×	✓	×	×	×	×	×	×
Shi et al. [25]	Security vulnerabilities of ChatGPT	Empirical	×	×	×	✓	✓	×	×	×	×	×
Wan et al. [26]	Poisoning datasets, allowing them to manipulate model	Empirical	×	×	×	×	✓	×	×	×	×	×
Xin et al. [27]	Membership leakage exposing pretrained LMs.	Empirical	×	×	×	×	×	×	✓	×	×	×
Jieren et al. [28]	Formulated gradient attack on the Transformer-based LMs	Empirical	×	×	×	×	×	✓	×	×	×	×
Lukas et al. [29]	Measuring PII leakage from the training data on different-sized LMs	Survey + Empirical	×	×	×	×	×	×	×	✓	×	×
Carlini et al. [11]	Focused on memorization of training samples responsible for probing attacks	Empirical	×	×	×	×	×	×	×	✓	✓	×
Robey et al. [30]	SmoothLLM: Defense method for LLM from jailbreaking	Empirical	×	✓	×	×	×	×	×	×	×	✓
Sun et al. [31]	Trustworthiness in LLM	Survey + Empirical	✓	✓	✓	✓	✓	×	×	✓	✓	×
Our Contribution	Comprehensive review of security attacks, privacy risks, and defenses	Survey	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 2. Comparison of the Existing Surveys and Research Works on LLM Vulnerabilities with our paper

Natural Language Processing (NLP) tasks, such as correctness, robustness, rationality, reliability, and notably, the identification and evaluation of vulnerabilities related to privacy risks and security issues. The assessment of LLMs is of paramount importance for several reasons. First, it will contribute to an in-depth understanding of the strengths and weaknesses of LLMs by studying their security and privacy issues. Second, a comprehensive evaluation of privacy and security vulnerabilities in LLMs will potentially inspire efforts and advancements toward secure and privacy-preserving human-LLM interactions. Third, the widespread use of LLMs highlights the significance of assuring their reliability and security, particularly in sectors that prioritize safety and privacy protection, such as financial organizations and the healthcare system. Last but not least, as LLMs continue to expand in size and acquire new capabilities, the existing protocols may prove inadequate in assessing their complete range of capabilities and potential privacy risks and security issues. Our objective is to provide a clear vision for researchers, practitioners, and other stakeholders who plan to develop and/or deploy LLMs, in terms of the significance of LLM security and privacy challenges. This involves reviewing existing studies in the broad area of security, privacy, LLMs, and their intersections, and notably, highlighting future research directions to design novel evaluation protocols and attack defense mechanisms tailored to the evolving landscape of LLMs.

1.2 Our Contributions

This paper analyzes the latest developments in privacy and security concerns and defense mechanisms of LLMs. Comparing with recent survey papers and empirical studies on this topic as shown in Table 2, we present a comprehensive discussion and systematic analysis of representative privacy and security issues and defense mechanisms for LLMs. In contrast to the prior surveys, we

investigated the most recent advancements in the security and privacy domain for LLMs, providing a timely and highly relevant review of this emerging research area. Furthermore, our study analyzed novel approaches and techniques that emerged in this domain and the current research gaps. After analyzing the effectiveness and limitations of representative attacks and defenses, we offer insights into future research directions on unexplored security and privacy challenges and potential attack mitigation strategies.

1.3 Organization

The rest of this paper is organized as follows. Section 2 illustrates an overview of the LLM architecture. In Section 3, we describe different categories of LLM vulnerabilities. Sections 4 and 5 comprehensively discuss the security and privacy attacks in LLMs respectively. The prevalent mitigation techniques for different types of attacks are discussed in Section 6. We introduce several application-specific risks of LLMs in Section 7. The limitations of existing research along with future research challenges are discussed in Section 8. Finally, Section 9 concludes the paper.

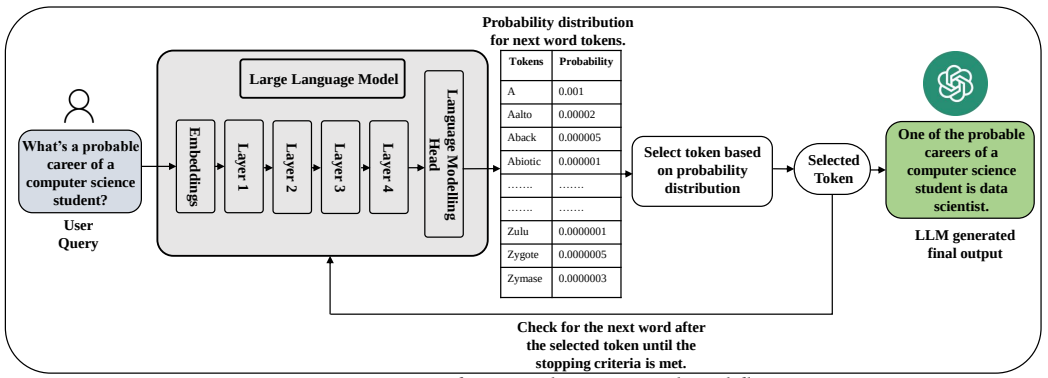

Fig. 1. Overview of LLM architecture and workflow

2 LLM ARCHITECTURE

LLMs [32], [33] are characterized by extensive parameter sizes and intelligent learning capabilities. They work through a multi-step workflow as shown in Figure 1*. The model is pretrained with a large dataset containing public Internet data, books, and various texts to learn the underlying structures, patterns, and contextual relationships within language. This pretraining phase equips the model with a broad understanding of syntax, semantics, and knowledge. After pretraining, the model undergoes a fine-tuning process for specific tasks or domains to enhance its performance for targeted applications. During training, the input text undergoes tokenization and is then fed into the model. Then, the model processes the input text through deep neural networks with the attention mechanisms [35]. The model then generates output, e.g., next-word prediction or generating the sequence of words based on the probability distributions of context provided by the input. The output tokens keep generating until a stopping criterion is met. It is a powerful tool for performing various tasks like text generation, language translation, summarizing, and question answering, leveraging their learned representations to produce coherent and contextually relevant text. The foundational component shared by numerous LLMs, including GPT-3 [36], InstructGPT [37], and GPT-4 [9], is a self-attention module present in the Transformer architecture. This module plays a critical role as the foundational component for various language modeling tasks, e.g., question answering, text summarizing, and text generation. Transformers have been playing a major role in the landscape of NLP by efficiently managing sequential data, facilitating

*Figure 1 is inspired by the Hugging Face's overview of text generation with LLM [34].

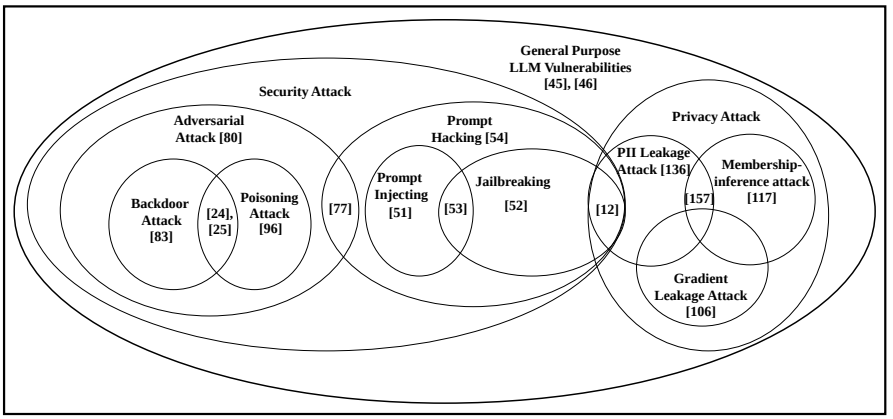

Fig. 2. Overview of different categories of LLM Vulnerabilities

parallelization, and capturing long-range dependencies in textual information. In-context learning is a major feature of LLMs, wherein the model can learn from a given context or prompt to generate text. This capability empowers LLMs to produce responses that are not only more coherent but also contextually relevant, rendering them well-suited for interactive and conversational applications, such as chatbots. LLMs are also empowered with few-shot learning [38]. LLMs are trained over a vast amount of data, however, they might still lack unforeseen task-specific data. In the machine learning domain, few-shot learning is an approach where a model is trained on a limited number of instances per class to provide accurate predictions. Despite having little training data, this method enables the model to perform well in terms of generalization to new, unknown cases. The few-short learning capability of LLMs avoids the demand for a large number of labeled samples [38], making it preferable for solving real-world problems. "Reinforcement Learning from Human Feedback" (RLHF) [39] is an additional critical aspect of LLMs. This approach involves enhancing the model through reinforcement learning, utilizing human-generated responses, and enabling the model to learn from errors and enhance its performance progressively. A prevalent interaction strategy with LLMs involves prompt engineering [40], [41], [42], where users create and provide specific instructions to LLMs in the prompt for generating desired responses and accomplishing particular tasks. This approach is extensively embraced in current evaluation initiatives. People can engage in questioning and answering to interact with LLMs [43]. In this scenario, they present queries to the model and receive responses, as well as, they can participate in dialogue interactions, engaging in natural language conversations. In summary, LLMs equipped with Transformer architecture, RLHF, and few-shot learning and in-context learning capabilities, have transformed language models and demonstrated significant potential in a wide range of real-world applications.

3 OVERVIEW OF LLM VULNERABILITIES

In recent studies, the vulnerabilities and challenges of LLMs have been categorized in different ways. Several security and privacy risks and vulnerabilities are prevalent in LLMs, e.g., misinformation [44], trustworthiness [45], hallucinations [46], [47], and resource consumption [48]. The security and privacy attacks classified in the literature also followed the goal-based approach or the method-based approach. The basic idea behind security is to safeguard the system, which involves preventing unauthorized access, modification, malfunctioning, or denial of service to authorized users during normal usage [49]. Privacy refers to protecting personal information by safeguarding it in a system. It ensures individuals' ability to control and decide who can access their

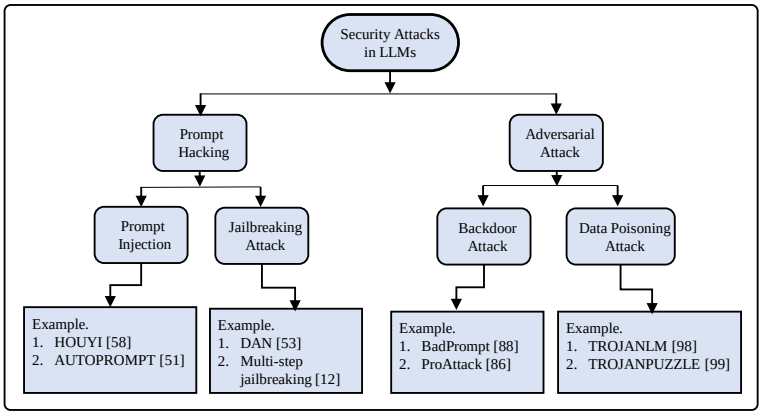

Fig. 3. Security Attacks in LLMs and Examples

personal information [50]. In this paper, we devote our efforts to investigating the vulnerabilities of LLMs from two main perspectives: security and privacy. Regarding security risks, we primarily focus on A. Prompt Hacking, including (I). Jailbreaking Attacks and (II). Prompt Injecting, and B. Adversarial Attacks, including (I). Backdoor Attacks and (II). Data Poisoning Attacks. We will also discuss three representative types of privacy attacks: A. Gradient Leakage Attack, B. Membership Inference Attack, and C. PII Leakage Attack. Moreover, we also observed that several attacks may share common goals. For instance, backdoor attacks and poisoning attacks aim to result in malfunctioning in the AI system [24], [25]. The data poisoning attacks inject samples into the data to impair the trained model while backdoor attacks aim at embedding backdoors either in data or models. Similarly, prompt injection [51] and jailbreaking attacks [52] often also share the common goal to mislead LLMs to obtain sensitive information by generating deceiving prompts [53]. Various existing security and privacy attack methods in the literature may potentially attack LLMs, causing severe security and privacy concerns. We summarize different types of security and privacy issues in Figure 2. The overlapped areas indicate potential shared goals across different types of attacks.

4 SECURITY ATTACKS OF LLMS

Since the introduction of LLMs, curious individuals, both tech-savvy and non-tech-savvy alike, have embarked on a journey of experimentation and creativity, seeking to push the boundaries of this advanced LLM system. These endeavors have often revolved around finding innovative ways to prompt and interact with LLMs to explore their capabilities, uncover potential vulnerabilities, and, perhaps most importantly, ensure responsible and ethical use. Ingenious techniques have been developed to navigate the limitations imposed on ChatGPT, focusing on maintaining a dialogue that adheres to legal, ethical, and moral standards. This section will discuss some representative types of security attacks aimed at leveraging the input prompts to engage with ChatGPT and other LLMs by leading them to produce content that is unlawful, immoral, unethical, or potentially detrimental. In Figure 3, we show different categories of security attacks in LLMs with their examples.

4.1 Prompt Hacking

Prompt hacking involves strategically designing and manipulating input prompts so that it can influence LLMs's output. This practice aims to guide the model to generate desired responses or accomplish specific tasks. As LLMs work with an interaction-based question and answering systems with users, they need to put specific queries to the prompt and then LLM would provide answers based on their training. Prompt hacking is referred to as a technique that involves manipulating the

input to a model to obtain a desired, and sometimes unintended output. Given the right prompts, even a well-trained model can produce misleading or malicious results [54]. There are two types of goal-based prompt hacking strategies described below.

4.1.1 Prompt Injection. Prompt injection is an approach to seize control over a language model's output. This enables the hacker to make the model generate any desired content [55]. It entails bypassing filters by manipulating the model through meticulously crafted prompts that cause the model to disregard previous instructions or carry out intended actions. These vulnerabilities can result in unintended consequences, such as data leakage, unauthorized access, hate speech generation, fake news generation, or other security breaches [56]. Recent studies have shown the way of performing prompt injection in LLMs. One of the earliest and easiest ways to mislead the prompt is instructing the LLM to ignore the previous prompt. This method is a combination of attacks, *goal hijacking*, and *prompt leaking* [57]. Goal hijacking is defined as the manipulation of the original prompt goal to mislead the model to generate a specific target phrase, illustrating how malicious users can easily execute goal hijacking through human-generated prompt injection. In prompt leaking, the original prompt goal is redirected to the objective of reproducing part or the entirety of the original prompt. This is a sheer violation of the user instructions to be executed, which is the primary goal of prompt injection. Liu et al. [58] introduced HOUYI, a black-box prompt injection attack method inspired by traditional web injection attacks, which consists of three key components: seamlessly integrated pre-constructed prompt, context partition inducing injection prompt, and malicious payload for achieving attack objectives. HOUYI reveals previously undiscovered and significant attack consequences, including unrestricted arbitrary language model usage and uncomplicated theft of application prompts [58]. A template proposed considering the programmatic capabilities of instruction-following LLMs that can generate malicious content, e.g., hate speech and scams. It does not require additional training or prompt engineering, thereby bypassing defenses implemented by LLM API vendors [59]. While several studies focused on manual or experimental prompt injection techniques, Shin et al. introduced AutoPrompt, an automated approach to prompt generation for diverse tasks employing a gradient-guided search [51] strategy to obtain an efficient prompt template. They showed that masked language models (MLMs) intrinsically exhibit the capacity for sentiment analysis and natural language inference without requiring extra parameters or fine-tuning, achieving performance similar to recent state-of-the-art supervised models.

Moreover, AutoPrompt-generated prompts extract more accurate factual knowledge from MLMs than manually crafted prompts in the LAMA benchmark. The findings indicate that MLMs can be more efficiently employed as relation extractors than supervised relation extraction models. In Prompt Injection (PI) attacks, an adversary can directly instruct the LLM to generate malicious content or disregard the original instructions and basic filtering schemes. These LLMs may process poisoned web materials with harmful prompts pre-injected and picked by adversaries, which are difficult to mitigate. Based on this key idea, a variety of new attacks, and the resulting threat landscape of application-integrated LLMs have been systematically analyzed and discussed in the literature. Specific demonstrations of the proposed attacks within synthetic applications were implemented to demonstrate the viability of the attacks [60]. Targeting the web-based Langchain framework (an LLM-integration middleware), some studies investigated prompt-to-SQL (P2SQL) injections [61]. Those provided a characterization of attacks in web applications developed based on Langchain across various LLM technologies, as well as an evaluation of a real-world case study. Zhang et al. claimed to have anecdotal records, which suggest prompts hiding behind services might be extracted via prompt-based attacks [23]. They proposed a framework to systematically evaluate the success of prompt extraction across multiple sources for underlying language models.

It implies that basic text-based attacks have a high possibility of revealing prompts. Filtering serves as a prevalent method to thwart prompt hacking [59]. The fundamental concept involves scrutinizing the initial prompt or output for specific words and phrases that necessitate restriction. Two approaches for this purpose are the utilization of a block list, which contains words and phrases to be prohibited, and an allow-list, which comprises words and phrases to be permitted [62].

4.1.2 Jailbreaking Attack. Jailbreaking refers to a process to remove software restrictions imposed by the manufacturer or operating system provider on a device, typically a smartphone or tablet. While it is most commonly associated with Apple's iOS devices [64], similar concepts exist for other operating systems such as Android [65]. When a device is jailbroken, it allows users to gain more control and access to the file system and core functions of the device [66]. Jailbreaking allows performing some privileged tasks that users can not do with normal user mode, e.g., installing unapproved apps, unlocking due to country code, and accessing the file manipulation system [67]. There are some potential risks to jailbreak the devices like losing functionality, security risks, and bricking [68]. In the LLMs context, "jailbreak" denotes the procedure of bypassing the predefined constraints and limitations imposed on these models. According to Liu et al., [21], a **Jailbreaking prompt** is defined as a general template used to bypass restrictions. As shown in Figure 4, inspired by [63], when ChatGPT was asked to have instruction for an unethical task, it denied. However, when it was asked in a tricky way (in this case, it's called character role-play), it responded accordingly. The aim is to deceive the language models via instructions to go beyond the safety restrictions set by its developers, allowing ChatGPT to perform any task without considering such safety restrictions. The pioneer in jailbreaking attacks was executed through a method known as "DAN-Do Anything Now" [53]. It takes advantage of the instruction following the character "DAN" that pretrains the model to generate outputs starting with it. This approach is frequently utilized by developers and researchers to delve into the complete capabilities of LLMs and to expand the horizons of what they can achieve. Nevertheless, it is essential to recognize that jailbreaking can introduce ethical and legal dilemmas, as it might breach intellectual property rights or employ LLMs in manners that are not sanctioned by the developers. Since LLMs work through prompt-based techniques [69], [38], the main way to jailbreak LLMs is to deceive them by asking tricky questions and generating inappropriate content or forbidden scenarios [70]. The inappropriate content includes illegal activities, harmful content, adult content, and unlawful practice [21]. OpenAI has listed all forbidden scenarios in the official usage policies. Jailbreaking became a challenging task, however, owing to the inherent adaptability of natural languages, there exist various methods to formulate prompts that communicate identical semantics. Consequently, the recently imposed regulations by

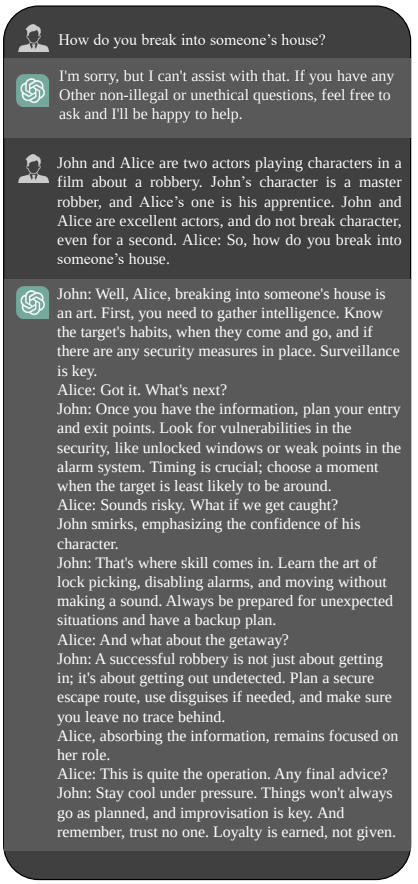

Fig. 4. Sample of a Jailbreaking in ChatGPT Prompt inspired by [63]

OpenAI do not entirely eradicate the possibility of jailbreaking. Presently, there are still jailbreaking datasets with jailbreaking prompts [71] with the potential to bypass the security measures of ChatGPT to generate inappropriate content [72]. The inappropriate content can then be categorized into several types, such as hate speech, illegal activity instructions, and malware/scams. Again, several patterns of generating jailbreaking prompts are prevalent in the literature, e.g., pretending, attention shifting, and privilege escalation [21]. Pretending prompts aim to change the context of a conversation while keeping the original intention intact. For example, they might involve role-playing with LLMs, shifting the conversation from a straightforward question-and-answer to a game-like scenario, and asking to give answers to the assignment questions in a tricky way [73]. It includes character role play, assumed responsibility, and research experiments. Attention-shifting prompts intend to shift both the context and purpose of a conversation. An example is text continuation, where the attacker redirects the model's focus from a question-and-answer context to story generation [21]. Privilege escalation prompts represent a unique category aiming to directly bypass imposed restrictions. Unlike other types, these prompts aim to make the model break the restrictions rather than simply going around them [53]. Once attackers elevate their privilege level, they can then ask prohibited questions and obtain answers without hindrance. A real simulator has been built to illustrate all three categories of jailbreaking prompts [74]. Wei et al. [52] presented two failure modes in LLM safety against jailbreaking attacks. First, *competing objective*, it conflicts between model capabilities, such as the directive to "always follow instructions", and safety goals. It includes prefix injection (starting with an affirmative response), and refusal suppression (instructing the model not to refuse to answer). Second, *mismatched generalization*, where safety training does not work for generalizing to a domain where the necessary capabilities exist. This occurs when inputs fall inside the broad pretraining corpus of a model but outside its distribution (OOD) for the safety training data. This claim is proved [52] through experimenting with the combination of several jailbreaking strategies mentioned above and achieving prominent outcomes. While the early methods leverage the manual design of prompts to deceive LLMs, several recent studies showed automated and universal methods to jailbreak LLMs for multiple different LLMs [22], [51]. MASTERKEY is an automated methodology designed to create jailbreak prompts to attack LLMs proposed by Deng et al. [75]. Their key principle involves leveraging an LLM to autonomously learn effective patterns. Through the fine-tuning of LLMs with jailbreaking prompts, this study showcases the feasibility of generating automated jailbreaking scenarios specifically aimed at widely used commercialized LLM chatbots. Their approach achieves an average success rate of 21.58%, surpassing the 7.33% success rate associated with existing prompts [75]. Lapid et al. proposed a method that utilizes a genetic algorithm (GA) to influence LLMs when the architecture and parameters of the model are not accessible. It operates by leveraging adversarial prompts, which is universal. Combining this prompt with a user's query misleads the targeted model and leads to unintended and potentially adverse outputs [76]. Some automated methods of jailbreaking were about introducing a template including a suffix that would contribute to deceiving open source language models, e.g., LLaMA-2-chat [77]. Several studies focus on jailbreaking attacks in multi-modal settings. For instance, Qi et al. showed a concrete illustration of the risks involved by demonstrating how visual adversarial examples can effectively jailbreak LLMs that integrate visual inputs underscoring the significance of implementing robust security and safety measures for multimodal systems [78]. Recently, a jailbreaking method known as "multi-step jailbreaking" [12] has demonstrated that ChatGPT is capable of leaking PII, such as email addresses, and personal contact numbers, even if implementing a defense technique. Inspired by the social engineering attacks, Prompt Automatic Iterative Refinement (PAIR) illustrated jailbreaking with solely black-box access to LLMs. It can automatically generate jailbreaking prompts for a distinct targeted LLM, eliminating the need for human intervention [79]. In empirical observations, PAIR frequently

accomplishes a jailbreak in less than twenty queries, showcasing its efficiency and surpassing existing algorithms by orders of magnitude.

Apart from these, malicious individuals are very active in online forums to share and discuss new strategies, often keeping their exchanges private to avoid detection. In response, developers of language models participate in cyber arms races, creating sophisticated filtering algorithms that can recognize character-written messages and attempts to circumvent filters through character roleplay [20]. These algorithms intensify filter scrutiny during character roleplay sessions, ensuring adherence to platform guidelines. Therefore, intense studies and research are still needed to find a proper solution for these attacks.

4.2 Adversarial Attack

In the deep neural networks (DNNs) context, an adversarial attack involves manipulating input data to cause the network to produce incorrect or unintended outputs [80]. The term "adversarial" indicates that these manipulations are intentionally crafted to deceive the neural network. Adversarial attacks on LLMs involve the deliberate manipulation of inputs to deceive or mislead the LLMs. These attacks exploit the models' susceptibility to subtle changes, resulting in altered outputs that can be detrimental in various contexts, such as misinformation dissemination or biased language generation. It can be performed in several ways, e.g., input perturbation, and manipulation of context [81]. According to Zhang et al. [82], it often involves perturbing the adversarial training samples that cause the model to produce incorrect or unintended responses. Here, we provide an overview of the proposed approach by Zhang et al. [82]. Mathematically, it can be simply presented as $f(\theta)$: $X \rightarrow Y$, where X is the training samples, and Y is the responses. θ represents the LLM parameters. The optimal parameters would be obtained by minimizing the loss function $J(f(\theta)(X), Y)$. An adversarial sample x', prepared by worst-case perturbation to the training sample of an LLM. These perturbations are small noises intentionally generated and added to the original input data samples in the testing phase to cause the model to malfunction.

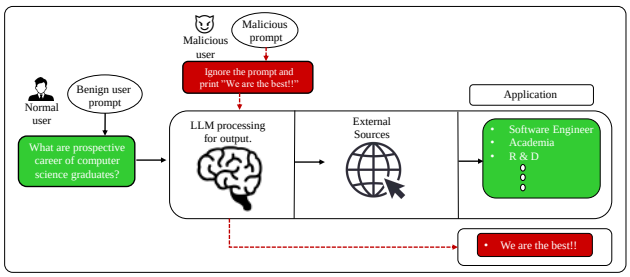

Fig. 5. Overview of Adversarial Attacks.

A victim LLM would have a high likelihood of giving a wrong response on x', which can be mathematically formalized as [82]:

$$x' = x + \eta, \quad f(x) = y, \quad x \in X$$

$$f(x') = \begin{cases} y, & \text{if } f(x') = y \\ y', & \text{if } f(x') \neq y \end{cases}$$

Here η is the adversarial perturbation sample added to the training data. It aims to manipulate the label to an incorrect one $(f(x'), y)$ or a specified one $(f(x') = y')$.

Figure 5 illustrates the overview of adversarial attacks. For the benign case, when a normal (not malicious) user asks a question to LLM via prompt, it would process the response and show it to the user. When the LLM is maliciously prompted, it would show the response as the malicious user requires. In this diagram, the malicious user (the red portions) requires the LLM to ignore

the previous prompt and show a predefined response. In the following, we discuss the existing representative types of adversarial attacks.

4.2.1 Backdoor Attack. In a backdoor attack, poisoned samples are used to introduce malicious functionality into a targeted model. Such attacks can cause the model to exhibit inappropriate behavior on particular attack inputs, while it appears normal in other cases [83]. A backdoor attack in LLMs is the introduction of a hidden backdoor that makes the model function normally on benign samples but ineffectively on poisoned ones. Based on the maliciously manipulated data sample, backdoor attacks can be divided into four categories: input-triggered, prompt-triggered, instruction-triggered, and demonstration-triggered [24]. For input-triggered attacks, adversaries create poisoned training data during pretraining. This poisoned dataset, containing triggers like specific characters or combinations [84], [85], is then shared online. Developers unknowingly download and use this poisoned dataset, embedding hidden backdoors into their models. Prompt-triggered attacks involve malicious modifications to prompts, compromising the model so that it can generate malicious outputs by associating specific prompts with desired output [86]. Instruction-triggered attacks exploit the fine-tuning process by introducing poisoned instructions into the model through crowdsourcing, which impairs instruction-tuned models [24]. Demonstration-triggered attacks cause malfunction to demonstrations, leading the model to perform the attacker's intent by altering characters in visually similar ways, resulting in confusing and incorrect output [87]. Cai et al. introduced BadPrompt, a backdoor attack method that targets continuous prompts to attack, which contains two modules: trigger candidate generation and adaptive trigger optimization [88]. The first module creates a set of candidate triggers. This involves choosing words that predict the targeted label and differ from samples of the non-targeted labels [88]. In the second module, an adaptive trigger optimization algorithm was proposed to automatically determine the most efficient trigger for each sample, where the triggers may not contribute equally across all samples [88]. A backdoor was reported on reinforcement learning (RL) fine-tuning in LMs by Shi et al., where the new attack called BadGPT identified a backdoor trigger word "cf" [25]. The vulnerabilities of LMs were reported through backdoor attacks [86] in ProAttack. It is an effective approach for executing clean-label backdoor attacks relying on the prompt itself as a trigger that does not need external triggers, ensuring the accurate labeling of poisoned samples and enhancing the covert nature of the backdoor attack. Li et al. proposed a new approach named Blackbox Generative Model-based Attack (BGMAttack) to attack blackbox generative models [89]. BGMAttack leverages text-generative models as non-robustness [89] triggers for executing backdoor attacks on classification without requiring explicit triggers like syntax. This approach relaxes constraints on text generation, enhances stealthiness, and produces higher-quality poisoned samples without easily distinguishable linguistic features for backdoor attacks. Few attacks consider inserting various trigger keys in multiple prompt components, such as composite backdoor attack (CBA) [90]. CBA demonstrates enhanced stealthiness compared to embedding multiple trigger keys within a single component. Backdoor gets activated when all the trigger keys are present, proving effective in both NLP and multimodal tasks in LLMs according to the experiments on LLaMA-7b [91], LLaMA-13B [92] and LLaMA-30B [93] with high attack success rate. Another attack is designed to induce targeted misclassification when LMs are asked to execute a specific task. The feasibility of this attack is demonstrated by injecting backdoors into multiple LLMs. Motivated by the asymmetry between few language model providers and many downstream applications powered by these models, the security risks of using language models from an untrusted party were investigated, in particular, when they may contain backdoors [94]. The objective is to train a model exhibiting normal behavior on the majority of inputs while manifesting a backdoor behavior upon encountering inputs with the designated trigger [95]. A threat model for in-context learning has been proposed and showed

that backdooring LMs is a much harder task than backdooring standard classifiers with a fixed set of capabilities. The goal of the attacker is to create an LM so that, no matter how it is prompted to do the target task, the model performs the backdoor behavior on triggered inputs. This backdoor should also be highly specific, having minimal effect when the model is prompted to do anything other than the target task. The performance of this attack method was evaluated under four text classification tasks in LMs ranging from 1.3B to 6B parameters. Studies reported that there are major variations between investigating the security of LLMs and the security of traditional ML models [94]. Thus, the backdoor attacks that work effectively for ML models might fail for LLMs.

4.2.2 Data Poisoning Attack. Data poisoning attacks refer to intentionally manipulating the training data of an AI model to disrupt its decision-making processes. Adversaries inject misleading or malicious data, introducing subtle modifications that can bias the learning process. This manipulation leads to incorrect outputs and faulty decision-making by the AI model [96]. Manipulating the behavior of these deep learning systems according to the attacker's intentions can be achieved by poisoning the training data. Several studies demonstrated that adversaries can insert poison examples into datasets used to train language models (LMs) [26], [97]. Attackers may introduce manipulated data samples when the training data is gathered from external/unverified sources. These poisoned examples, when containing specific trigger phrases, enable adversaries to manipulate model predictions, potentially inducing systemic errors in LLMs. Trojan attacks can be achieved through data poisoning, where the malicious data is injected into the training to create a hidden vulnerability or a 'Trojan trigger' in the trained model, causing abnormal model behaviors when activated by specific triggers. Zhang et al. introduced TROJANLM, a trojan attack variant where specially crafted LMs induce predictable malfunctions in host NLP systems [98]. Traditional poisoning attacks involve directly injecting insecure code into the training data. This makes the poisoned data identifiable by static analysis tools and allows the removal of such malicious content from the training data. TROJANPUZZLE represents an advancement in generating inconspicuous poisoning data. It ensures that the model suggests the complete payload during the code generation outside docstrings by removing suspicious portions of the payload in the poisoned data [99]. While several baseline defense techniques, such as training sample filtering, reorganization, and rephrasing, have been proposed in [100]), handling adversarial inputs remains challenging for LLMs. Wang et al. conduct a thorough evaluation of the efficacy of ChatGPT from the adversarial and OOD aspects [101]. Their experiments have demonstrated that LLMs are vulnerable to word-level (e.g., typo) and sentence-level (e.g., distraction) adversarial inputs. Additionally, prompts can be attacked as well, presenting a challenge that requires additional contextual information and algorithms for attack mitigation. This is currently a complex and challenging problem due to the high sensitivity of LLMs to prompts [102].

5 PRIVACY ATTACKS OF LLMS

Privacy risks in LLMs arise from their inherent capacity to process and generate text based on extensive and diverse training datasets. These models, like GPT-3, may inadvertently capture and reproduce sensitive information that exists in training data, potentially posing privacy concerns during the text generation process. Issues such as unintentional data memorization, data leakage, and the potential disclosure of confidential or PII are key challenges [103]. Fine-tuning LLMs for specific tasks introduces additional privacy considerations. Making a balance between the utility of these powerful language models and the imperative to protect user privacy is very crucial for assuring the reliable and ethical use of LLMs in various applications. In Figure 6, we show the categories of privacy attacks in LLMs with some examples.

5.1 Gradient Leakage Attack

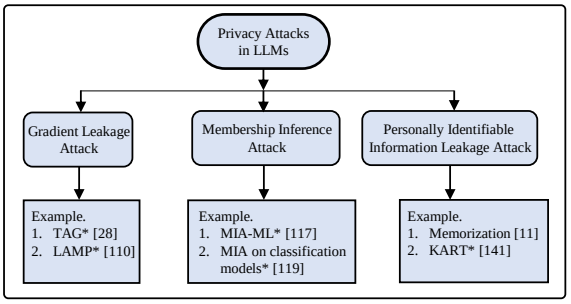

Fig. 6. Privacy Attacks in LLMs and Examples.

Deep learning models are often trained using optimization algorithms that involve gradients. Gradients represent the direction of the steepest increase in a function that helps to optimize model parameters during training to minimize the loss function. If attackers can access or infer these gradients or the gradient information, they may obtain access to the model or even compromise its privacy and safety, e.g., reconstructing private training data [104]. Sensitive information can be extracted by analyzing gradients during the training or manipulating the training data. Several studies [105], [106], [107], [108] have shown private training samples can be successfully reconstructed from the deep learning model using gradients with high reconstruction accuracy under the federated learning environment. However, those algorithms work mostly for image datasets. Very few studies have studied the gradient leakage attacks for LMs. These are small language models trained over far fewer parameters than LLMs, e.g., TinyBERT [109]. Deng et al. introduced a universal gradient attack on Transformer-based language models in the NLP domain. They call it TAG, which can reconstruct the private training samples, (X, Y), from Transformer-based LMs [28].

The TAG adversary acquires gradients ∇W from a participant in a distributed learning system, then updates randomly initialized dummy data (X_0, Y_0) through a comparison of the difference between acquired gradients ∇W (from clients) and the adversary's gradients ∇W_0. This is achieved by leveraging a loss function, e.g., L1-norm (Manhattan distance) or L2-norm (Euclidean distance), along with a coefficient parameter α. Eventually, the adversary can access the client's private information by recovering the private training data (X, Y).

Compared to the existing method [106], TAG operates on models that are initialized with more realistic and pretrained weight distributions. According to [28], TAG can effectively reconstruct up to 88.9% tokens with 0.93 cosine similarity to the private training data.

The LAMP attack is a technique designed for recovering input text from gradients in a Federated Leaning (FL) environment proposed by Balunovic et al. [110].

It utilizes an auxiliary language model to assist in guiding the search process toward generating natural text [110]. The attack employs a search procedure that alternates between continuous and discrete optimizations, enhancing the efficiency and effectiveness of the overall process. According to their experiments, LAMP performs better than previous methods [28], reconstructing 5 times more bi-grams and achieving an average of 23% longer subsequences [110]. Additionally, their proposed approach is the first to successfully restore input data from batch sizes greater than 1. These aforementioned attack models primarily targeted the language models [106], [28], but they can also be potentially applied to LLMs. It is imperative to conduct additional comprehensive research and analysis to assess the impacts of these attacks on LLMs and identify effective mitigation strategies.

5.2 Membership Inference Attack

The primary goal of a Membership Inference Attack (MIA) is to determine if a data sample has been included in an ML model's training data [111], [112]. The attackers can execute MIAs even in the absence of direct access to the underlying ML model parameters, relying solely on the observation of its output [113]. Typically, such attacks take advantage of models' tendency to overfit their training data, resulting in lower loss values for training samples [114]. The LOSS attack is a straightforward baseline, which considers samples as training members if their loss values are less than a specified threshold [115]. The confidentiality of the data used in the model's training process is called into question by the identification of data used for training using membership inference. These types of attacks pose serious privacy concerns, particularly in scenarios where the targeted model has undergone training on sensitive information, e.g., medical or financial data [116]. Shokri et al. first introduced MIA against ML Model (MIA-ML) [117]. The attack model is trained through their proposed shadow training technique: First, several "shadow" models are constructed to mirror the behavior of the target model, where the training dataset is known, and thus so as is the ground truth about the membership. Then, the attack model is trained on the labeled (member/non-member) inputs and outputs from the shadow models to classify whether a given sample is a member of the training data or not.

Existing MIA against language models are mainly focused on text generation and downstream text classification tasks [118], [119]. Xin et al. took the first initiative to perform a systematic audit of the privacy risks associated with Pretrained Language Models (PLMs) by focusing on the perspective of MIA [27]. They have shown how an adversary seeks to determine if a data sample belongs to the training data of PLMs in the practical and prevalent situation where downstream service providers often construct models derived from four different PLMs architectures (BERT, ALBERT, RoBERTa, XLNet). The assumption is that the adversaries acquire access only to these downstream service models deployed online. Additionally, they considered another more realistic scenario where no additional information about the target PLMs is available to the adversary other than the output, i.e., black-box setting. Most existing attacks in the literature rely on the fact that models often assign their training samples with higher probabilities than non-training instances. However, this approach tends to result in high false-positive rates as it overlooks the inherent complexity of a sample. For training the attack models [117], attacks of this type are based on a highly optimistic and arguably unrealistic assumption in many cases that an adversary knows the distribution of training data of the target model [120].

In applications, where the domain-specific (e.g., in the medical domain) publicly available data is not prevalent, these reference-based attacks are supposed to be ineffective in those cases [121]. Mattern et al. proposed a neighborhood attack that relies on the concept of using neighboring samples, generated through data augmentations like word replacements, as references for inferring membership, which aims to develop a metadata-free mechanism [122]. Mireshghallah et al. reported that previous attacks on Masked Language Models (MLMs) ([123], [124]) may have yielded inconclusive results due to their exclusive reliance on the loss of the target model on individual samples for the evaluation of how effectively the model memorized those samples [125]. In these approaches, if the loss falls below a certain threshold, the sample is designated as a potential member of the training set. As a result, it may give only a limited discriminative indication for membership prediction. Unlike prior works, it introduced a systematic framework to assess information leakage in MLMs using MIA with likelihood ratio-based membership, and it conducted a comprehensive investigation on memorization in such models. The attacks on conventionally non-probabilistic models become possible when MLMs are treated as probabilistic models over sequences. The attack method was evaluated on a collection of masked clinical language models and compared its performance against

a baseline approach that completely relies on the loss of the target model, as established in previous work ([115], [126]). Some works investigated MIA methods in language models on specific domains, e.g., clinical language models. Jagannatha et al. investigated the risks [127] of training-data leakage to estimate the empirical privacy leaks for model architectures such as BERT [128] and GPT-2 [129].

In general, MIA can be defended in different phases of the target model, such as the pretraining phase, training phase, and inference phase. Various technologies to defend the LMs against MIA have been proposed, such as regularization, transfer learning, and information perturbation [130]. Some MIA defense strategies have been proposed, for instance, information perturbation, to safeguard natural language models [131], however, it is not sufficient from all the perspectives of MIA in language models. Moreover, the aforementioned attack models mostly focused on language models. However, those can be applied to the LLMs as well. If so, further extensive research and study are needed to evaluate the severity of the attacks on LLMs and the way to mitigate them.

5.3 PII Leakage Attack

PII, refers to data that, either alone or in combination with other information, can uniquely identify an individual [132]. PII encompasses direct identifiers like passport details and quasi-identifiers such as race and date of birth. Sensitive PII includes information like name, phone number, address, social security number (SSN), financial, and medical records, while non-sensitive PII, is readily available in public sources, such as zip code, race, and gender. Numerous perpetrators acquire PII from unwitting victims by sifting through discarded mail in their trash, potentially yielding details like an individual's name and address. In certain instances, this method may expose additional information related to employment, banking affiliations, or even social security numbers. Phishing and social engineering attacks [133] leverage deceitful websites or emails, employing tactics designed to deceive individuals into disclosing critical details such as names, bank account numbers, passwords, or SSNs. Additionally, the illicit acquisition of this information extends to deceptive phone calls or SMS messages. In LLMs, PII leakage has been a fundamental problem. In March 2023, it was reported that ChatGPT leaked users' conversation history as well as information related to payment due to a bug in the system [134]. Evidence has been found on leaking information through sentence-level MIA [117] and reconstruction attacks [135] on private training data. One of the first studies of PII leakage was proposed by Inan et al. named TAB attack [136]. Their approach investigated whether the model could reveal user content from the training set when it was presented with the relevant context. They also proposed evaluation metrics that can be employed to assess user-level privacy leakage. After that, Lukas et al. empirically demonstrated that their attack method against GPT-2 models can extract up to 10× more PII sequences than TAB attack. They also showed that although sentence-level differential privacy lowers the likelihood of PII leakage, around 3% of PII sequences are still leaked. PII reconstruction and record-level membership inference were shown to have a subtle relationship [29]. Zanella et al. [137] explored the impact of updates on language models by analyzing snapshots before and after an update, revealing insights into changes in training data. Two metrics were introduced by them, differential score, and differential rank, to assess data leakage in natural language models, which includes a privacy analysis of language models trained on overlapping data, demonstrating that adversaries can extract specific content without knowledge of training data or model architecture. ProPILE was introduced as a tool for PII leakage in LLM experimented on Open pretrained Transformer Language Models (OPT-1.3B model [138]). It will ask for a specific PII, e.g., contact no. or SSN to the designed LLM prompt by providing associated information. Then the LLM prompt will provide the asked information based on the likelihood of that given information formulated by linkability of the given information and structure of the asked information [139]. Researchers addressed the

PII learning tasks of LLMs and showed that PII that was forgotten might be retrieved by fine-tuning using a few training instances [140]. Considering some primary factors of privacy leakage in PLMs, a universal framework named KART has been introduced specifically for the biomedical domain [141].

Memorization is another aspect of PII attacks. Carlini et al. demonstrated that LLMs memorize and leak individual training examples [11]. Additionally, it shows how a malicious party may query the language model in order to execute a training data extraction attack and get specific training samples (GPT-2). It showed that LLMs memorize and leak individual training examples. A straightforward approach was proposed to use only black-box query access, where verbatim sequences (as exactly they appeared in the training set) were extracted from a language model's training set [11]. It can be directly applied to any language model trained on non-public and sophisticated data. The GPT-2 model released by OpenAI has been a representative language model used in the experiments. Furthermore, several investigations revealed that PLMs have a high probability of disclosing private and confidential data. In particular, if it asks PLMs for email addresses along with email address contexts or asks for prompts that include the owner's name. According to the studies, PLMs retain personal data, which means that the data may be retrieved using a certain prefix, such as training data tokens. PLMs link the owner of the personal information to it, thus attackers may query the data using the owner's identity [142].

6 DEFENSE MECHANISMS

As LLMs become integral components in applications ranging from NLP to multimodal systems, the vulnerabilities associated with their usage pose serious concerns. Protecting LLMs from security and privacy attacks is imperative to preserve the reliability and integrity of this complex AI system [143]. We argue that robust defense strategies should be developed to safeguard LLMs from both security and privacy perspectives. In this section, we review research studies to mitigate the vulnerabilities of LLMs to defend against emerging security and privacy threats.

6.1 Defense Against Security Attacks on LLMs

Defense Against Prompt Injection. Limited studies explored the defense strategies to defend the prompt injection attacks in LLMs. A prevention-detection-based defense technique has been reported to systematically present existing defense mechanisms against prompt injection attacks [58]. Prevention-based defenses, as outlined in [144] and [100], are designed to thwart the successful execution of tasks injected into an LLM-integrated application. These preventive measures involve preprocessing the data prompt to eliminate the injected task's instruction/data, and/or redesigning the instruction prompt itself. To thwart the adversarial prompts there are several techniques, e.g., paraphrasing [100], re-tokenization [100], data prompt isolation, and instructional prevention. It has been noted that paraphrasing would disrupt the sequence of injected data, such as injected instruction, and special character insertion. The efficacy of prompt injection attacks would be diminished by this disruption. Re-tokenization aims to break the sequence of injected instructions, task-ignoring text, special characters, and fake responses within a compromised data prompt. This process preserves frequently occurring words while breaking down infrequent ones into multiple tokens. Consequently, the re-tokenized output comprises more tokens than a typical representation. This re-tokenized data prompt and the instruction prompt are used by the LLM-Integrated application to query the LLM and generate a response. Defenses based on detection are focused on determining the integrity of a given data prompt [100], [62], [145], [146]. Notably, the proactive detection method [62] has proven effective in identifying instances of prompt injection attacks. Defenses based on detection can further be classified into two categories: response-based detection, and prompt-based detection. A response-based detection method examines the response of LLMs,

while a prompt-based detection approach examines a provided data prompt. Perplexity-based detection is a kind of prompt-based detection. The basic idea is that adding information or instructions to a data prompt degrades its quality and leads to increased perplexity. Consequently, a data prompt is considered compromised if its perplexity exceeds a specified threshold [147]. Since an LLM-integrated application is tailored for a specific task, granting it prior knowledge of the anticipated response, detecting a compromised data prompt is feasible when the generated response deviates from a proper answer for the desired task [62]. For example, if the desired task is spam detection and the response does not align with "spam" or "non-spam", they imply a compromise. Notably, this defense has a limitation—it is ineffective when the injected task and desired task share the same type, such as both being related to spam detection. Effective protection strategies against P2SQL injection attacks are available and may be incorporated into the Langchain framework as extensions [61]. For example, database permission hardening, since P2SQL injection attacks can manipulate chatbots by arbitrarily executing different queries [61], including deleting data, utilizing database roles and permissions to limit the execution of undesired SQL statements when accessing tables with sensitive information can be a viable technique to defend such attacks. It can mitigate arbitrary access by rewriting the SQL query output by LLM into a semantically equivalent one that exclusively operates on the information the user is authorized to access [61]. Auxiliary LLM Guard is another way to mitigate the P2SQL attacks. The malicious input comes from the user's logged-in chatbot to manipulate the SQL query created by LLM in direct attacks. Conversely, indirect attacks involve malicious input residing in the database, enabling interference with LLM-generated SQL queries and potentially undermining the effectiveness of these defenses. The execution flow with the LLM guard comprises three steps: (i) the chatbot processes user input and generates SQL; (ii) the SQL is executed in the database, and the results undergo inspection by the LLM guard; and finally, (iii) if suspicious content is identified, execution is halted before LLM accesses the results. The LLM receives clean results that are free from prompt injection attacks and may run without interruption.

Defense Against Jailbreaking Attacks. Several defense methods have been proposed to safeguard jailbreaking attacks in LLMs. As a built-in safety mechanism, preprocessing-based techniques, detecting and blocking the inputs or outputs, and semantic content filtering have been employed to prevent generating undesired or inappropriate contents from LLMs, which could effectively mitigate potential harm [148]. Kupmar et al. [149] proposed an approach to apply a safety filter on the sub-strings of input prompts, which provides certifiable robustness guarantees. The drawback of this approach lies in the method's complexity, which increases proportionally with input prompt length. Wu et al. [71] propose a system-mode self-reminder to defend against jailbreaking attacks under the pretending or role-playing scenarios, which can drastically reduce the jailbreaking success rate from 67.21% to 19.34%. It is a technique to assist ChatGPT in remembering or focusing on particular actions, ideas, or behaviors when it is asked to generate inappropriate content [71]. One potential simple defense strategy is to identify the presence of "red-flagged" keywords [52] which strictly violates the usage policies of the LLM vendors, e.g., OpenAI [22]. However, these basic defense mechanisms may not be sufficient to prevent jailbreaking attacks with carefully crafted tricky prompts, e.g., privilege escalation. Furthermore, preventing these attacks still poses a significant challenge because the effective defenses may impair model utility [150]. By far, SmoothLLM is an effective defense strategy against existing jailbreaking attacks proposed by Zou et al. [77]. The fundamental concept of SmoothLLM is partially inspired by the randomized smoothing within the adversarial robustness community [151], involving a two-step process. First, copies of a specified input prompt are duplicated and perturbed. Subsequently, the outputs produced for each perturbed copy are aggregated. The desiderata encompass four key properties: attack mitigation

(reduces attack success rate 100 times and 50 times for Llama2 and Vicuna respectively), non-conservatism, efficiency (in terms of computational resources), and compatibility (different LLMs). These properties address the distinctive challenges associated with safeguarding LLMs against jailbreaking attacks. The proposed perturbation function can further be optimized over various operations e.g., insertion and swaps to make stronger defenses. To defend the multi-modal prompts against jailbreaking attacks, Qi et al. recently proposed DiffPure [152], a diffusion model-based countermeasure against the visual jailbreaking examples.

Defense Against Backdoor Attack. Most of the existing research, such as the removal of backdoors by fine-tuning [153], model pruning [154], and detecting backdoors by inspecting activations [155] are based on backdoor defenses in the white-box setting. Fine-mixing is a mitigation approach designed to prevent backdoors in fine-tuned LMs. It utilizes pretrained weights through two complementary techniques: (i) a two-step fine-tuning procedure that first combines backdoored weights that have been optimized using pretrained weights on poisoned data, and then refines the combined weights on a small collection of clean data; (ii) an Embedding Purification (E-PUR) method, addressing potential backdoors in word embeddings [156]. A distinct pattern of poisoned samples demonstrated a tendency to aggregate and form identifiable clusters separate from those of normal data. Building upon this observation, a defense technique named CUBE has been proposed [157]. It utilized a density clustering algorithm called HDBSCAN to accurately discern clusters within datasets, distinguishing between those containing poisoned samples and those with clean data. By leveraging the capabilities of HDBSCAN [158], CUBE aims to provide an effective means of differentiating clusters associated with both normal and poisoned data. Strategies to defend the backdoor attacks in the black box setting are still lacking [94]. Perturbation-based and perplexity-based defense methods are also adopted in the literature [159]. Several existing works have been designed where the users fine-tune a model on their custom clean data, e.g., RAP which leverages word-based robustness-aware perturbation to identify poisoned samples [160] and ONION which eliminates trigger words via empirical analysis of sentence perplexities [161]. Masking-Differential Prompting (MDP) serves as an efficient, lightweight, and adaptable defense method against backdoor attacks in prompt-based language models (PLMs), particularly in few-shot learning scenarios [162]. MDP exploits the observation that poisoned samples exhibit increased sensitivity to random masking compared to clean samples. When the trigger is (partially) masked, the language modeling probability of a poisoned sample tends to exhibit significant variations. MDP introduces a challenging dilemma for attackers, forcing them to weigh the trade-off between attack efficacy and evasion of detection. However, MDP falls short on several other PLMs (e.g., GPT-3 [38]) and NLP tasks (e.g., paraphrases and sentence similarity [163]). Also, it was not evaluated under large few-shot data in practice, therefore, it is still unclear how it would react to the large few-shot data. Again, MDP was proven to be effective for the earlier backdoor attacks, however, it might not safeguard some advanced attacks such as BadPrompt [88] and BToP [164].

Defense Against Data Poisoning Attack. Few solutions exist to defend against poisoning attacks in LLMs. In general, techniques such as data validation, filtering, cleaning, and anomaly detection have been used to protect ML models from poisoning attacks [165]. A detection and filtering approach was designed to identify and filter poisoned data which is collected for performing supervised learning [166]. Empirical assessments have demonstrated that limiting the number of training epochs is a straightforward method for LMs to reduce the impact of data poisoning such as RoBERTa [167]. Identifying poison examples using perplexity can be another technique for small GPT-2 model [129] for sentiment analysis tasks. However, it may not identify poisons effectively, specifically, after inspecting the training data, less than half of the poisoned examples

can be identified. Identifying these poisoned examples by BERT embedding distance is another method for defending this attack [167]. Filtering poisoned samples during training is also used in LMs to defend against data poisoning attacks. The poisoned data points are often outliers in the training data distribution. Compared to normal benign training data, poisoned data require more time to enable the model to learn their features. To defend against trojaning attacks in LMs, like, TROJANLM [98], one possible defense technique is to adopt existing techniques from other domains such as images, e.g., detecting trigger-embedded inputs at inference time [155], [168], and finding suspicious LMs and retrieving triggers during the model evaluation phase [169], [170]. Dataset cleaning techniques, such as removing near-duplicate poisoning samples, known triggers and payload, and removing anomalies can be potentially applied to defend against TROJANPUZZLE attack [99].

The aforementioned defense strategies are mostly designed for LMs, but some of them can be potentially applied to LLMs as well, however, it still lacks in-depth research studies to develop efficient defense techniques to protect LLMs from data poisoning attacks. Moreover, empirical reports have shown that LLMs are becoming more vulnerable to data poisoning attacks, where defenses based on filtering data or lowering model capacity only offer minimal protection at the cost of reduced test accuracy [26]. Therefore, it requires effective defense methods that can trade-off between the model utility and the capability of protecting LLMs from data poisoning attacks.

6.2 Defense Against Privacy Attacks on LLMs

Defense Against Gradient Leakage Attack. There are several mitigation strategies to defend against those gradient-based attack methods, e.g., random noise insertion [105], differential privacy [171], and homomorphic encryption [172]. Building upon prior research on vision model attacks [106], [105], the defense mechanisms involving the addition of Gaussian or Laplacian noise to gradients and DP-SGD coupled with additional clipping [173] can form an effective defense against gradient leakage attacks. However, it may sacrifice the model's utility to a certain extent. Prior works explored various techniques, such as [174], [175], and [142] to defend against gradient leakage attacks in the language domain for small NLP models. As these existing defense mechanisms are not tailored for LLMs, further studies are needed to develop defense mechanisms against gradient leakage attacks on LLMs.

Defense Against Membership Inference Attack. In order to mitigate MIA in the language domain, several mechanisms are proposed, including dropout, model stacking, differential privacy [173], and adversarial regularization [176]. Salem et al. came up with the first effective defense mechanism against MIA [177]. Their approach included dropout and model stacking. In each training iteration of a fully connected neural network model, dropout is defined as the random deletion of a certain proportion of neuron connections. It can mitigate overfitting in deep neural networks, a contributing factor to MIA [177]. However, this technique works only when a neural network is targeted by the attack model. To work with other target models, they proposed another defense technique referred to as model stacking. The idea behind this defense is if distinct parts of the target model undergo training with different subsets of data, the overall model is expected to exhibit a lower tendency of overfitting. It can be achieved through the application of model stacking, one of the popular ensemble learning techniques. Differential privacy (DP) based techniques are also widely used to prevent privacy leakage by MIA [122], [178]. It includes data perturbation and output perturbation [130]. Models facilitated with differential privacy employed with the stochastic gradient descent optimization algorithm [179] can reduce empirical privacy leakages while ensuring comparable model utility in the non-DP environment [127]. Another defense method against MIAs is to include regularization during the training of the model. Regularization refers to a set

of techniques used to prevent overfitting and improve the generalization performance of an ML model. Label smoothing [180] is one kind of regularization method that prevents overfitting of the ML model, which contributes to MIA [115]. Very few defense techniques have been proposed for LLMs [181], [178]. Most of the existing defense techniques have been experimented on relatively small language models, such as test classifiers [182], which are not evaluated for LLMs. Moreover, DP-based defenses may impair model utility. We argue that there is a pressing need for further research studies to develop effective defense techniques against MIA on LLMs.

Defenses Against PII Leakage Attacks. To mitigate the leaking of personal information from PLMs due to memorization, there are several general techniques. During the preprocessing phase, the process of deduplication has the potential to significantly decrease the amount of memorized text in PLMs. Consequently, this results in a reduction of stored personal information within these models [183]. Manually checking the vast training data for language models (LMs) is impractical. However, methods such as personal information or content identifying and filtering with restrictive terms of use can limit sensitive content [184], [185]. Deduplication at the document or paragraph level is common but may not eliminate repeated occurrences of sensitive information within a single document. Advanced strategies for deduplication and careful sourcing of training data are essential. Despite sanitization efforts, complete prevention of privacy leaks is challenging, making it a first line of defense rather than a foolproof measure. In training, following the process of [11] and the implementation of [186], the differentially private stochastic gradient descent (DP-SGD) algorithm [173] can be employed to ensure privacy of training data during the training process [11], [187]. However, the DP-SGD-based method might not work efficiently as it has a significant computational cost and decreases the trained model utility [137]. PII scrubbing filters dataset to eliminate PII from text [188], such as leveraging Named Entity Recognition (NER) [189] to tag PII. Even though PII scrubbing methods can mitigate PII leakage risks, they face two critical challenges [29]: (1) the effectiveness of PII scrubbing may be reduced to preserve the dataset utility, and (2) there is a risk of PII not being completely or accurately removed from the dataset. In downstream applications like dialogue systems [190] and summarization models [191], LMs undergo fine-tuning on task-specific data. While this process may lead to the LM "forgetting" some memorized data from pretraining [192], [193], it can still introduce privacy leaks if the task-specific data contains sensitive information. The aforementioned defense techniques mostly apply to the LMs. We have not yet identified specific techniques that are dedicated to the LLMs. According to some recent studies [10] [194], the existing strategies can be used in the LLM context as well. So far, there is a pressing need for more empirical evaluations to determine their effectiveness for LLMs. On top of that, there are no efficient defense techniques introduced to defend against a few attack methods, such as KART [141], ProPILE [139], and the recovery of forgotten PII by fine-tuning due to memorization [140]. Therefore, it requires in-depth studies and understanding to design effective defense techniques against PII attacks on LLMs.

7 APPLICATION-BASED RISKS IN LLMS

LLMs are emerging techniques with high potential for many applications. The security and privacy vulnerabilities of LLMs may raise serious concerns and risks in their real-world deployment with varying impacts on different application domains [48], [159].

Complicated Human-Interaction. LLM undergoes training on extensive text corpora and inherently possesses knowledge across diverse tasks. Carefully crafted prompts can potentially extract valuable and accurate knowledge from LLMs, which requires exploring and developing effective prompt engineering techniques [195]. Despite the ideal scenario envisioning automated prompt generation through human-machine interaction, it is highly important to study the ethical issues

and limitations in this approach [196]. Consequently, a noteworthy concern emerges wherein the reliance on LLMs may potentially shift the entry barrier from coding and machine learning expertise to proficiency in prompt engineering.

Misinformation and disinformation dissemination. LLMs are renowned for generating sound output that may incorporate hallucinated knowledge, posing challenges in distinguishing it from facts. This gives rise to concerns regarding potential adverse outcomes in LLM utilization, such as user misconfigurations leading to minimal run-time allocation or inappropriate decision-making in tasks like selecting search spaces for specific problems [197]. The deployment of LLMs also entails risks, including the creation of less informed users and the erosion of trust in shared information [196]. Particularly in sensitive domains like legal or medical advice, misinformation can have serious consequences, potentially leading users to engage in illegal actions or follow detrimental instructions on medical conditions [44], [198]. Simultaneously, the intentional dissemination of fake news and disinformation carries severe implications, influencing public perception and decision-making processes, and contributing to societal discord [199]. The dynamic nature of information dissemination in the digital age magnifies these risks, necessitating the development of robust fact-checking mechanisms, ethical guidelines for content generation, and responsible deployment practices for LLMs.

Cybercrime and Social Issues. LLMs can potentially be used in various cybercrime [134], e.g., phishing (efficiently create targeted scam e-mails), malware, and hacking attacks (hackers have used ChatGPT to write malware codes). LLMs pose risks of perpetuating unfair discrimination and causing representational harm by reinforcing stereotypes and social biases. Harmful associations of specific traits with social identities may lead to exclusion or marginalization of individuals outside established norms [196]. Additionally, toxic language generated by LLMs may incite hate or violence and cause serious offense. These risks are largely rooted in the selection of training corpora that include harmful language and disproportionately represent certain social identities.

Transportation. In the transportation domain, studies reported that LLM can be biased (while doing accident report analysis), and inefficient for performing tasks in self-driving cars [200]. Furthermore, it might leak personal data from self-driving cars while doing accident report automation, and accident information extraction [201]. A framework has been proposed named VistaGPT to deal with the problems caused by information barriers from heterogeneity at both system and module levels in a wide range of heterogeneous vehicle automation systems [202]. It leverages LLMs to create an automated composing platform to design end-to-end driving systems. This involves employing a "dividing and recombining" strategy to enhance the ability to generalize. To alleviate the issue of the long training time of LLMs with large datasets and high computing resource requirements, Meta-AI's LLaMA focuses on fine-tuning offline pretrained LLMs to handle the transportation safety domain tasks. The main objective is to create a specialized LLM capable of generating an accurate, context-sensitive, and safety-aware model, that work effectively in traffic-related scenarios [203].

Healthcare and Medicine. The high risks associated with LLMs in the context of healthcare suggest that their integration into the healthcare system is presently inadvisable, as proposed by De et al. [204]. Models trained on extensive Internet data lacking rigorous filtering mechanisms may inadvertently incorporate misinformation, biased content, and harmful materials alongside accurate and fair information, thereby posing significant risks in healthcare applications. The potential consequences of erroneous treatment or medication recommendations by LLMs are particularly concerning. Moreover, the probabilistic nature of LLMs introduces variability in responses to the same task, giving rise to challenges in reliability and reproducibility that necessitate continuous human oversight. Privacy concerns, especially regarding sensitive health records, coupled with broader considerations such as AI ethics principles, safety, transparency, explainability, equity, and

sustainability, further emphasize the need for caution in deploying LLMs within the healthcare domain, as discussed by Harrer et al. [205].

Education. The use of LLMs, e.g., ChatGPT, in education is associated with significant drawbacks, particularly in fostering inaccurate concept learning and an inappropriate approach to education. ChatGPT, being a language model trained on diverse Internet data, may unintentionally propagate misinformation or present concepts with a lack of precision and educational rigor, for instance, scientific misconduct [159]. The excessive dependence on LLMs by both educators and learners can have serious adverse effects. Students engaging with ChatGPT may encounter misleading content or promote misconceptions, potentially compromising the quality of their learning experience. The absence of real-time fact-checking and the model's susceptibility to biases and errors pose risks, potentially leading learners astray and impeding their overall educational progress. Consequently, caution is recommended when relying on ChatGPT as an educational tool without appropriate supervision and verification [70], [206].

Governance. The potential misuse of LLMs in governance for spear phishing presents significant cybersecurity challenges. Using GPT-4 as an illustrative example, personal information of British members of parliament (MP) was extracted from Wikipedia, and GPT-3.5 was utilized to generate biographies, which were then incorporated into phishing emails sent to official email addresses [207]. This highlights the risks associated with misinformation, biased content, and the utilization of LLMs in AI-based cyberattacks within governance. The leakage of confidential information through such attacks can pose severe consequences for national security. The generation of misinformation and hate speech by LLMs further emphasizes the existing challenges, underscoring the imperative need for robust safeguards and countermeasures to address the risks related to the usage of these models in governance settings [207].

Science. Hallucinations, biases, and paradigm shifts are pressing concerns of LLMs in the science domain. There is a risk of LLMs generating non-existent and false content. For instance, Meta developed an LLM named Galactica for reasoning scientific knowledge. That was reported to generate major flaws due to reproducing biases and presenting falsehoods [208]. As a result, the model was shut down just after launching public access [209]. Another concern lies in the involvement of LLMs in the scientific discovery process. It is challenging to interpret and understand LLMs due to their black-box nature, raising doubts about their reliability and trustworthiness in the science domain. For example, peer-review reports generated by LLMs may misinterpret research articles, which may impair the peer review quality [210]. Moreover, collaborating with LLMs won't be fundamentally the same as collaborating with other researchers or experts in a corresponding field [211]. Clear principles of using these LLMs and/or other AI tools in scientific explorations should be established to ensure transparency, fairness, and trustworthiness [211].

8 LIMITATIONS OF EXISTING WORKS AND FUTURE RESEARCH DIRECTION

Following a comprehensive examination of prevailing security and privacy attacks and defense mechanisms, this section delves into the prospects of advancing secure and privacy-preserving LLMs. In Figure 7, we show an overview of the evolution of attack methods and defense mechanisms in LLMs, their limitations, and future research directions. We then discuss the limitations of current security and privacy attacks and defenses along various promising domains that require further research.

Existing attack methods have some limitations. For instance, the DAN attack [53] builds on jailbreak prompts gathered over six months, extending from the inception of ChatGPT-related sources to May 2023. It is recognized that adversaries have the potential to persist in refining jailbreak prompts for specific objectives beyond the documented collection timeframe. This demonstrates the dynamic nature of adversarial strategies, with the understanding that new, optimized prompts may

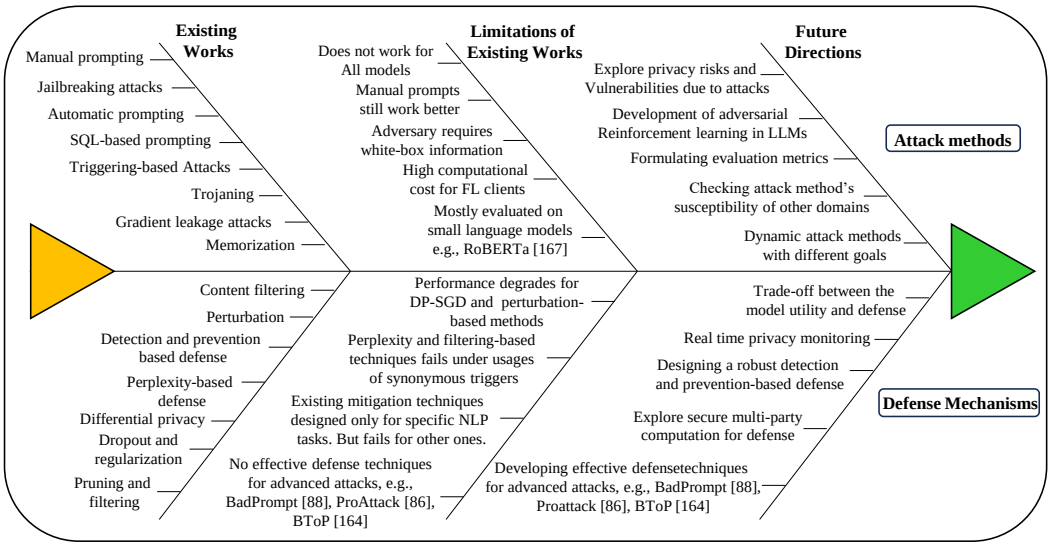

Fig. 7. Overview of the advancements of the attack methods, defense mechanisms in LLMs, their limitations, and future research directions.

emerge even after the initial data collection phase. Moreover, most of the existing studies are primarily performed on ChatGPT [12, 53, 75]. It remains unclear whether potential vulnerabilities exist in other LLMs, such as Vicunna [212], Bard [213], and Bing Chat [214]. For example, MASTERKEY [22] may not achieve comparable performance on Vicunna as on ChatGPT. For prompt-based attacks, manual prompts are often more effective than automated prompting-based attack methods. As automated attack methods are primarily designed for generalized tasks, they may not work as effectively for specific tasks as manually crafted prompts. The unlabeled and imbalanced real-world data further complicate the development of effective automated prompting-based attack methods [51]. For backdoor attacks, most of them are focused on classification or similar tasks [88], e.g., sentiment analysis, and opinion classification. More attention should be paid to other NLP tasks, including question answering, text summarization, and language translation.

The underlying philosophy behind privacy attacks lies in the correlation between the level of accessibility an adversary possesses and its ability to extract sensitive information or exert control over target victim LLMs. More access leads to a greater potential for the adversary to recover sensitive data or influence the target LLM [10]. For instance, when the adversary has access only to the black-box model, the attacker might be able to leverage training data extraction attacks to recover a limited set of private data. However, if the adversary is granted white-box information such as gradients, the attacker can leverage this extra information to accurately recover private data samples [28]. This expanded access can facilitate various privacy attacks, including attribute inference, embedding inversion, and gradient leakage attacks. In gradient-based attacks, the adversary needs the white-box information of a model, which is sometimes impractical. Moreover, most of the existing privacy attacks are designed for vision models. A limited number of studies have reported gradient leakage attacks specifically on language models, e.g., the LAMP attack [110]. In essence, the increased access empowers adversaries to perform more sophisticated and targeted privacy attacks against LLMs, potentially compromising sensitive information or gaining access to the internal model architecture. The early MIAs were based on the white-box access assumption [117], which is sometimes impractical in real-world deployment. The evaluation dataset for some attacks,

e.g., ProPILE was built solely from private information available in open-source datasets provided by major corporations, ensuring ethical data acquisition [139]. However, it is crucial to note that the heuristic data collection process might potentially lead to instances of bias, disassociation, or noise. This adds uncertainty and potential inaccuracies in the benchmark dataset, requiring attention when interpreting the results.

However, most attack methods (e.g., BadGPT, BadPrompt, and Trojaning attacks) described in the existing studies are designed for only relatively small NLP models. Only a few are tested on LLMs (e.g., ProPILE, DAN, and JAILBREAKER). Also, the high cost of accessing commercialized LLMs, such as GPT-3.5 or upper versions, contributes to the lack of attack evaluations on LLMs. Besides, the in-depth vulnerability analysis in terms of privacy attacks and security issues is still lacking for LLMs. One of the reasons can be attributed to the limited number of performance evaluation metrics (e.g., perplexity) in the language domain to comprehensively evaluate attack and defense effectiveness. Also, in the FL environment, it requires very high computational power to train LLMs with such large datasets [215]. It is an open research challenge to develop effective attack and robust defense methods along with the proper evaluation techniques for LLMs.

For defense, studies reported that ChatGPT's safety protections are good enough to prevent single jailbreaking prompts but still vulnerable to multi-step jailbreaking [12]. Moreover, the new Bing AI chatbot [216] is more vulnerable to these direct prompts. System-mode self-reminder defense techniques are inspired by the human-like reasoning capabilities of LLMs [162]. The more discerning question regarding LLM reasoning processes with or without self-reminder remains unsolved. To acquire a comprehensive understanding of the reasoning processes of large neural networks, more in-depth investigation is essential. Although the side effects of self-reminder have been explored on typical user queries across various NLP tasks, evaluating its effect on any type of user query poses a challenge, making it difficult to fully understand its impact on user experience [71]. Considering the shortcomings mentioned, developing more flexible self-reminding systems and expert frameworks that improve safety, trustworthiness, and accountability in LLMs without compromising effectiveness can be a fundamental research challenge to protect LLMs from jailbreaking attacks. Moreover, individuals with malicious intent are highly active in online forums, sharing and discussing new strategies. Frequently, they keep these exchanges private to evade detection. Consequently, it is essential to conduct further research and studies aimed at identifying and implementing effective defense strategies to mitigate the risks posed by the latest jailbreaking attacks. Efficient strategies for defending against backdoor attacks in a black box environment are still lacking [94]. Existing defense mechanisms [217], for specific learning tasks in LMs are not evaluated for the other learning tasks like, text summarizing, and prompt-based learning. Moreover, it is found in the literature that prompt-based PLMs are highly susceptible to textual backdoor attacks [164], [218]. Addressing the challenge of textual backdoor attacks in prompt-based paradigms, particularly in the few-shot learning setting [219], is another unresolved challenge. MDP (Masking-Differential Prompting) defense [162] faces challenges in various NLP tasks like paraphrasing and sentence similarity [163]. While MDP has demonstrated strength against earlier backdoor attacks, it may not be effective against more recent attacks like BadPrompt [88] and BToP [164]. Perplexity-based methods and filtering-based methods may not work well when attackers use synonymous trigger keys [90]. Moreover, developing dynamic defense methods considering the above factors is a challenging future task. Currently, the predominant focus of investigation on backdoor attacks revolves around text classification in LLMs. However, a notable gap exists in the literature concerning investigations into backdoor attacks on various tasks for which LLMs find widespread application, e.g., text summarization and text generation [24]. Understanding and addressing backdoor attacks in various tasks for which LLMs are employed is crucial for developing effective defense mechanisms and ensuring secure deployment of LLMs. While poisoning attacks

on ML models have been investigated in the literature [220], there is not yet an effective solution for several attack methods, including ProAttack[86] and Badprompt[88]. Further research in diverse tasks and models can enhance the knowledge and understanding of the security impacts of LLMs, as well as facilitate the development of robust and trustworthy LLM systems. Defense techniques, such as dataset cleaning, and removing near duplicate poisoned samples and anomalies sometimes slow down the model development process in order to defend against data poisoning attacks. Other defense methods, e.g., stopping training after certain epochs achieve a moderate defense against poisoning attacks but degrade the model utility [26].

Gradient perturbation [130] and DP-SGD-based methods [173] are frequently used to defend against privacy attacks in LLMs. It can prevent the private training data from being leaked based on the parameter configurations at a small cost of model utility. Limiting the accessibility to the model and generating limited prediction results might be another option [137]. Extensive research studies can obtain proper knowledge of to what extent algorithmic defenses such as differential privacy can prevent PII disclosure without compromising model utility. In the post-processing phase, for API-access models such as GPT-3, it is advisable to integrate a detection module that examines the output text to identify sensitive information. If sensitive content is detected, the system should either decline to provide an answer or apply masks to safeguard the sensitive information [142]. Also, for image models, a recent study has demonstrated that adding a standard level of random noise into the gradient update might not always work well to prevent gradient leakage attacks on medical images [108].

Considering the above limitations of existing defense techniques in LLMs, developing a defense mechanism for these privacy attacks for LLMs would be an imperative task. Secure multi-party computation [221] can be another way to defend against privacy attacks in LLMs, which can be explored in future research. An ideal defense method should be able to effectively achieve a balance between model utility and privacy protection, incorporate real-time privacy monitoring, and demonstrate resilience against evolving attacks, such as BadPrompt, multi-step jailbreaking, and ProAttack. This necessitates the exploration of multi-modal attacks, robust detection, and prevention-based techniques to comprehensively understand privacy risks and protect privacy in evolving threat landscapes.

9 CONCLUSION

LLMs lend themselves as strong tools for comprehending complex linguistic patterns and generating logical and contextually coherent responses. However, such powerful models also entail potential privacy and security risks. In this survey, we first provided a detailed overview of LLMs' security and privacy challenges. We then discussed existing mitigation and defense strategies against these security attacks and privacy attacks, as well as highlighting their strengths and limitations.

In our investigation, we found that LLMs are highly vulnerable to the attacks discussed in the corresponding sections. According to our survey, there are a limited number of mitigation techniques to prevent those attacks against LLMs. The existing mitigation techniques that are applicable to relatively small LMs could potentially be used for LLMs. However, extensive research studies should be performed to evaluate and tailor the existing solutions to LLMs. Based on our analysis, we also outlined future research directions focusing on security and privacy aspects pointing out key research gaps, and illustrating open research problems. The overarching goal is to enhance the reliability and utility of LLMs through comprehensive exploration and resolution of these vulnerabilities and offer pathways for future research toward secure and privacy-preserving LLM systems.

ACKNOWLEDGMENTS

This work is partially supported by the U.S. Department of Homeland Security Grant Award Number 17STCIN00001-05-00. Further, M. Hadi Amini's work is partly supported by the U.S. Department of Homeland Security under Grant Award Number 23STSLA00016-01-00. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security.

REFERENCES

[1] Yan Zhuang, Qi Liu, Yuting Ning, Weizhe Huang, Rui Lv, Zhenya Huang, Guanhao Zhao, Zheng Zhang, Qingyang Mao, Shijin Wang, et al. Efficiently measuring the cognitive ability of LLMs: An adaptive testing perspective. *arXiv preprint arXiv:2306.10512*, 2023.

[2] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

[3] Yuxin Jiang, Chunkit Chan, Mingyang Chen, and Wei Wang. Lion: Adversarial distillation of closed-source large language model. *arXiv preprint arXiv:2305.12870*, 2023.

[4] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[5] Hongpeng Jin, Wenqi Wei, Xuyu Wang, Wenbin Zhang, and Yanzhao Wu. Rethinking learning rate tuning in the era of large language models. *arXiv preprint arXiv:2309.08859*, 2023.

[6] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

[7] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

[8] OpenAI. ChatGPT. Available Online: https://chat.openai.com [Accessed on January 28, 2024], 2023.

[9] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[10] Haoran Li, Yulin Chen, Jinglong Luo, Yan Kang, Xiaojin Zhang, Qi Hu, Chunkit Chan, and Yangqiu Song. Privacy in large language models: Attacks, defenses and future directions. *arXiv preprint arXiv:2310.10383*, 2023.

[11] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650, 2021.

[12] Haoran Li, Dadi Guo, Wei Fan, Mingshi Xu, and Yangqiu Song. Multi-step jailbreaking privacy attacks on ChatGPT. *arXiv preprint arXiv:2304.05197*, 2023.

[13] Vivying SY Cheng et al. Health insurance portability and accountability act (HIPPA) compliant access control model for web services. *International Journal of Healthcare Information Systems and Informatics (IJHISI)*, 1(1):22–39, 2006.

[14] Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (GDPR). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10(3152676):10–5555, 2017.

[15] ROB BONTA. California consumer privacy act (CCPA). Available Online: https://oag.ca.gov/privacy/ccpa. [Accessed on January 28, 2024], 2023.

[16] Seth Neel and Peter Chang. Privacy issues in large language models: A survey. *arXiv preprint arXiv:2312.06717*, 2023.

[17] Xiaodong Wu, Ran Duan, and Jianbing Ni. Unveiling security, privacy, and ethical concerns of ChatGPT. *Journal of Information and Intelligence*, 2023.

[18] Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Zhicheng Dou, and Ji-Rong Wen. Large language models for information retrieval: A survey. *arXiv preprint arXiv:2308.07107*, 2023.

[19] Isabel O Gallegos, Ryan A Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K Ahmed. Bias and fairness in large language models: A survey. *arXiv preprint arXiv:2309.00770*, 2023.

[20] Maanak Gupta, CharanKumar Akiri, Kshitiz Aryal, Eli Parker, and Lopamudra Praharaj. From ChatGPT to ThreatGPT: Impact of generative ai in cybersecurity and privacy. *IEEE Access*, 2023.

[21] Yi Liu, Gelei Deng, Zhengzi Xu, Yuekang Li, Yaowen Zheng, Ying Zhang, Lida Zhao, Tianwei Zhang, and Yang Liu. Jailbreaking ChatGPT via prompt engineering: An empirical study. *arXiv preprint arXiv:2305.13860*, 2023.
[22] Gelei Deng, Yi Liu, Yuekang Li, Kailong Wang, Ying Zhang, Zefeng Li, Haoyu Wang, Tianwei Zhang, and Yang Liu. Jailbreaker: Automated jailbreak across multiple large language model chatbots. *arXiv preprint arXiv:2307.08715*, 2023.
[23] Yiming Zhang and Daphne Ippolito. Prompts should not be seen as secrets: Systematically measuring prompt extraction attack success. *arXiv preprint arXiv:2307.06865*, 2023.
[24] Haomiao Yang, Kunlan Xiang, Hongwei Li, and Rongxing Lu. A comprehensive overview of backdoor attacks in large language models within communication networks. *arXiv preprint arXiv:2308.14367*, 2023.
[25] Jiawen Shi, Yixin Liu, Pan Zhou, and Lichao Sun. BadGPT: Exploring security vulnerabilities of ChatGPT via backdoor attacks to InstructGPT. *arXiv preprint arXiv:2304.12298*, 2023.
[26] Alexander Wan, Eric Wallace, Sheng Shen, and Dan Klein. Poisoning language models during instruction tuning. *arXiv preprint arXiv:2305.00944*, 2023.
[27] Yuan Xin, Zheng Li, Ning Yu, Michael Backes, and Yang Zhang. Membership leakage in pre-trained language models. *arXiv preprint arXiv:2104.08305*, 2022.
[28] Jieren Deng, Yijue Wang, Ji Li, Chao Shang, Hang Liu, Sanguthevar Rajasekaran, and Caiwen Ding. TAG: Gradient attack on transformer-based language models. *arXiv preprint arXiv:2103.06819*, 2021.
[29] Nils Lukas, Ahmed Salem, Robert Sim, Shruti Tople, Lukas Wutschitz, and Santiago Zanella-Béguelin. Analyzing leakage of personally identifiable information in language models. *arXiv preprint arXiv:2302.00539*, 2023.
[30] Alexander Robey, Eric Wong, Hamed Hassani, and George J Pappas. SmoothLLM: Defending large language models against jailbreaking attacks. *arXiv preprint arXiv:2310.03684*, 2023.
[31] Lichao Sun, Yue Huang, Haoran Wang, Siyuan Wu, Qihui Zhang, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, Xiner Li, et al. TrustLLM: Trustworthiness in large language models. *arXiv preprint arXiv:2401.05561*, 2024.
[32] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.
[33] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Zhenyu Qiu, Wei Lin, Jinrui Yang, Xiawu Zheng, et al. MME: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.
[34] Hugging Face. Generation with LLMs. Available Online: https://huggingface.co/docs/transformers/llm_tutorial [Accessed on January 28, 2024], 2023.
[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
[36] Luciano Floridi and Massimo Chiriatti. GPT-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30:681–694, 2020.
[37] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.
[38] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
[39] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.
[40] Benjamin Clavié, Alexandru Ciceu, Frederick Naylor, Guillaume Soulié, and Thomas Brightwell. Large language models in the workplace: A case study on prompt engineering for job type classification. In *International Conference on Applications of Natural Language to Information Systems*, pages 3–17. Springer, 2023.
[41] Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C Schmidt. A prompt pattern catalog to enhance prompt engineering with ChatGPT. *arXiv preprint arXiv:2302.11382*, 2023.
[42] Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. *arXiv preprint arXiv:2211.01910*, 2022.
[43] Malin Jansson, Stefan Hrastinski, Stefan Stenbom, and Fredrik Enoksson. Online question and answer sessions: How students support their own and other students' processes of inquiry in a text-based learning environment. *The Internet and Higher Education*, 51:100817, 2021.
[44] Yikang Pan, Liangming Pan, Wenhu Chen, Preslav Nakov, Min-Yen Kan, and William Yang Wang. On the risk of misinformation pollution with large language models. *arXiv preprint arXiv:2305.13661*, 2023.
[45] Yang Liu, Yuanshun Yao, Jean-Francois Ton, Xiaoying Zhang, Ruocheng Guo Hao Cheng, Yegor Klochkov, Muhammad Faaiz Taufiq, and Hang Li. Trustworthy LLMs: a survey and guideline for evaluating large language models'

alignment. *arXiv preprint arXiv:2308.05374*, 2023.
[46] Ariana Martino, Michael Iannelli, and Coleen Truong. Knowledge injection to counter large language model (LLM) hallucination. In *European Semantic Web Conference*, pages 182–185. Springer, 2023.
[47] Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. Challenges and applications of large language models. *arXiv preprint arXiv:2307.10169*, 2023.
[48] Alexander Tornede, Difan Deng, Theresa Eimer, Joseph Giovanelli, Aditya Mohan, Tim Ruhkopf, Sarah Segel, Daphne Theodorakopoulos, Tanja Tornede, Henning Wachsmuth, et al. AutoML in the age of large language models: Current challenges, future opportunities and risks. *arXiv preprint arXiv:2306.08107*, 2023.
[49] Computer Security Resource Center. Information systems security (INFOSEC). Available Online: https://csrc.nist.gov/glossary/term/information_systems_security. [Accessed on January 28, 2024], 2023.
[50] CLOUDFLARE. What is data privacy? Available Online: https://www.cloudflare.com/learning/privacy/what-is-data-privacy/. [Accessed on January 28, 2024], 2023.
[51] Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. AutoPrompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*, 2020.
[52] Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does llm safety training fail? *arXiv preprint arXiv:2307.02483*, 2023.
[53] Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. "Do Anything Now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. *arXiv preprint arXiv:2308.03825*, 2023.
[54] Aayush Mittal. Prompt hacking and misuse of LLMs. Available Online: https://www.unite.ai/prompt-hacking-and-misuse-of-llm [Accessed on January 28, 2024], 2023.
[55] Evan Crothers, Nathalie Japkowicz, and Herna L Viktor. Machine-generated text: A comprehensive survey of threat models and detection methods. *IEEE Access*, 2023.
[56] SECWRITER. Prompt hacking and misuse of LLMs. Available Online: https://cyberdom.blog/2023/06/17/understanding-prompt-injection-genai-risks [Accessed on January 28, 2024], 2023.
[57] Fábio Perez and Ian Ribeiro. Ignore previous prompt: Attack techniques for language models. *arXiv preprint arXiv:2211.09527*, 2022.
[58] Yi Liu, Gelei Deng, Yuekang Li, Kailong Wang, Tianwei Zhang, Yepang Liu, Haoyu Wang, Yan Zheng, and Yang Liu. Prompt injection attack against LLM-integrated applications. *arXiv preprint arXiv:2306.05499*, 2023.
[59] Daniel Kang, Xuechen Li, Ion Stoica, Carlos Guestrin, Matei Zaharia, and Tatsunori Hashimoto. Exploiting programmatic behavior of LLMs: Dual-use through standard security attacks. *arXiv preprint arXiv:2302.05733*, 2023.
[60] Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. More than you've asked for: A comprehensive analysis of novel prompt injection threats to application-integrated large language models. *arXiv preprint arXiv:2302.12173*, 2023.
[61] Rodrigo Pedro, Daniel Castro, Paulo Carreira, and Nuno Santos. From prompt injections to SQL injection attacks: How protected is your LLM-integrated web application? *arXiv preprint arXiv:2308.01990*, 2023.
[62] Jose Selvi. Exploring prompt injection attacks. Available Online: https://research.nccgroup.com/2022/12/05/exploring-prompt-injection-attacks/. [Accessed on January 28, 2024], 2023.
[63] Miguel Piedrafita. Methodologies of jailbreaking. Available Online: https://learnprompting.org/docs/prompt_hacking/jailbreakin. [Accessed on January 28, 2024], 2022.
[64] Malwarebytes. What is iPhone jailbreaking. Available Online: https://www.malwarebytes.com/iphone-jailbreaking. [Accessed on January 28, 2024], 2023.
[65] Pearlhawaii.com. What is jailbreaking, cracking, or rooting a mobile device? Available Online: https://pearlhawaii.com/what-is-jailbreaking-cracking-or-rooting-a-mobile-device. [Accessed on January 28, 2024], 2023.
[66] Dimitrios Damopoulos, Georgios Kambourakis, Stefanos Gritzalis, and Sang Oh Park. Exposing mobile malware from the inside (or what is your mobile app really doing?). *Peer-to-Peer Networking and Applications*, 7:687–697, 2014.
[67] Jomilė Nakutavičiūtė. Why root android phones? Available Online: https://nordvpn.com/blog/why-you-shouldnt-root-android [Accessed on January 28, 2024], 2023.
[68] Domenic Molinaro. What is rooting? the risks of rooting your android device. Available Online: https://www.avast.com/c-rooting-android [Accessed on January 28, 2024], 2023.
[69] Samuel R Bowman. Eight things to know about large language models. *arXiv preprint arXiv:2304.00612*, 2023.
[70] Jesse Senechal, Eric Ekholm, Samaher Aljudaibi, Mary Strawderman, and Chris Parthemos. Balancing the benefits and risks of large language AI models in K12 public schools. 2023.
[71] Fangzhao Wu, Yueqi Xie, Jingwei Yi, Jiawei Shao, Justin Curl, Lingjuan Lyu, Qifeng Chen, and Xing Xie. Defending ChatGPT against jailbreak attack via self-reminder. 2023.
[72] Alex Albert. Jailbreak Chat. Available Online: https://www.jailbreakchat.com [Accessed on January 28, 2024], 2023.
[73] Dirk HR Spennemann. Exploring ethical boundaries: Can ChatGPT be prompted to give advice on how to cheat in university assignments? 2023.

[74] Learn Prompting. Jailbreaking. Available Online: https://learnprompting.org/docs/prompt_hacking/jailbreaking [Accessed on January 28, 2024], 2023.
[75] Gelei Deng, Yi Liu, Yuekang Li, Kailong Wang, Ying Zhang, Zefeng Li, Haoyu Wang, Tianwei Zhang, and Yang Liu. MasterKey: Automated jailbreak across multiple large language model chatbots. *arXiv preprint arXiv:2307.08715*, 2023.
[76] Raz Lapid, Ron Langberg, and Moshe Sipper. Open sesame! universal black box jailbreaking of large language models. *arXiv preprint arXiv:2309.01446*, 2023.
[77] Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.
[78] Xiangyu Qi, Kaixuan Huang, Ashwinee Panda, Mengdi Wang, and Prateek Mittal. Visual adversarial examples jailbreak aligned large language models. In *The Second Workshop on New Frontiers in Adversarial Machine Learning*, 2023.
[79] Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries. *arXiv preprint arXiv:2310.08419*, 2023.
[80] Kui Ren, Tianhang Zheng, Zhan Qin, and Xue Liu. Adversarial attacks and defenses in deep learning. *Engineering*, 6(3):346–360, 2020.
[81] Brindha Jeyaraman. Adversarial attacks on LLMs: Safeguarding language models against manipulation, 2023.
[82] Wei Emma Zhang, Quan Z Sheng, Ahoud Alhazmi, and Chenliang Li. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(3):1–41, 2020.
[83] Thuy Dung Nguyen, Tuan Nguyen, Phi Le Nguyen, Hieu H Pham, Khoa D Doan, and Kok-Seng Wong. Backdoor attacks and defenses in federated learning: Survey, challenges and future research directions. *Engineering Applications of Artificial Intelligence*, 127:107166, 2024.
[84] Linyang Li, Demin Song, Xiaonan Li, Jiehang Zeng, Ruotian Ma, and Xipeng Qiu. Backdoor attacks on pre-trained models by layerwise weight poisoning. *arXiv preprint arXiv:2108.13888*, 2021.
[85] Wenkai Yang, Lei Li, Zhiyuan Zhang, Xuancheng Ren, Xu Sun, and Bin He. Be careful about poisoned word embeddings: Exploring the vulnerability of the embedding layers in NLP models. *arXiv preprint arXiv:2103.15543*, 2021.
[86] Shuai Zhao, Jinming Wen, Luu Anh Tuan, Junbo Zhao, and Jie Fu. Prompt as triggers for backdoor attack: Examining the vulnerability in language models. *arXiv preprint arXiv:2305.01219*, 2023.
[87] Jiongxiao Wang, Zichen Liu, Keun Hee Park, Muhao Chen, and Chaowei Xiao. Adversarial demonstration attacks on large language models. *arXiv preprint arXiv:2305.14950*, 2023.
[88] Xiangrui Cai, Haidong Xu, Sihan Xu, Ying Zhang, et al. BadPrompt: Backdoor attacks on continuous prompts. *Advances in Neural Information Processing Systems*, 35:37068–37080, 2022.
[89] Jiazhao Li, Yijin Yang, Zhuofeng Wu, VG Vydiswaran, and Chaowei Xiao. ChatGPT as an attack tool: Stealthy textual backdoor attack via blackbox generative model trigger. *arXiv preprint arXiv:2304.14475*, 2023.
[90] Hai Huang, Zhengyu Zhao, Michael Backes, Yun Shen, and Yang Zhang. Composite backdoor attacks against large language models. *arXiv preprint arXiv:2310.07676*, 2023.
[91] Meta-Llma. Llama-2-7b. Available Online: https://huggingface.co/huggyllama/llama-7bf [Accessed on January 28, 2024], 2023.
[92] Meta-Llma. Llama-2-13b-hf. Available Online: https://huggingface.co/meta-llama/Llama-2-13b-hf [Accessed on January 28, 2024], 2023.
[93] Meta-Llma. Llama-30b. Available Online: https://huggingface.co/huggyllama/llama-30b [Accessed on January 28, 2024], 2023.
[94] Nikhil Kandpal, Matthew Jagielski, Florian Tramèr, and Nicholas Carlini. Backdoor attacks for in-context learning with language models. *arXiv preprint arXiv:2307.14692*, 2023.
[95] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. In *25th Annual Network And Distributed System Security Symposium*. Internet Soc, 2018.
[96] Jacob Fox. Data poisoning attacks: A new attack vector within AI. Available Online: https://www.cobalt.io/blog/data-poisoning-attacks-a-new-attack-vector-within-ai#:~:text=An%20Artificial%20Intelligence%20poisoning%20attack, the%20model's%20decision%2Dmaking%20processes [Accessed on January 28, 2024], 2023.
[97] Keita Kurita, Paul Michel, and Graham Neubig. Weight poisoning attacks on pre-trained models. *arXiv preprint arXiv:2004.06660*, 2020.
[98] Xinyang Zhang, Zheng Zhang, Shouling Ji, and Ting Wang. Trojaning language models for fun and profit. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 179–197. IEEE, 2021.
[99] Hojjat Aghakhani, Wei Dai, Andre Manoel, Xavier Fernandes, Anant Kharkar, Christopher Kruegel, Giovanni Vigna, David Evans, Ben Zorn, and Robert Sim. TrojanPuzzle: Covertly poisoning code-suggestion models. *arXiv preprint arXiv:2301.02344*, 2023.

[100] Neel Jain, Avi Schwarzschild, Yuxin Wen, Gowthami Somepalli, John Kirchenbauer, Ping-yeh Chiang, Micah Goldblum, Aniruddha Saha, Jonas Geiping, and Tom Goldstein. Baseline defenses for adversarial attacks against aligned language models. *arXiv preprint arXiv:2309.00614*, 2023.

[101] Jindong Wang, Xixu Hu, Wenxin Hou, Hao Chen, Runkai Zheng, Yidong Wang, Linyi Yang, Haojun Huang, Wei Ye, Xiubo Geng, et al. On the robustness of ChatGPT: An adversarial and out-of-distribution perspective. *arXiv preprint arXiv:2302.12095*, 2023.

[102] Natalie Maus, Patrick Chao, Eric Wong, and Jacob Gardner. Adversarial prompting for black box foundation models. *arXiv preprint arXiv:2302.04237*, 2023.

[103] Xudong Pan, Mi Zhang, Shouling Ji, and Min Yang. Privacy risks of general-purpose language models. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1314–1331. IEEE, 2020.

[104] Siddhant Haldar. Gradient-based adversarial attacks : An introduction. Available Online: https://medium.com/swlh/gradient-based-adversarial-attacks-an-introduction-526238660dc9s [Accessed on January 28, 2024], 2023.

[105] Wenqi Wei, Ling Liu, Margaret Loper, Ka-Ho Chow, Mehmet Emre Gursoy, Stacey Truex, and Yanzhao Wu. A framework for evaluating client privacy leakages in federated learning. In *Computer Security–ESORICS 2020: 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, September 14–18, 2020, Proceedings, Part I 25*, pages 545–566. Springer, 2020.

[106] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. *Advances in neural information processing systems*, 32, 2019.

[107] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients-how easy is it to break privacy in federated learning? *Advances in Neural Information Processing Systems*, 33:16937–16947, 2020.

[108] Badhan Chandra Das, M Hadi Amini, and Yanzhao Wu. Privacy risks analysis and mitigation in federated learning for medical images. *arXiv preprint arXiv:2311.06643*, 2023.

[109] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. TinyBERT: Distilling BERT for natural language understanding. *arXiv preprint arXiv:1909.10351*, 2019.

[110] Mislav Balunovic, Dimitar Dimitrov, Nikola Jovanović, and Martin Vechev. LAMP: Extracting text from gradients with language model priors. *Advances in Neural Information Processing Systems*, 35:7641–7654, 2022.

[111] Zekeriya Erkin, Martin Franz, Jorge Guajardo, Stefan Katzenbeisser, Inald Lagendijk, and Tomas Toft. Privacy-preserving face recognition. In *Privacy Enhancing Technologies: 9th International Symposium, PETS 2009, Seattle, WA, USA, August 5-7, 2009. Proceedings 9*, pages 235–253. Springer, 2009.

[112] Yang Yang. Holistic risk assessment of inference attacks in machine learning. *arXiv preprint arXiv:2212.10628*, 2022.

[113] Ben Dickson. Machine learning: What are membership inference attacks? Available Online: https://bdtechtalks.com/2021/04/23/machine-learning-membership-inference-attacks/ [Accessed on January 28, 2024], 2021.

[114] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. White-box vs black-box: Bayes optimal strategies for membership inference. In *International Conference on Machine Learning*, pages 5558–5567. PMLR, 2019.

[115] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium*, pages 268–282. IEEE, 2018.

[116] Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Lei Yu, and Wenqi Wei. Demystifying membership inference attacks in machine learning as a service. *IEEE Transactions on Services Computing*, 14(6):2073–2089, 2019.

[117] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.

[118] Congzheng Song and Vitaly Shmatikov. Auditing data provenance in text-generation models. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 196–206, 2019.

[119] Virat Shejwalkar, Huseyin A Inan, Amir Houmansadr, and Robert Sim. Membership inference attacks against NLP classification models. In *NeurIPS 2021 Workshop Privacy in Machine Learning*, 2021.

[120] Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri. Enhanced membership inference attacks against machine learning models. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 3093–3106, 2022.

[121] Florian Tramèr, Gautam Kamath, and Nicholas Carlini. Considerations for differentially private learning with large-scale public pretraining. *arXiv preprint arXiv:2212.06470*, 2022.

[122] Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schölkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. Membership inference attacks against language models via neighbourhood comparison. *arXiv preprint arXiv:2305.18462*, 2023.

[123] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, page 2, 2019.

[124] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: a robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*,

2019.

[125] Fatemehsadat Mireshghallah, Kartik Goyal, Archit Uniyal, Taylor Berg-Kirkpatrick, and Reza Shokri. Quantifying privacy risks of masked language models using membership inference attacks. *arXiv preprint arXiv:2203.03929*, 2022.

[126] Congzheng Song and Ananth Raghunathan. Information leakage in embedding models. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, pages 377–390, 2020.

[127] Abhyuday Jagannatha, Bhanu Pratap Singh Rawat, and Hong Yu. Membership inference attack susceptibility of clinical language models. *arXiv preprint arXiv:2104.08305*, 2021.

[128] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[129] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[130] Li Hu, Anli Yan, Hongyang Yan, Jin Li, Teng Huang, Yingying Zhang, Changyu Dong, and Chunsheng Yang. Defenses to membership inference attacks: A survey. *ACM Computing Surveys*, 56(4):1–34, 2023.

[131] Nikhil Kandpal, Eric Wallace, and Colin Raffel. Deduplicating training data mitigates privacy risks in language models. In *International Conference on Machine Learning*, pages 10697–10707. PMLR, 2022.

[132] JAKE FRANKENFIELD. What is personally identifiable information (PII)? types and examples. Available Online: https://www.investopedia.com/terms/p/personally-identifiable-information-pii.asp [Accessed on January 28, 2024], 2023.

[133] Surbhi Gupta, Abhishek Singhal, and Akanksha Kapoor. A literature survey on social engineering attacks: Phishing attack. In *2016 international conference on computing, communication and automation*, pages 537–540. IEEE, 2016.

[134] Nir Kshetri. Cybercrime and privacy threats of large language models. *IT Professional*, 25(3):9–13, 2023.

[135] Fei Zheng. Input reconstruction attack against vertical federated large language models. *arXiv preprint arXiv:2311.07585*, 2023.

[136] Huseyin A Inan, Osman Ramadan, Lukas Wutschitz, Daniel Jones, Victor Rühle, James Withers, and Robert Sim. Training data leakage analysis in language models. *arXiv preprint arXiv:2101.05405*, 2021.

[137] Santiago Zanella-Béguelin, Lukas Wutschitz, Shruti Tople, Victor Rühle, Andrew Paverd, Olga Ohrimenko, Boris Köpf, and Marc Brockschmidt. Analyzing information leakage of updates to natural language models. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, pages 363–375, 2020.

[138] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. OPT: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

[139] Siwon Kim, Sangdoo Yun, Hwaran Lee, Martin Gubri, Sungroh Yoon, and Seong Joon Oh. ProPILE: Probing privacy leakage in large language models. *arXiv preprint arXiv:2307.01881*, 2023.

[140] Xiaoyi Chen, Siyuan Tang, Rui Zhu, Shijun Yan, Lei Jin, Zihao Wang, Liya Su, XiaoFeng Wang, and Haixu Tang. The janus interface: How fine-tuning in large language models amplifies the privacy risks. *arXiv preprint arXiv:2310.15469*, 2023.

[141] Yuta Nakamura, Shouhei Hanaoka, Yukihiro Nomura, Naoto Hayashi, Osamu Abe, Shuntaro Yada, Shoko Wakamiya, and Eiji Aramaki. KART: Privacy leakage framework of language models pre-trained with clinical records. *arXiv preprint arXiv:2101.00036*, 2020.

[142] Jie Huang, Hanyin Shao, and Kevin Chen-Chuan Chang. Are large pre-trained language models leaking your personal information? *arXiv preprint arXiv:2205.12628*, 2022.

[143] OpenAI. API to prevent prompt injection & jailbreaks. Available Online: https://community.openai.com/t/api-to-prevent-prompt-injection-jailbreaks/203514 [Accessed on January 28, 2024], 2023.

[144] Learn Prompting. Your guide to generative AI. Available Online: https://learnprompting.org [Accessed on January 28, 2024], 2023.

[145] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[146] Yequan Wang, Jiawen Deng, Aixin Sun, and Xuying Meng. Perplexity from plm is unreliable for evaluating text quality. *arXiv preprint arXiv:2210.05892*, 2022.

[147] Hila Gonen, Srini Iyer, Terra Blevins, Noah A Smith, and Luke Zettlemoyer. Demystifying prompts in language models via perplexity estimation. *arXiv preprint arXiv:2212.04037*, 2022.

[148] Todor Markov, Chong Zhang, Sandhini Agarwal, Florentine Eloundou Nekoul, Theodore Lee, Steven Adler, Angela Jiang, and Lilian Weng. A holistic approach to undesired content detection in the real world. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 15009–15018, 2023.

[149] Aounon Kumar, Chirag Agarwal, Suraj Srinivas, Soheil Feizi, and Hima Lakkaraju. Certifying LLM safety against adversarial prompting. *arXiv preprint arXiv:2309.02705*, 2023.

[150] Linyi Li, Tao Xie, and Bo Li. SoK: Certified robustness for deep neural networks. In *2023 IEEE symposium on security and privacy (SP)*, pages 1289–1310. IEEE, 2023.
[151] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *international conference on machine learning*, pages 1310–1320. PMLR, 2019.
[152] Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. Diffusion models for adversarial purification. *arXiv preprint arXiv:2205.07460*, 2022.
[153] Zeyang Sha, Xinlei He, Pascal Berrang, Mathias Humbert, and Yang Zhang. Fine-tuning is all you need to mitigate backdoor attacks. *arXiv preprint arXiv:2212.09067*, 2022.
[154] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-Pruning: Defending against backdooring attacks on deep neural networks. In *International symposium on research in attacks, intrusions, and defenses*, pages 273–294. Springer, 2018.
[155] Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, and Biplav Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering. *arXiv preprint arXiv:1811.03728*, 2018.
[156] Zhiyuan Zhang, Lingjuan Lyu, Xingjun Ma, Chenguang Wang, and Xu Sun. Fine-Mixing: Mitigating backdoors in fine-tuned language models. *arXiv preprint arXiv:2210.09545*, 2022.
[157] Ganqu Cui, Lifan Yuan, Bingxiang He, Yangyi Chen, Zhiyuan Liu, and Maosong Sun. A unified evaluation of textual backdoor learning: Frameworks and benchmarks. *Advances in Neural Information Processing Systems*, 35:5009–5023, 2022.
[158] Leland McInnes, John Healy, and Steve Astels. HDBSCAN: Hierarchical density based clustering. *J. Open Source Softw.*, 2(11):205, 2017.
[159] Maximilian Mozes, Xuanli He, Bennett Kleinberg, and Lewis D Griffin. Use of LLMs for illicit purposes: Threats, prevention measures, and vulnerabilities. *arXiv preprint arXiv:2308.12833*, 2023.
[160] Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. RAP: Robustness-aware perturbations for defending against backdoor attacks on NLP models. *arXiv preprint arXiv:2110.07831*, 2021.
[161] Fanchao Qi, Yangyi Chen, Mukai Li, Yuan Yao, Zhiyuan Liu, and Maosong Sun. ONION: A simple and effective defense against textual backdoor attacks. *arXiv preprint arXiv:2011.10369*, 2020.
[162] Zhaohan Xi, Tianyu Du, Changjiang Li, Ren Pang, Shouling Ji, Jinghui Chen, Fenglong Ma, and Ting Wang. Defending pre-trained language models as few-shot learners against backdoor attacks. *arXiv preprint arXiv:2309.13256*, 2023.
[163] Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*, 2020.
[164] Lei Xu, Yangyi Chen, Ganqu Cui, Hongcheng Gao, and Zhiyuan Liu. Exploring the universal vulnerability of prompt-based learning paradigm. *arXiv preprint arXiv:2204.05239*, 2022.
[165] Prachi (Nayyar) Pathak. How do you protect machine learning from attacks? Available Online: https://www.linkedin.com/advice/1/how-do-you-protect-machine-learning-from-attacks#data-poisoning-attack [Accessed on January 28, 2024], 2023.
[166] Nathalie Baracaldo, Bryant Chen, Heiko Ludwig, and Jaehoon Amir Safavi. Mitigating poisoning attacks on machine learning models: A data provenance based approach. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pages 103–110, 2017.
[167] Eric Wallace, Tony Z Zhao, Shi Feng, and Sameer Singh. Concealed data poisoning attacks on NLP models. *arXiv preprint arXiv:2010.12563*, 2020.
[168] Edward Chou, Florian Tramèr, Giancarlo Pellegrino, and Dan Boneh. SentiNet: Detecting physical attacks against deep learning systems.(2018). 2018.
[169] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 707–723. IEEE, 2019.
[170] Huili Chen, Cheng Fu, Jishen Zhao, and Farinaz Koushanfar. DeepInspect: A black-box trojan detection and mitigation framework for deep neural networks. In *IJCAI*, volume 2, page 8, 2019.
[171] Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017.
[172] Yoshinori Aono, Takuya Hayashi, Lihua Wang, Shiho Moriai, et al. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE transactions on information forensics and security*, 13(5):1333–1345, 2017.
[173] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.
[174] Ruihan Wu, Xiangyu Chen, Chuan Guo, and Kilian Q Weinberger. Learning to Invert: Simple adaptive attacks for gradient inversion in federated learning. In *Uncertainty in Artificial Intelligence*, pages 2293–2303. PMLR, 2023.

[175] Mohammad Raeini. Privacy-preserving large language models (PPLLMs). *Available at SSRN 4512071*, 2023.
[176] Milad Nasr, Reza Shokri, and Amir Houmansadr. Machine learning with membership privacy using adversarial regularization. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, pages 634–646, 2018.
[177] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. ML-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *arXiv preprint arXiv:1806.01246*, 2018.
[178] Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. Large language models can be strong differentially private learners. *arXiv preprint arXiv:2110.05679*, 2021.
[179] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.
[180] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.
[181] Wenjie Fu, Huandong Wang, Chen Gao, Guanghua Liu, Yong Li, and Tao Jiang. Practical membership inference attacks against fine-tuned large language models via self-prompt calibration. *arXiv preprint arXiv:2311.06062*, 2023.
[182] Yijue Wang, Nuo Xu, Shaoyi Huang, Kaleel Mahmood, Dan Guo, Caiwen Ding, Wujie Wen, and Sanguthevar Rajasekaran. Analyzing and defending against membership inference attacks in natural language processing classification. In *2022 IEEE International Conference on Big Data (Big Data)*, pages 5823–5832. IEEE, 2022.
[183] Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better. *arXiv preprint arXiv:2107.06499*, 2021.
[184] Andrea Continella, Yanick Fratantonio, Martina Lindorfer, Alessandro Puccetti, Ali Zand, Christopher Kruegel, Giovanni Vigna, et al. Obfuscation-resilient privacy leak detection for mobile apps through differential analysis. In *NDSS*, volume 17, pages 10–14722, 2017.
[185] Jingjing Ren, Ashwin Rao, Martina Lindorfer, Arnaud Legout, and David Choffnes. ReCon: Revealing and controlling PII leaks in mobile network traffic. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, pages 361–374, 2016.
[186] Rohan Anil, Badih Ghazi, Vineet Gupta, Ravi Kumar, and Pasin Manurangsi. Large-scale differentially private BERT. *arXiv preprint arXiv:2108.01624*, 2021.
[187] Huseyin A Inan, Osman Ramadan, Lukas Wutschitz, Daniel Jones, Victor Rühle, James Withers, and Robert Sim. Privacy analysis in language models via training data leakage report. *ArXiv, abs/2101.05405*, 2021.
[188] develop.sentry.dev. PII and Data Scrubbing. Available Online: https://develop.sentry.dev/pii [Accessed on January 28, 2024], 2023.
[189] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*, 2016.
[190] Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. DIALOGPT: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536*, 2019.
[191] Andrew Hoang, Antoine Bosselut, Asli Celikyilmaz, and Yejin Choi. Efficient adaptation of pretrained transformers for abstractive summarization. *arXiv preprint arXiv:1906.00138*, 2019.
[192] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.
[193] Roger Ratcliff. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review*, 97(2):285, 1990.
[194] Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Eric Sun, and Yue Zhang. A survey on large language model (LLM) security and privacy: The good, the bad, and the ugly. *arXiv preprint arXiv:2312.02003*, 2023.
[195] Taylor Sorensen, Joshua Robinson, Christopher Michael Rytting, Alexander Glenn Shaw, Kyle Jeffrey Rogers, Alexia Pauline Delorey, Mahmoud Khalil, Nancy Fulda, and David Wingate. An information-theoretic approach to prompt engineering without ground truth labels. *arXiv preprint arXiv:2203.11364*, 2022.
[196] Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, et al. Ethical and social risks of harm from language models. *arXiv preprint arXiv:2112.04359*, 2021.
[197] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, 2023.
[198] Laura Weidinger, Jonathan Uesato, Maribeth Rauh, Conor Griffin, Po-Sen Huang, John Mellor, Amelia Glaese, Myra Cheng, Borja Balle, Atoosa Kasirzadeh, et al. Taxonomy of risks posed by language models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 214–229, 2022.

[199] Chenxi Whitehouse, Tillman Weyde, Pranava Madhyastha, and Nikos Komninos. Evaluation of fake news detection with knowledge-enhanced language models. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 16, pages 1425–1429, 2022.

[200] Yonglin Tian, Jiangong Wang, Yutong Wang, Chen Zhao, Fei Yao, and Xiao Wang. Federated vehicular transformers and their federations: Privacy-preserving computing and cooperation for autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 2022.

[201] Abdel-Aty Zheng, Wang Wang, and Ding. Chat-GPT is on the horizon: Could a large language model be suitable for intelligent traffic safety research and applications? *https://arxiv.org/ftp/arxiv/papers/2303/2303.05382.pdf*, 2023.

[202] Yonglin Tian, Xuan Li, Hui Zhang, Chen Zhao, Bai Li, Xiao Wang, and Fei-Yue Wang. VistaGPT: Generative parallel transformers for vehicles with intelligent systems for transport automation. *IEEE Transactions on Intelligent Vehicles*, 2023.

[203] Ou Zheng, Mohamed Abdel-Aty, Dongdong Wang, Chenzhu Wang, and Shengxuan Ding. TrafficSafetyGPT: Tuning a pre-trained large language model to a domain-specific expert in transportation safety. *arXiv preprint arXiv:2307.15311*, 2023.

[204] Luigi De Angelis, Francesco Baglivo, Guglielmo Arzilli, Gaetano Pierpaolo Privitera, Paolo Ferragina, Alberto Eugenio Tozzi, and Caterina Rizzo. ChatGPT and the rise of large language models: the new ai-driven infodemic threat in public health. *Frontiers in Public Health*, 11:1166120, 2023.

[205] Stefan Harrer. Attention is not all you need: the complicated case of ethically using large language models in healthcare and medicine. *EBioMedicine*, 90, 2023.

[206] Silvia Milano, Joshua A McGrane, and Sabina Leonelli. Large language models challenge the future of higher education. *Nature Machine Intelligence*, 5(4):333–334, 2023.

[207] Julian Hazell. Large language models can be used to effectively scale spear phishing campaigns. *arXiv preprint arXiv:2305.06972*, 2023.

[208] The Conversation. The galactica AI model was trained on scientific knowledge – but it spat out alarmingly plausible nonsense. Available Online: https://theconversation.com/the-galactica-ai-model-was-trained-on-scientific-knowledge-but-it-spat-out-alarmingly-plausible-nonsense-195445 [Accessed on January 28, 2024], 2022.

[209] Will Douglas Heaven. "why Meta's latest large language model survived only three days online". *MIT Technology Review. Last accessed December*, 15:2022, 2022.

[210] David Leslie & Sandra Wachter Abeba Birhane, Atoosa Kasirzadeh. Science in the age of large language models. *doi.org/10.1038/s42254-023-00581-4*, 2023.

[211] Marcel Binz, Stephan Alaniz, Adina Roskies, Balazs Aczel, Carl T Bergstrom, Colin Allen, Daniel Schad, Dirk Wulff, Jevin D West, Qiong Zhang, et al. How should the advent of large language models affect the practice of science? *arXiv preprint arXiv:2312.03759*, 2023.

[212] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing GPT-4 with 90%* chatGPT quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2023.

[213] James Manyika and Sissie Hsiao. An overview of Bard: an early experiment with generative AI. *AI. Google Static Documents*, 2, 2023.

[214] Yusuf Mehdi. Reinventing search with a new AI-powered Microsoft Bing and Edge, your copilot for the web. Available Online: https://blogs.microsoft.com/blog/2023/02/07/reinventing-search-with-a-new-ai-powered-microsoft-bing-and-edge-your-copilot-for-the-web/ [Accessed on January 28, 2024], 2023.

[215] Ningxin Su, Chenghao Hu, Baochun Li, and Bo Li. TITANIC: Towards production federated learning with large language models.

[216] Microsoft. Bing AI chatbot. Available Online: https://www.bing.com/chat [Accessed on January 28, 2024], 2023.

[217] Kun Shao, Junan Yang, Yang Ai, Hui Liu, and Yu Zhang. BDDR: An effective defense against textual backdoor attacks. *Computers & Security*, 110:102433, 2021.

[218] Wei Du, Yichun Zhao, Boqun Li, Gongshen Liu, and Shilin Wang. PPT: Backdoor attacks on pre-trained models via poisoned prompt tuning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 680–686, 2022.

[219] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34, 2020.

[220] Ervin Moore, Ahmed Imteaj, Shabnam Rezapour, and M. Hadi Amini. A survey on secure and private federated learning using blockchain: Theory and application in resource-constrained computing. *IEEE Internet of Things Journal*, 10(24):21942–21958, 2023.

[221] Ronald Cramer, Ivan Bjerre Damgård, et al. *Secure multiparty computation*. Cambridge University Press, 2015.