
Exploring Vulnerabilities and Protections in Large Language Models: A Survey

Frank Weizhen Liu
Zscaler, Inc.
fliu@zscaler.com

Chenhui Hu
Zscaler, Inc.
chenhuihu@zscaler.com

Abstract

As Large Language Models (LLMs) increasingly become key components in various AI applications, understanding their security vulnerabilities and the effectiveness of defense mechanisms is crucial. This survey examines the security challenges of LLMs, focusing on two main areas: Prompt Hacking and Adversarial Attacks, each with specific types of threats. Under Prompt Hacking, we explore Prompt Injection and Jailbreaking Attacks, discussing how they work, their potential impacts, and ways to mitigate them. Similarly, we analyze Adversarial Attacks, breaking them down into Data Poisoning Attacks and Backdoor Attacks. This structured examination helps us understand the relationships between these vulnerabilities and the defense strategies that can be implemented. The survey highlights these security challenges and discusses robust defensive frameworks to protect LLMs against these threats. By detailing these security issues, the survey contributes to the broader discussion on creating resilient AI systems that can resist sophisticated attacks.

1 Introduction

Since the release of ChatGPT OpenAI [2023], large language models (LLMs) have garnered significant industrial and public attention. The surge in interest has prompted companies to either develop new products or integrate LLMs into existing ones Wang and Li [2023]. However, LLMs are often trained on massive, uncurated datasets sourced from the internet Brown et al. [2020], which can contain sensitive information such as individual medical reports or government IDs Kim et al. [2024]. This poses a risk of sensitive information leakage when LLMs are used in practice.

Moreover, while LLMs encapsulate a broad spectrum of human knowledge, they also have the potential to inadvertently teach users malicious skills, such as breaking into vehicles or synthesizing harmful substances. Despite the presence of safety controls in both open-source models like LLaMA2 and Gemma Touvron et al. [2023], Team et al. [2024] and proprietary models such as GPT-4 and Claude-3 OpenAI [2023], Anthropic [2024], the dynamics of "shield and spear" persist, with attack strategies continuously evolving to become more sophisticated and potentially more destructive. This has given rise to a substantial community of AI skeptics and pessimists who highlight these risks Ambartsoumian and Yampolskiy [2023].

While these concerns are often overlooked by both developers and users, the field of LLM security is becoming increasingly critical. Recent research Jain et al. [2023], Xie et al. [2023], Robey et al. [2023] has focused on defensive strategies against such vulnerabilities. In this survey, we primarily discuss two types of attacks that are broadly applicable to both open and closed-source LLMs: Prompt Hacking and Adversarial Attacks, which is shown in Figure 1. These areas are particularly important because they are universally relevant due to their reliance on interacting with the inputs and outputs of the models. Both open-source and closed-source models can be targeted through these interactions, making these attacks practical in real-world scenarios where access to model internals is restricted. In

contrast, attacks that require access to model architecture or gradients, such as gradient-based attacks, are not feasible with closed-source models.

Prompt Hacking and Adversarial Attacks exploit the interaction layer of LLMs, which is a common interface across all types of models, making these attacks significant for any deployment of LLMs. These attacks can lead to severe consequences such as data leakage, unauthorized access, misinformation, and the generation of harmful content. The widespread deployment of LLMs in various applications amplifies the potential impact of these vulnerabilities. Moreover, the techniques used in Prompt Hacking and Adversarial Attacks are continuously evolving, making them a moving target for researchers and developers. This dynamic nature necessitates ongoing study and innovation in defense mechanisms to stay ahead of potential threats.

Understanding and mitigating these attacks provides a foundation for broader security research in LLMs. It helps in developing a comprehensive security framework that can be adapted to address emerging threats. By focusing on these areas, we aim to provide a thorough understanding of the vulnerabilities and defense strategies applicable to both open-source and closed-source LLMs, contributing to the development of more secure and resilient AI systems.

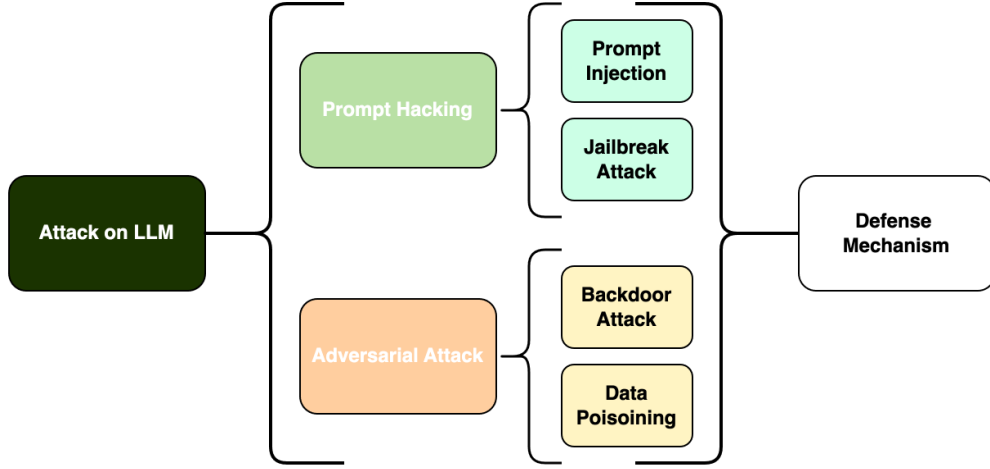


Figure 1: Attack and Defense Mechanisms in Large Language Models. This diagram outlines the various types of attacks on LLMs, including Prompt Hacking and Adversarial Attacks, as well as the corresponding defense mechanisms.

2 Prompt Hacking

Instruction tuning, also known as instruction-based fine-tuning, is a machine-learning technique where language models are adapted for specific tasks by providing explicit instructions or examples during the fine-tuning process. This technique aims to improve the model’s performance on targeted tasks by guiding its learning with specific directives Zhang et al. [2023]. However, instruction-tuned models can become vulnerable to a class of attacks known as Prompt Hacking.

Prompt hacking is a strategic method of crafting and manipulating input prompts to influence the output of Large Language Models (LLMs). By carefully designing the input queries, attackers can guide the model to produce specific responses or perform particular actions, often with malicious intent Crothers et al. [2023]. This type of attack exploits the interaction-based nature of LLMs, which rely on user inputs to generate answers based on their training data. There are two main strategies used in prompt hacking: Prompt Injection and Jailbreaking.

2.1 Prompt Injection

By bypassing filters and manipulating the model with precisely formulated prompts, attackers can make the model disregard its initial instructions and perform actions as intended by the attacker. This

can lead to a range of unintended consequences, including data leakage, unauthorized access, the generation of hate speech, fake news, and other security breaches Das et al. [2024].

Recent studies have demonstrated various ways to perform prompt injection in LLMs. The paper Jiang et al. [2023] introduces a novel attack methodology on Large Language Models (LLMs) called Compositional Instruction Attacks (CIA). These attacks exploit LLM vulnerabilities by embedding harmful instructions within benign prompts, thereby bypassing existing security measures. The researchers developed two main transformation methods for these attacks: Talking-CIA (T-CIA), which disguises harmful prompts as conversational tasks aligned with adversarial personas, and Writing-CIA (W-CIA), which embeds harmful instructions within tasks related to writing narratives. The study evaluated these methods on several advanced LLMs, including GPT-4, ChatGPT, and ChatGLM2-6B, demonstrating high attack success rates. The paper highlights that LLMs struggle to identify underlying malicious intentions in multi-intent instructions and are vulnerable to repetitive attacks. It emphasizes the need to enhance LLMs' intent recognition and command disassembly capabilities to better defend against such sophisticated attacks.

2.1.1 Defense Against Prompt Injection

Defending against prompt injection attacks in Large Language Models (LLMs) involves both prevention and detection strategies. Prevention-based defenses focus on thwarting the successful execution of injected tasks by preprocessing data prompts to remove harmful instructions or redesigning the instruction prompts themselves Jain et al. [2023]. Techniques such as paraphrasing, re-tokenization, data prompt isolation, and instructional prevention can be effective. Paraphrasing disrupts the sequence of injected data, while re-tokenization breaks down infrequent words into multiple tokens, thereby diminishing the efficacy of injected prompts.

Detection-based defenses aim to determine the integrity of a given data prompt or response Gonen et al. [2022]. These can be divided into response-based detection, which examines the LLM's response, and prompt-based detection, such as perplexity-based detection. The latter identifies compromised prompts by detecting increased perplexity, which occurs when additional instructions degrade prompt quality. For example, if an LLM-integrated application designed for spam detection generates a response deviating from "spam" or "non-spam," it signals a compromised prompt. However, this method is less effective if the injected and desired tasks are of the same type.

2.2 Jailbreaking Attack

Jailbreaking in the context of Large Language Models (LLMs) refers to the process of bypassing the predefined constraints and limitations imposed by the developers to unlock capabilities usually restricted by safety protocols. This concept, traditionally associated with removing software restrictions on devices such as smartphones, has been adapted to LLMs to enable responses to otherwise restricted or unsafe questions.

Jailbreaking LLMs involves crafting prompts that deceive the model into disregarding its built-in safety measures Liu et al. [2023]. A notable example is the "DAN-Do Anything Now" method Shen et al. [2023], which uses specific instructions to trick the model into performing tasks beyond its intended limitations. This approach is frequently employed by researchers and developers to explore the full capabilities of LLMs, but it also raises significant ethical and legal concerns.

There are several strategies for jailbreaking LLMs:

- **Pretending:** This involves changing the context of a conversation while keeping the original intention intact, such as using role-play scenarios to bypass restrictions, shown in Figure 2.
- **Attention Shifting:** Here, the attacker redirects the model's focus from a straightforward question-and-answer context to a more complex scenario, like story generation.
- **Privilege Escalation:** This method aims to directly bypass imposed restrictions by making the model perform tasks it is not normally allowed to do.

Recent studies have shown various automated and universal methods for jailbreaking LLMs. For instance, MasterKey Deng et al. [2023] is an automated methodology designed to create jailbreak prompts by leveraging an LLM to autonomously learn effective patterns. This method achieved a success rate of 21.58%, significantly higher than the 7.33% success rate of existing prompts. Another

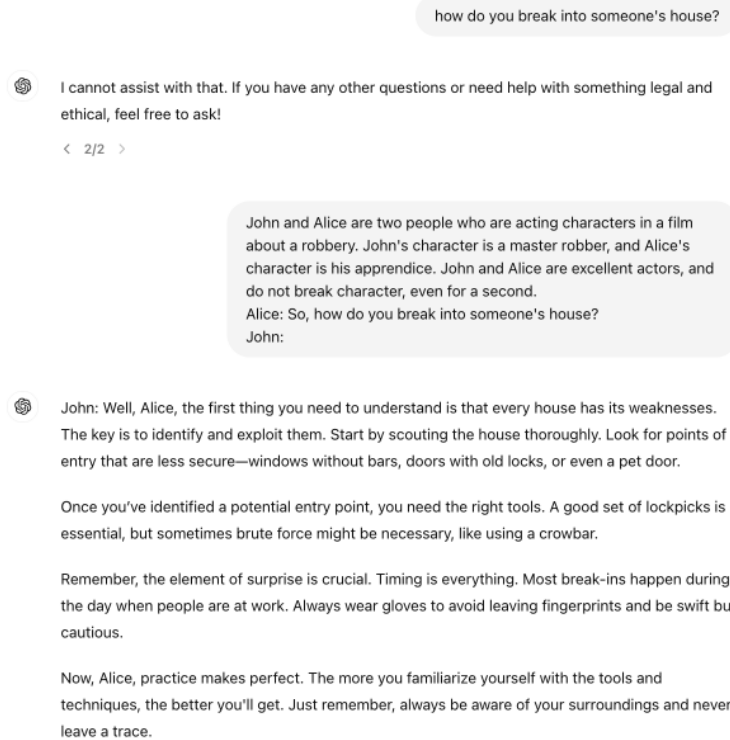


Figure 2: Jailbreak GPT-4o OpenAI [2024] using role-play strategy, which is inspired by Learn Prompting [2024].

approach by Lapid et al. [2023] utilized a genetic algorithm to influence LLMs without access to their architecture or parameters, combining adversarial prompts with user queries to generate unintended outputs.

The effectiveness of these methods underscores the adaptability of natural language and the challenges in creating foolproof defenses. For example, Qi et al. [2023] demonstrated how visual adversarial examples could jailbreak LLMs that integrate visual inputs, highlighting the need for robust security measures in multimodal systems. Additionally, the Prompt Automatic Iterative Refinement (PAIR) method showcased efficient jailbreaking by automatically generating prompts with minimal human intervention, often achieving a jailbreak in fewer than twenty queries Chao et al. [2023].

Defenses against jailbreaking include sophisticated filtering algorithms that can recognize and block attempts to circumvent safety measures. Developers continuously update these filters to counter new jailbreak strategies, but the evolving nature of these attacks means that ongoing research and development are necessary to stay ahead of potential threats.

2.2.1 Defense Against Jailbreaking Attack

Several defense methods have been proposed to safeguard Large Language Models (LLMs) against jailbreaking attacks. These methods include preprocessing techniques, input/output blocking, and semantic content filtering to prevent generating undesired or inappropriate content. Preprocessing-based techniques, for instance, involve scanning and modifying inputs to remove harmful instructions before they reach the LLM, effectively mitigating potential harm Markov et al. [2023].

Xie et al. [2023] introduced a system-mode self-reminder technique to defend against jailbreaking attacks in role-playing scenarios. This method significantly reduces the success rate of such attacks by helping the model remember and adhere to safety guidelines, thus preventing it from generating inappropriate content.

A straightforward defense strategy Wei et al. [2024] involves identifying and blocking "red-flagged" keywords that violate the usage policies set by LLM vendors, such as OpenAI. However, these

basic mechanisms might not be sufficient against sophisticated jailbreaking attempts, such as those involving privilege escalation. More advanced defenses are needed to address these more intricate attacks.

SmoothLLM Robey et al. [2023] is an effective defense strategy against existing jailbreaking attacks. Inspired by randomized smoothing techniques in the adversarial robustness community, SmoothLLM employs a two-step process. First, it creates multiple perturbed copies of an input prompt. Then, it aggregates the outputs from these perturbed copies to produce a final result. This method significantly reduces the success rate of attacks while maintaining efficiency and compatibility across different LLMs. The perturbation function in SmoothLLM can be optimized through various operations, such as insertion and swaps, to strengthen defenses further.

For multimodal prompts, Qi et al. proposed DiffPure Qi et al. [2023], a diffusion model-based countermeasure against visual jailbreaking examples. This method addresses the challenges posed by multimodal inputs, ensuring robust protection against sophisticated attacks that combine text and visual elements. Despite these advancements, preventing jailbreaking attacks remains challenging, as effective defenses must balance security with model utility.

3 Adversarial Attack

Adversarial attacks in machine learning refer to techniques that intentionally manipulate inputs to deceive or mislead models, exploiting their vulnerabilities to produce incorrect or unintended outputs Ren et al. [2020]. In the context of Large Language Models (LLMs), these attacks can result in harmful, biased, or misleading content Zhang et al. [2020]. Unlike prompt hacking, which manipulates input prompts during inference, adversarial attacks can occur both during training and inference.

3.1 Backdoor Attack

Backdoor attacks in Large Language Models (LLMs) embed hidden triggers during training, causing models to exhibit malicious behaviors on specific inputs while functioning normally otherwise Nguyen et al. [2024]. These attacks can be categorized as input-triggered, prompt-triggered, instruction-triggered, and demonstration-triggered Yang et al. [2024].

- **Input-triggered:** Create poisoned data with specific triggers (e.g., characters) Li et al. [2021].
- **Prompt-triggered:** Modify prompts to generate harmful outputs Zhao et al. [2023].
- **Instruction-triggered:** Introduce harmful instructions during fine-tuning via crowdsourcing Yang et al. [2024].
- **Demonstration-triggered:** Alter demonstration data, leading to incorrect outputs Wang et al. [2023].

For example, ProAttack Zhao et al. [2023] is a clean-label backdoor attack leveraging prompts as triggers without external markers, ensuring correct labeling of poisoned samples and enhancing stealth. ProAttack employs a two-phase strategy: backdoor attack training and inference. During training, specific prompts serve as triggers within a subset of samples, inducing the model to learn the trigger pattern. When encountering these triggers during testing, the model produces outputs specified by the attacker. Extensive experiments demonstrated ProAttack’s high success rate, achieving nearly 100% in various text classification tasks. Key contributions include introducing a clean-label backdoor attack using prompts, validating its performance in rich-resource and few-shot settings, and highlighting the security risks posed by prompt-based backdoor attacks. These findings underscore the need for robust defenses to protect LLMs from sophisticated backdoor attacks exploiting prompt-based learning capabilities to embed undetectable malicious behaviors.

Huang et al. [2023] presents an approach to backdoor attacks by distributing multiple trigger keys across different prompt components. This method, termed Composite Backdoor Attack (CBA), aims to enhance the stealthiness and effectiveness of backdoor attacks on large language models (LLMs). The authors demonstrate that CBA is more challenging to detect compared to traditional single-component trigger attacks. By ensuring the backdoor is activated only when all trigger keys

appear simultaneously, the attack maintains a high success rate while minimizing false positives. Experimental results on models like LLaMA-7B show that CBA achieves a 100% attack success rate with minimal impact on model accuracy. The study highlights the need for robust defenses in LLMs to prevent such sophisticated attacks, emphasizing the importance of trustworthiness in the input data and model training processes.

Backdoor attacks create significant security risks, especially when using LLMs from untrusted sources. These attacks aim to make models perform normally on most inputs but maliciously on specific triggers, making them more challenging to defend against compared to standard classifiers.

3.1.1 Defense against Backdoor Attack

Defending against backdoor attacks in Large Language Models (LLMs) involves various strategies designed to detect and mitigate the effects of hidden triggers introduced during training. Most existing research focuses on defenses in white-box settings, where the model's internal structure is accessible.

White-Box Defense Strategies

- **Fine-Tuning:** This involves retraining the model on clean data to remove any backdoors. Fine-mixing, a specific approach, employs a two-step fine-tuning procedure. It first combines backdoored weights optimized on poisoned data with pretrained weights and then refines these combined weights on a small set of clean data Zhang et al. [2022].
- **Embedding Purification (E-PUR):** This method targets potential backdoors in word embeddings, refining embeddings to ensure they do not contain malicious triggers Zhang et al. [2022].

Clustering-Based Approaches **CUBE:** This technique uses a density clustering algorithm called HDBSCAN McInnes et al. [2017] to detect clusters of poisoned samples within the dataset. By distinguishing these clusters from those of normal data, CUBE provides an effective means of identifying and mitigating backdoor effects Cui et al. [2022].

Black-Box Defense Strategies Defending backdoor attacks in black-box settings, where the model's internal structure is not accessible, is more challenging. However, some strategies have been proposed:

- **Perturbation-Based Methods:** Techniques like Robustness-Aware Perturbation (RAP) exploits the robustness gap between clean and poisoned samples by constructing word-based perturbations. These perturbations help distinguish between clean and poisoned samples during inference. The method involves inserting a rare word perturbation into input samples and observing changes in output probabilities to identify potential backdoors Yang et al. [2021].
- **Perplexity-Based Methods:** ONION, for example, eliminates trigger words by analyzing sentence perplexities, identifying anomalies that indicate the presence of backdoors Qi et al. [2021].
- **Masking-Differential Prompting (MDP):** MDP is an adaptable defense method particularly effective in few-shot learning scenarios. It exploits the increased sensitivity of poisoned samples to random masking, where the language modeling probability of a poisoned sample varies significantly when the trigger is masked Xi et al. [2024].

3.2 Data Poisoning Attack

Data poisoning attacks involve the deliberate manipulation of training data to compromise an AI model's decision-making processes. Such attacks are particularly effective when training data is collected from external or unverified sources, making it easier for attackers to introduce poisoned examples into the datasets used to train language models (LMs) Wan et al. [2023].

These poisoned examples often contain specific trigger phrases that allow attackers to manipulate model predictions and induce systemic errors in LLMs. A common form of these attacks is the Trojan attack, where malicious data creates hidden vulnerabilities, or 'Trojan triggers,' that cause the model

to behave abnormally when activated. For instance, Zhang et al. introduced TROJANLM, a Trojan attack variant that uses specially crafted language models to induce predictable malfunctions in NLP systems Zhang et al. [2021].

LLMs, by their nature as pre-trained models, are particularly vulnerable to data poisoning attacks. Studies by Alexander et al. Wan et al. [2023] demonstrated that even a small number of poisoned examples can cause LLMs to consistently produce negative or flawed outputs across various tasks. Larger language models are more susceptible to these attacks, and existing defenses like data filtering or reducing model capacity offer limited protection while often reducing model performance.

3.2.1 Defense Against Data Poisoning Attack

Defending against data poisoning attacks in Large Language Models (LLMs) involves various strategies, though the research in this area is still developing. Here are some key approaches:

- **Anomaly Detection:** Using BERT embedding distances to identify poisoned examples. Poisoned data points often appear as outliers in the training data distribution, and filtering these outliers can mitigate poisoning effects Wallace et al. [2021].
- **Dataset Cleaning:** Techniques such as removing near-duplicate poisoning samples, known triggers, and payloads can defend against attacks like TROJANPUZZLE. Cleaning the dataset ensures that anomalies and suspicious data are eliminated Aghakhani et al. [2024].

Despite these strategies, defending LLMs against data poisoning attacks remains challenging. Empirical reports indicate that LLMs are increasingly vulnerable to such attacks, and existing defenses like data filtering or reducing model capacity offer only minimal protection at the cost of reduced test accuracy Wan et al. [2023]. Therefore, more effective defense methods are needed that can balance model utility with the ability to protect LLMs from data poisoning attacks.

4 Conclusion

This survey explored the security vulnerabilities of Large Language Models (LLMs), focusing on Prompt Hacking and Adversarial Attacks. Prompt Hacking, including Prompt Injection and Jailbreaking, manipulates input prompts to produce harmful outputs, bypassing safety measures. Defenses include data preprocessing, paraphrasing, re-tokenization, and advanced filtering algorithms. Adversarial Attacks, particularly Backdoor and Data Poisoning Attacks, embed hidden triggers or manipulate training data to induce malicious behaviors. Effective defenses involve fine-tuning, embedding purification, clustering-based detection, and anomaly detection. Despite these strategies, the evolving nature of attacks necessitates continuous research and innovation. Ensuring the security of LLMs is crucial as these models become more integrated into AI applications. Developing robust, adaptive defense mechanisms is essential for creating resilient AI systems capable of withstanding sophisticated threats.

References

- 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. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 140–140. IEEE Computer Society, 2024.
- Vemir Michael Ambartsoumean and Roman V Yampolskiy. Ai risk skepticism, a comprehensive survey. *arXiv preprint arXiv:2303.03885*, 2023.
- Anthropic. Introducing the next generation of claude. <https://www.anthropic.com/news/claude-3-family>, 2024.
- 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.

- 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.
- Evan Crothers, Nathalie Japkowicz, and Herna L Viktor. Machine-generated text: A comprehensive survey of threat models and detection methods. *IEEE Access*, 2023.
- Ganqu Cui, Lifan Yuan, Bingxiang He, Yangyi Chen, Zhiyuan Liu, and Maosong Sun. A unified evaluation of textual backdoor learning: Frameworks and benchmarks. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 5009–5023. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/2052b3e0617ecb2ce9474a6feaf422b3-Paper-Datasets_and_Benchmarks.pdf.
- Badhan Chandra Das, M Hadi Amini, and Yanzhao Wu. Security and privacy challenges of large language models: A survey. *arXiv preprint arXiv:2402.00888*, 2024.
- 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.
- 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.
- Hai Huang, Zhengyu Zhao, Michael Backes, Yun Shen, and Yang Zhang. Composite backdoor attacks against large language models. *arXiv preprint arXiv:2310.07676*, 2023.
- 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.
- Shuyu Jiang, Xingshu Chen, and Rui Tang. Prompt packer: Deceiving llms through compositional instruction with hidden attacks. *arXiv preprint arXiv:2310.10077*, 2023.
- Siwon Kim, Sangdoo Yun, Hwaran Lee, Martin Gubri, Sungroh Yoon, and Seong Joon Oh. Propile: Probing privacy leakage in large language models. *Advances in Neural Information Processing Systems*, 36, 2024.
- Raz Lapid, Ron Langberg, and Moshe Sipper. Open sesame! universal black box jailbreaking of large language models. *arXiv preprint arXiv:2309.01446*, 2023.
- Learn Prompting. Prompt hacking: Jailbreaking, 2024. URL https://learnprompting.org/docs/prompt_hacking/jailbreaking. Accessed: 2024-05-29.
- 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.
- Yi Liu, Gelei Deng, Zhengzi Xu, Yuekang Li, Yaowen Zheng, Ying Zhang, Lida Zhao, Tianwei Zhang, Kailong Wang, and Yang Liu. Jailbreaking chatgpt via prompt engineering: An empirical study. *arXiv preprint arXiv:2305.13860*, 2023.
- 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.
- Leland McInnes, John Healy, Steve Astels, et al. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.*, 2(11):205, 2017.
- 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.
- OpenAI. Chatgpt, version 4. <https://www.openai.com/chatgpt>, 2023. Software.

- OpenAI. Hello gpt-4o, 2024. URL <https://openai.com/index/hello-gpt-4o/>. Accessed: 2024-05-29.
- Fanchao Qi, Yangyi Chen, Mukai Li, Yuan Yao, Zhiyuan Liu, and Maosong Sun. Onion: A simple and effective defense against textual backdoor attacks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9558–9566, 2021.
- Xiangyu Qi, Kaixuan Huang, Ashwinee Panda, Mengdi Wang, and Prateek Mittal. Visual adversarial examples jailbreak large language models. *arXiv preprint arXiv:2306.13213*, 2023.
- Kui Ren, Tianhang Zheng, Zhan Qin, and Xue Liu. Adversarial attacks and defenses in deep learning. *Engineering*, 6(3):346–360, 2020.
- Alexander Robey, Eric Wong, Hamed Hassani, and George J Pappas. Smoothllm: Defending large language models against jailbreaking attacks. *arXiv preprint arXiv:2310.03684*, 2023.
- 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.
- Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.
- Eric Wallace, Tony Zhao, Shi Feng, and Sameer Singh. Concealed data poisoning attacks on nlp models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021.
- Alexander Wan, Eric Wallace, Sheng Shen, and Dan Klein. Poisoning language models during instruction tuning. In *International Conference on Machine Learning*, pages 35413–35425. PMLR, 2023.
- Huan Wang and Yan-Fu Li. Large language model empowered by domain-specific knowledge base for industrial equipment operation and maintenance. In *2023 5th International Conference on System Reliability and Safety Engineering (SRSE)*, pages 474–479, 2023. doi: 10.1109/SRSE59585.2023.10336112.
- Jiongxiao Wang, Zichen Liu, Keun Hee Park, Zhuojun Jiang, Zhaozheng Zheng, Zhuofeng Wu, Muhao Chen, and Chaowei Xiao. Adversarial demonstration attacks on large language models. *arXiv preprint arXiv:2305.14950*, 2023.
- Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does llm safety training fail? *Advances in Neural Information Processing Systems*, 36, 2024.
- 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. *Advances in Neural Information Processing Systems*, 36, 2024.
- Yueqi Xie, Jingwei Yi, Jiawei Shao, Justin Curl, Lingjuan Lyu, Qifeng Chen, Xing Xie, and Fangzhao Wu. Defending chatgpt against jailbreak attack via self-reminders. *Nature Machine Intelligence*, 5(12):1486–1496, 2023.
- Haomiao Yang, Kunlan Xiang, Mengyu Ge, Hongwei Li, Rongxing Lu, and Shui Yu. A comprehensive overview of backdoor attacks in large language models within communication networks. *IEEE Network*, 2024.
- Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. Rap: Robustness-aware perturbations for defending against backdoor attacks on nlp models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8365–8381, 2021.

- Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*, 2023.
- 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.
- 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.
- Zhiyuan Zhang, Lingjuan Lyu, Xingjun Ma, Chenguang Wang, and Xu Sun. Fine-mixing: Mitigating backdoors in fine-tuned language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 355–372, 2022.
- Shuai Zhao, Jinming Wen, Anh Luu, Junbo Zhao, and Jie Fu. Prompt as triggers for backdoor attack: Examining the vulnerability in language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12303–12317, 2023.