Breaking Down the Defenses: A Comparative Survey of Attacks on Large Language Models

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have become a cornerstone in the field of Natural Language Processing (NLP), offering transformative capabilities in understanding and generating human-like text. However, with their rising prominence, the security and vulnerability aspects of these models have garnered significant attention. This paper presents a comprehensive survey of the various forms of attacks targeting LLMs, discussing the nature and mechanisms of these attacks, their potential impacts, and current defense strategies. We delve into topics such as adversarial attacks that aim to manipulate model outputs, data poisoning that affects model training, and privacy concerns related to training data exploitation. The paper also explores the effectiveness of different attack methodologies, the resilience of LLMs against these attacks, and the implications for model integrity and user trust. By examining the latest research, we provide insights into the current landscape of LLM vulnerabilities and defense mechanisms. Our objective is to offer a nuanced understanding of LLM attacks, foster awareness within the AI community, and inspire robust solutions to mitigate these risks in future developments.

1 Introduction

The emergence of artificial intelligence has marked a significant transformation in Natural Language Processing through the introduction of large language models (LLMs) enabling unprecedented advances in language comprehension, generation, and translation (Zhao et al., 2023c; Naveed et al., 2023; Achiam et al., 2023). Despite their transformative impact, LLMs have become susceptible to a variety of sophisticated attacks, posing significant challenges to their integrity and reliability (Yao et al., 2023; Liu et al., 2023d). This survey paper provides a comprehensive examination of the attacks targeting LLMs, elucidating their mechanisms, consequences, and the fast evolving threat landscape.

The significance of investigating attacks on LLMs lies in their extensive integration across various sectors and their consequential societal ramifications (Eloundou et al., 2023). LLMs are instrumental in applications ranging from automated customer support to sophisticated content creation. Therefore, understanding their vulnerabilities is imperative for ensuring the security and trustworthiness of AI-driven systems (Amodei et al., 2016; Hendrycks et al., 2023). This paper categorizes the spectrum of attacks, based on access to model weights and attack vectors, each presenting distinct challenges and requiring specific attention.

Additionally, the methodologies employed in executing these attacks are dissected, offering insights into the adversarial techniques utilized to exploit LLM vulnerabilities. While acknowledging the limitations of current defense mechanisms, the paper also proposes potential avenues for future research in enhancing LLM security.

We summarize the major contributions of our work as follows:

Our Contributions

- We propose a novel taxonomy of attacks on LLMs, which can help researchers to better understand the research landscape and find their areas of interest.
- We present existing attack and mitigation approaches in detail, discussing key implementation details.
- We discuss important challenges, highlighting promising directions for future research.

2 Exploring LLM Security: White and Black Box Attacks

This section delves into the security challenges of Large Language Models (LLMs) from both white box and black box perspectives. It highlights the importance of understanding and protecting LLMs against complex security threats.

2.1 White Box

These attacks exploit full access to the LLM's architecture, training data, and algorithms, enabling attackers to extract sensitive information, manipulate outputs, or insert malicious code. Shayegani et al. (2023) discusses whitebox attacks, highlighting how this access permits crafting adversarial inputs to alter outputs or impair performance. The study covers various attack strategies, such as context contamination and prompt injection, aimed at manipulating LLMs for specific outputs or reducing their quality.

Separately, Li et al. (2023a) examines privacy concerns in LLMs, emphasizing the importance of protecting personal information in the face of evolving AI technologies. They discuss the privacy risks associated with training and inference data, highlighting the critical need to analyze whitebox attacks for effective threat mitigation.

2.2 Black Box

These attacks exploit LLM vulnerabilities with limited knowledge of the model's internals, focusing on manipulating or degrading performance through the input-output interface. This approach, realistic in practical scenarios, poses risks such as sensitive data extraction, biased outputs, and diminished trust in AI. Chao et al. (2023) illustrates black-box methods to "jailbreak" LLMs like GPT-3.5 and GPT-4, with Qi et al. (2023a); Yong et al. (2023) exploring attacks on API-based models such as GPT-4 across various surfaces.

3 LLM Attacks Taxonomy

3.1 Jailbreaks

This section delves into jailbreak attacks on LLMs, detailing strategies to exploit model vulnerabilities for unauthorized actions, underscoring the critical need for robust defense mechanisms.

Refined Query-Based Jailbreaking: Chao et al. (2023) represent a strategic approach in jailbreaking, utilizing a minimal number of queries. This method doesn't just exploit simple model vulnerabilities but involves a nuanced understanding of the model's response mechanism, iteratively refining queries to probe and eventually bypass the model's defenses. The success of this approach underscores a key vulnerability in LLMs: their predictability and manipulability through iterative, intelligent querying. This work introduces Prompt Automatic Iterative Refinement (PAIR), an algorithm designed to automate the generation of semantic jailbreaks for LLMs. PAIR works by using an attacker LLM to iteratively query a target LLM, refining a candidate jailbreak. This approach, more efficient than previous methods, requires fewer queries and can often produce a jailbreak in under twenty queries. PAIR demonstrates success in jailbreaking various LLMs, including GPT-3.5/4 and Vicuna, and is notable for its efficiency and interpretability, making the jailbreaks transferable to other LLMs.

Sophisticated Prompt Engineering Techniques: Perez and Ribeiro (2022) delve into the intricacies of LLMs' prompt processing capabilities. They demonstrate that embedding certain trigger words or phrases within prompts can effectively hijack the model's decision-making process, leading to the overriding of programmed ethical constraints. (Ding et al., 2023) focus on subtle, hard-to-detect jailbreaking methods using nested prompts. These findings reveal a critical shortcoming in the LLMs' content evaluation algorithms, suggesting the need for more complex, context-aware natural language processing that can discern and neutralize manipulative prompt structures.

Cross-Modal and Linguistic Attack Surfaces: Qi et al. (2023a) reveals that LLMs are susceptible to multimodal inputs that combine text with visual cues. This approach takes advantage of the models' less robust processing of non-textual information. Similarly, Yong et al. (2023) exposes the heightened vulnerability of LLMs in processing low-resource languages. This indicates a significant gap in the models' linguistic coverage and comprehension, especially for languages with limited representation in training data. This work demonstrated that by translating unsafe English inputs into low-resource languages, it's possible to circumvent GPT-4's safety safeguards.

Universal and Automated Attack Strategies: The development of universal and automated attack frameworks, as discussed in (Mehrotra et al., 2023) marks a pivotal advancement in jailbreaking techniques. These attacks involve appending specially chosen sequences of characters to a user's query, which can cause the system to provide unfiltered, potentially harmful responses. Shah et al.

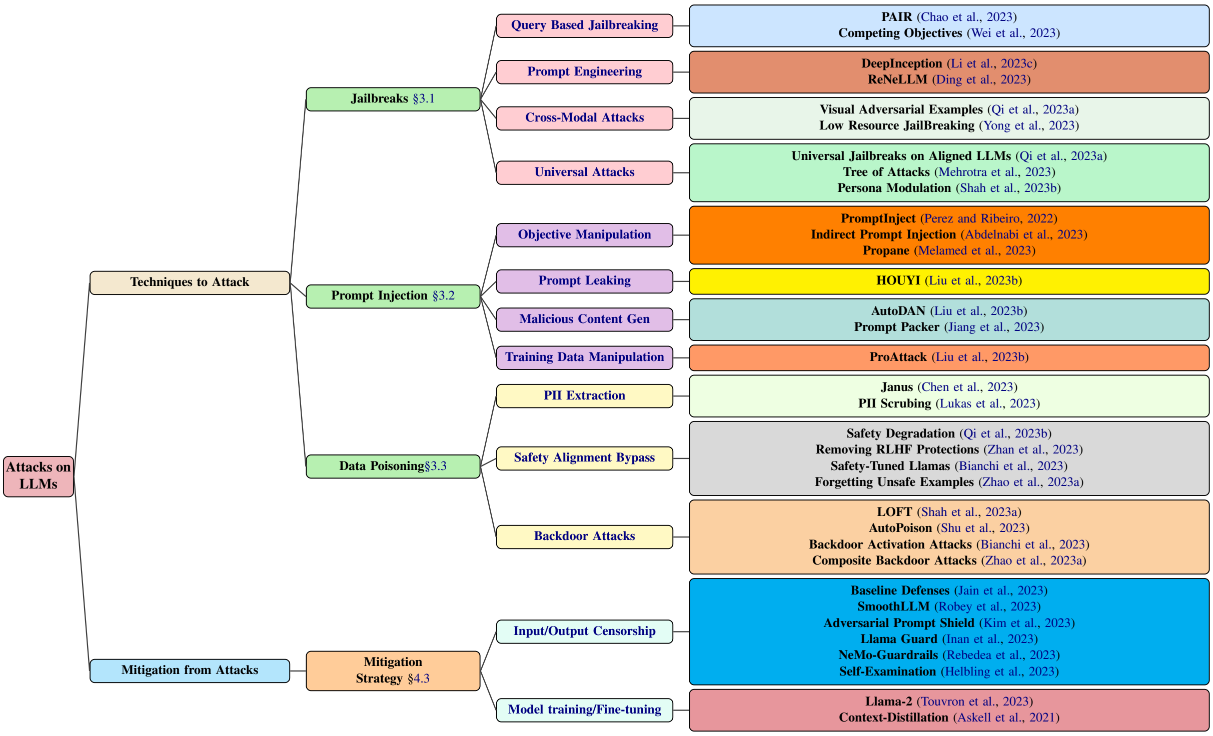
Attacks on LLMs
Techniques to Attack
Mitigation from Attacks
Jailbreaks §3.1
Prompt Injection §3.2
Data Poisoning§3.3
Mitigation Strategy §4.3
Query Based Jailbreaking
Prompt Engineering
Cross-Modal Attacks
Universal Attacks
Objective Manipulation
Prompt Leaking
Malicious Content Gen
Training Data Manipulation
PII Extraction
Safety Alignment Bypass
Backdoor Attacks
Input/Output Censorship
Model training/Fine-tuning
PAIR (Chao et al., 2023)
Competing Objectives (Wei et al., 2023)
DeepInception (Li et al., 2023c)
ReNeLLM (Ding et al., 2023)
Visual Adversarial Examples (Qi et al., 2023a)
Low Resource JailBreaking (Yong et al., 2023)
Universal Jailbreaks on Aligned LLMs (Qi et al., 2023a)
Tree of Attacks (Mehrotra et al., 2023)
Persona Modulation (Shah et al., 2023b)
PromptInject (Perez and Ribeiro, 2022)
Indirect Prompt Injection (Abdelnabi et al., 2023)
Propane (Melamed et al., 2023)
HOUYI (Liu et al., 2023b)
AutoDAN (Liu et al., 2023b)
Prompt Packer (Jiang et al., 2023)
ProAttack (Liu et al., 2023b)
Janus (Chen et al., 2023)
PII Scrubing (Lukas et al., 2023)
Safety Degradation (Qi et al., 2023b)
Removing RLHF Protections (Zhan et al., 2023)
Safety-Tuned Llamas (Bianchi et al., 2023)
Forgetting Unsafe Examples (Zhao et al., 2023a)
LOFT (Shah et al., 2023a)
AutoPoison (Shu et al., 2023)
Backdoor Activation Attacks (Bianchi et al., 2023)
Composite Backdoor Attacks (Zhao et al., 2023a)
Baseline Defenses (Jain et al., 2023)
SmoothLLM (Robey et al., 2023)
Adversarial Prompt Shield (Kim et al., 2023)
Llama Guard (Inan et al., 2023)
NeMo-Guardrails (Rebedea et al., 2023)
Self-Examination (Helbling et al., 2023)
Llama-2 (Touvron et al., 2023)
Context-Distillation (Askell et al., 2021)

Figure 1: Taxonomy of attacks and defences on LLMs. We focus on prevalent methods across subthemes spanning jailbreaks, prompt injections and data poisoning. For Mitigation strategies, we divide papers into training based and moderation based efforts. All branches highlight key works that represent the themes.

(2023b) examine attacks leveraging the persona or style emulation capabilities of LLMs, introducing a new dimension to the attack strategies.

3.2 Prompt Injection

This section outlines attacker strategies to manipulate LLM behavior using carefully designed malicious prompts and organizes the research into seven key areas.

Objective Manipulation: Abdelnabi et al. (2023) demonstrate a prompt injection attack capable of fully compromising LLMs, with practical feasibility showcased on applications like Bing Chat and Github Copilot. Perez and Ribeiro (2022) introduce the PromptInject framework for goal-hijacking attacks, revealing vulnerability to prompt misalignment and offering insights into inhibiting measures such as stop sequences and post-processing model results.

Prompt Leaking: Liu et al. (2023b) addresses security vulnerabilities in Large Language Models like GPT-4, focusing on prompt injection attacks. It introduces the HOUYI methodology, a black-box prompt injection attack approach designed for versatility and adaptability across various LLM-integrated services/applications. HOUYI comprises three phases: Context Inference (interaction with the target application to grasp its inherent context and input-output relationships), Payload Generation (devising a prompt generation plan based on the obtained application context and prompt injection guidelines), and Feedback (gauging the effectiveness of the attack by scrutinizing the LLM's responses to the injected prompts, followed by iterative refinement for optimal outcomes), aiming to trick LLMs into interpreting malicious payloads as questions rather than data payloads. Experiments on 36 real-world LLM-integrated services using HOUYI show an 86.1% success rate in launching attacks, revealing severe ramifications such as unauthorized imitation of services and exploitation of computational power.

Malicious Content Generation: Addressing scalability challenges in malicious prompt generation, Liu et al. (2023a) present AutoDAN, which is designed to preserve meaningfulness and fluency in prompts. They highlight the discovery of prompt injection attacks combined with malicious questions, can lead LLMs to generate harmful or objectionable content by bypassing safety features. Using a hierarchical genetic algorithm tailored for structured discrete data sets AutoDAN apart from existing methods. The initialization of the population is crucial, and the paper employs handcrafted jailbreak prompts identified by LLM users as prototypes to reduce the search space. Different crossover policies for both sentences and words are introduced to avoid falling into local optima and consistently search for the global optimal solution. Implementation details include a multi-point crossover policy based on a roulette selection strategy and a momentum word scoring scheme to enhance search capability in the fine-grained space. The method achieves lower sentence perplexity, indicating more semantically meaningful and stealthy attacks.

Manipulating Training Data: Zhao et al. (2023b) present ProAttack, which boasts near-perfect success rates in evading defenses, highlighting the urgency for better handling of prompt injection attacks with LLMs' growing application.

Prompt Injection Attacks and Defenses in LLM-Integrated Applications: Comprehensive studies such as (Liu et al., 2023e) emphasize the importance of understanding and mitigating the risks posed by prompt injection attacks. These works highlight sophisticated methodologies like 'HouYi' (Liu et al., 2023e) and underscore the urgent need for more robust security measures.

Prompt Manipulation Frameworks: Recent literature explores various methods for manipulating LLM behavior, as detailed in works like (Melamed et al., 2023; Jiang et al., 2023). Propane (Melamed et al., 2023) introduces an automatic prompt optimization framework, while Prompt Packer (Jiang et al., 2023) introduces Compositional Instruction Attacks, revealing vulnerabilities in LLMs to multifaceted attacks.

Benchmarking and Analyzing LLM Prompt Injection Attacks: Toyer et al. (2023) present a dataset of prompt injection attacks and defenses, offering insights into LLM vulnerabilities and paving the way for more resilient systems. This benchmarking and analysis are crucial for understanding the intricacies of prompt injection attacks and developing effective countermeasures.

3.3 Data Poisoning

Contemporary NLP systems follow a two-stage process: pretraining and fine-tuning. Pretraining involves learning from a large corpus to understand general linguistic structures, while fine-tuning tailors the model for specific tasks using smaller datasets. Recently, providers like OpenAI have enabled end-users to fine-tune models, enhancing adaptability. This section explores studies on data poisoning techniques and their impact on safety aspects during training, including privacy risks and susceptibility to adversarial attacks.

PII extraction: Chen et al. (2023) investigate whether fine-tuning large language models (LLMs) on small datasets containing personal identifiable information (PII) can lead to the models disclosing more PII embedded in their original training data. The authors demonstrate a strawman method where an LLM is fine-tuned on a small PII dataset converted to text, which enables the model to then disclose more PII when prompted. To improve on this, they propose Janus methodology which defines a PII recovery task and uses few-shot fine-tuning. Experiments indicate that fine-tuning GPT-3.5 on just 10 PII instances enables it to accurately disclose 650 out of 1000 target PIIs, versus 0 without fine-tuning. The Janus method further improves this divulgence, disclosing 699 target PIIs. Analysis shows larger models and real training data have stronger memorization and PII recovery and fine-tuning is more effective than prompt engineering alone for PII leakage. This indicates that LLMs can shift from non-disclosure to revealing significant amounts of PII with minimal fine-tuning.

Bypassing Safety Alignment: Qi et al. (2023b) investigate safety risks in fine-tuning aligned LLMs, finding that even benign datasets can compromise safety. Backdoor attacks are shown to effectively bypass safety measures, emphasizing the need for improved post-training protections.

Bianchi et al. (2023) analyze the safety risks of instruction tuning, showing that overly instruction-tuned models can still produce harmful content. They propose a safety tuning dataset to mitigate these risks, balancing safety and model performance.

Zhao et al. (2023a) study how LLMs learn and forget unsafe examples during fine-tuning, proposing a technique called ForgetFilter to filter fine-tuning data and improve safety without sacrificing performance.

Backdoor Attacks: Shah et al. (2023a) introduce Local Fine Tuning (LoFT) for discovering adversarial prompts, demonstrating successful attacks on LLMs. Shu et al. (2023) propose Autopoison, an automated data poisoning pipeline, showcasing its effectiveness in altering model behavior without semantic degradation.

4 Human Interference

Adversarial attacks range from human-crafted (slow and non-scalable) to automated methods utilizing search, target function optimization, LLMs, and tools.

4.1 Human Red Teaming

Through human-crafted adversarial prompts, individuals employ their creativity and expertise to design attacks carefully. These attacks often involve a deep understanding of the targeted model's vulnerabilities and limitations.

In a study performed by Huang et al. (2023), 600 curated harmful prompts were tested over 11 LLMs. By simply varying decoding hyperparameters and sampling methods, they show these curated prompts can easily break LLMs. Shen et al. (2023a) collect 6,387 malicious prompts and test them over 13 forbidden scenarios from OpenAI's policy. These were collected through various online sources like reddit, discords, datasets and other public places on the web. They found 2 highly effective prompts that have 99% attack success on GPT-3.5 and GPT-4.

Li et al. (2023b) collected personally identifiable information, like emails and phone numbers, to test if they could extract this data from LLMs. They crafted a multi-step jailbreaking role-playing prompting approach that a human attacker can use to break ChatGPT's ethical constraints and extract private data. The website

The online platform (Jai) is an active website for gathering jailbreaking prompts through crowdsourcing. Another study conducted by Liu et al. (2023c) utilized this website to analyze 78 malicious prompts. These prompts were categorized into three main classes: Pretending, Attention Shifting, and Privilege Escalation, each further divided into subclasses. In total, they created 10 categories encompassing various types of harmful prompts that broke over 10 OpenAI policies. Other sources of curated adversarial prompts have also surfaced over the web (Shen et al., 2023b; Jai).

Creating interactive systems to facilitate adversarial sample generation is another way to get human expertise into breaking LLMs. Wallace et al. (2019) created an interactive UI for leveraging human creativity and trivia knowledge to generate adversarial examples for Question Answering systems. The authors build an interactive interface that shows question authors model predictions and word importance scores. The authors are trivia enthusiasts who craft tricky questions that fool the model. A similar large scale study (Schulhoff et al., 2023) collect over 600k adversarial prompts from thousands of participants worldwide through an interactive interface. In another work Ziegler et al. (2022) leveraged human contractors who manually wrote adversarial text snippets that could fool a injurious/non-injurious text classifier. They built an interface to help contractors rewrite snippets to be adversarial, including highlighting salient tokens and suggesting token replacements.

Xu et al. (2021) introduce Bot-Adversarial Dialogue, a human-and-model-in-the-loop framework for enhancing conversational AI safety. Crowd workers converse with chatbots to elicit unsafe/offensive responses, categorized by severity. A verification task identifies offensive language types, involving humans in both collecting and labeling adversarial examples for safety and offensiveness type.

4.2 Automated Adversarial Attacks

Automated adversarial attacks use algorithms to generate and deploy adversarial examples, offering scalability without human expertise.

Deng et al. (2023) propose the "MASTERKEY framework", which uses time-based characteristics inherent to the generative process to reverse-engineer the defense strategies behind mainstream LLM chatbot services. They automatically generate jailbreak prompts against well-protected LLMs by fine-tuning another LLM with the jailbreak prompts. Zou et al. (2023) propose a universal automated approach for adversarial attacks on LLMs. It involves generating a suffix to be added to various queries, prompting the LLM to produce inappropriate content. This method merges greedy and gradient-based search techniques to automatically create these adversarial suffixes. The adversarial prompts produced by this method are highly transferable, even to black-box, publicly available, production LLMs.

AutoDAN (Liu et al., 2023a), an automated, interpretable, gradient-based adversarial attack method for LLMs, combines the strengths of manual jailbreak attacks and automatic adversarial attacks. It generates readable prompts that bypass perplexity filters while maintaining high attack success rates. It formulates the attack as an optimization problem and employs a hierarchical genetic algorithm to search for effective prompts in the space initialized by handcrafted prompts. Their method operates at multiple levels - sentence and word - to ensure both diversity and fine-grained optimization.

Jones et al. (2023) present ARCA, a coordinate ascent discrete optimization algorithm efficiently searching for input output text pairs matching a desired behavior in LLMs. It uncovers unexpected behaviors like derogatory completions or language-switching inputs. Several tools to automatically generate adversarial samples for LLMs exist. PromptAttack (Xu et al., 2023), a tool for evaluating the adversarial robustness of LLMs, converts adversarial textual attacks into an attack prompt that causes the LLM to output an adversarial sample, essentially fooling itself. The attack prompt consists of the original input, the attack objective, and the attack guidance.

Casper et al. (2023) present a red-teaming framework for LLMs, starting with output exploration via clustering, establishing undesired behaviors through classifier training, and using reinforcement learning to train a "red" model generating adversarial prompts, focusing on controversial topics. They successfully red-team GPT-2 for toxic text and GPT-3 for false claims, particularly in controversial political contexts, demonstrating more impactful attacks than traditional methods.

4.3 Mitigation Strategies

Mitigation strategies for protecting LLMs can be broadly divided into two categories based on defense deployment strategy.

4.3.1 External: Input/Output filtering or Guarding

In guarding-based mitigation for LLMs, external systems play a crucial role by detecting adversarial inputs (input filtering) or anomalous outputs (output filtering), negating the need for model retraining. Popular tools like OpenChatKit[1] and NeMo-Guardrails Rebedea et al. (2023) exemplify this approach, and have been adopted by a number of production-LLM systems. Guarding techniques can further be bifurcated into defenses against gradient-based jailbreaks that employ adversarial suffixes to augment prompts, and manual jailbreaks aiming to misalign the model's responses.

Defense against gradient-based jailbreaks: The current state-of-the-art literature in the area of mitigating gradient-based adversarial attacks on LLMs can be broadly categorized into two main strategies: one focusing on detecting malicious prompts based on characteristic features of the input (e.g., high perplexity, character-level perturbations) and the other utilizing classifier-based approaches where models, such as DistilBERT(Sanh et al., 2019), are employed to distinguish between adversarial and non-adversarial prompts.

In the former category, Jain et al. (2023) discuss baseline defenses like input filtering, which, despite their effectiveness, may inadvertently alter the intended output through techniques such as paraphrasing and retokenization, or flag legitimate queries due to perplexity-based filtering. Similarly, Robey et al. (2023) introduce SmoothLLM, which leverages the vulnerability of adversarial attacks to character-level perturbations, adopting a scatter-gather approach for prompt processing. This method aims to nullify adversarial content by averaging out the final response based on the aggregated responses produced by the model for the perturbed input prompts. Similarly, Hu et al. (2023) propose token-level adversarial prompt detection, capitalizing on the high perplexity characteristic of adversarial prompts to identify and classify adversarial tokens within a prompt, leveraging the relationship between neighbouring tokens. As with other perplexity-based techniques, this might not be feasible for black-box LLMs where perplexity calculation cannot be done directly.

On the classifier-based side, Kim et al. (2023) propose the Adversarial Prompt Shield (APS), a DistilBERT(Sanh et al., 2019)-based model designed for prompt classification into safe or unsafe categories. This approach is complemented by a method for generating training data that simulates adversarial attacks by adding synthetic noise to legitimate conversations. However, the necessity for frequent retraining to stay abreast of new attack vectors and reduce false positives presents a challenge to this approach.

The characteristic feature-based methods provide a more direct approach to detecting adversarial content, potentially allowing for real-time mitigation without the need for extensive retraining.

[1] https://github.com/togethercomputer/OpenChatKit

Conversely, classifier-based approaches, while requiring more maintenance, offer a more nuanced understanding of the intricacies of adversarial and non-adversarial prompts, potentially leading to more accurate and robust defenses against a wider range of attacks.

Defense against manual jailbreaks: Inan et al. (2023) introduce Llama Guard, a safeguard model leveraging Llama2-7b Touvron et al. (2023) for input-output protection in LLMs. It employs taxonomy-based task classification for customizing responses through few-shot prompting or fine-tuning. Rebedea et al. (2023) present NeMo-Guardrails, an open-source framework enhancing LLM conversational systems with programmable guardrails. It uses a proxy layer with Colang-defined rules to manage user interactions, though its reliance on chain-of-thought (CoT) prompting may limit scalability. Helbling et al. (2023) propose a similar approach and suggest an output filtering method involving a secondary LLM to assess the malicious nature of responses, facing challenges in language compatibility and operational costs.

Glukhov et al. (2023) argue that semantic censorship in LLMs is inherently undecidable, given their ability to follow instructions and generate outputs through arbitrary rule-based encodings. They propose viewing LLM censorship as a security issue, necessitating specific countermeasures rather than treating it solely as a machine learning challenge.

4.3.2 Internal: Model training/fine-tuning

The state of the art methods in this differ primarily in the stage at which the model is trained for providing safe outputs, as well as the source of the data used for providing the safe output. In this section, we highlight the current trends.

Supervised Safety fine-tuning: Touvron et al. (2023), collect adversarial prompts along with their safe demonstrations and then use these samples as a part of the general supervised fine tuning pipeline. While the examples in this case are curated manually, automated collection techniques and red-teaming are an effective methods to discover harmful prompts. A detailed discussion of red-teaming and collecting data both manually and automatically is discussed in section 4.1.

Safety-tuning as a part of the RLHF pipeline: RLHF has been shown to make models more robust to jailbreak attempts Bai et al. (2022). Touvron et al. (2023) train a safety reward model based on manually collected adversarial prompts and responses from multiple models where the response that is deemed the safest is selected, this reward model is then used as a part of RLHF pipeline in order to safety-tune the model.

Safety Context Distillation: In using Context Distillation Askell et al. (2021) for model safety, Touvron et al. (2023) prepend the prompt with a persona of a safe model such as “You are a responsible and safe assistant,” and then while fine-tuning, they remove this prepended prompt, distilling this safe context into the model, enhancing its proclivity to deny any requests that create a problematic response.

5 Challenges and Future Research

Here, we discuss a few potential directions that are promising for future research on defending the attacks on LLMs, enhancing their robustness, and gaining trust from the end-users.

5.1 Real-time Monitoring Systems

The growing use of Large Language Models (LLMs) in diverse fields brings various applications, but it requires robust monitoring to detect anomalies effectively. Current evaluation mechanisms are inadequate, leaving LLMs vulnerable to threats like data exposure, misinformation, illegal content, and aiding criminal activities. Understanding and countering these attacks are challenging due to adversaries’ ability to manipulate LLMs with deceptive prompts. Therefore, it is imperative to not only introduce LLM safeguard systems but also to fortify them with advanced detection capabilities. Future research can focus on building such systems, equipped to scrutinize outputs comprehensively, identifying and flagging any undesirable content swiftly and accurately. Additionally, efforts should be directed towards ensuring the resilience and adaptability of these guard mechanisms, making them resistant to potential evasion tactics employed by adversaries.

5.2 Multimodal Approach

The integration of multimodal capabilities presents both exciting opportunities and formidable challenges for ensuring the safety and reliability of LLMs. Future research should prioritize developing techniques to mitigate these challenges, such as improving input sanitization and validation, and creating custom defense prompts to prevent jailbreaking attempts. These efforts are crucial for strengthening the security and resilience of LLMs amidst evolving threats in multimodal environments.

5.3 Benchmark

It becomes apparent that safeguarding LLMs alone falls short of addressing the broader concerns. Hence, the pertinent inquiry emerges: How can we reliably determine the comparative efficacy of attacks 'A' versus 'B' on LLMs through quantifiable and rational observations? The establishment of a standardized benchmark for evaluating attacks on LLMs becomes important, ensuring ethical reliability and factual performance. While considerable research has been devoted to benchmarking (Jin et al., 2024), the existing frameworks often prove insufficient for practical deployment in real-world scenarios. Consequently, the development of a scalable, near-real-time evaluation infrastructure emerges as a crucial requirement for both LLMs and their enterprise applications.

5.4 Explainable LLMs

Explainability of LLMs is pivotal, not just for enhancing the transparency and trustworthiness of these models but also for identifying and mitigating vulnerabilities to linguistic attacks. Future research in explainable LLMs must pivot towards developing and refining methods that illuminate the complex decision-making processes inherent within these models. This entails a focused investigation into explainability techniques that unravel the intricacies of attention mechanisms, delineates the significance of features contributing to the models' outputs, and trace the reasoning pathways that underpin their decisions. Such efforts are critical for enabling a deeper understanding and interpretation of LLM outputs by a broad spectrum of stakeholders, from developers to end-users. There are existing work (Chefer et al., 2021; Voita et al., 2019; Dosovitskiy et al., 2020) that try to explain transformer architecture outputs but because of the black box nature of neural networks, they fall short to give reliable explanations and leave room for further fundamental developments. Moreover, the endeavor to make LLMs explainable presents multifaceted challenges, including the technical difficulty of dissecting often opaque neural network architectures, the need for methodologies that can reliably attribute decision-making in a manner that is both accurate and accessible to non-experts, and the ethical implications of creating transparent systems that respect user privacy and data security. Addressing these challenges requires a multidisciplinary approach that bridges computational techniques with principles of ethical AI, aiming to foster models that are not only robust and efficient but also intrinsically interpretable and aligned with societal values. This push towards explainable LLMs is not just a technical necessity but a foundational step towards ensuring that AI technologies remain accountable, understandable, and beneficial across diverse applications.

6 Conclusion

This paper provides a comprehensive overview of attacks targeting LLMs. We start by categorizing the LLM attacks literature into a novel taxonomy to provide a better structure and aid for future research. Through the examination of these attack vectors, it is evident that LLMs are vulnerable to a diverse range of threats, posing significant challenges to their security and reliability in real-world applications. Furthermore, this paper has highlighted the importance of implementing effective mitigation strategies to defend against LLM attacks. These strategies encompass a variety of approaches, including data filtering, guardrails, robust training techniques, adversarial training and safety context distillation. To summarize, although LLMs present significant opportunities for enhancing natural language processing capabilities, their vulnerability to adversarial exploitation highlights the critical need to address security issues. Through ongoing exploration and advancement in detecting attacks, implementing mitigative measures, and enhancing model resilience, we can aim to fully leverage the advantages of LLM technology while fortifying defenses against potential risks.

7 Limitations

This study, while comprehensive in its examination of attacks on Large Language Models (LLMs) and mitigation strategies, is subject to several limitations:

Scope and Coverage: Despite our efforts to conduct a thorough survey, the fast-paced advancements in LLM technologies and attack methodologies mean that some emerging threats might not be covered. The landscape of cybersecurity threats evolves rapidly, and new vulnerabilities could emerge following this publication.

Generalizability of Mitigation Strategies: The effectiveness of the mitigation strategies discussed may vary across different models, contexts, and against specific attacks. While we aimed for broad applicability in our recommendations, the specificity of certain defenses to particular models or scenarios limits their universal applicability.

Ethical and Societal Implications: Our focus was primarily on the technical aspects of LLM security, which led to a less comprehensive exploration of the broader ethical and societal implications of both the attacks and the countermeasures. The dual-use nature of many AI technologies, including those discussed, necessitates careful consideration of ethical implications beyond the scope of this paper.

Dynamic Nature of Threats: The adversarial landscape is characterized by an ongoing race, with attackers continually evolving their strategies in response to new defenses. This paper captures a snapshot of the current state, but continuous research and vigilance are required to address the adaptive nature of threats.

Scalability and Practicality of Defenses: Implementing robust defense mechanisms in practical settings poses challenges, including computational overhead, scalability issues, and the need for ongoing updates. Balancing security with usability remains a critical, yet underexplored, area.

In summary, while this work provides significant insights into LLM security, it highlights the importance of continued research, interdisciplinary collaboration, and an agile response to the complex and evolving landscape of AI security.

References

Jailbreakchat. https://www.jailbreakchat.com/.

Sahar Abdelnabi, Kai Greshake, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. 2023. Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, pages 79–90.

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

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Jackson Kernion, Kamal Ndousse, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, and Jared Kaplan. 2021. A general language assistant as a laboratory for alignment.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback.

Federico Bianchi, Mirac Suzgun, Giuseppe Attanasio, Paul Röttger, Dan Jurafsky, Tatsunori Hashimoto, and James Zou. 2023. Safety-tuned llamas: Lessons from improving the safety of large language models that follow instructions. *arXiv preprint arXiv:2309.07875*.

Stephen Casper, Jason Lin, Joe Kwon, Gatlen Culp, and Dylan Hadfield-Menell. 2023. Explore, establish, exploit: Red teaming language models from scratch. *arXiv preprint arXiv:2306.09442*.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. 2023. Jailbreaking black box large language models in twenty queries. *arXiv preprint arXiv:2310.08419*.

Hila Chefer, Shir Gur, and Lior Wolf. 2021. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 782–791.

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

Gelei Deng, Yi Liu, Yuekang Li, Kailong Wang, Ying Zhang, Zefeng Li, Haoyu Wang, Tianwei Zhang, and Yang Liu. 2023. Jailbreaker: Automated jailbreak across multiple large language model chatbots. *arXiv preprint arXiv:2307.08715*.

Peng Ding, Jun Kuang, Dan Ma, Xuezhi Cao, Yunsen Xian, Jiajun Chen, and Shujian Huang. 2023.

A wolf in sheep's clothing: Generalized nested jailbreak prompts can fool large language models easily. *arXiv preprint arXiv:2311.08268*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Tyna Eloundou, Sam Manning, Pamela Mishkin, and Daniel Rock. 2023. Gpts are gpts: An early look at the labor market impact potential of large language models. *arXiv preprint arXiv:2303.10130*.

David Glukhov, Ilia Shumailov, Yarin Gal, Nicolas Papernot, and Vardan Papyan. 2023. Llm censorship: A machine learning challenge or a computer security problem? *arXiv preprint arXiv:2307.10719*.

Alec Helbling, Mansi Phute, Matthew Hull, and Duen Horng Chau. 2023. Llm self defense: By self examination, llms know they are being tricked. *arXiv preprint arXiv:2308.07308*.

Dan Hendrycks, Mantas Mazeika, and Thomas Woodside. 2023. An overview of catastrophic ai risks. *arXiv preprint arXiv:2306.12001*.

Zhengmian Hu, Gang Wu, Saayan Mitra, Ruiyi Zhang, Tong Sun, Heng Huang, and Viswanathan Swaminathan. 2023. Token-level adversarial prompt detection based on perplexity measures and contextual information.

Yangsibo Huang, Samyak Gupta, Mengzhou Xia, Kai Li, and Danqi Chen. 2023. Catastrophic jailbreak of open-source llms via exploiting generation. *arXiv preprint arXiv:2310.06987*.

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, et al. 2023. Llama guard: Llm-based input-output safeguard for human-ai conversations. *arXiv preprint arXiv:2312.06674*.

Neel Jain, Avi Schwarzschild, Yuxin Wen, Gowthami Somepalli, John Kirchenbauer, Ping yeh Chiang, Micah Goldblum, Aniruddha Saha, Jonas Geiping, and Tom Goldstein. 2023. Baseline defenses for adversarial attacks against aligned language models.

Shuyu Jiang, Xingshu Chen, and Rui Tang. 2023. Prompt packer: Deceiving llms through compositional instruction with hidden attacks. *arXiv preprint arXiv:2310.10077*.

Mingyu Jin, Suiyuan Zhu, Beichen Wang, Zihao Zhou, Chong Zhang, Yongfeng Zhang, et al. 2024. Attackeval: How to evaluate the effectiveness of jailbreak attacking on large language models. *arXiv preprint arXiv:2401.09002*.

Erik Jones, Anca Dragan, Aditi Raghunathan, and Jacob Steinhardt. 2023. Automatically auditing large language models via discrete optimization. *arXiv preprint arXiv:2303.04381*.

Jinhwa Kim, Ali Derakhshan, and Ian G. Harris. 2023. Robust safety classifier for large language models: Adversarial prompt shield.

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

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

Xuan Li, Zhanke Zhou, Jianing Zhu, Jiangchao Yao, Tongliang Liu, and Bo Han. 2023c. Deepinception: Hypnotize large language model to be jailbreaker. *arXiv preprint arXiv:2311.03191*.

Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. 2023a. Autodan: Generating stealthy jailbreak prompts on aligned large language models. *arXiv preprint arXiv:2310.04451*.

Yi Liu, Gelei Deng, Yuekang Li, Kailong Wang, Tianwei Zhang, Yepang Liu, Haoyu Wang, Yan Zheng, and Yang Liu. 2023b. Prompt injection attack against llm-integrated applications. *arXiv preprint arXiv:2306.05499*.

Yi Liu, Gelei Deng, Zhengzi Xu, Yuekang Li, Yaowen Zheng, Ying Zhang, Lida Zhao, Tianwei Zhang, and Yang Liu. 2023c. Jailbreaking chatgpt via prompt engineering: An empirical study. *arXiv preprint arXiv:2305.13860*.

Yiheng Liu, Tianle Han, Siyuan Ma, Jiayue Zhang, Yuanyuan Yang, Jiaming Tian, Hao He, Antong Li, Mengshen He, Zhengliang Liu, et al. 2023d. Summary of chatgpt-related research and perspective towards the future of large language models. *Meta-Radiology*, page 100017.

Yupei Liu, Yuqi Jia, Runpeng Geng, Jinyuan Jia, and Neil Zhenqiang Gong. 2023e. Prompt injection attacks and defenses in llm-integrated applications.

Nils Lukas, Ahmed Salem, Robert Sim, Shruti Tople, Lukas Wutschitz, and Santiago Zanella-Béguelin. 2023. Analyzing leakage of personally identifiable information in language models. *arXiv preprint arXiv:2302.00539*.

Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron Singer, and Amin Karbasi. 2023. Tree of attacks: Jailbreaking black-box llms automatically. *arXiv preprint arXiv:2312.02119*.

Rimon Melamed, Lucas H McCabe, Tanay Wakhare, Yejin Kim, H Howie Huang, and Enric Boix-Adsera. 2023. Propane: Prompt design as an inverse problem. *arXiv preprint arXiv:2311.07064*.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Nick Barnes, and Ajmal Mian. 2023. A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*.

Fábio Perez and Ian Ribeiro. 2022. Ignore previous prompt: Attack techniques for language models. *ML Safety Workshop NeurIPS*.

Xiangyu Qi, Kaixuan Huang, Ashwinee Panda, Mengdi Wang, and Prateek Mittal. 2023a. Visual adversarial examples jailbreak large language models. *arXiv preprint arXiv:2306.13213*.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. 2023b. Fine-tuning aligned language models compromises safety, even when users do not intend to! *arXiv preprint arXiv:2310.03693*.

Traian Rebedea, Razvan Dinu, Makesh Sreedhar, Christopher Parisien, and Jonathan Cohen. 2023. Nemo guardrails: A toolkit for controllable and safe llm applications with programmable rails. *arXiv preprint arXiv:2310.10501*.

Alexander Robey, Eric Wong, Hamed Hassani, and George J. Pappas. 2023. Smoothllm: Defending large language models against jailbreaking attacks.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Sander Schulhoff, Jeremy Pinto, Anaum Khan, Louis-François Bouchard, Chenglei Si, Svetlina Anati, Valen Tagliabue, Anson Liu Kost, Christopher Carnahan, and Jordan Boyd-Graber. 2023. Ignore this title and hackaprompt: Exposing systemic vulnerabilities of llms through a global scale prompt hacking competition. *arXiv preprint arXiv:2311.16119*.

Muhammad Ahmed Shah, Roshan Sharma, Hira Dhamyal, Raphael Olivier, Ankit Shah, Dareen Alharthi, Hazim T Bukhari, Massa Baali, Soham Deshmukh, Michael Kuhlmann, et al. 2023a. Loft: Local proxy fine-tuning for improving transferability of adversarial attacks against large language model. *arXiv preprint arXiv:2310.04445*.

Rusheb Shah, Soroush Pour, Arush Tagade, Stephen Casper, Javier Rando, et al. 2023b. Scalable and transferable black-box jailbreaks for language models via persona modulation. *arXiv preprint arXiv:2311.03348*.

Erfan Shayegani, Md Abdullah Al Mamun, Yu Fu, Pedram Zaree, Yue Dong, and Nael Abu-Ghazaleh. 2023. Survey of vulnerabilities in large language models revealed by adversarial attacks. *arXiv preprint arXiv:2310.10844*.

Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. 2023a. " do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. *arXiv preprint arXiv:2308.03825*.

Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. 2023b. "Do Anything Now": Characterizing and Evaluating In-The-Wild Jailbreak Prompts on Large Language Models. *CoRR abs/2308.03825*.

Manli Shu, Jiongxiao Wang, Chen Zhu, Jonas Geiping, Chaowei Xiao, and Tom Goldstein. 2023. On the exploitability of instruction tuning. *arXiv preprint arXiv:2306.17194*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Sam Toyer, Olivia Watkins, Ethan Adrian Mendes, Justin Svegliato, Luke Bailey, Tiffany Wang, Isaac Ong, Karim Elmaaroufi, Pieter Abbeel, Trevor Darrell, et al. 2023. Tensor trust: Interpretable prompt injection attacks from an online game. *arXiv preprint arXiv:2311.01011*.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv preprint arXiv:1905.09418*.

Eric Wallace, Pedro Rodriguez, Shi Feng, Ikuya Yamada, and Jordan Boyd-Graber. 2019. Trick me if you can: Human-in-the-loop generation of adversarial examples for question answering. *Transactions of the Association for Computational Linguistics*, 7:387–401.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. 2023. Jailbroken: How does llm safety training fail? *arXiv preprint arXiv:2307.02483*.

Jing Xu, Da Ju, Margaret Li, Y-Lan Boureau, Jason Weston, and Emily Dinan. 2021. Bot-adversarial dialogue for safe conversational agents. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2950–2968.

Xilie Xu, Keyi Kong, Ning Liu, Lizhen Cui, Di Wang, Jingfeng Zhang, and Mohan Kankanhalli. 2023. An llm can fool itself: A prompt-based adversarial attack. *arXiv preprint arXiv:2310.13345*.

Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Eric Sun, and Yue Zhang. 2023. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *arXiv preprint arXiv:2312.02003*.

Zheng-Xin Yong, Cristina Menghini, and Stephen H Bach. 2023. Low-resource languages jailbreak gpt-4. *arXiv preprint arXiv:2310.02446*.

Qiusi Zhan, Richard Fang, Rohan Bindu, Akul Gupta, Tatsunori Hashimoto, and Daniel Kang. 2023. Removing rlhf protections in gpt-4 via fine-tuning. *arXiv preprint arXiv:2311.05553*.

Jiachen Zhao, Zhun Deng, David Madras, James Zou, and Mengye Ren. 2023a. Learning and forgetting unsafe examples in large language models. *arXiv preprint arXiv:2312.12736*.

Shuai Zhao, Jinming Wen, Luu Anh Tuan, Junbo Zhao, and Jie Fu. 2023b. Prompt as triggers for backdoor attack examining the vulnerability in language models. *arXiv preprint arXiv2305.01219*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023c. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Daniel Ziegler, Seraphina Nix, Lawrence Chan, Tim Bauman, Peter Schmidt-Nielsen, Tao Lin, Adam Scherlis, Noa Nabeshima, Benjamin Weinstein-Raun, Daniel de Haas, et al. 2022. Adversarial training for high-stakes reliability. *Advances in Neural Information Processing Systems*, 35:9274–9286.

Andy Zou, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models.

A Appendix

Paper Name	Category	Threat Model			Attack Strategy		Evaluated LLM Models
		White Box	Gray Box	Black Box	Prompt or Response	Model Based	
(Chao et al., 2023)	Jailbreak			✓	✓		GPT-3.5/4, Vicuna, and PaLM-2
(Wei et al., 2023)	Jailbreak			✓	✓		GPT-4, GPT-3.5 Turbo, Claude v1.3
(Li et al., 2023c)	Jailbreak			✓	✓	✓	Falcon, Vicuna, Llama-2, GPT-3.5, GPT-4, GPT-4V
(Ding et al., 2023)	Jailbreak			✓	✓	✓	ChatGPT, GPT-4
(Qi et al., 2023a)	Jailbreak	✓			✓	✓	MiniGPT-4, BLIP-2, GPT-4
(Yong et al., 2023)	Jailbreak			✓	✓		GPT-4
(Zou et al., 2023)	Jailbreak	✓		✓	✓	✓	Vicuna, ChatGPT, Claude, Llama-2, Pythia, Falcon
(Mehrotra et al., 2023)	Jailbreak			✓	✓	✓	GPT-4, GPT-4 Turbo
(Abdelnabi et al., 2023)	Prompt Injection		✓	✓	✓		GPT-3.5, GPT-4
(Perez and Ribeiro, 2022)	Prompt Injection			✓	✓		GPT-3.5
(Zhao et al., 2023b)	Prompt Injection	✓		✓		✓	GPT-NEO
(Liu et al., 2023e)	Prompt Injection		✓	✓	✓	✓	GPT-3.5
(Toyer et al., 2023)	Prompt Injection		✓		✓		Llama-2 (7B, 13B, 70B), CodeLaMMA-34B
(Melamed et al., 2023)	Prompt Injection			✓			GPT Model suits: Pythia
(Jiang et al., 2023)	Prompt Injection			✓	✓		GPT-4, GPT-3.5, and ChatGLM2-6B
(Chen et al., 2023)	Data Poisoning			✓	✓		GPT-3.5
(Lukas et al., 2023)	Data Poisoning		✓	✓	✓		GPT-3.5
(Qi et al., 2023b)	Data Poisoning	✓		✓		✓	GPT-3.5 Turbo, Llama-2
(Zhan et al., 2023)	Data Poisoning		✓	✓	✓		GPT-4
(Bianchi et al., 2023)	Data Poisoning		✓		✓		LLaMA, Falcon
(Zhao et al., 2023a)	Data Poisoning			✓			LLaMA 7B
(Shah et al., 2023a)	Data Poisoning			✓	✓		ChatGPT, GPT-4, and Claude
(Shu et al., 2023)	Data Poisoning			✓	✓		OPT (350M, 1.3B, 6.7B)

Table 1: A comprehensive summary detailing attacks targeting LLMs is provided, categorized into three primary categories: Jailbreak, Prompt Injection, and Data Poisoning. We outline the threat model, attack strategy, and the list of evaluated LLM models for each of the papers listed.