Protecting Large Language Models from malicious prompts

Victor Mazanov, Artyom Grishin, Oleg Shchendrigin

Innopolis University

July 4, 2024

I Introduction

The development of AI is occurring at an accelerated pace on a daily basis, with a growing emphasis on the personalization of AI and the globalization of this technology. Large Language Model (LLM) is an AI algorithm that uses neural network methods with many parameters to process text, generate pictures, write code, etc. However, there are inherent risks associated with the future, as LLM systems are currently susceptible to hacking by users through various methods and techniques. This vulnerability has been highlighted in recent studies [1], [2]. Consequently, the initial challenge is to develop strategies to secure LLM systems from being overused and to make the models safe for use. The issue of jailbreaking LLM has persisted since it is exceedingly challenging to provide clear instructions (pipeline) on how to prevent LLM from issuing responses containing malicious content. The objective of our survey is to investigate a defensive strategy against attempts to outsmart the model by forcing it to produce a result containing content that should not be seen.

II Literature Review

The subject of AI security is a matter of considerable concern. As trustworthy AI should be ethical, lawful, and robust [3], it is essential to guarantee that the LLM's responses are controllable and do not cause harm to people. It is of the utmost importance to address the safety and ethical issues that may arise during the pursuit of an LLM degree. By taking these precautions, it is possible to minimize the potential for harm and to fully exploit the benefits of this qualification [4]. Gupta et al. [1] demonstrate the weaknesses of ChatGPT, showing successful attacks on it with Jailbreak methods and reverse psychology. Nevertheless, model design can also be a cause of privacy leakage in neural networks [5].

A. Ways to attack AI

Attackers can manage the large language models using jailbreaks to breakout security actions, reply to unlabeled or controversial requests, and gain access to confidential data [6]. In addition, adversaries employ prompt injections to retrieve up to 67% of the training data,

which contains sensitive information. Furthermore, Wu et al. [7] mention two main types of model attacks. Prompt injections disorient the LLM and cause it to perform unintentional actions, such as returning personal information. LLMs are susceptible to assaults in which an intruder can penetrate the preparation information pipeline and infuse poisoning data. With a training data poisoning attack, an attacker can circumvent the fine-tuning phase of the LLM and exploit the model's inherent weaknesses. In a poisoning attack, an infiltrator corrupts the training data by inserting pre-composed false data samples that disrupt the entire training process. These samples can compromise the final accuracy or cause the model to make incorrect predictions altogether [8], [9], [10].

On the other hand, Kuzlu et al. [11] demonstrated that AI can result in the disclosure of private information in IoT input attacks. In model poisoning, intruders alter the original model with one that has been pre-trained on poisoned data. This allows attackers to obtain private information and cause damage to the company. Membership Inference Attacks (MIA) are useful to steal user data based on leaked model databases. In that instance, the attackers are able to obtain some data about specific users, but the sources contain only partial information. This information can be exploited by brute force attacks, such as data resetting and password spraying [9], [12]. Model Extraction Attacks (MEA) represent a particular sort of assault where an intruder tries to duplicate the machine learning model for subsequent remote deployment and testing. This is done with the intention of identifying vulnerabilities and exploiting them in the original LLM [9], [12].

Eventually, intruders use many methods to attack LLM's, such as jailbreaks, prompt injections, poisoning data attacks, and model poisoning.

B. Ways to protect AI

Researchers [13] propose several methods for combating adversarial attacks. The first one is adversarial training, which involves training the model on both original and adversarial data. The second is input reconstruction, which is effective against black box attacks but less so against gray box attacks. Its objective is to clean up the input. Furthermore, Yao et al. [2] show three strategies: The first one, instruction processing, aims to pre-process the prompt from the user in order to "clean" dangerous places, thereby preventing attempts to attack the model. One option is to mask the user's prompt and then have other LLMs

rebuild it, predicting this prompt and issuing it as a new prompt. The second strategy, malicious detection, aims to process the prompt by LLM in the generation of answers. And the third one, generation processing, checks responses to the LLM and removes any potential hazards.

Meanwhile, researchers [14] present a self-reminder system that demonstrates efficiency against jailbreaks of LLM. The results of the Xie et al. study show that using self-reminders reduces the overall percentage of successful jailbreaks from 67.21% to 19.34% on ChatGPT. Moreover, Taddeo et al. [15] recommend three requirements for the development of reliable AI cybersecurity systems: a) in-house development to limit supply chain risks; b) adversarial training against refined attack models; and c) parallel monitoring against cloned control systems to detect divergences.

As the result, several methods have been found to combat attacks on LLM: adversarial training, input reconstruction, instruction processing, malicious detection, self-reminders, and parallel monitoring.

A review of the literature reveals that the topic of LLM security is of significant importance from the perspective of data privacy. Our research focused on the security of via prompts, as this is an area where there is a clear research gap: the development of a system to protect LLMs from jailbreaking.

III Methodology

The Generation Processing strategy [2] will be employed to safeguard the LLM. When the model is attempting to respond, it is necessary to verify that the answer remains within the control area and does not contain any hazards. This necessitates the use of another LLM model to double-check the responses of the first LLM. This approach is less resource-intensive, as it only requires the creation of prompts for the validation model. However, this approach will result in a reduction in the overall efficiency of the model [14]. The creation of such prompts will require approximately two months, as it will be necessary to ascertain the precise methodology for evaluating another model and to identify the specific criteria that may indicate a jailbroken answer. Additionally, another two months will be devoted to creating a cross-checking procedure. Finally, writing up the research will take approximately one month

to recheck everything. The proposed methodology does not necessitate the allocation of a budget, as all requisite technologies are open source, with a sufficient quantity of which being 5,000 prompts, which can be found in articles on the topic of defense against attacks via prompts. For example, this article [14] contains open-source data. The deductive research approach is employed in order to test a hypothesis through the gathering of research data and information.

IV Anticipated Results

The objective is to identify patterns in attacks on LLMs via prompts, which will be presented in a table. This approach is intended to develop smart and complex prompts that can protect LLM from jailbreaks and malicious penetrations. Moreover, the expected results would include a list of patterns by which LLM can be tricked and the working approach with validation LLM. A comparison of the proposed strategy with other available options [14] will be presented in numerical form. The results of this study will assist in the protection of LLM from abuse. Should our study prove successful, our methodology can be applied to any LLM.

V Discussion

The results of this research will benefit the founders and users of LLM. By increasing the security of the model, the performance and impact of LLM on the world will be enhanced. Users will be reassured by the privacy guarantees offered by LLM, while developers will no longer have to worry about the potential for undefined behavior. The safety of LLM will facilitate the development of an ethical policy for AI and encourage the investment of resources in the advancement of personalized AI. However, it should be noted that this study is limited by the absence of tests to assess the impact on response speed and accuracy. To address these limitations, it would be necessary to conduct a separate study involving large comparisons on large benchmarks.

References

[1] M. Gupta, C. Akiri, K. Aryal, E. Parker, and L. Praharaj, "From ChatGPT to ThreatGPT: Impact of Generative AI in Cybersecurity and Privacy," *IEEE Access*, vol. 11, no. 1, pp. 80 218–80 245, Aug. 2023, Accesed: May 4, 2024, DOI: 10.1109/access.2023.3300381. [Online]. Available: http://dx.doi.org/10.1109/ACCESS.2023.3300381.

[2] Y. Yao, J. Duan, K. Xu, Y. Cai, Z. Sun, and Y. Zhang, "A Survey on Large Language Model (LLM) Security and Privacy: The Good, The Bad, and The Ugly," *HighConfidence Comput*, vol. 4, no. 2, Jun. 2024, Accesed: May 4, 2024, DOI: 10.1016/j.hcc.2024.100211. [Online]. Available: http://dx.doi.org/10.1016/j.hcc.2024.100211.

[3] N. Díaz-Rodríguez, J. Del Ser, M. Coeckelbergh, M. López de Prado, E. Herrera-Viedma, and F. Herrera, "Connecting the dots in trustworthy Artificial Intelligence: From AI principles, ethics, and key requirements to responsible AI systems and regulation," *Inf. Fusion*, vol. 99, no. 101896, Nov. 2023, Accesed: May 4, 2024, DOI: 10.1016/j.inffus.2023.101896. [Online]. Available: http://dx.doi.org/10.1016/j.inffus.2023.101896.

[4] C. Zhang, J. Chen, J. Li, Y. Peng, and Z. Mao, "Large language models for human–robot interaction: A review," *Biomimetic Intell. Robot.*, vol. 3, no. 4, Dec. 2023, Accesed: May 4, 2024, DOI: 10.1016/j.birob.2023.100131. [Online]. Available: http://dx.doi.org/10.1016/j.birob.2023.100131.

[5] Y. Li et al., "Model architecture level privacy leakage in neural networks," *Sci China Inf Sciences*, vol. 67, no. 3, Oct. 2023, Accesed: May 4, 2024, DOI: 10.1007/s11432-022-3507-7. [Online]. Available: http://dx.doi.org/10.1007/s11432-022-3507-7.

[6] Y. Wang, Y. Pan, M. Yan, Z. Su, and T. H. Luan, "A Survey on ChatGPT: AI–Generated Contents, Challenges, and Solutions," *IEEE Open J Comput Soc*, vol. 4, no. 1, pp. 280–302, Aug. 2023, Accesed: May 4, 2024, DOI: 10.1109/ojcs.2023.3300321. [Online]. Available: http://dx.doi.org/10.1109/OJCS.2023.3300321.

[7] X. Wu, R. Duan, and J. Ni, "Unveiling security, privacy, and ethical concerns of ChatGPT," *J. Inf. Intell.*, vol. 2, no. 2, pp. 102–115, Mar. 2024, Accesed: May 4, 2024, DOI: 10.1016/j.jiixd.2023.10.007. [Online]. Available: http://dx.doi.org/10.1016/j.jiixd.2023.10.007.

[8] M. K. Puttagunta, S. Ravi, and C. Nelson Kennedy Babu, "Adversarial examples: attacks and defences on medical deep learning systems," *Multimedia Tools Appl*, vol. 82, no. 22, pp. 33 773–33 809, Mar. 2023, Accesed: May 4, 2024, DOI: 10.1007/s11042-023-14702-9. [Online]. Available: http://dx.doi.org/10.1007/s11042-023-14702-9.

[9] A. Kuppa and N.-A. Le-Khac, "Adversarial XAI Methods in Cybersecurity," *IEEE Trans Inf Forensics Secur*, vol. 16, no. 1, pp. 4924–4938, Oct. 2021, Accesed: May 4, 2024, DOI: 10.1109/tifs.2021.3117075. [Online]. Available: http://dx.doi.org/10.1109/TIFS.2021.3117075.

[10] R. S. Sangwan, Y. Badr, and S. M. Srinivasan, "Cybersecurity for AI Systems: A Survey," *J Cybersecurity Privacy*, vol. 3, no. 2, pp. 166–190, May 2023, Accesed: May 4, 2024, DOI: 10.3390/jcp3020010. [Online]. Available: http://dx.doi.org/10.3390/jcp3020010.

[11] M. Kuzlu, C. Fair, and O. Guler, "Role of Artificial Intelligence in the Internet of Things (IoT) cybersecurity," *Discover Internet Things*, vol. 1, no. 1, Feb. 2021, Accesed: May 4, 2024, DOI: 10.1007/s43926-020-00001-4. [Online]. Available: http://dx.doi.org/10.1007/s43926-020-00001-4.

[12] H. Sun, T. Zhu, Z. Zhang, D. Jin, P. Xiong, and W. Zhou, "Adversarial Attacks Against Deep Generative Models on Data: A Survey," *IEEE Trans Knowl Data Eng*, vol. 35, no. 4, pp. 3367–3388, Apr. 2023, Accesed: May 4, 2024, DOI: 10.1109/tkde.2021.3130903. [Online]. Available: http://dx.doi.org/10.1109/TKDE.2021.3130903.

[13] P. Laykaviriyakul and E. Phaisangittisagul, "Collaborative Defense-GAN for protecting adversarial attacks on classification system," *Expert Syst with Appl*, vol. 214, no. 118957, Mar. 2023, Accesed: May 4, 2024, DOI: 10.1016/j.eswa.2022.118957. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2022.118957.

[14] Y. Xie et al., "Defending ChatGPT against jailbreak attack via self-reminders," *Nature Mach Intell*, vol. 5, no. 12, pp. 1486–1496, Dec. 2023, Accesed: May 4, 2024, DOI: 10.1038/s42256-023-00765-8. [Online]. Available: http://dx.doi.org/10.1038/s42256-023-00765-8.

[15] M. Taddeo, T. McCutcheon, and L. Floridi, "Trusting artificial intelligence in cybersecurity is a double-edged sword," *Nature Mach Intell*, vol. 1, no. 12, pp. 557–560, Nov. 2019, Accesed: May 4, 2024, DOI: 10.1038/s42256-019-0109-1. [Online]. Available: http://dx.doi.org/10.1038/s42256-019-0109-1.