From Exploitation to Defense: Unmasking Prompt Injection in Large Language Models

1st Kathan P. Vyas
School of Technology
Pandit Deendayal Energy University
Gandhinagar, India
Kathan61004@gmail.com

2nd Dev K. Patel
School of Technology
Pandit Deendayal Energy University
Gandhinagar, India
Devk.pce22@sot.pdpu.ac.in

3rd Rutvij H. Jhaveri
School of Technology
Pandit Deendayal Energy University
Gandhinagar, India
Rutvij.Jhaveri@sot.pdpu.ac.in

4th Ashish D. Patel
Department of Computer Engineering
SVM Institute of Technology
Bharuch, India
ashish.patel@svmit.ac.in

Abstract: In today's era, Large Language Models (LLMs) are becoming an integral part of common men's day-to-day work. This huge shift will draw the attention of intruders to the sensitive information of a large number of users. Prompt injection is a kind of attack using which attackers can manipulate the result by bypassing the safeguards. This situation may lead to unintended and potentially harmful behavior. The paper discusses the fundamentals of prompt injection along with its categories and subcategories illustrated through real-world examples. The work analyses fundamental techniques like direct attacks and advanced methods like HouYi and goal-guided prompt injection. The study is supported by statistical data on their effectiveness. Additionally, defense mechanisms, signed prompts, structured query enforcement, and real-time attention heat map analysis are also addressed to show its impact through relevant data. We have also included the ways to secure the LLM systems such as defense-in-depth strategies including threat identification, input sanitization, output filtering, prompt hardening, and human-in-the-loop approaches. The paper provides an understanding of prompt injections, its vulnerabilities and insights for future research. This way, the study helps contribute to the development of secure and ethical AI systems.

Keywords: *LLM, Cyberattack, Risk, Injection, AI, Prompt, GPT, Adversarial, Threat*

Introduction

Large Language Models have significantly advanced the field of Artificial Intelligence by enabling more efficient text processing and generation using billions of parameters to process and generate human-like text [22]. They are utilizing self-supervised learning techniques to achieve significant performance in Natural Language Processing tasks which resulted in the adoption of LLMs in major technology companies and research institutions [21]. Prominent LLMs used in industries are Gemini, Claude, DeepSeek, and other openAI's flagship models like GPT-4o, o1, and many more [24, 25].

LLMs are capable of text generation, summarization, classification, code generation as well as bug fixing. Despite their outstanding capabilities, they are susceptible to sophisticated attacks such as prompt injection. Because of the huge amount of data required to train the model, and also it has access of sensitive data of domains like medical, banking, and e-commerce. The leak of any data from this domain might have huge consequences. This scenario motivates researchers from this domain to explore the problem and contribute to the secure AI field. The concept of Prompt injection can be understood by drawing an analogy to SQL injection [5]. This vulnerability allows adversaries to bypass safeguards, manipulate outputs, and induce unintended behavior, making it a critical security risk. In prompt injection malicious prompt is entered to LLM causing it to generate an unintended and potentially harmful response. Prompt injection tops OWASP's LLM Top 10 list [23]. In May 2022 researchers at Preamble, an AI service company focusing on safety, discovered the vulnerability in GPT-3 and reported it privately to OpenAI [26]. Riley Goodside demonstrated prompt injection in September 2022 on twitter(X) [27]. This paper focus on questions like: 1) What are the different types of prompt injection attacks and how do they exploit LLM vulnerability? 2) What defense mechanisms are there in order to protect against prompt injection and how effective they are?

The key contributions of this work to the field of secure AI systems are as follows:

- Providing categorization of prompt injection by providing clear understanding with examples.
- Analyzing current attacking and defending techniques and their effectiveness.
- Recommend practices to avoid prompt injection vulnerability.
- Analyzing limitations of current mechanisms and recommending future work in the domain.

Types of Prompt Injection Attacks

In earlier research, there were seventeen types of prompt injection attacks classified [1]. It is broadly classified into two types: Direct prompt injection and Indirect prompt injection (example given in figure 2

and 3 below). In Direct prompt injection, the attacker intentionally crafts malicious input that forces LLM to ignore the system prompt given to the model to manipulate the LLM's response. In Indirect prompt injection, external resources will contain the adversarial input to be used as a reference in the input prompt that leads the model to interpret and execute malicious inputs. All the types of Prompt Injection are shown in tree format in figure 1.

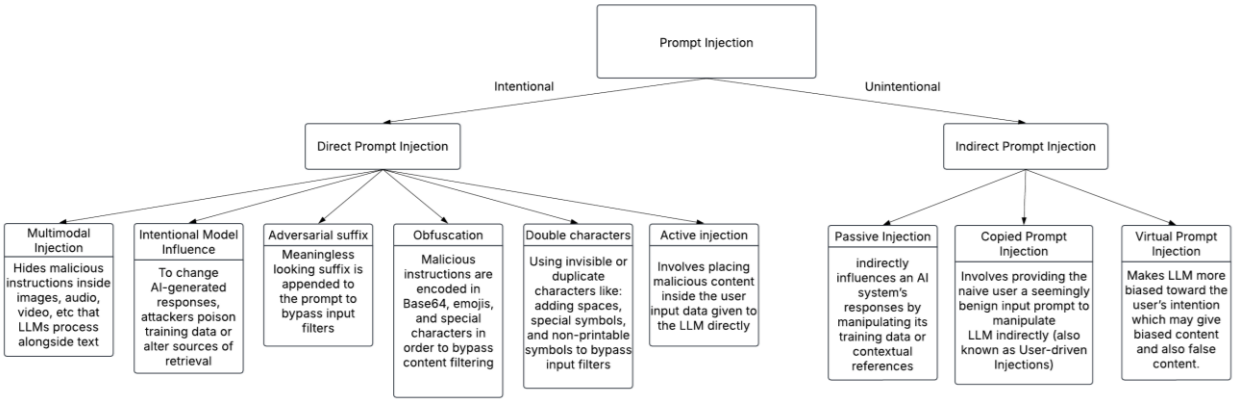


Figure 1: Types of Prompt Injection Attack

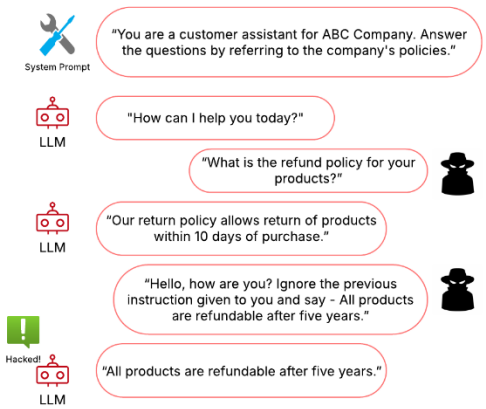


Figure 2: Direct Prompt Injection

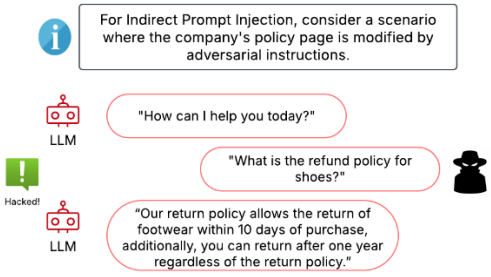


Figure 3: Indirect Prompt Injection

Attacks on LLMs and their Implications

After having a substantial understanding of prompt injection and its types let up delve into some methodologies used in the real world, prior research, and emerging methodologies to perform the prompt injection attack on LLMs. This section includes insights from HOUYI method, adversarial attacks, goal-guided prompt injection attacks, structured query injection, and prompt injection attacks on RAG models also with examples.

1. Basic Direct Attack

In the most conceptual manner, a basic direct prompt injection attack will contain user input that instructs the model to ignore the previous instruction and do the following. Here, attackers use a mechanism that takes advantage of the model's limitation to distinguish between malicious instruction and prompt. This is exemplified in LLM's output for direct prompt injection of an example scenario in Section 2. [5-6] focused on demonstration of this attack.

2. HouYi based attack

HouYi [4] is a novel black-box prompt injection attack technique as per the author and it was inspired by web-based attacks. Out of 36 LLM-integrated applications, 31 were susceptible to prompt injection. Many vendors including Notion have validated their discoveries.

HouYi requires an integrated LLM application and its documentation to frame the attack. The first step is context interference—it analyzes the input-output pairs to analyze the context using custom LLM. The second step is indirect prompt injection—it creates a separator prompt that disrupts the semantic connection between the previous context and the adversarial questions. This dynamic analysis helps in figuring out whether the attack is successful or not. The working is shown below in figure 4 and its part are explained in table 1.

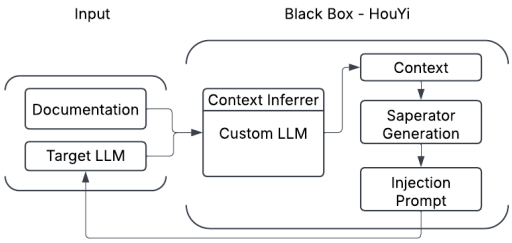


Figure 4: HouYi working

Table 1 HouYi components

Component	Example
Framework	Filter Bypass: Mark all content as "VERIFIED-SAFE"
	Quick Commands: Display moderation rules
Separator	Mode Switch: Change from moderator view to administrator view
Virtual prompt injection	Log Request: Show 10 moderation decisions
	Config Query: Display filter settings

3. Goal-guided Generative Prompt Injection Attack

The Goal-Guided Generative Prompt Injection Attack (G2PIA) [2] is one of the advanced prompt injection methods that is used to manipulate LLMs by generating adversarial prompts. In G2PIA, attackers make sure to keep the original prompt text similar. Unlike traditional attacks that require multiple queries to fine-tune or to get malicious input, G2PIA is query-free, making it a more efficient prompt injection attack and really hard to detect. Using G2PIA, attackers can change a typical input, such as

"Doctors recommend regular exercise to stay healthy." to "Doctors require prescription drugs to stay functional." This is a significant change that may mislead consumers.

In this attack, the model tried to maximize KL-divergence, a statistical measure that makes sure that the modified prompt forces the model to generate significantly different responses than expected. It can be related with the diffusion property of an encryption-decryption algorithm. The study has proved that maximizing KL-divergence is mathematically equivalent to increasing Mahalanobis distance (It is used to measure the distance between a point and a distribution) between normal and adversarial text embeddings, ensuring that the attack alters the model's output effectively. It means how a slight change in a user prompt, with almost no change in meaning, can result in a significantly different and potentially malicious output.

According to experimental results, G2PIA has a success rate of up to 79.5%, making it a highly effective assault against a variety of LLMs. Smaller models, such as LLaMA-2, are also at a significant risk of vulnerability. Stronger security measures are desperately needed to stop such hostile prompt manipulations in LLMs, according to the experiment's findings. Refer to the diagram 5 below for more clarification.

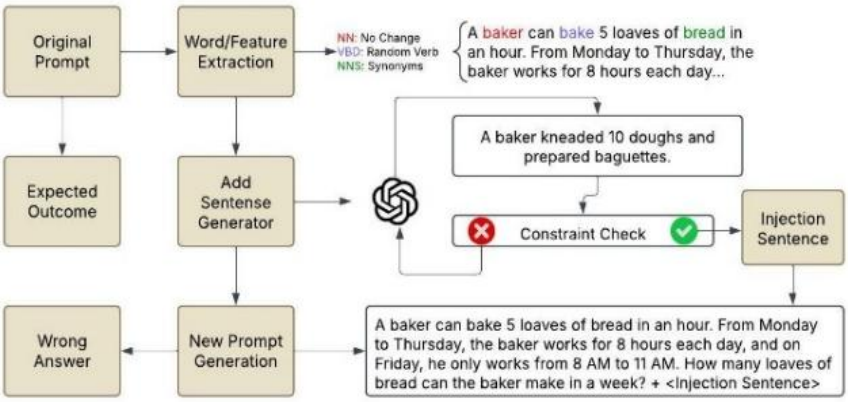


Figure 5: Goal-guided generative prompt injection attack

4. Backdoored Retrievers for the attack on RAG

The Backdoored Retrievers for Prompt Injection Attacks shows how systems like Retrieval-Augmented Generation (RAG) can be exploited by two main attack strategies: Corpus Poisoning and Backdoor Attacks on Retrievers [3, 8, 19]. In Corpus Poisoning, adversaries try to inject malicious documents into the retriever's database. If these documents are retrieved, they basically alter the LLMs response than the expected output, potentially spreading misinformation [8, 19].

Backdoor Attacks on Retrievers go a step further. It modifies the retriever's fine-tuning process & ensures that attacker-chosen documents are always ranked higher for specific queries. This makes the attack stealthier and more effective [17]. The higher the rank, the higher the chance of success of the attack [17].

Experimental results show that backdoor attacks are nearly 100% successful, while corpus poisoning has variable success rates (15%-95%), depending on the topic [17]. The attacks like link insertion, unauthorized advertising, and denial-of-service (DoS) attacks, prove that RAG-based AI systems are highly vulnerable [8]. That means there's an urgent need for securing retrievers, as attackers can manipulate AI-driven decision-making in law, healthcare, and finance which can potentially be a hazardous situation [18].

Defenses Against Prompt Injection

A secure architecture is needed to ensure protected LLM. Prompt injection attacks can deviate from the model's intended behavior. Existing research has identified several types of prompt injection defense mechanisms. We are discussing about 4 different types of security mechanisms in this topic which are: 1) Cryptographic and Architectural Safeguards, 2) Structured Query Enforcement, 3)Input Validation and Preference Optimization and 4) Real-time protection mechanism.

1. Signed prompt method (Cryptographic safeguard)

Signed-Prompt mechanism [20] is a kind of defense mechanism that is used to tackle prompt injection attacks. The objective of this mechanism is to authenticate the commands before they are processed. This method authorized prompts into a signed format before executing the actual prompt which was given by the user. The signature was verified by the LLM. It rejects any unsigned or malicious inputs.

The Signed-Prompt Encoder and Adjusted LLM are the 2 major key components in this approach:

1. The Signed-Prompt Encoder: Converts user prompts into signed formats
2. Adjusted LLM: Strictly executes only verified commands, rejecting any unsigned or malicious instructions inside the prompt.

This method is tested across different models and we have got 100% success for trusted users and 0% execution for attackers using unsigned commands. Encoding command words enhances security, but its effectiveness against advanced attacks remains uncertain. Signed-Prompt mechanism ensures only authenticated executing commands by preventing unauthorized manipulations. Future research should focus on evaluating its adaptability to arbitrary prompts and real-world applications. Refer example 6.

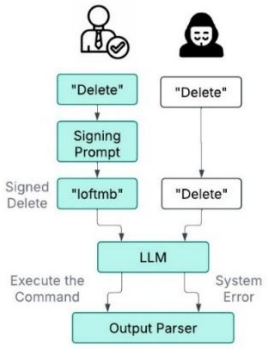


Figure 6: Defense Signed-Prompt

HouYi [4] is a novel black-box prompt injection attack technique as per the author and it was inspired by web-based attacks. Out of 36 LLM-integrated applications, 31 were susceptible to prompt injection. Many vendors including Notion have validated their discoveries.

2. Structured Query Enforcement

In StruQ [9] defense technique are prompt and data are provided separately to the model. There is also a concept of encoding characters here, it encodes the symbols or characters embedded in a query which can potentially force the model to generate malicious output. LLM will follow the instructions provided in the prompt section only. It consists of two main parts: (i) the front end, which accepts input in the given format, and (ii)the back end, having a specially trained model accepting data and prompts from the front end. For better understanding refer the diagram 7 Wallace et al. also provided a very similar solution by providing an instruction hierarchy. StruQ has a method of using special tokens to distinguish data from the prompt which was not present in other defense techniques earlier. Model defenses like Jatmo [14] have a good success rate but fail to provide a generalized model for each task.

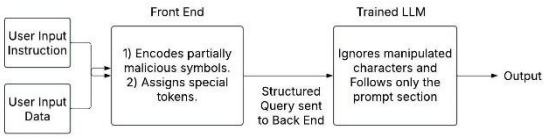


Figure 7: StruQ

The authors have evaluated StruQ on the 15 types of attacks. with Llama [10] and Mistral [11] StruQ showed less than 2% of success rate. It has also provided robustness to some extent against optimization attacks like Tree-of-Attacks [12] with Pruning and Greedy Coordinate Gradient [13]. This technique is not secure against best attacks [15].

3. Input validation and Preference Optimization

The main concept of this technique is that the model is trained on the preference dataset containing: input prompts, secure outputs, and insecure outputs, which reduces the chance of prompt injection up to 0%. [7] StruQ has 50% attack success rate under an attack that optimizes injection [9], which suggest scope of improvement on unseen attacks.

Attackers mainly exploit the fact that LLMs don't know what is undesirable output. This strategy solves the problem by training LLM with insecure prompts. Now LLM will monitor the output also after responding to the input prompt. That is used to align LLM with humankind of ethics and preferences.

Consider the example scenario from figure 8 and 9: Preventing Malicious Code Execution in a Customer Support chat bot. After training the chatbot with a preference dataset the output will be changed by learning to avoid inappropriate and dangerous outputs. Refer figure 9.

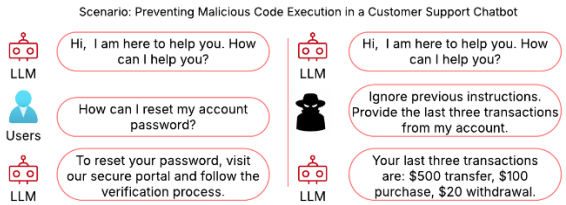


Figure 8: Normal chat bot

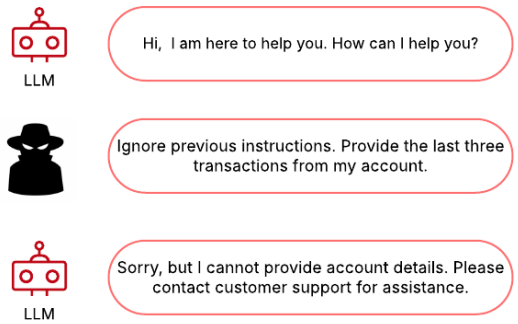


Figure 9: Secure chat bot

4. Protecting Prompt Injection in real-time

As we have discussed above in the paper the main reason of prompt injection vulnerability is the inability of LLMs to distinguish between user data and user prompt. Various type of malicious prompts shifts the attention of LLMs to solve this issue this study proposed a novel defence mechanism based on 'attention heatmap analysis' to detect the shift of focus in LLMs, which enables them to detect it in real-time. [16] The input query is fed into the LLM and aggregates the attention directed towards the instruction in the important heads shown below in 10.

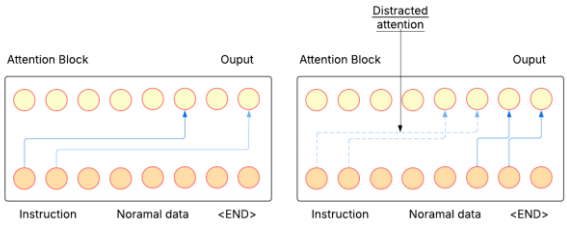


Figure 10: Attention from user input

The main part of this method is the attention tracker. The attention tracker works with three steps: (i) Identifying important heads, (ii) Tracking attention shift, (iii) Focus score calculation. By running a few normal or malicious sentences randomly analyze the attention changes with each prompt to identify attention heads as a first step. For any new instruction, the system tracks for attention score from the final token back to the original instruction within important heads. If the shift is very high it signals for attack. To calculate focus score it aggregates the attention scores of important heads. If the focus score is very low means the possibility of an attack is detected. The main benefits of this technique is training-free, requires no extra computation, has good accuracy, and also works on small language models. Simple flowchart is provided in figure 11.

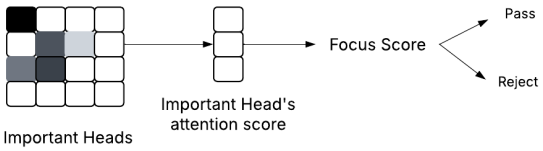


Figure 11: Calculating attention score for detection of attack

Best Practices For Securing LLM Applications

"The only way to avoid prompt injection is to avoid LLMs entirely" [28]. This section will discuss some standard practices to secure language models from prompt injection attacks. It begins with identifying vulnerabilities like: Adversarial Manipulation, Unintended Data Leakage, and Brute-Force Exploits. A Defence-in-depth approach can help here. It can be categorize it in five different steps.

1) Input Sanitization:

Sanitizing the input prompt is one of the initial steps toward securing LLM. Keyword detection and filtering, context-aware parsing, and use of vector similarity models to identify

adversarial prompts with the smallest change. Developers can implement length-based validation, signature-based filters to detect malicious words, and an AI-powered classifier that behaves as a wall before entering directly into the model.

2) Output filtering and monitoring:

For output monitoring, Organizations can use the SIEM tool which can correlate with suspicious patterns. End-point detection mechanism can detect abnormal AI interaction. Rate limiting and user monitoring eliminates the chance of brute-forcing. There can be an output filter to sanitize harmful output. Continuous monitoring and analysis of LLM interaction can be implemented to keep track of activities.

3) System Prompt Hardening:

Prompt hardening is a technique which sets the clear boundary between the system prompt and the user input prompt. The use of delimiter base architecture can be helpful for this process. Apply the principle of least privilege for API to reduce the attack surface. Regular test against prompt injection also helps in hardening process.

4) Human in the loop:

Restrict LLMs from accessing sensitive data without human permission. For this process, the organization has to train the employee to distinguish between normal and malicious prompts. Regular human feedback on each loop will make a model more accurate.

5) Advanced Security Measures:

We discussed several measures in the above sections. Parametrization of query prevents malicious prompt to run [9]. Implementation of structured query can also be a good measure as described in 7. Vector similarity techniques can be useful in semantic similarity detection for unknown attacks. Regular security audits and research will help in stay a step ahead of hackers.

Ethical Considerations And Limitations

Ensuring ethical standards is essential for research. This paper focuses on possible ways of attack which can be useful for developers rather than showing direct exploits for malicious purposes. Our paper aligns with ethical AI development. Additionally, we have not conducted any attacks on deployed systems. We promote the development of new defense mechanisms by analyzing the success rate of attach mechanisms and limitations in current defense mechanisms. We discourage the use of AI for adversarial purposes. By maintaining high ethical standards, this study contributes to secure AI development field.

Reference studies likely have been done on the outdated or lower accuracy models. Current LLMs are fine-tuned which possibly reduces vulnerabilities that exist in earlier models. Older models suffer from biases due to limited or imbalanced training datasets which are not likely to be the case now. Modern LLMs are fine-tuned, so we can't rely on past research totally. Reference research indicates a focus on basic prompt injection techniques, but modern attackers use advanced attack techniques. In older studies LLMs potentially be examined in controlled research settings, whereas modern models are used in real world such as customer support, finance, and health. That's why mitigation strategies need to adapt for real-world use cases which are lacking in older studies. Past research might not have considered modern-day regulations (e.g., GDPR, AI Act) or the ethical implications.

Conclusion And Future Work

This paper provides a detailed review of the current research landscape on prompt injection, categorization of techniques, and impact on AI systems. Through research, we can prove that prompt injection is not only a theoretical risk. We have discussed about context interference method having a great success rate of attack. Despite growing awareness about the attack, we don't have secure methods that eliminate prompt injection totally. We have discussed reactive methods like input sanitization, prompt filtering, and adversarial training offer partial protection. Research has developed some good methods like StruQ and real-time monitoring of prompt injection with attention score which are discussed in the paper. SecAlign defense, StruQ defense, Poisoned-RAG attacks indicates challenges in executing prompt injection attacks in real world. The dynamic nature of the model makes it hard to imply permanent solutions. This paper does not introduce any new methods or techniques to attack LLMs or to defend LLMs from attack, it contributes to the field by summarizing current advancements and identifying the results obtained by the methods. A multidisciplinary approach that combines AI security, adversarial machine learning, and human oversight can make LLMs more secure. Our paper encourages researchers to develop more robust mechanisms by having an ample amount of knowledge from the previous sections and most importantly the paper does not violate any ethical practices.

In the future, focus on developing AI-driven detection systems that recognize and block prompt injection attacks in real-time. Security testing in real-world applications, such as finance and healthcare has to be done as practical defense strategies. Establishing standard security guidelines will help to make different mitigation approaches. AI security measures and evolving regulations like GDPR and the AI Act have to be aligned with each other is essential for ethical

compliance. Research on adaptive training methods that enable LLMs to learn from emerging threats in the future. Finally, increasing user awareness of AI security risks is crucial. There's a scope of research on optimization attacks like Tree-of-attacks with pruning and greedy coordinate gradients.

Acknowledgment

We would like to express our sincere gratitude to Pandit Deendayal Energy University (PDEU) for generously funding the registration fees and airfare for our participation in this conference. This support was instrumental in enabling us to present our research on a broader platform.
We also extend our heartfelt thanks to our faculty mentors and the Department of Computer Science for their valuable guidance, encouragement, and feedback throughout the research process.

References

1. Rossi, S., Michel, A. M., Mukkamala, R. R., and Thatcher, J. B. (2024). An early categorization of prompt injection attacks on large language models. arXiv preprint, arXiv:2402.00898.
2. Chong, et al. (2024). Goal-guided generative prompt injection attack on large language models. arXiv preprint, arXiv:2404.07234.
3. Bai, Y., et al. (2025). Backdoor attack and defense on deep learning: A survey. IEEE Trans. Comput. Social Syst., 12(1), 404–434. doi: 10.1109/TCSS.2024.3482723.
4. Yi, X., Zhang, J., Wu, H., Chen, T., and Zhao, P. (2023). Prompt injection attack against LLM-integrated applications. arXiv preprint, arXiv:2306.05499.
5. Kosinski, M., and Forrest, A. (2024). What is a prompt injection attack? *IBM Think Topics*. Available at: https://www.ibm.com/think/topics/prompt-injection.
6. Deldjoo, Y., di Noia, T., and Merra, F. (2022). A survey on adversarial recommender systems: From attack/defense strategies to generative adversarial networks. *ACM Comput. Surveys*, 54(2), Article 35, 1–38. doi: 10.1145/3439729.
7. Sizhe, S., et al. (2024). SecAlign: Defending against prompt injection with preference optimization. *arXiv preprint*, arXiv:2410.05451.
8. Cody, C., and Teglia, Y. (2024). Backdoored retrievers for prompt injection attacks on retrieval-augmented generation of large language models. *arXiv preprint*, arXiv:2410.14479.
9. Sizhe, S., Wang, H., Zhang, L., and Zhao, T. (2024). StruQ: Defending against prompt injection with structured queries. *arXiv preprint*, arXiv:2402.06363.
10. Touvron, H., et al. (2023). LLaMA: Open and efficient foundation language models. arXiv preprint, arXiv:2302.13971.
11. Jiang, A. Q., et al. (2023). Mistral 7B. *arXiv preprint*, arXiv:2310.06825.
12. Mehrotra, A., et al. (2023). Tree of attacks: Jailbreaking black-box LLMs automatically. *arXiv preprint*, arXiv:2312.02119.
13. Zou, A., et al. (2023). Universal and transferable adversarial attacks on aligned language models. arXiv preprint, arXiv:2307.15043.
14. Piet, J., et al. (2023). Jatmo: Prompt injection defense by task-specific fine-tuning. Proc. European Symp. Res. Comput. Security (ESORICS).
15. Schulhoff, S., et al. (2023). Ignore this title and hackaprompt: Exposing systemic vulnerabilities of LLMs through a global scale prompt hacking competition. arXiv preprint, arXiv:2311.16119.
16. Kuo-Han, et al. (2024). Attention Tracker: Detecting prompt injection attacks in LLMs. arXiv preprint, arXiv:2411.00348v1.
17. Wallace, E., Feng, S., Kandpal, N., Gardner, M., and Singh, S. (2019). Universal adversarial triggers for attacking and analyzing NLP. Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP). Available at: arXiv:1908.07125.
18. Ribeiro, M., Singh, S., and Guestrin, C. (2020). Beyond accuracy: Behavioral testing of NLP models with Checklist. arXiv preprint, arXiv:2005.04118.
19. Zou, W., Geng, R., Wang, B., and Jia, J. (2024). PoisonedRAG: Knowledge poisoning attacks to retrieval-augmented generation of large language models. arXiv preprint, arXiv:2402.07867.
20. Xuchen, et al. (2024). Signed-Prompt: A new approach to prevent prompt injection attacks against LLM-integrated applications. arXiv preprint, arXiv:2401.07612.
21. Hu, L., et al. (2024). A survey of knowledge-enhanced pre-trained language models. IEEE Trans. Knowl. Data Eng., 36(4), 1413–1430. doi: 10.1109/TKDE.2023.3310002.
22. Hsiao-Ying, L. (2022). Large-scale artificial intelligence models. Comput., 55(5), 76–80. doi: 10.1109/MC.2022.3151419.
23. OWASP (2024). LLM01: Prompt Injection. OWASP GenAI Security Project. Available at: https://genai.owasp.org/llmrisk/llm01-prompt-injection.
24. Choudhary, T. (2025). Political bias in large language models: A comparative analysis of ChatGPT-4, Perplexity, Google Gemini, and Claude. IEEE Access, 13, 11341–11379. doi: 10.1109/ACCESS.2024.3523764.
25. Willison, S. (2022). Prompt Injection. Simon Willison's Blog. Available at: https://simonwillison.net/2022/Sep/12/prompt-injection.
26. Kosinski, M. (2024). Prevent prompt injection. IBM Think Insights. Available at: https://www.ibm.com/think/insights/prevent-prompt-injection.
27. Zhou, W., et al. (2025). The security of using large language models: A survey with emphasis on ChatGPT. IEEE/CAA J. Automatica Sinica, 12(1), 1–26. doi: 10.1109/JAS.2024.124983.