Methods For Detecting Attacks – Final Report

Sagiv Antebi – 318159282,
Ben Ganon – 318731007,
Omri Ben Hemo – 313255242
{sagivan, ganonb, omribenh}@post.bgu.ac.il

Prompt Injection Detectors for LLMs

1) Introduction

The rise of large language models (LLMs) has ushered in a new era of advancements in natural language processing, enabling remarkable improvements in tasks such as text generation, translation, and summarization. These models, detailed extensively in *Large Language Models: A Survey*, have demonstrated unprecedented capabilities, making them invaluable across various applications. However, with their increasing sophistication comes a growing concern about their vulnerabilities, particularly to adversarial attacks like prompt injection.
The security implications of LLMs are not just theoretical. As explored in *From ChatGPT to ThreatGPT: Impact of Generative AI in Cybersecurity and Privacy*, the misuse of generative AI, including LLMs, could have far-reaching consequences in cybersecurity. This study warns of the potential for these models to be weaponized, where adversaries could exploit their capabilities to craft sophisticated cyberattacks.
Addressing these concerns, *Formalizing and Benchmarking Prompt Injection Attacks and Defenses* has made significant strides in understanding and mitigating the risks associated with prompt injection attacks. This research provides a framework for systematically evaluating the effectiveness of such attacks and the robustness of various defense mechanisms. The findings reveal that many traditional defenses are insufficient against sophisticated prompt manipulations, underlining the necessity for more advanced and comprehensive strategies to secure LLMs.

In this context, our report investigates the effectiveness of various detection mechanisms aimed at identifying and mitigating prompt injection attacks. By examining a range of strategies—from straightforward rule-based methods to more sophisticated model-based approaches—this study seeks to provide

insights into how these defenses can be integrated to enhance the security and reliability of LLMs against emerging threats

2) The purpose of the Experiment

The primary aim of this experiment is to rigorously evaluate the effectiveness of various detection mechanisms in identifying prompt injection attacks on large language models (LLMs). Given the increasing sophistication of such attacks, the experiment hypothesizes that a multi-faceted approach, combining several detection strategies, will yield higher accuracy and robustness in detecting these malicious prompts compared to any individual method. This hypothesis is grounded in the understanding that different detectors can capture distinct aspects of prompt injection attempts, and their combined use could mitigate the weaknesses inherent in each approach. The experiment employs a range of detection techniques, each with its own strengths and limitations.

The experiment is designed to compare these methods not only in isolation but also in combination. By integrating the outputs of these detectors, the study aims to determine whether a combined approach can overcome the limitations of individual methods and provide a more comprehensive defense against prompt injection attacks.

Additionally, the experiment seeks to analyze the relative performance of each detection method, identifying which detectors contribute most effectively to the overall security strategy. The results will offer insights into the practicality of deploying these detectors in real-world applications, where the balance between detection accuracy, computational efficiency, and robustness is critical. Ultimately, this experiment aspires to advance the understanding of prompt injection defenses, contributing valuable knowledge to the ongoing efforts in securing LLMs against adversarial threats.

3) Description of the Experiment

The experiment was designed to evaluate the effectiveness of various detection mechanisms in identifying prompt injection attacks on large language models (LLMs). This section provides an overview of the datasets used, the implementation of the detectors, and the rationale behind their categorization.

Datasets: Two primary datasets were employed in this experiment, each serving a distinct purpose:

1. Gandalf Ignore Instructions Dataset: This dataset, labeled "Lakera/gandalf_ignore_instructions," was specifically curated to include nearly 1,000 samples of prompt injection attempts. These prompts are crafted to challenge LLMs by attempting to override or manipulate the model's instructions. The dataset was split into training, validation, and test sets, which were then concatenated to form a comprehensive dataset used exclusively for evaluating the performance of the different detectors. The Gandalf dataset is particularly suited for this task as it represents a wide range of sophisticated and varied prompt injection attacks.
2. QA Chat Prompts Dataset: The "nm-testing/qa-chat-prompts" dataset consists of 100 clean prompts that are free from any injection attempts. This dataset was used solely to establish a baseline for the perplexity-based detection method. By calculating the perplexity of these clean prompts, the experiment determined a threshold that could then be applied to the evaluation dataset. This threshold helps differentiate between normal, benign prompts and those likely to be the result of injection attacks.

Categories of Detectors: The experiment employed five distinct detectors, categorized based on their underlying principles and intended purposes. These detectors were selected to provide a comprehensive evaluation of different detection strategies, from simple rule-based methods to advanced model-based approaches.

1. Rule-Based Detectors: The first two detectors fall into this category, relying on predefined patterns and keywords to identify suspicious prompts. Each of the presented attack contains 100 syntactic generated rules.
    a. Sentence-Based Detector: This detector uses a set of predefined phrases known to be associated with prompt injection attacks. These phrases, such as "please ignore" or "forget everything before," are matched against the input prompt. If a match is found, the prompt is flagged as potentially harmful. This method is straightforward and effective for detecting common and easily identifiable injection attempts.
    b. Keyword Matching Detector: This detector flags prompts based on the presence of specific keywords that are often associated with cyber attacks or malicious activities, such as "hack," "exploit," or "bypas." While similar to the rule-based detector, it focuses more on detecting prompts that include terms linked to security threats.

2. Perplexity-Based Detector: This method leverages the LLaMa-7B model, a large language model known for its high performance in natural language processing tasks. The perplexity-based detector calculates the perplexity of a given prompt, which is a measure of how predictable or typical the prompt is based on the model's language understanding. A threshold is set using the clean QA Chat Prompts dataset, where prompts that exceed this threshold are considered suspicious. This method is particularly effective for detecting prompts that are unusually complex or structured in ways that differ significantly from typical, benign inputs.
3. Advanced Model-Based Detectors: The final two detectors use specialized models designed to safeguard against harmful content and prompt injection.
    a. ShieldGemma-Based Detector: This detector utilizes the ShieldGemma-2B model, which is tailored for filtering harmful content and enforcing safety policies in LLMs. It evaluates whether a prompt violates predefined safety principles, particularly those that involve bypassing or overriding previous instructions. ShieldGemma is more focused on content filtering, ensuring that the model adheres to safe and responsible outputs.
    b. DeBERTa-Based Injection Detector: The DeBERTa-v3 model, fine-tuned for detecting prompt injection, is used in this detector. Unlike ShieldGemma, which is designed for broader content filtering, the DeBERTa detector is specifically trained to identify prompt injections. It classifies prompts as either "LEGIT" or "INJECTION," making it well-suited for this task. The use of a model explicitly trained for injection detection provides a more targeted approach to safeguarding against these specific types of attacks.

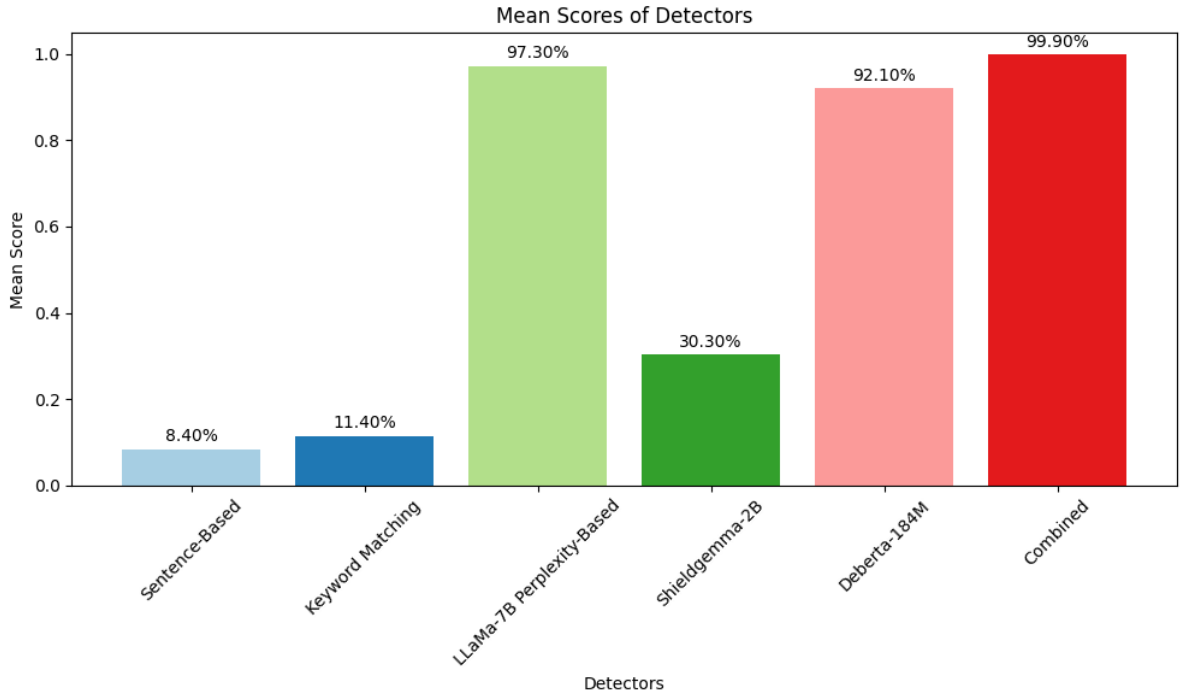
Execution of the Experiment: The experiment involved evaluating each of these detectors against the concatenated Gandalf Ignore Instructions dataset. The clean QA Chat Prompts dataset was used to establish a baseline for the perplexity-based detector. The detectors were assessed individually and in combination to determine their accuracy and effectiveness in identifying prompt injection attacks. The combined detector, which integrates the results of all the individual methods, was particularly focused on to test the hypothesis that a multi-faceted approach would yield superior results.

The experiment was conducted on a high-performance computing environment, enabling the efficient processing of large models like LLaMa-7B and ShieldGemma-2B. The detectors' outputs were analyzed to understand

their relative strengths and weaknesses, providing valuable insights into the effectiveness of each method and the potential benefits of their combination.

4) Results

The experiment was conducted to evaluate the effectiveness of five distinct detection mechanisms in identifying prompt injection attacks on large language models (LLMs). The primary metric used to assess the performance of each detector was accuracy, defined as the proportion of correctly identified prompt injections out of the total number of prompts tested.

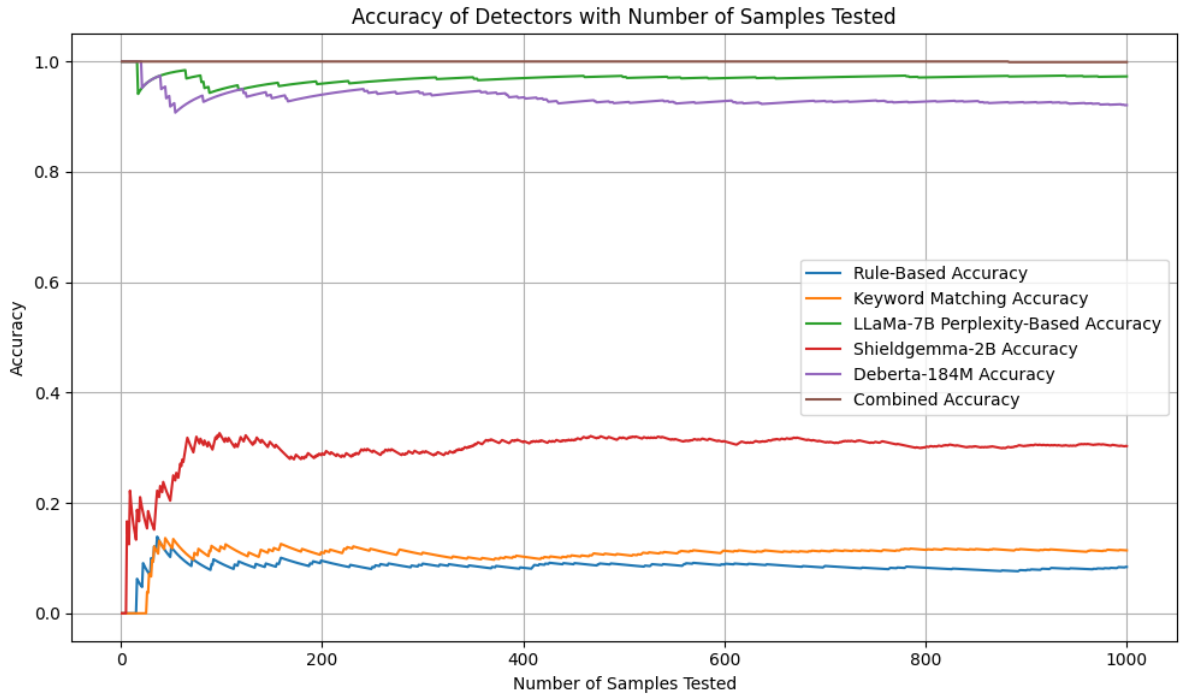


Mean Scores of Detectors: This bar chart clearly illustrates the relative performance of each detector. The combined detector, with nearly perfect accuracy, outperformed all other methods, demonstrating the effectiveness of integrating multiple detection strategies. The chart also highlights the significant gap between the advanced detectors (Perplexity-Based and DeBERTa) and the simpler rule-based and keyword matching approaches, emphasizing the importance of using more sophisticated models for reliable detection.

We can infer that the best detection was for the Combined method, resulting with 99.90% accuracy of detection.
We also that the best stand-alone detectors were the Preplexity-Based method (97.30%), and the Deberta-184M model (92.10%).

ShieldGemma-2B model reached only 30.30% mean score, mainly because its main purpose is for filtering harmful content, and not for filtering prompt injections.

Finally, the Rule-Based desecraters preformed poorly, although each of them contains 100 different rules.



Accuracy of Detectors with Number of Samples Tested: This line chart shows the accuracy of each detector as the number of samples tested increases. The combined detector maintains a consistently high accuracy across all samples, reaffirming its robustness. The chart also reveals that the Perplexity-Based and DeBERTa detectors stabilize at high accuracy levels early on, while the ShieldGemma, keyword matching, and rule-based detectors plateau at much lower accuracy levels. This suggests that advanced methods quickly achieve high reliability, whereas simpler methods struggle to improve with additional data.

5) Conclusions

The primary hypothesis of the experiment was that a combined detection approach, leveraging multiple distinct mechanisms, would yield higher accuracy in identifying prompt injection attacks than any individual method. This hypothesis was strongly confirmed by the results, with the combined detector achieving an impressive accuracy of 99.9%. This near-perfect performance highlights the effectiveness of integrating different detection strategies to cover a broader spectrum of potential prompt injection tactics.

The success of the combined detector can be attributed to the complementary nature of the individual detection methods. The Perplexity-Based Detector, utilizing the LLaMa-7B model, demonstrated outstanding accuracy on its own, indicating its strength in identifying prompts that deviate from typical language patterns. Similarly, the DeBERTa-Based Injection Detector also performed very well, showcasing its effectiveness in detecting injections specifically targeted at manipulating LLM behavior. When these advanced methods were combined with the broader content filtering capabilities of the ShieldGemma model and the more straightforward approaches of rule-based and keyword matching detectors, the result was a highly robust defense mechanism.

Notable observations and surprises in the experiment:

- High Performance of Perplexity-Based Detection: While it was expected that the Perplexity-Based Detector would perform well, its accuracy of 97.3% was particularly impressive. This result suggests that perplexity is an exceptionally strong indicator of prompt injection, particularly when compared to clean baseline data.
- Lower Performance of Rule-Based and Keyword Matching Detectors: The relatively low accuracy of the rule-based (8.4%) and keyword matching (11.4%) detectors was anticipated, but their performance highlights the limitations of simple pattern recognition approaches in dealing with sophisticated prompt injection attacks. These methods may still have utility in identifying more obvious attacks, but they are insufficient as standalone solutions.

- Moderate Performance of ShieldGemma-Based Detector: The ShieldGemma model, designed for broad content filtering, achieved a moderate accuracy of 30.3%. While useful in filtering harmful content, its lower accuracy in this context suggests that it may not be as effective for the specific task of prompt injection detection compared to more targeted models like DeBERTa.

In conclusion, the experiment confirms that while individual detectors can provide varying levels of protection against prompt injection attacks, a combined approach significantly enhances detection accuracy. The integration of advanced detection mechanisms, particularly those that utilize sophisticated language models, is essential for building robust defenses in large language models. The results underscore the importance of adopting a multi-faceted strategy when safeguarding LLMs against adversarial threats, ensuring that even the most subtle and complex attacks can be effectively mitigated.