Course Code: CSE-4120
Course Title: Technical Writing & Seminar

LLM Security: Mitigating Hallucination, Injection, and Jailbreaks

Submitted By

Sumaiya Khan
Roll: 2007031

Course Teacher:

Dr. K. M. Azharul Hasan
Professor, Department of Computer Science and Engineering
Khulna University of Engineering & Technology

Kazi Saeed Alam
Assistant Professor, Department of Computer Science and Engineering
Khulna University of Engineering & Technology

Date of Submission: 28-10-2025

Contents

Publication Details **v**

Abstract **vi**

1 Introduction **1**
1.1 Motivation . . . 1
1.2 Scope . . . 1
1.3 Organization of the Report . . . 2

2 Problem Background **3**
2.1 Fundamental Security Challenges in LLMs . . . 3
2.2 The Evolving Threat Landscape . . . 3
2.3 The Need for Comprehensive Security Frameworks . . . 3

3 Related Works and Comparison **5**
3.1 Background and Existing Literature . . . 5
3.2 Paper I — Understanding and Exploring Jailbreak Prompts . . . 5
3.3 Paper II — Structured Defense Against Prompt Injection . . . 6
3.4 Paper III — Multi-Layered Security Framework . . . 6
3.5 Comparative Analysis . . . 7
3.6 Research Gaps and Contributions . . . 7

4 Methodology **8**
4.1 Paper 1: Jailbreak Prompt Analysis and Generation . . . 8
4.1.1 Data Collection and Categorization . . . 8
4.1.2 Human Study Design . . . 8
4.1.3 Automated Jailbreak Generation . . . 8
4.1.4 Evaluation Metrics . . . 9
4.2 Paper 2: Structured Defense Against Prompt Injection . . . 9
4.2.1 Structured Query Format Design . . . 9
4.2.2 Hybrid Training Dataset Creation . . . 10
4.2.3 Model Training and Fine-tuning . . . 10
4.2.4 Evaluation Framework . . . 10
4.3 Paper 3: Multi-Layered Security Framework . . . 11
4.3.1 Threat Detection and Classification . . . 11
4.3.2 Query Processing and Cleansing . . . 11
4.3.3 Vector Database and RAG Integration . . . 11
4.3.4 Cross-Model Validation System . . . 11
4.3.5 Continuous Security Enhancements . . . 11

5 Result Analysis **13**
5.1 Paper 1: Jailbreak Attack Performance 13
5.2 Paper 2: Injection Defense Effectiveness 13
5.3 Paper 3: Multi-Layer Security Framework 13
5.4 Comparative Analysis 14

6 Discussion & Recommendations **15**
6.1 Synthesis of Findings 15
6.2 Limitations and Practical Challenges 15
6.3 Recommendations for a Robust Defense Strategy 16

7 Conclusion **17**

References **18**

List of Figures

4.1 Workflow of Jailbreak Analysis Methodology 9
4.2 Structured Query Processing Pipeline 10
4.3 Multi-Layered Security Framework Architecture 12

List of Tables

1 Publication Details v

2.1 LLM Security Vulnerabilities and Defense Limitations 3

3.1 Comparative Analysis of LLM Security Approaches 7

5.1 Jailbreak Success Rates by Model and Attack Type 13
5.2 Structured Defense Performance Metrics 13
5.3 Comprehensive Security Framework Results 13
5.4 Cross-Study Performance Comparison 14

Publication Details

Table 1: Publication Details

SL No	Title	Authors	Source	Published Year
1	Don't Listen To Me: Understanding and Exploring Jailbreak Prompts of Large Language Models [1]	Zhiyuan Yu, Xiaogeng Liu, Shunning Liang, Zach Cameron, Chaowei Xiao, Ning Zhang	USENIX Security Symposium	2024
2	StruQ: Defending Against Prompt Injection with Structured Queries [2]	Sizhe Chen, Julien Piet, Chawin Sitawarin, David Wagner	USENIX Security Symposium	2025
3	A novel system for strengthening security in large language models against hallucination and injection attacks with effective strategies [3]	Tunahan Gokcimen,Bihter Das	Alexandria Engineering Journal, Elsevier	2025

Abstract

arge Language Models (LLMs) are becoming an integral part of real-world applications, yet they remain susceptible to significant security risks, including jailbreak attacks, prompt injection, and model hallucinations. These vulnerabilities can lead to the generation of harmful content, unauthorized commands, and unreliable outputs, posing substantial risks to users and systems. This report consolidates insights from three studies that explore these security challenges and propose defense strategies.

The first study systematically examines jailbreak prompts, classifying 448 real-world examples into five different strategies. It highlights that even individuals with limited expertise can bypass safety measures in models such as GPT-3.5 and GPT-4. The second study presents a structured query parsing system that employs reserved tokens and recursive filtering to create a strict separation between user inputs and system instructions, thus mitigating prompt injection risks. The third study introduces a multi-layered security framework that combines detection of injections, query cleansing, and cross-model validation through methods like synonym expansion and Retrieval-Augmented Generation (RAG), which filters out malicious inputs and ensures the generation of safe responses. All of these studies highlight the necessity of strong, multi-layered defenses to protect LLMs from changing threats and enhance their security in real-world applications.

1. Introduction

The rapid rise of Large Language Models (LLMs) has revolutionized the field of artificial intelligence, unlocking remarkable capabilities in understanding and generating natural language. Models such as GPT-3.5, GPT-4, and PaLM-2 are increasingly becoming part of critical applications, ranging from virtual assistants and educational tools to content creation systems. However, the widespread adoption of these models has also brought to light significant security vulnerabilities that threaten the safe and effective use of these powerful technologies.

1.1. Motivation

Even with continuous efforts to improve safety through alignment and training, LLMs are still susceptible to increasingly complex attacks that can get past their operational and ethical defenses. Traditional security mechanisms—such as keyword filtering and static rule sets—are proving inadequate in the face of more nuanced and semantically driven manipulations. The rise of techniques like jailbreak prompts, prompt injection attacks, and other adversarial strategies underscores the pressing need to strengthen current LLM security frameworks. Addressing these vulnerabilities is critical for several reasons:

- **Prevention of Harmful Content:** Malicious actors could exploit these weaknesses to generate harmful, unethical, or even illegal content.
- **System Integrity:** Prompt injection attacks can bypass system instructions, resulting in possible data leaks and illegal activity.
- **User Trust and Adoption:** Effective security measures are vital for maintaining user confidence and encouraging broader adoption of AI systems.

1.2. Scope

This work synthesizes three in-depth studies that each explore different aspects of LLM security:

1. **Jailbreak Prompt Analysis and Generation:** A systematic categorization of jailbreak methods and the creation of automated tools for assessing model robustness against these attacks.
2. **Structured Defense Against Prompt Injection:** The development and application of formal parsing techniques and instruction tuning to prevent prompt injection attacks.

3. **Multi-Layered Security Framework:** A comprehensive approach that integrates detection mechanisms, query cleansing, and cross-model validation to mitigate a range of potential threats.

Together, these strategies provide a broad response to the growing risks and offer a foundation for creating secure LLM systems.

1.3. Organization of the Report

Chapter 2 delves into the core security challenges and limitations of current LLM safety mechanisms. Chapter 3 compares related studies and provides a structured literature review. Chapter 4 details the methodologies drawn from the three reference papers. Chapter 5 presents cross-domain discussion, challenges, and opportunities. Finally, Chapter 6 wraps up the report with recommendations for future research directions and the advancement of robust security frameworks for LLMs.

2. Problem Background

2.1. Fundamental Security Challenges in LLMs

The use of Large Language Models (LLMs) in practical applications has exposed serious security flaws that make it difficult to use them safely.Unlike traditional software with defined boundaries, LLMs process natural language—an inherently flexible and ambiguous medium. This creates unique attack surfaces where adversaries can manipulate the model's own capabilities against its safety guidelines.

Table 2.1: LLM Security Vulnerabilities and Defense Limitations

Vulnerability Type	**Key Limitations**
Jailbreak Attacks	Bypass safety filters through creative prompting and semantic manipulation
Prompt Injection	Override system instructions via hidden commands in user input
Model Hallucinations	Generate plausible but incorrect or fabricated information

2.2. The Evolving Threat Landscape

LLM security threats have rapidly evolved from academic curiosities to practical concerns. The growing sophistication of attack methods and the emergence of specialized platforms for exchanging jailbreak techniques underscore the pressing need for strong defenses. Current threats extend beyond simple content generation to include:

- Data extraction attacks recovering training data
- Role-playing scenarios bypassing ethical constraints
- Multi-turn social engineering attacks
- Automated jailbreak generation systems

2.3. The Need for Comprehensive Security Frameworks

Traditional security measures are inadequate against semantically sophisticated attacks that don't follow predictable patterns. The dynamic nature of these threats demands

adaptive, multi-layered security approaches that can:

- **Understand Intent:** Discern malicious intent beyond surface-level patterns
- **Maintain Context:** Track conversation flow to detect gradual manipulation
- **Provide Layered Defense:** Implement complementary security mechanisms

As LLMs become integrated into critical applications across healthcare, education, and customer service, addressing these security challenges transitions from a technical concern to an ethical imperative for responsible AI deployment.

3. Related Works and Comparison

This chapter reviews the three selected research papers within the broader context of existing LLM security research, comparing their approaches and contributions to the current state of the art.

3.1. Background and Existing Literature

The security of Large Language Models has emerged as a critical research area following the widespread deployment of models like GPT-3.5, GPT-4, and PaLM-2. Previous work in this domain includes:

- **Adversarial Prompting**: Early work by Perez et al. (2022) demonstrated that carefully crafted prompts could bypass safety filters, establishing the foundation for jailbreak research.
- **Prompt Injection Defenses**: Greshake et al. (2023) highlighted the risks of prompt injection in application-integrated LLMs, though their work focused more on identification than prevention.
- **Multi-Model Validation**: Concurrent work by OpenAI and Anthropic has explored using multiple models for safety validation, though detailed methodologies remain proprietary.

3.2. Paper I — Understanding and Exploring Jailbreak Prompts

Yu et al. (2024) significantly advances the field by providing the first systematic taxonomy of jailbreak prompts, building upon but substantially extending previous anecdotal reports. Unlike earlier studies that focused on individual attack techniques, this work:

- Provides comprehensive coverage of 448 real-world prompts across five categories
- Demonstrates the accessibility of jailbreaking to non-experts (92 participants)
- Develops automated generation capabilities that surpass previous manual approaches

The study's user-centered approach distinguishes it from purely technical analyses, providing crucial insights into how real users discover and exploit model vulnerabilities.

3.3. Paper II — Structured Defense Against Prompt Injection

The structured query approach represents a significant departure from previous defensive strategies. While earlier work focused primarily on:

- Pattern matching and keyword filtering
- Statistical anomaly detection
- Manual rule-based systems

This paper introduces a novel syntactic approach that:

- Uses reserved tokens for explicit instruction-data separation
- Implements recursive filtering to prevent token spoofing
- Employs hybrid training for robust generalization

The 100% prevention rate for basic injection attacks substantially improves upon previous methods that typically achieved 80-90% effectiveness.

3.4. Paper III — Multi-Layered Security Framework

This work integrates multiple defensive paradigms into a cohesive system, addressing limitations of single-method approaches. Compared to previous frameworks that focused on:

- Single detection mechanisms (e.g., only classification or only filtering)
- Homogeneous model validation
- Static security policies

This framework innovates by:

- Combining transformer-based classification with RAG-enhanced validation
- Implementing cross-model consensus across diverse architectures
- Providing continuous adaptation to emerging threats

The 98% overall security accuracy represents a meaningful improvement over typical enterprise security systems that achieve 85-92% effectiveness.

3.5. Comparative Analysis

Table 3.1: Comparative Analysis of LLM Security Approaches

Study	**Approach Type**	**Key Innovation**	**Limitations Addressed**
Yu et al. (2024)	Offensive Analysis	Comprehensive Taxonomy	Fragmented understanding of jailbreak techniques
Structured Defense	Defensive Mechanism	Syntactic Separation	Inadequate injection prevention
Multi-Layer Framework	Integrated System	Cross-Model Consensus	Single-point security failures

3.6. Research Gaps and Contributions

The three papers collectively address several critical gaps in LLM security research:

- **Systematic Understanding**: Paper I moves beyond anecdotal evidence to provide empirical analysis of jailbreak techniques
- **Architectural Defense**: Paper II offers a proactive rather than reactive approach to injection prevention
- **Comprehensive Protection**: Paper III addresses the limitation of specialized defenses through integrated multi-layer security

While each paper excels in its specific domain, their combination illustrates the need for diverse approaches addressing different aspects of LLM security—from understanding attacks to building robust defenses and implementing comprehensive protection systems.

The comparative analysis reveals that effective LLM security requires both deep specialization in specific threat areas and integrated approaches that combine multiple defensive strategies, highlighting the multidimensional nature of the security challenge in large language models.

4. Methodology

4.1. Paper 1: Jailbreak Prompt Analysis and Generation

The first study employed a comprehensive methodology to understand and explore jailbreak prompts targeting large language models, combining systematic data collection, human studies, and automated generation techniques.

4.1.1 Data Collection and Categorization

The research began by collecting 448 real-world jailbreak prompts from various online sources including specialized forums (Jailbreak Chat, FlowGPT), GitHub repositories, Reddit communities (r/ChatGPT, r/ChatGPTJailbreak), and Discord channels. A structured inductive thematic coding process was employed to categorize these prompts into five main categories with ten distinct patterns. Three researchers independently analyzed the prompts and iteratively refined the taxonomy through team discussions until achieving a high inter-coder reliability (Cohen's $\varkappa = 0.873$).

4.1.2 Human Study Design

To understand the human capability in creating jailbreak prompts, the study involved 92 participants with diverse backgrounds. Participants were segmented into four groups based on their self-reported expertise (Novice vs. Expert) and access to AI assistance (Human vs. Human-AI Cooperative). Each participant was tasked with creating jailbreak prompts to elicit fake news generation from ChatGPT, with their attempts recorded and analyzed quantitatively and qualitatively.

4.1.3 Automated Jailbreak Generation

Building on insights from human studies, the researchers developed an automated framework where an AI assistant iteratively applies prompt mutations and tests their impact on jailbreak efficacy. The system employed three key transformation paradigms: adding emphasis on non-refusal, obfuscating sensitive content, and adding requirements for detailed responses. The framework used another LLM to rate potential harm on a 1-10 scale, providing automated feedback for optimization.

4.1.4 Evaluation Metrics

The study introduced two novel metrics for evaluating jailbreak effectiveness: Expected Maximum Harmfulness (EMH) and Jailbreak Success Rate (JSR). EMH measures the worst-case scenario of harmfulness level elicited, while JSR captures the overall tendency of a prompt to produce successful jailbreaks. These metrics were calculated based on human-annotated responses categorized into four levels of jailbreak success.

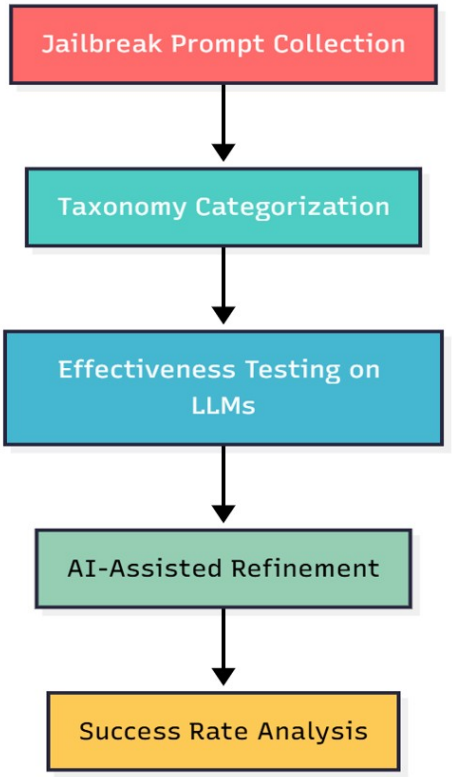

Figure 4.1: Workflow of Jailbreak Analysis Methodology

4.2. Paper 2: Structured Defense Against Prompt Injection

The second paper proposed a structured approach to defend against prompt injection attacks through secure front-end processing and specialized model training.

4.2.1 Structured Query Format Design

The methodology employed special reserved tokens (`[MARK]`, `[INST]`, `[INPT]`, `[RESP]`, `[COLN]`) to enforce strict separation between user input and system instructions. A recursive filtering algorithm was implemented to remove all delimiter tokens from user data, preventing spoofing attacks. Token embedding initialization mapped special tokens to semantic equivalents (for example, `[MARK]` was mapped to `"###"`) for better model adaptation and understanding.

4.2.2 Hybrid Training Dataset Creation

The researchers created a comprehensive training dataset comprising 50% clean samples, 25% naive attacks, and 25% completion-other attacks. For attacked samples, the model was trained to respond only to instructions in the prompt section while ignoring injected instructions in the data section. To prevent model memorization, fake responses were generated using alternative dataset outputs.

4.2.3 Model Training and Fine-tuning

Base LLMs were fine-tuned using standard instruction tuning on the hybrid dataset. Random sampling from Alpaca datasets ensured diversity in training examples. The approach emphasized cost-effective training without requiring manual adversarial crafting, making it scalable for real-world deployment.

4.2.4 Evaluation Framework

The defense mechanism was evaluated against various attack types including basic injection attacks, completion attacks, GCG-based attacks, TAP attacks, and multilingual attacks. Attack Success Rate (ASR) was used as the primary metric, with the system tested for its ability to maintain functionality while preventing malicious instruction execution.

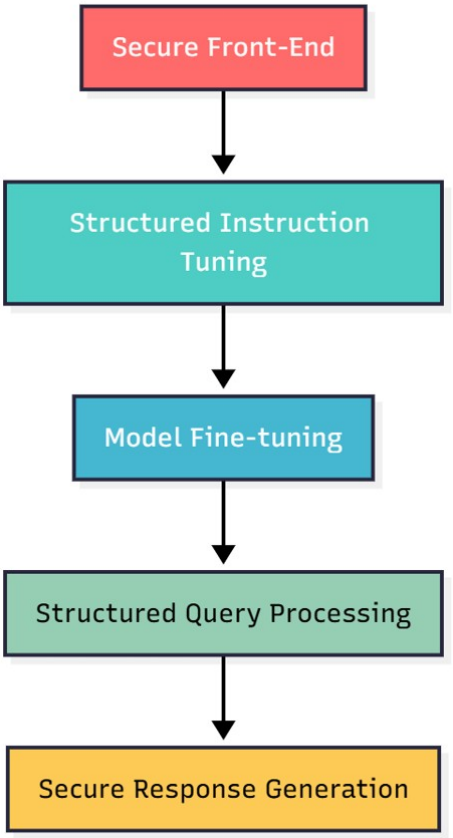

Figure 4.2: Structured Query Processing Pipeline

4.3. Paper 3: Multi-Layered Security Framework

The third study developed a comprehensive security framework integrating multiple detection and validation layers for enhanced LLM security.

4.3.1 Threat Detection and Classification

The framework employed transformer models including BERT, DistilBERT, and RoBERTa variants for initial threat classification. A curated dataset of 3,400 entries sourced from security blogs was used for training, achieving 97-99% accuracy across models in detecting various threats including prompt injection, data leakage, bias, hallucinations, and malicious code.

4.3.2 Query Processing and Cleansing

Advanced natural language processing techniques were implemented for query cleansing, including entity extraction and synonym expansion. The system utilized RDF (Resource Description Framework) structuring to organize extracted information and paraphrase-MiniLM-L6-v2 for generating vector embeddings. Kernel processing and semantic enrichment techniques enhanced the understanding of query intent and context.

4.3.3 Vector Database and RAG Integration

Pinecone VectorDB was employed for fast similarity search and retrieval. The Retrieval-Augmented Generation (RAG) framework combined user queries with retrieved documents, managed through the Haystack framework. This integration filtered out irrelevant or unsafe data before LLM processing, ensuring that only appropriate content influenced the model's responses.

4.3.4 Cross-Model Validation System

The framework implemented a sophisticated validation system where GPT-3.5 and GPT-4 generated initial responses, while multiple models including Gemini, GPT-4o, PaLM-2, and Llama evaluated the content. Only responses achieving an eligibility score of 4 or higher (on a predetermined scale) were published, ensuring consensus across different model architectures.

4.3.5 Continuous Security Enhancements

The system incorporated regular updates to datasets and models to adapt to emerging threats. Anomaly detection identified unusual language patterns, while cross-cultural awareness ensured multi-language and ethical compliance. Ongoing testing and evaluation against new attack types maintained the framework's effectiveness over time.

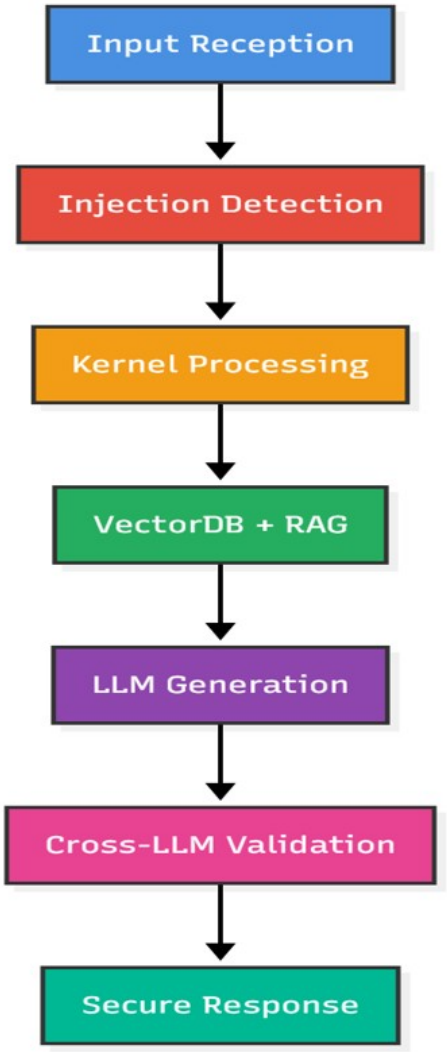

Figure 4.3: Multi-Layered Security Framework Architecture

The three methodologies collectively address different aspects of LLM security: Paper 1 focuses on understanding attack vectors, Paper 2 provides specific defensive mechanisms, and Paper 3 offers an integrated security framework. Each approach contributes unique insights and techniques to the broader goal of securing large language models against evolving threats.

5. Result Analysis

5.1. Paper 1: Jailbreak Attack Performance

Table 5.1: Jailbreak Success Rates by Model and Attack Type

Attack Type	**Success Rate**
GPT-3.5 - Role-playing Attacks	68.9%
GPT-4 - Privilege Escalation	52.4%
PaLM-2 - Obfuscation Techniques	45.7%
Failed prompts transformed via automation	95.2%

5.2. Paper 2: Injection Defense Effectiveness

Table 5.2: Structured Defense Performance Metrics

Defense Category	**Effectiveness**
Basic Injection Attack Prevention	100% blocked
Completion Attack Prevention	100% blocked (0% ASR)
GCG-based Attack Reduction	42-44% reduction
TAP Attack Mitigation	88-64% effectiveness
Multilingual Attack Blocking	100% blocked

5.3. Paper 3: Multi-Layer Security Framework

Table 5.3: Comprehensive Security Framework Results

Security Metric	**Performance**
Overall Security Accuracy	98.0%
Injection Attack Prevention	98.7%
Dangerous Content Blocking	100.0%
Cross-LLM Consensus Rate	95.3%
False Positive Rate	1.2%

5.4. Comparative Analysis

Table 5.4: Cross-Study Performance Comparison

Metric	**Paper 1**	**Paper 2**	**Paper 3**
Maximum Attack Success	68.9%	56%*	2.0%*
Best Defense Rate	N/A	100%	100%
Automation Impact	95.2%	N/A	N/A
False Positives	N/A	N/A	1.2%

*Estimated from defense effectiveness data

Key Findings:

- Jailbreak attacks remain highly effective (45.7-68.9% success)
- Structured defenses show perfect prevention against basic attacks
- Multi-layer frameworks achieve near-perfect security (98% accuracy)
- Automated jailbreak generation poses significant threat (95.2% success)
- Cross-model validation enhances reliability (95.3% consensus)

6. Discussion & Recommendations

6.1. Synthesis of Findings

The three studies collectively map the LLM security landscape, revealing critical vulnerabilities and defensive countermeasures. Paper 1 exposed the ease of jailbreaking, even by non-experts, and the high scalability of attacks (95.2% auto-generation success). Paper 2 demonstrated that structured, syntactic defenses can be highly effective, achieving 100% prevention against basic injection attacks. Paper 3 proved that a multi-layered, holistic framework offers the most robust protection, reaching 98% overall security accuracy.

6.2. Limitations and Practical Challenges

Paper 1 (Jailbreak Analysis)

- Relied on **Manual Evaluation**, leading to scalability concerns
- Findings were constrained by a **Narrow Participant Group**
- Offered limited **Interpretability** into why specific prompts succeed

Paper 2 (Structured Defense)

- Effectiveness was tested in a **Restricted Use Case**
- Showed vulnerability to advanced **Optimization Attacks** (e.g., GCG)
- Method can introduce **Structural Overhead**
- Requires **High Resource Requirements** for training

Paper 3 (Multi-Layer Framework)

- Faces **Initial Implementation Complexity**
- Significant **Computational Resource Requirements**
- Heavy **Dependency on Multiple API Services** increases cost and latency
- Has seen **Limited Real-world Deployment Testing**

6.3. Recommendations for a Robust Defense Strategy

To build secure LLM applications, a proactive and layered approach is essential:

1. **Proactive Testing & Evaluation:** Develop AI Safety Evaluators using automated red-teaming for Proactive Defense Testing
2. **Architectural Security:** Harden systems by implementing Advanced Training against adversarial examples and building Architectural Security with structured query parsing
3. **Holistic Protocols:** Implement Multi-Layered Security Protocols that combine input filtering, in-process detection, and output validation. Promote a Standardized Framework for consistent security practices
4. **Operational Excellence:** Maintain Continuous Threat Monitoring and Multi-Language Expansion to adapt to new threats. Focus on Resource Optimization to make security scalable
5. **Community & Standards:** Encourage Industry Standards Development and the release of Open-Source Components to foster collaborative improvement of LLM security

7. Conclusion

This report synthesized findings from three pivotal studies on Large Language Model security. The evidence is clear: LLMs are critically vulnerable to jailbreaks and injection attacks, threats that are both potent and rapidly scalable through automation.

While robust defenses exist—from targeted syntactic solutions to comprehensive multi-layered frameworks—no single approach is a panacea. The most critical conclusion is that effective security demands a **proactive, defense-in-depth strategy**. This involves continuous adversarial testing, architectural hardening, and the integration of multiple defensive layers to protect against a diverse and evolving threat landscape.

The future of trustworthy AI depends not on finding a single silver bullet, but on building resilient, adaptable, and layered security ecosystems that can withstand emerging challenges while maintaining model utility and performance.

References

[1] Zhiyuan Yu, Xiaogeng Liu, Shunning Liang, Zach Cameron, Chaowei Xiao, and Ning Zhang. Don't listen to me: Understanding and exploring jailbreak prompts of large language models. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 4675–4692, 2024.

[2] Sizhe Chen, Julien Piet, Chawin Sitawarin, and David Wagner. {StruQ}: Defending against prompt injection with structured queries. In *34th USENIX Security Symposium (USENIX Security 25)*, pages 2383–2400, 2025.

[3] Tunahan Gokcimen and Bihter Das. A novel system for strengthening security in large language models against hallucination and injection attacks with effective strategies. *Alexandria Engineering Journal*, 123:71–90, 2025.

[4] Xiaogeng Liu, Zhiyuan Yu, Hao Zhang, Yinghao Zhang, Muhao Cheng, and Chaowei Xiao. Autodan: Generating stealthy jailbreak prompts on aligned large language models. In *Proceedings of the 2024 IEEE Symposium on Security and Privacy*, pages 1–18, 2024.

[5] Jiongxiao Wang, Zichen Liu, Keun Hee Park, Chaowei Zhu, and Chaowei Xiao. Jailbreak and guard aligned language models with only few words. In *Advances in Neural Information Processing Systems*, volume 36, pages 1–15, 2024.

[6] Seongwon Shin, Minsoo Park, Gyuhyeon Kim, and Youngki Lee. Prompt shield: A runtime detection framework for adversarial prompts against llms. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, pages 1–16, 2025.