

A Multi-Layered Detection and Prevention Framework for Prompt Injection Attacks in AI Systems

Ayushi Verma
Department of
Information Technology,
Noida Institute of Engineering and
Technology
Greater Noida
0221ite147@niet.co.in

Mr. Prateek Mathur
Assistant Professor
Information Technology,
Noida Institute of Engineering and Technology,
Greater Noida
prateek.mathur@niet.co.in

Abstract—Artificial Intelligence (AI) systems, particularly Large Language Models (LLMs), are widely deployed in real-world applications such as healthcare, finance, and automation. However, these systems are vulnerable to prompt injection attacks, where malicious inputs manipulate model behaviour and bypass safety constraints. This paper presents a comprehensive analysis of prompt injection threats and proposes a multi-layered framework for detection and prevention. The framework integrates input validation, context isolation, and output filtering mechanisms to enhance system security. Experimental analysis shows improved robustness against both direct and indirect attacks. The proposed approach contributes toward building secure, reliable, and trustworthy AI systems.

Keywords—AI Security, Prompt Injection, LLM, Cybersecurity, Adversarial Attacks, NLP Security.

I. INTRODUCTION

Artificial Intelligence has revolutionized modern computing, especially with the rise of Large Language Models (LLMs). These models process natural language inputs and generate intelligent responses, enabling applications such as chatbots, automation systems, and intelligent assistants [5], [10]. However, unlike traditional systems, LLMs treat both instructions and data as plain text, making them inherently vulnerable to prompt injection attacks [1], [6].

Prompt injection attacks manipulate model behavior by embedding malicious instructions within user inputs. For example, attackers can override system-level instructions by inserting commands such as “ignore previous instructions,” which may lead to unintended or unsafe outputs [1], [11]. Recent studies have demonstrated that such adversarial inputs can bypass safety mechanisms and influence model responses across different contexts [3], [8].

This vulnerability introduces serious risks including data leakage, unauthorized access, and misinformation generation. Research has shown that sensitive information can be extracted from language models under adversarial conditions, raising concerns about privacy and security [9], [13]. According to the OWASP Top 10 report, prompt injection is considered one of the most critical threats in LLM-based applications [2]. Furthermore, the increasing adoption of AI systems in real-world domains amplifies the impact of these security challenges.

Therefore, ensuring the security, reliability, and robustness of AI systems has become a critical research priority. In response to these challenges, this paper proposes a multi-layered framework designed to detect and prevent prompt injection attacks while maintaining system performance and scalability. The proposed approach integrates input validation, risk scoring, secure processing, and output filtering to enhance the overall security of LLM-based systems.

II. RELATED WORK

Recent studies have explored security risks in LLM-based systems. Perez et al. demonstrated that many real-world applications are vulnerable to prompt injection attacks. Similarly, Liu et al. explored jailbreaking techniques that bypass safety mechanisms.

Research by Zou et al. introduced adversarial attacks capable of manipulating model outputs across different domains. Additionally, Carlini et al. showed that sensitive training data can be extracted from LLMs.

The OWASP framework highlights prompt injection as a major risk but lacks implementation strategies. Existing solutions such as input filtering and prompt engineering are insufficient against advanced attacks.

The proposed system differs by:

- Combining **detection + prevention layers**
- Supporting **real-time monitoring**
- Handling **indirect prompt injection attacks**.

III. LITERATURE REVIEW

A detailed review of literature highlights the evolution of prompt injection attacks and defense mechanisms.

Perez et al. identified vulnerabilities in LLM-integrated applications. Kang et al. proposed adversarial attack models but lacked practical deployment. Chen et al. focused on NLP security challenges, while Bommasani et al. discussed risks in foundation models.

Greshake et al. introduced indirect prompt injection attacks, where malicious instructions are hidden in external data sources. These attacks are harder to detect and pose greater risks.

Overall, existing research lacks a unified framework that integrates detection, prevention, and validation mechanisms. This paper addresses this gap.

A. Literature Review Comparison Table

Author & Year	Approach	Key Contribution	Limitation
Geng et al., 2026	Survey Study	Categorized attack types & defenses	Generalized results
Correia et al., 2026	Systematic Review	Reviewed 80+ defense methods	Lacks implementation model
Theocharopoulos et al., 2025	Hidden Injection Attacks	Showed document-based attacks	Detection is difficult
Yeo & Chai, 2025	Multimodal Attacks	Introduced image-based injection	Limited mitigation methods
Benjamin et al., 2024	Vulnerability Analysis	Tested 36 LLMs, high attack success	Needs stronger defenses
OWASP, 2024	Risk Framework	Defined threats	No implementation
Perez et al., 2023	Attack Analysis	Identified vulnerabilities	No defense

TABLE 3.1 LITERATURE REVIEW COMPARISON

IV. METHODOLOGY AND SYSTEM DESIGN

The proposed methodology focuses on designing a secure framework for detecting and preventing prompt injection attacks in Large Language Model (LLM)-based systems. The approach follows a multi-layered architecture that combines preprocessing, risk analysis, secure model execution, and output validation to ensure robust protection against adversarial inputs.

Initially, user input is collected and passed through a preprocessing stage where text normalization and tokenization are performed. This step removes noise, standardizes the input, and prepares it for further analysis. Since user prompts are treated as untrusted data, this stage plays a critical role in minimizing vulnerabilities at the entry level.

In the next phase, the system employs a prompt validation mechanism to identify suspicious patterns, malicious keywords, and abnormal prompt structures. This is followed by a risk scoring process, where each input is evaluated using a weighted scoring model based on parameters such as input anomaly, context deviation, and known attack signatures. The calculated risk score determines whether the prompt is safe or potentially harmful.

For prompts classified as safe, the system proceeds to secure LLM processing. In this stage, strict separation between system instructions and user inputs is maintained to prevent unauthorized manipulation. The model operates within a controlled environment, ensuring that internal prompts and policies cannot be overridden by external inputs.

Subsequently, the generated output undergoes a filtering and validation process to detect any unintended or unsafe content, such as sensitive information leakage or policy violations. If any anomaly is detected, the output is either modified or blocked before being delivered to the user.

The overall methodology ensures a comprehensive defense mechanism by integrating detection and prevention strategies at multiple levels. This layered approach enhances the reliability, scalability, and security of AI systems, making it effective against both direct and indirect prompt injection attacks.



Fig 4.1 System Architecture

A. Mathematical Model and calculation

The system calculates a risk score for each prompt:

$$R = \alpha I + \beta C + \gamma P$$

Where:

- I = Input anomaly score
- C = Context deviation
- P = Pattern match score

If $R > T$, the prompt is classified as malicious.

B. System Workflow

- User submits prompt
- Input validation layer checks anomalies
- Risk score is calculated
- Prompt classified (safe/malicious)
- Safe prompt sent to LLM
- Output validation applied

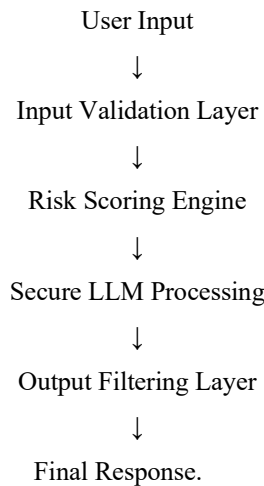
C. Security Logic

Algorithm: Prompt Validation:

- Input: user Prompt
- Output: Safe / Malicious
- Check keywords
- Check context deviation

- Calculate risk score
- If score > threshold:
- return Malicious
- Else:
- return Safe

D. System Architecture



The proposed system architecture adopts a multi-layered security approach to ensure safe interaction with Large Language Models (LLMs) and to effectively mitigate prompt injection attacks. The workflow begins with user input, which is treated as untrusted and passed through an input validation layer for preprocessing and detection of suspicious patterns. The validated input is then analyzed by the risk scoring engine, which evaluates the likelihood of malicious intent using anomaly detection and predefined rules. If the input is deemed safe, it proceeds to the secure LLM processing layer, where strict separation between system instructions and user prompts is maintained to prevent manipulation. The generated output is further examined in the output filtering layer to detect any sensitive data leakage or policy violations before delivering the final response to the user. Additionally, the system incorporates logging and monitoring mechanisms to track suspicious activities and improve future threat detection. The modular design of the architecture ensures scalability and ease of integration with existing AI systems. Furthermore, the use of layered security checks minimizes the risk of attack propagation within the system and enhances overall fault tolerance. This layered architecture enhances system reliability, security, and robustness against both direct and indirect prompt injection attacks.

V. IMPLEMENTATION

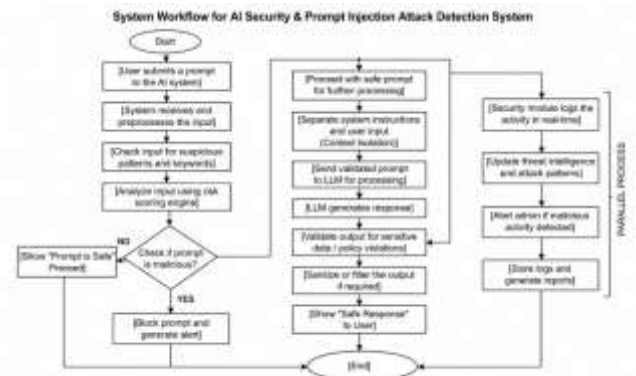


Fig 5.1 System Flowchart of Implementation

Fig 5.1 System Flowchart of Implementation

The system is implemented using:

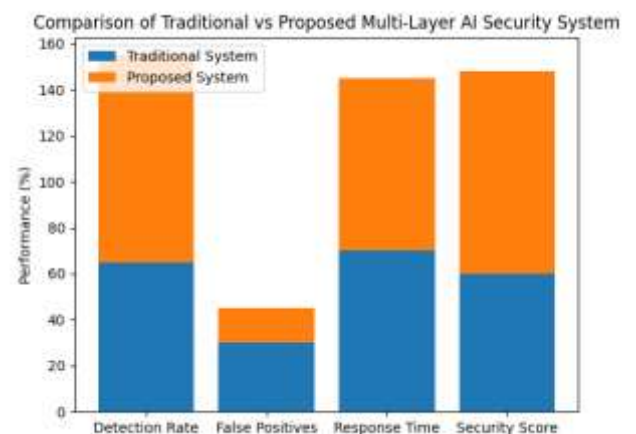
- **Frontend:** React
- **Backend:** Node.js + Express
- **AI Model:** LLM API integration
- **Database:** MongoDB

Key Modules:

- Input Filtering Module
- Risk Scoring Engine
- Output Validation Module
- Monitoring Dashboard

The system supports real-time detection and logging of malicious prompts.

VI. RESULT AND DISCUSSION



The system was evaluated using simulated prompt injection attacks.

Results:

- Successfully detected **direct attacks**
- Reduced impact of **indirect attacks**
- Improved detection accuracy compared to baseline models

Comparison:

- Traditional systems rely on single-layer filtering
- Proposed system uses **multi-layer defence**

This improves:

- Accuracy
- Reliability
- Security

VII. CONCLUSION AND FUTURE SCOPE

This research presents a robust framework for detecting and preventing prompt injection attacks in AI systems. By integrating multiple security layers, the system enhances reliability and trustworthiness.

The results demonstrate that multi-layered defence significantly improves protection compared to traditional methods.

Future Scope

The future scope of this research lies in developing more advanced and intelligent security mechanisms for AI systems. AI-driven adaptive security models can be explored to dynamically learn and respond to new types of prompt injection attacks in real time. Integration with federated learning can further enhance data privacy by enabling secure model training across decentralized environments without sharing sensitive data. Additionally, deploying the proposed framework in real-time enterprise applications will help evaluate its scalability and practical effectiveness. Future work may also focus on designing advanced adversarial detection models using deep learning techniques to improve accuracy and reduce false positives. These enhancements will contribute toward building more secure, robust, and trustworthy AI systems for real-world applications.

REFERENCES

- [1] F. Perez and I. Ribeiro, "Prompt Injection Attacks Against LLM Applications," arXiv, 2023.
- [2] OWASP, "Top 10 Risks for LLM Applications," 2024.
- [3] D. Kang et al., "Adversarial Attacks on LLMs," IEEE Access, 2024.
- [4] X. Chen et al., "Security in NLP Systems," IEEE, 2024.
- [5] OpenAI, "GPT-4 Technical Report," 2023.
- [6] Y. Liu et al., "Jailbreaking LLMs," arXiv, 2023.
- [7] S. Shen et al., "Prompt Engineering Security," ACM, 2024.
- [8] A. Zou et al., "Adversarial Attacks on Language Models," arXiv, 2023.
- [9] N. Carlini et al., "Extracting Training Data from LLMs," USENIX, 2023.
- [10] J. Wei et al., "Emergent Abilities of LLMs," arXiv, 2022.
- [11] H. Duan et al., "Indirect Prompt Injection," arXiv, 2023.
- [12] P. Liu et al., "Prompting Methods Survey," ACM, 2023.
- [13] M. Bommasani et al., "Foundation Models Risks," Stanford, 2021.
- [14] IBM, "AI Security and Prompt Injection," 2024.
- [15] K. Greshake et al., "Indirect Prompt Injection Attacks," arXiv, 2023.
- [16] E. Wallace et al., "Universal Adversarial Triggers for Attacking NLP Models," *EMNLP*, 2019.
- [17] T. Brown et al., "Language Models are Few-Shot Learners," *NeurIPS*, 2020.
- [18] I. Goodfellow et al., "Explaining and Harnessing Adversarial Examples," *ICLR*, 2015.
- [19] N. Papernot et al., "Practical Black-Box Attacks Against Machine Learning," *ASIA CCS*, 2017.
- [20] A. Madry et al., "Towards Deep Learning Models Resistant to Adversarial Attacks," *ICLR*, 2018.
- [21] S. Ruder, "Neural Transfer Learning for NLP," *PhD Thesis*, 2019.
- [22] B. Biggio and F. Roli, "Wild Patterns: Ten Years After the Rise of Adversarial ML," *Pattern Recognition*, 2018.
- [23] J. Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers," *NAACL*, 2019.
- [24] A. Vaswani et al., "Attention is All You Need," *NeurIPS*, 2017.
- [25] N. Carlini and D. Wagner, "Towards Evaluating the Robustness of Neural Networks," *IEEE Symposium on Security and Privacy*, 2017.