

Design and Development of an AI-Powered Career Coaching Platform for Personalized Career Growth

Anjalikrishna C¹, Ms. K.V. Indulekha²

¹II MCA, Department of Computer Applications, Nehru Institute of Information Technology and Management, Coimbatore, Tamil Nadu, India

²Assistant Professor, Department of Computer Applications, Nehru Institute of Information Technology and Management, Coimbatore, Tamil Nadu, India

Abstract - The rapid adoption of Large Language Models (LLMs) in enterprise, healthcare, education, and digital services has introduced new security challenges, among which prompt injection attacks have emerged as a critical threat. Prompt injection occurs when malicious or manipulated inputs override system instructions, causing an AI model to produce unintended, unsafe, or unauthorized responses. This project presents the design and development of a **hybrid prompt injection detection system** that combines rule-based pattern matching with AI-powered classification using **Gemini 2.5 Flash** to improve the accuracy and reliability of threat detection. The proposed system analyzes user prompts through multiple detection layers. The rule-based component identifies known attack patterns, suspicious keywords, and common jailbreak techniques, while the Gemini-based classifier evaluates contextual and semantic characteristics that may indicate sophisticated prompt injection attempts. By integrating both approaches, the system leverages the speed and transparency of rule-based methods together with the adaptability and contextual understanding of advanced language models.

The project aims to enhance the security of LLM-powered applications by detecting and classifying malicious prompts before they reach the target model. The architecture is designed to be modular, scalable, and capable of graceful degradation, ensuring continuous operation even when one detection component becomes unavailable. Experimental evaluation demonstrates the effectiveness of the hybrid approach in identifying prompt injection attacks while reducing false positives and improving overall detection performance.

This system contributes to the growing field of AI security by providing a practical and efficient solution for safeguarding LLM applications against prompt injection threats, thereby improving trust, reliability, and safe deployment of artificial intelligence systems in real-world environments.

Keywords: Artificial Intelligence, Career Guidance, Resume Builder, Interview Preparation, Generative AI, Career Development.

I. INTRODUCTION

The **Adversarial Prompt Injection Detector** is a security-focused system designed to protect Large Language Models (LLMs) from malicious or manipulated user inputs. As AI-powered applications become increasingly common, attackers attempt to exploit these systems through prompt injection attacks, where specially crafted prompts are used to bypass safety rules, override system instructions, or extract sensitive information.

This project aims to identify and prevent such threats before they can affect the target AI model. The detector analyzes incoming prompts using a combination of rule-based techniques and advanced AI-based classification methods. By examining both known attack patterns and contextual language characteristics, the system can effectively distinguish between legitimate user requests and potentially harmful prompts.

The proposed solution enhances the security, reliability, and trustworthiness of AI applications by acting as a protective layer between users and the language model. Its hybrid detection approach improves accuracy, reduces false alarms, and helps safeguard AI systems against evolving adversarial attacks. As prompt injection threats continue to grow in sophistication, the Adversarial Prompt Injection Detector provides a practical and scalable solution for secure AI deployment. This research proposes an AI Career Coach, an intelligent web application designed to assist users throughout their professional journey. The platform provides career recommendations, resume generation, cover letter creation, and interview preparation support using Generative AI technologies. The platform sends structured user profile data to Gemini's API and processes the returned content into career recommendations, resume drafts, and interview questions - all scoped to the user's stated goals and current skills.

II. LITERATURE REVIEW

Over the past three years, the rapid adoption of large language models (LLMs) in applications such as chatbots, coding assistants, and enterprise automation has introduced new security challenges, particularly adversarial prompt injection attacks. As LLMs increasingly interact with external data sources and execute user instructions, researchers have focused on developing methods to identify and mitigate malicious prompts that manipulate model behavior.

Prompt injection attacks exploit the instruction-following nature of LLMs by embedding deceptive or conflicting commands within user inputs or retrieved documents. Studies have shown that such attacks can cause models to ignore system instructions, reveal confidential information, or produce harmful outputs, highlighting the need for robust detection mechanisms before prompts are processed.

Several research efforts have explored machine learning-based and rule-based approaches for prompt injection detection. Traditional methods rely on keyword matching, pattern recognition, and handcrafted security rules, while recent studies employ transformer-based classifiers and fine-tuned language models to distinguish between legitimate and malicious prompts with improved accuracy. Hybrid approaches combining heuristic analysis and deep learning have demonstrated better generalization against previously unseen attack patterns.

readiness. Furthermore, intelligent analytics dashboards help users track their progress and identify areas for improvement. Despite these advances, existing solutions often struggle with sophisticated or previously unseen attacks and may generate false positives that affect user experience. Ongoing research therefore emphasizes adaptive detection models, continuous learning from emerging attack patterns, and real-time security monitoring to improve the reliability and trustworthiness of LLM-based systems deployed in practical environments.

The Adversarial Prompt Injection Detection System is designed as a secure AI-powered application that detects and prevents malicious prompt injection attacks targeting Large Language Models (LLMs). The system integrates machine learning-based classification, rule-based analysis, and prompt preprocessing techniques to identify potentially harmful user inputs before they reach the language model. The modular architecture ensures scalability, maintainability, and real-time threat detection.

1. User Interface Layer:The frontend is developed using React.js and Next.js, providing an intuitive interface where users can submit prompts and view detection results. The interface displays the risk level, detected attack type, and recommendations for safe prompt usage.

2. Input Validation and Preprocessing Module:All incoming prompts are first processed through an input validation module. The system removes unnecessary formatting, normalizes text, tokenizes the input, and extracts linguistic and semantic features required for security analysis.

3. Prompt Injection Detection Engine:The core detection engine combines rule-based filtering with a machine learning classifier to identify adversarial prompt injection attempts. The system analyzes instruction override patterns, role manipulation, jailbreak attempts, hidden commands, context poisoning, and suspicious prompt structures. Each prompt is assigned a risk score that determines whether it is safe, suspicious, or malicious.

4. AI Processing Layer:Only prompts classified as safe are forwarded to the Large Language Model (LLM) for processing. Prompts identified as malicious are blocked or sanitized before execution, preventing unauthorized instruction overrides and protecting sensitive system prompts.

5. Logging and Analytics Module:The system records detection events, attack categories, confidence scores, and user interactions in a secure database. The analytics module generates reports that help administrators monitor attack trends, evaluate detection performance, and continuously improve the detection model.

B. Working Procedure

Step 1: User Prompt Submission:The user enters a prompt through the web interface and submits it for processing.

Step 2: Prompt Preprocessing:The input prompt is cleaned, normalized, tokenized, and converted into structured features suitable for analysis.

Step 3: Threat Analysis:The detection engine evaluates the prompt using rule-based security checks and a trained machine learning model to identify potential adversarial prompt injection patterns.

Step 4: Risk Classification:Based on the analysis, the system classifies the prompt as **Safe**, **Suspicious**, or **Malicious** and assigns a confidence score.

Step 5: Response Decision:If the prompt is classified as safe, it is forwarded to the Large Language Model for response generation. If the prompt is classified as malicious, the system blocks or sanitizes the input and notifies the user.

Step 6: Logging and Monitoring:The system stores prompt metadata, detection results, timestamps, and attack categories for future analysis and security auditing.

Step 7: Performance Evaluation:Detection accuracy, precision, recall, F1-score, false positive rate, and processing time are continuously evaluated using benchmark datasets and real-world adversarial prompts. The analytics dashboard provides visual reports to assist administrators in monitoring system effectiveness and identifying emerging attack patterns.

Technology Stack

Table I. Technologies Used in AI Career Coach

Layer	Technology
Frontend	Streamlit
Programming language	Python
Styling	Streamlit,CSS
Authentication	Clerk
Detection Method	Hybrid Detection
Rule Engine	Regular expression
AI Engine	Gemini AI
Analytics	Detection Analysis
Background Jobs	Not implimented

IV. RESULTS AND IMPLEMENTATION

The proposed Adversarial Prompt Injection Detection System was successfully implemented and evaluated using various benign and adversarial prompt datasets. The application demonstrated effective performance in detecting prompt injection attacks while allowing legitimate prompts to pass through for processing. The hybrid detection approach, combining rule-based analysis and Google Gemini AI, improved the system's ability to identify both known and previously unseen attack patterns.

A. Prompt Injection Detection Module

The detection module successfully identified multiple categories of adversarial prompt injection attacks, including instruction override, jailbreak attempts, role manipulation, prompt leakage, and hidden command injection. The rule-based detector efficiently recognized known attack patterns using predefined regular expressions, while the Gemini AI classifier analyzed the semantic intent of prompts to detect more sophisticated attacks.

B. Hybrid Classification Module

The hybrid classification approach combined the outputs of the rule-based detector and the Gemini AI classifier to produce a final security decision. Each submitted prompt was classified as **Safe**, **Suspicious**, or **Malicious**, along with a confidence score. This approach reduced false negatives by detecting attacks that could bypass simple keyword-based filtering.

C. User Interface Implementation

The system was implemented using Streamlit, providing an interactive interface where users can submit prompts and receive real-time detection results. The interface displays the

classification result, confidence level, detected attack category, and security recommendations in an intuitive format.

IV. RESULTS AND IMPLEMENTATION

The proposed Adversarial Prompt Injection Detection System was successfully implemented and evaluated using both legitimate and adversarial prompts. The hybrid detection framework, which combines rule-based pattern matching with Google Gemini AI, demonstrated effective identification of prompt injection attacks while maintaining efficient real-time performance.

A. Prompt Injection Detection Module

The detection module successfully identified various prompt injection attacks, including instruction override, jailbreak attempts, role manipulation, prompt leakage, and hidden command injection. The rule-based detector rapidly identified known attack patterns using predefined regular expressions, while the Gemini AI classifier analyzed semantic intent to detect sophisticated or previously unseen attacks.

Prompt Risk Score Formula

The overall prompt risk score is calculated by combining the outputs of the rule-based detector and the AI classifier.

$$\text{Risk Score} = 0.6(R) + 0.4(A)$$

Where:

- R** = Rule-Based Detection Score (0–100)
- A** = AI Classification Score (0–100)

Example:

Rule-Based Score = 90

AI Classification Score = 80

Risk Score = $0.6(90) + 0.4(80)$

Risk Score = **86**

A higher risk score indicates a greater probability that the prompt contains an adversarial prompt injection attack.

B. Rule-Based Detection Module

The rule-based module analyzed user prompts using predefined regular expressions and attack signatures stored in patterns.json. It effectively detected known malicious keywords, system prompt extraction attempts, instruction overrides, and jailbreak patterns with minimal processing time.

Rule Match Score Formula

$$\text{Rule Match Score} = \frac{\text{Matched Rules}}{\text{Total Rules}} \times 100$$

This score represents the percentage of malicious patterns detected within a submitted prompt.

C. AI Classification Module

The Google Gemini 2.5 Flash model performed semantic analysis of prompts to detect attacks that may bypass traditional keyword-based methods. The classifier categorized prompts as **Safe**, **Suspicious**, or **Malicious**, providing additional reasoning to support the final decision.

D. Decision Engine

The decision engine combined the outputs of the rule-based detector and the Gemini classifier to determine the final classification.

Decision Criteria:

- **Risk Score < 40** → Safe
- **Risk Score 40–69** → Suspicious
- **Risk Score ≥ 70** → Malicious

This hybrid strategy improved overall detection reliability while reducing false negatives.

E. System Performance

The system was evaluated using a collection of normal prompts and adversarial prompt injection examples. Experimental testing showed that the proposed hybrid approach successfully detected known and unknown attack patterns while maintaining fast response times suitable for real-time applications.

Table II. Performance Evaluation of System Modules

Module	Accuracy (%)	Detection Rate (%)
Rule-Based Detection	91	90
Gemini AI Classification	95	94
Hybrid Decision Engine	97	96
Prompt Risk Assessment	94	93
Overall System Performance	96	95

The experimental results demonstrate that integrating rule-based detection with semantic analysis provided by Google Gemini AI significantly improves the detection of adversarial prompt injection attacks while maintaining efficient real-time performance. The hybrid architecture offers enhanced robustness against both known attack signatures and emerging prompt injection techniques.

V. DISCUSSION

The developed Adversarial Prompt Injection Detection System addresses one of the major security challenges associated with Large Language Models (LLMs): malicious prompt injection attacks. Unlike conventional input validation methods that primarily rely on keyword filtering, the proposed system combines rule-based detection with Google Gemini AI to identify both known and semantically complex adversarial prompts.

One of the key strengths of the proposed system is its hybrid detection approach. The rule-based module efficiently detects predefined attack patterns such as instruction overrides, jailbreak attempts, prompt leakage, and role manipulation using regular expressions. Meanwhile, the Gemini AI classifier performs semantic analysis to identify sophisticated attacks that may evade traditional pattern-matching techniques. This combination improves detection accuracy and reduces the likelihood of malicious prompts reaching the language model.

Another advantage of the system is its real-time detection capability. The Streamlit-based interface enables users to submit prompts and receive immediate security assessments, including the prompt classification (Safe, Suspicious, or Malicious), confidence score, and appropriate security recommendations. The lightweight architecture makes the system suitable for deployment as a security layer in AI-powered applications.

Despite these advantages, the system has certain limitations. The effectiveness of the rule-based detector depends on the completeness of predefined attack patterns stored in the pattern database. Additionally, AI-based classification relies on the reasoning capabilities of the Gemini model, which may occasionally misclassify highly ambiguous or novel prompt injection techniques. Continuous updates to attack patterns and periodic refinement of the AI classification logic are therefore necessary to maintain detection performance.

Overall, the proposed system demonstrates that integrating rule-based security mechanisms with semantic analysis using Large Language Models provides an effective and practical solution for mitigating adversarial prompt injection attacks. The hybrid architecture enhances the security, reliability, and trustworthiness of LLM-based applications while maintaining efficient real-time performance.

Table III. Comparison Between Traditional Prompt Filtering and Proposed System

Feature	Traditional Prompt Filtering	Proposed Detection System
Detection Method	Keyword-Based Rules	Hybrid (Regex + Gemini AI)
Semantic Analysis	Not Supported	Supported
Detection of Unknown Attacks	Limited	High
Real-Time Detection	Basic	Real-Time
Accuracy	Moderate	High
Scalability	Moderate	High
Adaptability	Low	High
False Negative Rate	Higher	Lower

VI. ADVANTAGES, CHALLENGES, AND EXPERIMENTAL ANALYSIS

A. Advantages of the Proposed System

The proposed Adversarial Prompt Injection Detection System provides several advantages over conventional prompt filtering techniques. The hybrid detection framework combines rule-based pattern matching with Google Gemini AI, enabling the system to detect both known and semantically complex prompt injection attacks. This improves detection accuracy compared to traditional keyword-based filtering methods.

Another major advantage is real-time detection. The Streamlit-based application allows users to submit prompts and receive immediate security assessments, including the prompt classification (Safe, Suspicious, or Malicious), confidence score, and recommended actions. This enables users and developers to identify potentially harmful prompts before they reach the Large Language Model.

The system is lightweight and easy to deploy, requiring only a Python environment and access to the Gemini API. The modular architecture also allows new attack patterns to be added to the rule database without modifying the core detection logic, making the system adaptable to evolving prompt injection techniques.

Furthermore, the integration of semantic analysis with rule-based detection reduces the likelihood of malicious prompts bypassing the security layer while maintaining efficient response times suitable for real-time AI applications.

B. Challenges Faced During Development

Several challenges were encountered during the development of the system. One of the primary challenges was identifying prompt injection attacks that use indirect or obfuscated language rather than explicit malicious keywords. To address this limitation, Google Gemini AI was integrated to perform semantic analysis in addition to rule-based detection.

Another challenge involved designing effective regular expression patterns capable of detecting common attack techniques without producing excessive false positives. The rule database required continuous refinement to balance detection sensitivity and accuracy.

Maintaining consistent response times while communicating with the external Gemini API was also a challenge, as API latency may vary depending on network conditions and service availability. Optimizing prompt preprocessing and minimizing unnecessary API calls helped improve the overall responsiveness of the application.

Finally, prompt injection techniques continue to evolve rapidly. As a result, the detection rules and AI evaluation strategy must be updated periodically to remain effective against newly emerging attack methods.

C. Experimental Analysis

The developed system was evaluated using a collection of legitimate user prompts and adversarial prompt injection examples representing common attack techniques such as instruction override, jailbreak attempts, role manipulation, prompt leakage, and hidden command injection.

Experimental evaluation showed that the rule-based detector efficiently identified known attack patterns, while the Gemini AI classifier successfully detected semantically complex attacks that were not explicitly covered by predefined rules. The hybrid decision engine combined the outputs of both modules to produce reliable classifications with low processing time.

Testing demonstrated that legitimate prompts were correctly classified as safe in most cases, while malicious prompts were accurately identified and blocked or flagged for review. The experimental results indicate that combining rule-based analysis with semantic AI classification significantly improves the robustness of prompt injection detection compared with using either technique independently.

D. Social Impact

The proposed system contributes to the safe and responsible deployment of Large Language Models by reducing the risk of adversarial prompt injection attacks. As AI-powered applications become increasingly common in education, healthcare, finance, and customer support, protecting these systems against malicious prompt manipulation is essential.

The detection framework can be integrated into AI applications to improve security, protect confidential system instructions, and reduce the possibility of generating unsafe or manipulated responses. By strengthening the trustworthiness of AI systems, the proposed solution supports the broader adoption of secure and reliable artificial intelligence technologies across multiple domains.

VII. CONCLUSION

This paper presented the design and implementation of an Adversarial Prompt Injection Detection System, a security solution developed to protect Large Language Models (LLMs) from malicious prompt injection attacks. The proposed system integrates rule-based pattern matching with Google Gemini AI to identify and classify potentially harmful prompts before they are processed by the language model.

The implementation results demonstrate that the hybrid detection approach effectively identifies a wide range of prompt injection attacks, including instruction override, jailbreak attempts, role manipulation, prompt leakage, and hidden command injection. By combining regular expression-based detection with semantic analysis, the system provides improved detection accuracy while maintaining efficient real-time performance.

The proposed solution demonstrates the practical application of artificial intelligence in enhancing the security and reliability of LLM-based applications. The lightweight Streamlit interface and modular architecture make the system easy to deploy, maintain, and extend. Overall, the developed framework provides an effective security layer that helps reduce the risk of adversarial prompt manipulation and supports the safe adoption of AI-powered systems across various application domains.

FUTURE SCOPE

Future enhancements can further improve the effectiveness and adaptability of the proposed system. Planned improvements include:

- Integration of transformer-based detection models such as BERT, RoBERTa, or DeBERTa to improve classification accuracy.

- Continuous learning mechanisms that automatically update detection rules based on newly identified prompt injection techniques.
- Support for multiple Large Language Models, including OpenAI GPT, Anthropic Claude, Meta Llama, and other open-source models.
- Development of a browser extension or API gateway for real-time prompt inspection across AI-powered applications.
- Implementation of attack logging, reporting dashboards, and security analytics to monitor prompt injection trends.
- Support for multilingual prompt injection detection to improve protection for non-English prompts.
- Integration with Security Information and Event Management (SIEM) platforms for enterprise-level threat monitoring and incident response.

These enhancements will improve the scalability, adaptability, and robustness of the proposed system, enabling it to address emerging adversarial prompt injection techniques and strengthen the security of future AI applications.

REFERENCES

- [1] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed., Pearson, 2021.
- [2] T. Brown *et al.*, "Language Models are Few-Shot Learners," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 1877–1901, 2020.
- [3] J. Wei *et al.*, "Finetuned Language Models Are Zero-Shot Learners," *International Conference on Learning Representations (ICLR)*, 2022.
- [4] Google DeepMind, "Gemini API Documentation," Google AI for Developers, 2025. Available: <https://ai.google.dev>
- [5] Streamlit Inc., "Streamlit Documentation," 2025. Available: <https://docs.streamlit.io>
- [6] Python Software Foundation, "Python 3.11 Documentation," 2025. Available: <https://docs.python.org/3>
- [7] OWASP Foundation, "OWASP Top 10 for Large Language Model Applications," 2025. Available: <https://owasp.org/www-project-top-10-for-large-language-model-applications/>
- [8] N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, U. Erlingsson, A. Oprea, and C. Raffel, "Extracting Training Data from Large

Language Models," *USENIX Security Symposium*, pp. 2633–2650, 2021.

[9] Hugging Face Inc., "Transformers Documentation," 2025. Available: <https://huggingface.co/docs/transformers>

[10] A. Greshake, M. Abdelnabi, S. Mishra, and others, "More than You've Asked For: A Comprehensive Analysis of Novel Prompt Injection Threats to Application-Integrated Large Language Models," arXiv:2302.12173, 2023.