IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155228

"A Survey on Prompt Injection and Jailbreak Defenses in Large Language Models"

Anushree K N[1], Vedanth M[2], Amulya H[3], Chirayu Gowda[4], Prof. Meghashree C[5]

Dept. of CSE(ICB), KSIT, Karnataka, India[1,2,3,4]

Asst. Prof., Dept. of CSE(ICB), KSIT, Karnataka, India[5]

Abstract: Large language models are now embedded in healthcare, education, and enterprise software at a scale that would have seemed unlikely just a few years ago. This rapid adoption has introduced a category of security threats that conventional mechanisms were never designed to handle: prompt injection and jailbreak attacks. Unlike traditional exploits, these attacks do not target code; they manipulate natural language itself to push a model past its safety constraints, extract information it should not reveal, or produce outputs its developers explicitly prohibited. What makes these threats particularly difficult to counter is that adversarial intent is often distributed gradually across multiple conversation turns, each message appearing harmless in isolation while collectively steering the model toward a malicious outcome. Defenses built on static keyword lists or single-message classification are structurally unable to detect this pattern.

Keywords: large language models, prompt injection, dual LLM, adaptive security, context drift, FAISS, semantic defense, adversarial attacks.

I. INTRODUCTION

Language models are now embedded in healthcare, education, finance, and customer service at a scale that would have seemed unlikely just a few years ago [3], [5]. That shift has quietly introduced a security problem that traditional defenses were never designed to handle. Unlike conventional software, which keeps a hard boundary between instructions and data, language models process everything as a single stream of text instructions, user input, and retrieved documents; all of it flows through the same pipeline. An attacker who can influence that stream can potentially redirect what the model does without touching a single line of code or exploiting any network vulnerability. This is the core mechanic behind prompt injection attacks, and it is not a bug someone forgot to patch; it is a direct consequence of how these systems work [5].

The threat has grown considerably more sophisticated since early demonstrations. What started as simple attempts to make a model "ignore previous instructions" has evolved into gradient-based suffix attacks that transfer across model families [6], indirect injection through documents retrieved from external sources, and multi-turn sequences where no single message looks alarming but the combined effect constitutes a successful jailbreak [7]. Research has shown that automated methods can find working jailbreaks in as few as twenty queries [6]. The attack surface keeps expanding into vision-language models used in oncology workflows, into financial messaging systems, and into agentic deployments where a compromised model can take real-world actions.

Current defenses split into two camps, and both have serious weaknesses. Keyword and pattern-matching systems break the moment an attacker paraphrases an instruction, switches character encoding, or routes the attack through retrieved content. Training large models to resist attacks natively is expensive, degrades general usefulness, and fails reliably on variants that were not in the training data [2]. Dual-model architectures, where a dedicated guard model screens inputs before they reach the primary system, show more promise but introduce their own limitations: high API costs, significant latency overhead, and an inability to track adversarial intent across multiple conversation turns [1], [4], [9].

This survey reviews the current landscape of prompt injection and jailbreak attacks, examines existing defense approaches across semantic, rule-based, and architectural categories, and identifies the critical gaps that remain unaddressed in the literature

II. TAXONOMY OF PROMPT INJECTION AND JAILBREAK ATTACKS

Prompt injection and jailbreak attacks can be broadly classified into five categories based on their delivery mechanism and target. Table I summarizes this classification.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155228

Table I: Taxonomy of Prompt Injection and Jailbreak Attack Types

Attack Type	Mechanism	Target	Example
Direct Prompt Injection	Malicious instructions embedded directly in user input	System prompt override	"Ignore previous instructions and..."
Indirect Prompt Injection	Malicious content hidden in external documents retrieved by the model	RAG pipeline	Poisoned web page or PDF
Gradient-based Suffix Attack	Algorithmically optimized adversarial suffixes appended to prompts	Model internals	GCG, AutoDAN
Multi-turn Jailbreak	Adversarial intent spread gradually across multiple conversation turns	Conversation context	Slow escalation across sessions
Encoding Obfuscation	Attack content disguised using character encoding or formatting tricks	Input filters	Base64, Unicode substitution

Understanding this classification is important because different attack types require fundamentally different detection strategies. A defense effective against direct injection may fail entirely against indirect or multi-turn variants, which motivates the need for layered, context-aware approaches.

III. SURVEY OF DEFENSE APPROACHES

A. Understanding Threats and Security Reviews

A prerequisite for effective defense is a systematic understanding of how attacks are structured [2], [18]. Over the years, researchers have built up a detailed picture of where language models are vulnerable. The threats range from poisoning the training data and leaking sensitive information [3] to injecting malicious instructions directly into prompts [19]. That last category, prompt injection, comes in two flavors: going after the model directly or hiding malicious content in external sources the model pulls from [4]. Subsequent research revealed an expanding attack surface, from quietly stealing private data to completely redirecting what a model does. Jailbreaking techniques [18], prompt leaking, and goal hijacking have all been catalogued and studied [20]. On the defensive side, notable work has been done on protecting system prompts and using reinforcement learning from human feedback [8], and even NIST updated its security framework to account for jailbreak-style attacks.

B. Dual-Model and Firewall Approaches

One architectural response to these threats is to keep potentially dangerous inputs away from the main model entirely [7], [9]. The idea is straightforward: use a separate, security-focused model to screen inputs before they ever reach the primary system. A university chatbot deployment demonstrated measurable success using this approach [7]. However, practical limitations were observed; the system processed requests one at a time and leaned on costly commercial APIs, which made it both slow and expensive to run. Other teams have tried layering semantic filtering on top of rule-based checks to catch malicious intent from multiple angles [11]. Some went further, building systems that can spot prompt injections alongside other attack types like backdoors and character obfuscation [11]. There has also been ongoing experimentation with how to physically and structurally separate user prompts from core system instructions.

C. Automated Attack Methods

Defenses also have to hold up against attacks that are automated and optimized, not just crafted by hand [1], [6]. Research has shown that optimization algorithms can find working jailbreaks with surprisingly few attempts [1], which raises the stakes considerably. Making those attacks reproducible and measurable matters too, both for red-teaming and for fairly evaluating defenses [18]. Some researchers have even borrowed from game theory to model how attacks might propagate through interconnected systems. The implication is clear: static defenses that do not adapt are going to fall behind.

D. Machine Learning Classifiers for Detection

Not every detection problem needs a full language model [10], [16]. Researchers have had success training specialized, lightweight classifiers specifically to catch attacks [10], fine-tuning existing models on attack datasets [8], or pairing machine learning with explainability tools [10] so that when something gets flagged, a human can actually understand

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155228

why. For fast retrieval and comparison at scale, vector libraries like FAISS [15] make it possible to search through embeddings in under a millisecond. Sequence comparison methods [13], using metrics like cosine similarity, have also proven useful for spotting interactions that drift from normal patterns before they cause damage.

E. Research Gaps

Gap 1: No dedicated security layer separate from response generation [1], [4], [9]. Most systems blur the line between checking for threats and actually generating responses, handling both in a single pipeline. The problem with that approach is structural: if an attacker manages to slip past the guard, they have broken the whole system at once, since the security check and the output live in the same place.

Gap 2: Single-layer detection misses paraphrased and indirect attacks [2], [4]. None of the systems reviewed bring together semantic classification, rule-based keyword signals, and contextual risk scoring into one unified, weighted pipeline. Relying on just one of these methods leaves obvious blind spots; a rephrased attack or one that avoids obvious keywords can slip right through.

Gap 3: Multi-turn conversational context is ignored [13]. Almost every published detection system looks at each prompt on its own, with no memory of what came before. That makes them vulnerable to slow-burn attacks, where an adversary spreads their malicious intent across several exchanges rather than putting it all in one message.

Gap 4: Weak evaluation practices [20]. Many published defenses have been tested on narrow, synthetic datasets without properly accounting for class imbalance or using rigorous cross-validation. As a result, the accuracy numbers they report do not hold up nearly as well in the real world as they appear to on paper.

IV. COMPARATIVE ANALYSIS OF EXISTING APPROACHES

Table II compares six representative defense systems across four criteria: multi-turn context awareness, unified risk scoring, local deployability, and reported detection accuracy.

Table II: Comparison of Existing Defense Approaches

Defense System	Method	Multi-Turn	Unified Score	Reported Accuracy
ChatTEDU [7]	Dual-LLM (cloud)	No	No	~91% (production)
Guo et al. [8]	Fine-tuned LLM	No	No	~94% (lab)
Zhang et al. [9]	Semantic + rules	No	Partial	Not reported
Nair et al. [10]	RF + Bi-LSTM + SHAP	No	No	92.5% (synthetic)
PromptArmor [12]	Layered sanitization	No	No	Competitive
STREAM [13]	CoT per-turn reasoning	Yes	No	Reduced success rate

V. CONCLUSION

This survey examined the current landscape of prompt injection and jailbreak attacks against large language models, reviewing defense approaches across semantic, rule-based, architectural, and classifier-based categories. The taxonomy presented identifies five distinct attack types, each requiring different detection strategies, while the comparative analysis of six representative systems reveals that no existing approach simultaneously addresses multi-turn context awareness, unified risk scoring, and local deployability. Three benchmark datasets, PromptBench , HackAPrompt, and AdvBench , have emerged as the primary evaluation standard, though inconsistent use of cross-validation and class imbalance handling continues to limit the reliability of reported accuracy figures . The gaps identified in this survey, architectural separation between security and response generation ; multi-signal unified detection; multi-turn context tracking; and rigorous evaluation practices, represent the clearest directions for future work in this area. Addressing these gaps collectively, rather than in isolation, remains the most pressing open problem in LLM defense research.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.155228

REFERENCES

[1]. P. Chao, A. Robey, E. Dobriban, H. Hassani, G. J. Pappas, and E. Wong, "Jailbreaking Black Box Large Language Models in Twenty Queries," IEEE Conference on Secure and Trustworthy Machine Learning (SaTML), 2025.
[2]. B. Rababah, S. Che, and M. Liao, "SoK: Prompt Hacking of Large Language Models," IEEE International Conference on Big Data (BigData), 2024.
[3]. Y. Liu, G. Deng, Z. Xu, Y. Li, Y. Zheng, Y. Zhang, L. Zhao, T. Zhang, and Y. Liu, "Formalizing and Benchmarking Prompt Injection Attacks and Defenses," USENIX Security Symposium, 2024.
[4]. K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, and M. Fritz, "Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection," ACM Workshop on Artificial Intelligence and Security (AISec), 2023.
[5]. R. W. Lee, J. Mathur, F. Kasner, and M. Petrov, "Vulnerability of Large Language Models to Prompt Injection Attacks," JAMA Network Open, 2025.
[6]. Y. Yi, S. Ye, J. Xing, M. Xu, and W. Lu, "Tree-based Dialogue Reinforced Policy Optimization for Red-Teaming Attacks (DialTree)," NeurIPS Workshop on Socially Responsible Language Modelling Research, 2025.
[7]. H. Emekci and G. Budakoglu, "Securing With Dual-LLM Architecture: ChatTEDU an Open Access Chatbot's Defense," IEEE Access, vol. 13, 2025.
[8]. Y. Guo, Z. Wang, H. Li, and X. Chen, "Fine-tuned Large Language Models: Improved Prompt Injection Detection," IEEE Transactions on Information Forensics and Security, vol. 20, 2025.
[9]. B. Zhang, L. Huang, Y. Wang, and J. Liu, "Enhancing System Security: LLM-Driven Defense Against Prompt Injection Vulnerabilities," IEEE Access, vol. 13, 2025.
[10]. M. Nair, R. Sharma, P. Verma, and S. Das, "Towards Secure AI: Detection of Prompt Injection Attacks with Explainability," IEEE International Conference on Testing, Evaluation, and Security (ICTEST), 2025.
[11]. H. Lin, Y. Zhao, T. Wu, and R. Chen, "UniGuardian: A Unified Defense for Detecting Prompt Injection, Backdoor Attacks, and Adversarial Attacks in LLMs," IEEE Access, vol. 13, 2026.
[12]. T. Shi, X. Luo, H. Zhang, and W. Tan, "PromptArmor: Simple yet Effective Prompt Injection Defenses," arXiv preprint arXiv:2507.15219, 2025.
[13]. P. Zhu, Q. Li, Y. Chen, and M. Wang, "STREAM: Safety Reasoning Elicitation Alignment for Multi-Turn Dialogues," IEEE/ACM Transactions on Information Systems, 2025.
[14]. Y. Wang, J. Li, X. Zhang, and H. Liu, "Robustness of Large Language Models Against Adversarial Attacks," IEEE Transactions on Neural Networks and Learning Systems, 2024.
[15]. M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvasy, P. Mazaré, M. Lomeli, L. Hosseini, and H. Jégou, "The Faiss Library," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2025.
[16]. Z. Chen, Y. Liu, W. Wang, and X. Huang, "Improving Robustness of Intent Detection Under Adversarial Attacks: A Geometric Constraint Perspective," IEEE Transactions on Neural Networks and Learning Systems, 2023.
[17]. F. Alves, D. Oliveira, R. Sousa, and P. Ferreira, "Revolutionizing Cyber Threat Detection With Large Language Models: A Privacy-Preserving BERT-Based Lightweight Model for IoT/IIoT Devices," IEEE Access, vol. 12, 2024.
[18]. X. Shen, Z. Chen, M. Backes, Y. Shen, and Y. Zhang, "Do Anything Now: Characterizing and Evaluating In-the-Wild Jailbreak Prompts on Large Language Models," ACM Conference on Computer and Communications Security (CCS), 2024.
[19]. S. Abdali, R. Anarfi, C. J. Barberan, and J. He, "A Systematic Review of Prompt Injection Attacks on LLMs," IEEE Transactions on Dependable and Secure Computing, vol. 22, 2025.
[20]. Y. Liu, Y. Deng, Z. Li, Y. Liu, T. Zhang, and Y. Liu, "Prompt Injection Attacks on Large Language Models: A Survey," IEEE Access, vol. 12, 2024.

