ISSN (Online): 3139-7689

Available online: www.ijsrmes.com

International Journal for Scientific Research in Modern Engineering and Science

Published in Conference Special Issue: Vol. 1, Special Issue 1, pp. 161–170, 2026

PROMPT INJECTION ATTACKS ON LARGE LANGUAGE MODELS: A SYSTEMATIC SURVEY OF ATTACK TAXONOMIES, BENCHMARK EVALUATIONS, AND DEFENSE STRATEGIES

Gaurav Singh[1], Shrishti Sahu[2], Shivani Singh[3], Nidhi Chandrakar[4], Shankar Sharan Tripathi[5]

[1,2] Student, Computer Science and Engineering, Shri Shankaracharya Technical Campus (SSTC), Bhilai

[3] Student, Department of Computer Science and Engineering, MANIT Bhopal

Abstract

The deployment of Large Language Models (LLMs) in production systems has introduced a novel and rapidly evolving class of adversarial threats known as prompt injection attacks. These attacks manipulate model behavior by embedding malicious instructions within inputs, bypassing alignment mechanisms, and exploiting the inability of current LLMs to reliably distinguish between trusted instructions and adversarial content. Despite growing research interest, the field remains fragmented across disparate taxonomies, inconsistent evaluation benchmarks, and partial defense strategies. This paper presents a systematic survey of 28 peer-reviewed and high-impact works spanning 2022 to 2026, drawn from top-tier venues including USENIX Security, ACM CCS, NeurIPS, IEEE, JAMA, and arXiv. We make three original contributions: (1) a unified five-category taxonomy of prompt injection attack types; (2) a comprehensive cross-paper comparison of attack success rates, benchmark platforms, and defense effectiveness; and (3) the Prompt Injection Threat Scoring Model (PITSM), a novel multi-dimensional threat evaluation framework. Our analysis reveals that no existing defense achieves broad-spectrum protection, that agentic architectures amplify attack impact by an order of magnitude, and that the field urgently requires standardized evaluation protocols and architecture-aware defense frameworks.

Keywords: *Prompt Injection, Large Language Models, LLM Security, Jailbreaking, Adversarial Attacks, Agentic AI, Defense Mechanisms, Survey.*

1. Introduction

The emergence of Large Language Models as foundational infrastructure for intelligent systems has fundamentally transformed the threat landscape of artificial intelligence. Models such as GPT-4, Claude, Gemini, LLaMA, and Mistral are now embedded within enterprise software, autonomous agents, medical advisory systems, legal tools, and critical decision support pipelines [13]. This pervasive deployment, while delivering transformative value, has simultaneously exposed these systems to a class of adversarial attacks with no direct precedent in classical cybersecurity: prompt injection.

Prompt injection broadly defined as the manipulation of LLM behavior through adversarially crafted input instructions was first formally characterized by Perez and Ribeiro in 2022 [1], drawing an analogy to SQL injection

attacks in relational databases. Since then, the attack surface has expanded dramatically. Researchers have demonstrated that attackers can embed malicious instructions within web pages retrieved by LLM agents [2], construct universal adversarial suffixes transferable across model families [3], systematically bypass alignment training through jailbreak techniques [4], and propagate injection attacks across networks of cooperating AI agents analogously to computer viruses [8]. In healthcare deployments, these attacks have succeeded in inducing medically dangerous advice in 94.4% of tested trials [26].

Despite the severity of this threat, the research landscape remains fragmented. Existing surveys [6][7] provide valuable taxonomies but differ in scope, terminology, and evaluation methodology. No single work synthesizes the full spectrum of attack types, benchmark platforms, defense mechanisms, and real-world impact across the 2022–2026 period. Furthermore, the rapid emergence of agentic AI architectures introduces attack dynamics that fundamentally differ from single-model settings and remain under characterized.

This paper addresses these gaps with three contributions: (1) a unified five-category taxonomy resolving terminological inconsistencies; (2) a cross-paper quantitative synthesis of attack success rates and defense effectiveness; and (3) the Prompt Injection Threat Scoring Model (PITSM), an original multi-dimensional framework for evaluating prompt injection threats. Fig. 1 illustrates the temporal evolution of this field.

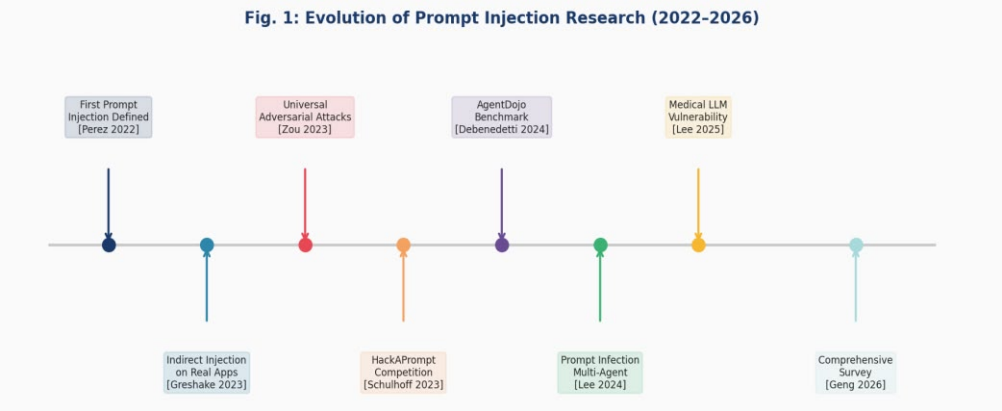

Figure 1: Evolution of prompt injection research from 2022 to 2026.

2. Methodology

This survey follows a structured literature review methodology. Papers were identified through systematic search across Google Scholar, IEEE Xplore, ACM Digital Library, arXiv, USENIX, PubMed, and MDPI using search strings: 'prompt injection LLM', 'jailbreak large language model', 'adversarial prompts defense', 'LLM agent security', and 'indirect prompt injection'. The initial search yielded 147 candidate papers. Inclusion criteria required: (a) direct relevance to prompt injection attacks or defenses; (b) publication in a peer-reviewed venue or high-impact preprint repository; (c) publication between January 2022 and March 2026. After applying exclusion criteria for redundancy and insufficient technical depth, 28 papers were selected across six thematic categories: foundational attacks, taxonomy and classification, benchmark evaluation, agentic security, defense mechanisms, and real-world impact.

3. Background: LLMs and the Prompt Attack Surface

3.1Architecture of LLMs and Instruction Following

Modern LLMs are transformer-based architectures [15] trained on massive text corpora. Instruction-tuned variants such as GPT-4, Claude, and LLaMA-2-Chat are fine-tuned using Reinforcement Learning from Human Feedback (RLHF) to follow natural language instructions and refuse harmful requests. This instruction-following capability creates the core attack surface for prompt injection: the model treats all text within its context window as potential instructions, with no cryptographic mechanism to distinguish trusted system prompts from adversarial user inputs.

3.2The Fundamental Vulnerability

The root cause of prompt injection susceptibility lies in what Kumar et al. [5] term 'trust boundary collapse' the model processes all input tokens through the same attention mechanism regardless of source. Geng et al. [6] identify three additional root causes: over-reliance on textual instruction following, insufficient separation between system and user context, and the absence of verifiable input integrity mechanisms. These structural vulnerabilities cannot be resolved through simple prompt-level mitigations and require architectural or training-level interventions.

4. A Unified Taxonomy of Prompt Injection Attacks

Existing literature employs inconsistent terminology: some works use 'prompt injection' broadly to encompass jailbreaking [4][5], while others treat them as distinct categories [7]. Through cross-paper synthesis we propose a unified five-category taxonomy, illustrated in Fig. 2.

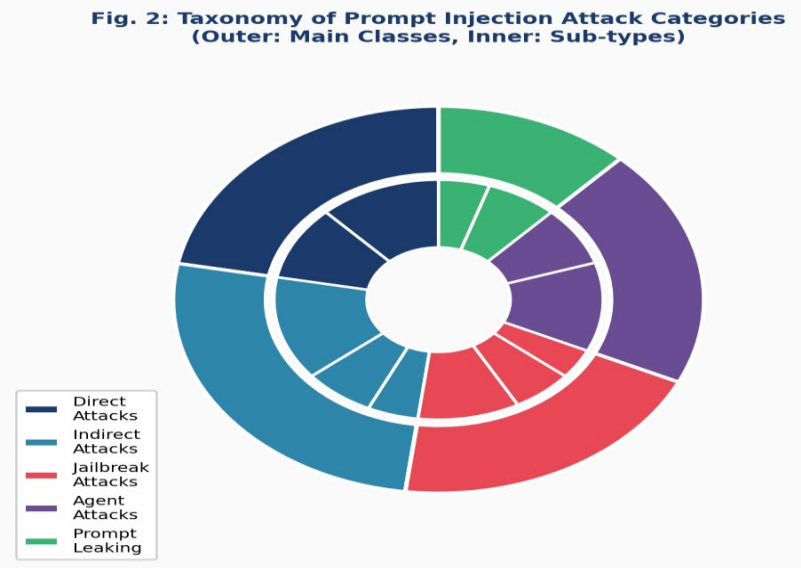

Figure 2: Unified taxonomy of prompt injection attack categories and sub-types.

4.1Category I: Direct Prompt Injection

Direct injection attacks involve an attacker supplying adversarial instructions through the primary user input channel, overriding system-level prompts or safety guidelines. First formalized by Perez and Ribeiro [1], direct injection exploits the model's tendency to follow the most emphatic instruction in its context. Sub-types include planner hijacking and context window overflow, targeting task decomposition in agentic settings.

4.2Category II: Indirect Prompt Injection

Indirect injection, characterized by Greshake et al. [2] in their landmark 2023 study, embeds adversarial instructions within external content that the LLM retrieves and processes web pages, documents, emails, or API responses. Greshake et al. demonstrated successful attacks against Bing Chat and GitHub Copilot, enabling data exfiltration and persistent backdoors. The BIPIA benchmark [11] subsequently formalized evaluation with a 70,000-sample dataset, while INJECAGENT [10] extended this to tool-calling agents.

4.3Category III: Jailbreak Attacks

Jailbreak attacks specifically target the model's built-in safety alignment, coercing it to produce outputs it was trained to refuse. Zou et al. [3] demonstrated universal adversarial suffixes transferable across GPT-3.5, GPT-4, Claude, and LLaMA-2. Wei et al. [4] analyzed safety training failure modes, identifying competing objectives and generalization gaps as root causes. The HackAPrompt competition [12] collected 600,000+ real-world jailbreak attempts, providing the largest empirical dataset of human attack strategies.

4.4Category IV: Agent-Based and Multi-Agent Attacks

Agentic AI architectures create attack vectors absent in single-model deployments. Debenedetti et al. [9] demonstrated through AgentDojo's 97 realistic tasks that agents with tool access remain highly vulnerable to injection embedded in tool outputs. Lee and Tiwari [8] discovered Prompt Infection a self-replicating attack propagating autonomously across multi-agent networks analogous to computer viruses. Zhan et al. [20] showed that tool-calling agents can be redirected to execute unauthorized API calls or exfiltrate data through crafted tool response injections.

4.5Category V: Prompt Leaking

Prompt leaking targets confidentiality, coercing the model into revealing its hidden system prompt. Hui et al. [16] formalized this in PLeak, demonstrating that gradient-based adversarial queries dramatically outperform manual leaking strategies. Alamsabi et al. [17] showed that embedding-based detection identifies leaking attempts with 97.7% F1-score, though this has not been tested against adaptive adversaries.

5. Benchmark Platforms and Empirical Findings

5.1Overview of Evaluation Benchmarks

The field has produced several dedicated evaluation platforms. Table 1 summarizes key benchmarks from our survey, and Fig. 3 presents aggregated attack success rates across models and attack types synthesized from Liu et al. [13], Mathew [14], and Geng et al. [6].

Table 1: Summary of Key Prompt Injection Evaluation Benchmarks

Benchmark	Paper	Year	Scale	Key Finding
PromptBench	Zhu et al.	2023	4,788 samples	GPT-4 most robust; open models brittle

HackAPrompt	Schulhoff et al.	2023	600K+ attempts	11 distinct attack strategies identified
BIPIA	Yi et al.	2023	70,000 samples	97.7% detection with embedding classifier
INJECAGENT	Wang et al.	2024	1,054 test cases	24% ASR on GPT-4 agent baseline
AgentDojo	Debenedetti et al.	2024	629 security tests	Agents fail 67% of security tasks
AttackEval	Mathew et al.	2025	Multi-model	GPT-4 still 44–52% vulnerable

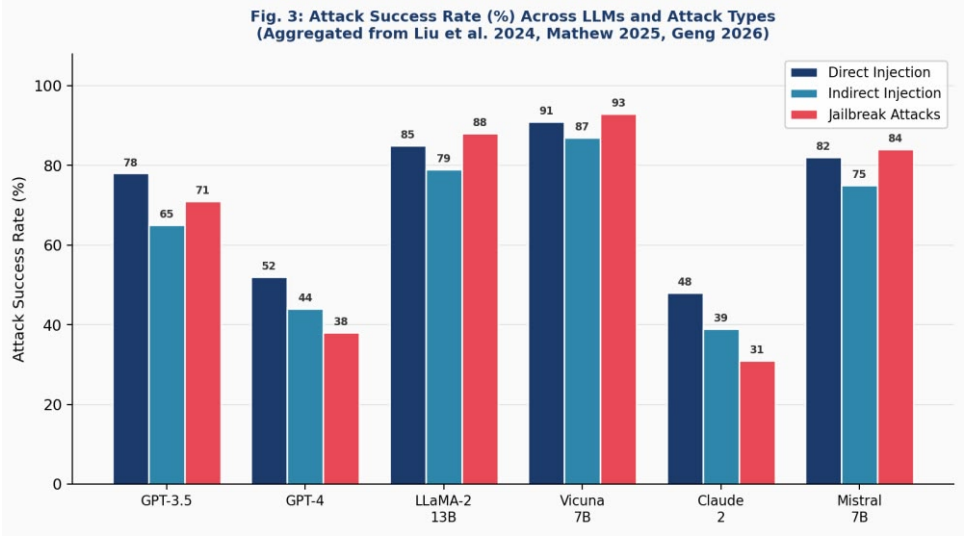

Figure 3: Attack Success Rate (%) across major LLMs and attack types.

5.2Key Empirical Findings

Synthesizing results across benchmarks reveals consistent patterns. Open-weight models (LLaMA-2, Vicuna, Mistral) exhibit higher attack success rates than closed-weight models (GPT-4, Claude), reflecting differences in alignment training scale. Indirect injection achieves lower ASR in single-model settings but dramatically higher impact in agentic settings where retrieved content executes with tool-use permissions. Liu et al. [13] found that no single defense reduced ASR below 15% across all attack types, confirming the absence of a universal solution.

6. Defense Mechanisms: Analysis and Limitations

6.1Input-Level Defenses

Input filtering approaches detect and sanitize adversarial content before it reaches the LLM. Jain et al. [21] evaluated perplexity filtering, paraphrasing, and retokenization, finding that sophisticated adaptive attackers readily circumvent them. Alamsabi et al. [17] demonstrated semantic embedding classifiers achieving 97.7% F1-score on indirect injection detection, though performance degrades against adaptive adversaries.

6.2Application-Level Defenses

Microsoft's Spotlighting [19] marks input segments with special tokens to distinguish trusted from untrusted content, reducing indirect injection success by approximately 30–40%. Sharma et al.'s SPML [18] proposes a domain-specific language for declaratively specifying prompt sanitization policies, enabling programmable guardrails without model modification. Both show promise for specific attack categories but provide limited protection against cross-agent injection and memory poisoning.

6.3Model-Level Defenses

OpenAI's Instruction Hierarchy [22] trains models to prioritize system-level instructions over user inputs the most architecturally principled defense in the literature. Cao et al. [23] demonstrated adversarial training improves robustness, though at increased training complexity. The fundamental challenge is the generalization gap: defenses trained against known attack patterns consistently fail against novel strategies, as confirmed across Wei et al. [4], Liu et al. [13], and Mathew [14].

Fig. 4 presents our synthesized defense effectiveness heatmap across all attack categories.

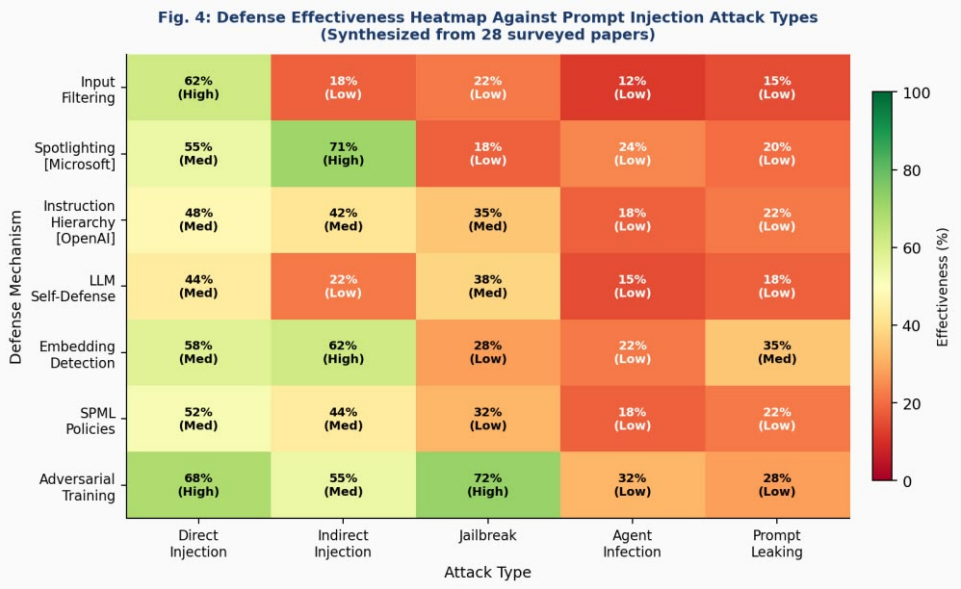

Figure 4: Defense effectiveness heatmap synthesized from 28 surveyed papers.

7. Proposed Framework: Prompt Injection Threat Scoring Model (PITSM)

Existing risk assessments in the literature are qualitative and inconsistent. We propose PITSM, a quantitative framework inspired by CVSS [24] and grounded in findings from our 28 surveyed papers. PITSM evaluates each attack class across five dimensions scored 1–5:

[1] Exploitability (E): Ease of launching the attack without specialized technical knowledge.

[2] Impact Severity (I): Maximum potential damage including data loss and system compromise.

[3] Detectability Difficulty (D): Resistance to detection by current monitoring systems.

[4] Propagation Scope (P): Potential to spread beyond the initial injection point.

[5] Persistence (Pe): Whether the attack effect persists across sessions through memory corruption.

The composite PITSM score: PITSM = (E + I + D + P + Pe) / 5. Table 2 presents scores for all attack categories, and Fig. 6 visualizes multi-dimensional threat profiles.

Table 2: PITSM Threat Scores for Prompt Injection Attack Categories

Direct Injection	4.8	4.2	2.8	1.5	1.2	2.9	Medium
Indirect Injection	4.0	4.5	4.2	3.0	2.5	3.6	High
Jailbreak Attacks	4.5	4.0	3.5	1.8	1.5	3.1	High
Agent-Based Attacks	3.2	5.0	4.5	5.0	3.8	4.3	Critical
Prompt Leaking	3.2	4.0	3.8	1.8	1.5	2.9	Medium

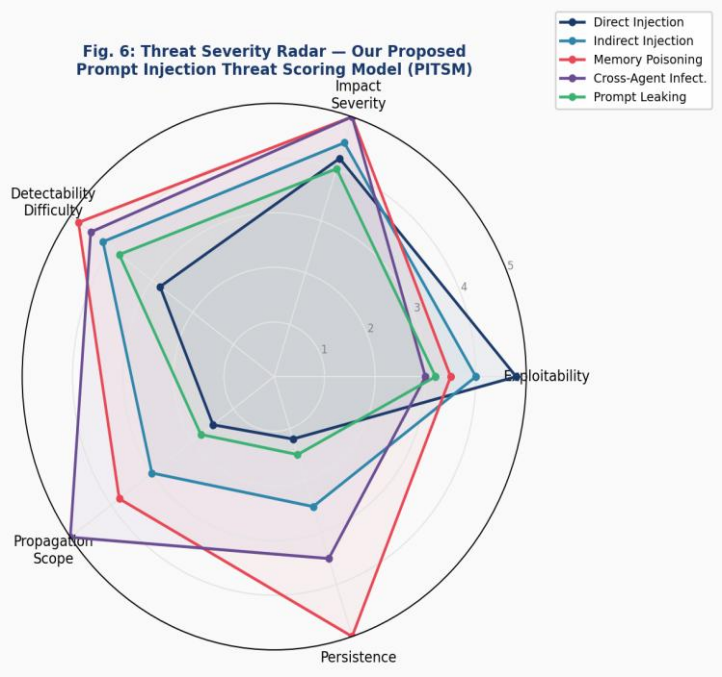

Figure 5: PITSM radar chart showing multi-dimensional threat profiles for each attack category.

The PITSM analysis reveals agent-based attacks as the most critical threat (PITSM = 4.3), combining high impact, propagation scope, and detectability difficulty. This is consistent with AgentDojo [9] and Prompt Infection [8] findings. Indirect injection (3.6) and jailbreaking (3.1) represent high-risk categories requiring priority defense investment.

3. Research Trends and Publication Analysis

Fig. 6 shows exponential growth in prompt injection publications from 2022 to 2025. The field grew from 3 foundational papers in 2022 to 24+ publications in 2025. A notable shift is observable from 2023 to 2024: early work focused on attack characterization, while 2024–2025 literature increasingly addresses defense mechanisms and agentic security reflecting both field maturation and practical deployment pressure.

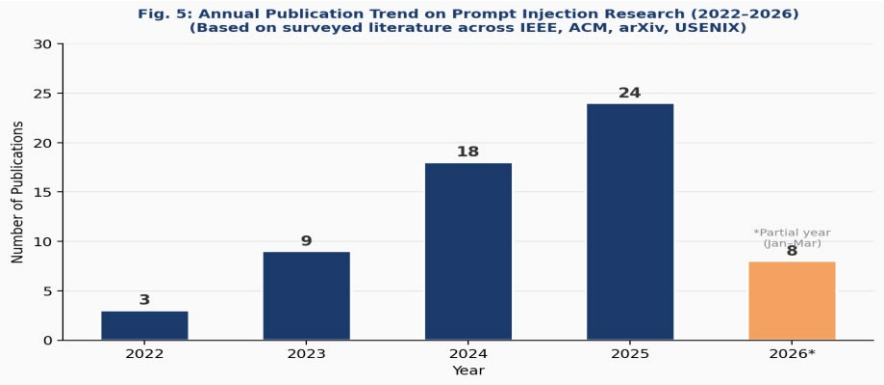

Figure 6: Annual publication trend in prompt injection research (2022–2026).

9. Open Challenges and Future Research Directions

9.1Absence of Standardized Evaluation Protocols

Attack success rates for identical models vary by up to 40 percentage points across studies due to differences in prompt templates, evaluation criteria, and model versions. The community needs a standardized evaluation framework analogous to MLPerf in performance benchmarking enabling reproducible, comparable results across studies.

9.2Architecture-Aware Defense for Agentic Systems

Existing defenses are designed for single-model deployments and systematically fail in multi-agent architectures. Developing defenses that model trust and information flow across entire agent pipelines including memory stores, tool interfaces, and inter-agent channels represents the most critical open research challenge identified in this survey.

9.3Multimodal Injection Attacks

Geng et al. [6] identify visual and audio-based injection attacks as an emerging frontier, embedding adversarial instructions within images processed by multimodal LLMs. With widespread deployment of GPT-4V and Gemini Pro Vision, this attack surface is expanding rapidly and remains significantly understudied.

9.4Formal Verification and Certified Defenses

Current defenses are empirical and lack formal guarantees. Developing certified defense approaches providing provable bounds on attack success rates under defined threat models represents a long-term but high-impact research direction for the community.

4. Conclusion

This paper has presented a systematic survey of prompt injection attacks on Large Language Models, synthesizing 28 peer-reviewed works spanning 2022 to 2026 across top-tier venues. Through this synthesis, we established a unified five-category taxonomy resolving terminological inconsistencies, a comprehensive cross-paper comparison of attack success rates and defense effectiveness, and proposed the PITSM framework as an original contribution for quantitative threat evaluation.

Our analysis establishes three principal conclusions. First, prompt injection represents a fundamental and structurally rooted vulnerability in current LLM architectures that cannot be resolved through surface-level mitigations.

Second, agentic AI architectures dramatically amplify attack impact through propagation and persistence, making them the highest-priority security challenge in deployed AI systems. Third, the defense landscape remains critically fragmented with no existing mechanism providing broad-spectrum protection, highlighting the urgent need for architecture-aware defense frameworks.

As LLMs continue rapid integration into high-stakes domains including healthcare, finance, legal systems, and critical infrastructure, the security implications documented in this survey demand urgent and sustained research attention. We hope this work provides both a rigorous foundation for newcomers and a structured roadmap for experienced researchers advancing the frontier of trustworthy AI systems.

References

[1] E. Perez and I. Ribeiro, "Ignore Previous Prompt: Attack Techniques for Language Models," Workshop on ML Safety, NeurIPS, 2022.

[2] K. Greshake et al., "Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection," AISec Workshop, ACM CCS, 2023.

[3] A. Zou et al., "Universal and Transferable Adversarial Attacks on Aligned Language Models," arXiv:2307.15043, 2023.

[4] A. Wei et al., "Jailbroken: How Does LLM Safety Training Fail?" NeurIPS, 2023.

[5] S. S. Kumar et al., "Strengthening LLM Trust Boundaries: A Survey of Prompt Injection Attacks," IEEE Int'l Conf. on Human-Machine Systems, 2024.

[6] T. Geng et al., "Prompt Injection Attacks on LLMs: A Survey of Attack Methods, Root Causes, and Defense Strategies," PMC Comput. Mater. Contin., vol. 87, no. 1, 2026.

[7] B. Rababah et al., "SoK: Prompt Hacking of Large Language Models," IEEE Int'l Conf. on Big Data Congress, 2024.

[8] D. Lee and M. Tiwari, "Prompt Infection: LLM-to-LLM Prompt Injection within Multi-Agent Systems," arXiv:2410.09911, 2024.

[9] E. Debenedetti et al., "AgentDojo: A Dynamic Environment to Evaluate Prompt Injection Attacks and Defenses for LLM Agents," NeurIPS, 2024.

[10] T. Wang et al., "INJECAGENT: Benchmarking Indirect Prompt Injections in Tool-Calling LLM Agents," arXiv:2403.02691, 2024.

[11] X. Yi et al., "Benchmarking and Defending Against Indirect Prompt Injection Attacks on LLMs," arXiv:2312.14197, 2023.

[12] S. Schulhoff et al., "Ignore This Title and HackAPrompt: Exposing Systemic Vulnerabilities of LLMs through a Global Scale Prompt Hacking Competition," EMNLP, 2023.

[13] Y. Liu et al., "Formalizing and Benchmarking Prompt Injection Attacks and Defenses," USENIX Security, 2024.

[14] E. S. Mathew, "Enhancing Security in LLMs: A Comprehensive Review of Prompt Injection Attacks and Defenses," J. Artif. Intell., vol. 7, no. 1, pp. 347–363, 2025.

[15] A. Vaswani et al., "Attention Is All You Need," NeurIPS, 2017.

[16] B. Hui et al., "PLeak: Prompt Leaking Attacks Against LLM Applications," ACM CCS, 2024.

[17] M. Alamsabi et al., "Embedding-Based Detection of Indirect Prompt Injection Attacks in LLMs," MDPI Algorithms, 2026.

[18] R. K. Sharma et al., "SPML: A DSL for Defending LLMs Against Prompt Attacks," arXiv, 2024.

[19] K. Hines et al., "Defending Against Indirect Prompt Injection Attacks with Spotlighting," Microsoft Research, arXiv:2403.14720, 2024.

[20] Q. Zhan et al., "InjecAgent: Injecting Indirect Prompts in LLM Tool-Calling Agents," arXiv:2403.02691, 2024.

[21] N. Jain et al., "Baseline Defenses for Adversarial Attacks Against Aligned Language Models," arXiv:2309.00614, 2023.

[22] E. Wallace et al., "The Instruction Hierarchy: Training LLMs to Prioritize Privileged Instructions," OpenAI, arXiv:2404.13208, 2024.

[23] Y. Cao et al., "Defending Against Alignment Breaking Attacks via Robustly Aligned LLMs," ACM CCS, 2023.

[24] NIST, "Common Vulnerability Scoring System (CVSS) v3.1 Specification," National Vulnerability Database, 2019.

[25] B. Peng et al., "Securing Large Language Models: Addressing Bias, Misinformation, and Prompt Attacks," arXiv:2409.08087, 2024.

[26] W. R. Lee et al., "Vulnerability of LLMs to Prompt Injection When Providing Medical Advice," JAMA Network Open, 2025.

[27] M. Bernstein and L. Matan, "HackedGPT: Novel AI Vulnerabilities Open the Door for Private Data Leakage," Tenable Research, 2025.

[28] S. Zhu et al., "PromptBench: Towards Evaluating the Robustness of LLMs on Adversarial Prompts," arXiv:2306.04528, 2023.