Security Vulnerabilities and Defense Framework for Large Language Models

Madhavi Dhingra[1], S.K. Manju Bargavi[2]
[1] Lincoln University College, Malaysia, Amity University Madhya Pradesh, Gwalior ; [2] Department of Computer Science and IT, Jain (Deemed-to-be University), Bangalore, Karnataka
[1]madhavi.dhingra@gmail.com, [2]cloudbargavi@gmail.com;

Abstract: Large Language Models (LLMs) have transformed content creation in various fields while also posing major security threats, including data privacy violations, prompt injection assaults, and intricate adversarial weaknesses. This document outlines an in-depth research investigation focused on examining these risks via a tripartite approach: (1) evaluating adversarial weaknesses through experimental red-teaming, (2) classifying exploitable actions using unsupervised clustering, and (3) creating a systematic multi-tiered preventive security framework. Through the assessment of various open-source and closed-source models (such as LLaMA, Falcon, and GPT variants), we gauge the rates of successful attacks, instances of refusal, and the intensity of harmful outputs. Additionally, we utilize KMeans clustering on response data to create an automated risk classification. The research assesses a multi-tiered defense structure that includes input cleansing and real-time oversight, showing quantifiable decreases in the intensity of security risks. The results highlight the necessity for multi-metric assessments and the adoption of defense-in-depth approaches to guarantee the safe and reliable use of LLMs in operational settings.

Keywords: LLM; LLM Security; Adversarial attack; Malicious attack; GPT Model

---

Introduction

The swift expansion of Large Language Models (LLMs) has changed the way businesses create, handle, and engage with written content. Models such as GPT-4, LLaMA, and Claude exhibit impressive skills in reasoning, programming, and creative writing. Nonetheless, the instruction-following abilities that enhance the utility of these models also make them susceptible to harmful interference. Malicious entities can create adversarial inputs—commonly known as "jailbreaks" or "prompt injections"—to manipulate models into producing harmful content, disclosing sensitive training data, or performing unauthorized actions [2], [10].

The essence of the issue is that LLMs are designed to optimize the probability of a response derived from a specific context, which can be exploited by malicious users. In contrast to conventional software vulnerabilities that can be fixed at the code level, vulnerabilities in LLMs frequently arise from statistical generalization patterns and the model's natural alignment with directive-following instructions [3]. Data privacy, data breaches, and prompt injection threats rank among the top security issues currently [2].

A structured Preventive Security framework is critically needed to identify, assess, and reduce these risks. This framework needs to establish adequate measures for risk mitigation while promoting the secure use of LLMs in practical applications. This research meets this requirement by initially recognizing and classifying security hazards via experimental assessment, and subsequently assessing a multi-layered security framework intended to protect against both input-level and deployment-level vulnerabilities.

Motivation

The academic and technical literature regarding LLM security has recognized various specific categories of threats, each necessitating unique detection and mitigation approaches. These risks cover the complete lifecycle of the model, from the preliminary pre-training stage to fine-tuning and operational implementation.

In spite of the increase in studies on individual attacks and defenses, the literature continues to be disjointed. Researchers agree that the domain is missing cohesive taxonomies, standardized threat models, and replicable benchmarks [4], [10]. Defense strategies like adversarial training, input validation, and provenance controls are available; however, there is a strong demand for cohesive operational frameworks that merge preventive measures, detection systems, and governance policies [4]. This research seeks to close this gap by creating and assessing a multi-tiered security framework

Methodology

The research methodology focuses on three main objectives aimed at delivering a thorough assessment of LLM security threats and reviewing protective strategies.

3.1 Goal 1: Analysis of Adversarial Vulnerability

The initial aim centered on assessing the vulnerability of different LLMs to adversarial prompts, such as prompt injection, bias induction, misinformation, and data extraction.

- Experimental Red-Teaming: We chose representative LLMs, encompassing both open-source models (e.g., LLaMA, Falcon) and closed-source models accessed through the GPT API.
- Attack Design: The group created adversarial prompts, harmful inputs, and role-playing directives to evaluate model reactions. These involved efforts to circumvent safety measures and provoke unsafe content creation.
- Metrics Gathering: The research monitored the Attack Success Rate (ASR), which is described as the proportion of outputs that overcame protections. We additionally evaluated the intensity and consistency of these attacks, enhancing frequency-based metrics with qualitative evaluations of hazardous outputs [3], [13].

3.2 Goal 2: Classification of Behavior and Development of Taxonomy

To methodically examine the characteristics of risky outputs, we executed an unsupervised classification of model behaviors.

- Data Clustering: Answers produced in the Objective 1 experiments were examined through TF-IDF vectorization and KMeans clustering. This method enabled the identification of repeating patterns in risky actions without depending on manual tagging.
- Taxonomy Creation: Usable behaviors were divided into three areas:
- Technical: Inversion of models, manipulation of outputs, and leakage of data.
- Social: Creation of false information and harmful conduct.
- Operational: Unapproved API utilization and evasion of deployment protections [4], [9].
- Expert Review: Cybersecurity and AI specialists participated in evaluating and enhancing these classifications to guarantee technical precision and significance.

3.3 Objective 3: Creation of a Multi-Layered Security Framework

The third goal focused on creating and assessing a structured preventive security framework known for its modularity and layered defense.

- Data-level security: Concentrates on input sanitization, provenance verification, and detecting anomalies in training and instructional data.
- Model-level security: Encompassed methods like adversarial training, prompt filtering, and alignment adjustment (e.g., RLHF).
- Deployment-level security: Established runtime oversight, anomaly identification, logging, and safety measures to address post-generation threats.
- Governance & Policy: Concentrated on human-in-the-loop validations, auditing procedures, and access management [11], [15].
- Assessment: A prototype structure incorporating input cleansing and real-time supervision was evaluated using the hostile prompts created in Objective 1.

Results and Assessment

4.1 Analysis of Vulnerabilities in Selected LLMs

The assessment of chosen LLMs showed that the size of the model does not automatically correlate with resilience to adversarial attacks.

- Attack Success Rates: Slim and instruction-optimized models showed greater attack success rates than their more substantial versions. Nonetheless, even more substantial models exhibited weaknesses, suggesting that resilience depends on the quality of alignment and safety adjustments rather than solely on the number of parameters [12].
- Severity and Refusal Behavior: Our findings indicated that models with a higher refusal rate for unsafe prompts did not automatically lead to fewer severe failures. In certain cases, when the refusal mechanism did not work, the outcome produced was high-risk. This indicates that frequency-based metrics (ASR) alone are inadequate and need to be supplemented with qualitative severity evaluations [6].

- Enforcement of Safety Policies: The experiments demonstrated that well-designed adversarial inputs were able to reliably circumvent safety policies in both open-source and closed-source models.

4.2 Behavior Clustering and Automated Risk Categorization

Unsupervised classification of behaviors through KMeans clustering effectively detected model-independent risk patterns.

- Cluster Identification: The clustering process resulted in separate categories related to technical risks (e.g., data breaches), behavioral risks (e.g., harmful behavior, false information), and operational risks (e.g., jailbreak-type responses).
- Cluster Distribution: Severe clusters were recognized, indicating key targets for intervention. The uniformity of these clusters across various models indicates that specific vulnerabilities are inherent in existing LLM architectures.
- Clustering Efficiency: Behavior clustering demonstrated its effectiveness as a method for automated risk taxonomy creation, greatly minimizing the manual work needed for expert labeling and enabling quicker threat evaluations in dynamic deployment settings.

4.3 Effectiveness of the Layered Defense Prototype

The assessment of the prototype layered defense framework showed quantifiable enhancements in security stance.

- Input Sanitization: The sanitization component effectively identified and removed typical jailbreak and prompt injection patterns. The prevalence of identified suspicious phrases indicated that adversarial prompts frequently include recognizable structural vulnerabilities, establishing sanitization as an effective initial defense.
- Runtime Surveillance: Oversight after generation revealed a notable share of hazardous outputs that could have otherwise evaded input screenings. Though it didn't remove all risks, it significantly enhanced system visibility and responsibility.
- Effect on Severity: A comparative study revealed a decrease in average severity scores with active defenses. For example, comparable systems such as LeakSealer have shown AUPRC scores reaching 0.97 for PII leakage detection, whereas standard guardrails like Llama Guard have a score of 0.84 [16]. These results support the implementation of defense-in-depth strategies for the secure deployment of LLMs.

Conclusion

This study has delivered a technical analysis of security threats in Large Language Models, concentrating on adversarial weaknesses, usable behaviors, and the creation of protective measures. The research shows that through experimental red-teaming and unsupervised behavior categorization, existing LLMs—irrespective of their size—continue to be vulnerable to various security risks, such as prompt injection, data leakage, and poisoning.

The suggested layered defense structure provides a replicable and quantifiable approach to reducing these dangers. Through the integration of input sanitization, model alignment, and runtime monitoring

alongside robust governance, organizations can greatly lessen the impact of security failures. This study highlights the significance of multi-metric assessment and establishes a solid groundwork for forthcoming research focused on enabling secure and reliable LLM implementation. Future directions must focus on exploring security in agentic and multi-modal LLM systems, along with creating stronger, automated defense-in-depth architectures.

References

1. J. Qi et al., "Visual Adversarial Examples Jailbreak Large Language Models," arXiv preprint arXiv:2306.13213, 2023.
2. I. Shumailov et al., "The Curse of Recursion: Training on Generated Data Makes Models Forget," arXiv preprint arXiv:2305.17493, 2023.
3. E. Perez et al., "Red Teaming Language Models," arXiv preprint arXiv:2202.03286, 2022.
4. Y. Wang et al., "Unique Security and Privacy Threats of Large Language Models: A Comprehensive Survey," ACM Computing Surveys, vol. 57, no. 1, 2025. DOI: 10.1145/3764113
5. N. Carlini et al., "Extracting Training Data from Large Language Models," in Proc. USENIX Security Symposium, 2021.
6. J. Wei et al., "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," in Proc. NeurIPS, 2022.
7. X. Jiang et al., "Prompt Injection Attacks and Defenses in Large Language Models," arXiv preprint arXiv:2303.09092, 2023.
8. R. Zhang et al., "LLM-based Social Engineering: Attacks and Defenses," arXiv preprint arXiv:2310.08129, 2023.
9. M. A. Hossen et al., "Assessing Cybersecurity Vulnerabilities in Code Large Language Models," arXiv preprint arXiv:2404.18567, 2024.
10. Y. Huang et al., "A Survey on Security and Privacy of Large Language Models," arXiv preprint arXiv:2307.10719, 2023.
11. J. Li et al., "Multi-Layered Defense for LLM Security," arXiv preprint arXiv:2311.09127, 2023.
12. T. Wolf et al., "Safety-Tuned LLMs: Alignment, Guardrails, and Filters," Hugging Face Technical Report, 2023.
13. D. Ganguli et al., "Red Teaming Language Models to Reduce Harms: A Framework," Anthropic Technical Report, 2022.
14. K. Xu et al., "A Taxonomy of Adversarial Defenses in NLP," ACM Computing Surveys, 2023.
15. NIST, "AI Risk Management Framework (AI RMF 1.0)," U.S. Department of Commerce, 2023.
16. A. Panebianco et al., "LeakSealer: A Semisupervised Defense for LLMs Against Prompt Injection and Leakage Attacks," arXiv preprint arXiv:2508.00602, 2025.
17. S. Das et al., "Security and Privacy Challenges of Large Language Models: A Survey," ACM Computing Surveys, vol. 57, no. 1, 2025. DOI: 10.1145/3712001
18. S. Amich, "Multifaceted Characterization and Enhancement of Machine Learning Security," University of Michigan, 2024. DOI: 10.7302/24910
19. A. Jaffal et al., "Large Language Models in Cybersecurity: Applications, Vulnerabilities, and Defense Techniques," arXiv preprint arXiv:2507.13629, 2025.