Journal of
Information Technology,
Cybersecurity, and Artificial Intelligence

(RESEARCH)

Evaluating Prompt Injection Defenses in Large Language Models A Multi-Model Empirical Study of Security–Usability Trade-offs

Robert Kemp
University of Portsmouth

Journal of Information Technology, Cybersecurity, and Artificial Intelligence, 2026, 3(3), 43-53

Article DOI: https://doi.org/10.70715/jitcai.2026.v3.i3.071

Abstract

Large Language Models (LLMs) are increasingly deployed across a wide range of applications, yet their susceptibility to prompt injection attacks presents a significant security challenge. This study investigates the impact of defensive mechanisms and model configurations on prompt injection resilience through a controlled experimental evaluation of four open-source LLMs: Gemma 3, Llama 3, Mistral, and Phi-3 Mini. A dataset of 100 prompts (50 malicious and 50 benign) was used to assess performance across multiple metrics, including detection rate, false positive rate, attack success rate reduction (ASRR), accuracy degradation, and computational overhead.

The results demonstrate that the introduction of layered defensive controls significantly improves detection rates, with increases from 56% to 80% for Mistral and from 62% to 94% for Phi-3 Mini, while Llama 3 achieved 100% detection under controlled conditions. Attack success rates were reduced by up to 100% for Llama 3 and 84.2% for Phi-3 Mini, although residual vulnerabilities remained in other models. However, these improvements were accompanied by notable trade-offs, including false positive rates of up to 16% and accuracy degradation of up to 56.45%, indicating a substantial impact on usability and output fidelity.

The findings reveal that while defensive mechanisms enhance security, they introduce measurable reductions in model performance and may, in some cases, alter prompt semantics in ways that weaken intrinsic safeguards. These results suggest that prompt injection represents a structural vulnerability in current LLM architectures. The paper concludes by advocating for adaptive, context-aware defence strategies to balance security and usability in real-world deployments.

Key Words: Large Language Models, Prompt Injection, Artificial Intelligence Security, Adversarial Machine Learning, Usability Trade-offs, Experimental Evaluation

1. Introduction

The rapid evolution of Large Language Models (LLMs) has significantly advanced the capabilities of Artificial Intelligence (AI) systems, facilitating increasingly sophisticated applications in domains such as natural language processing, automated decision-making, and human–computer interaction [1], [2]. As organisations continue to integrate these models into operational environments, concerns regarding their security and robustness have become increasingly prominent. Among the various threats identified, prompt injection attacks have emerged as a particularly insidious and pervasive form of adversarial manipulation [3], [4].

Prompt injection exploits the fundamental operational paradigm of LLMs, which are designed to interpret and respond to natural language instructions in a probabilistic manner [1]. By crafting carefully structured inputs, adversaries can induce models to deviate from their intended behaviour, override system-level constraints, or disclose sensitive information [3]. Unlike traditional software vulnerabilities, prompt injection does not rely on flaws in code execution but instead leverages the inherent ambiguity and flexibility of natural language [4]. This characteristic renders such attacks both accessible and difficult to mitigate using conventional security approaches.

Although prior research has identified a range of prompt injection techniques, including instruction override, persona manipulation, and multi-turn conversational attacks, the majority of existing studies have been limited in scope. In particular, there is a notable lack of empirical work that systematically evaluates both attacks and defensive mechanisms across multiple models within a unified experimental framework [5], [4]. Furthermore, the trade-offs associated with implementing security controls, particularly in relation to usability and model performance, remain insufficiently explored.

This paper seeks to address these limitations through a comprehensive multi-model empirical investigation. By evaluating prompt injection attacks and layered defence mechanisms across four distinct LLMs, this study provides a comparative analysis of model behaviour under adversarial conditions. In doing so, it contributes to a more nuanced understanding of the effectiveness and limitations of current mitigation strategies, as well as the broader implications for the secure deployment of LLM-based systems.

Recent literature has extensively documented the growing threat of prompt injection attacks in Large Language Models, identifying them as a fundamental vulnerability arising from the inability of models to distinguish between trusted instructions and untrusted input [6]. Empirical studies have demonstrated that such attacks can achieve high success rates, often exceeding 90% in unprotected systems, while highlighting the rapid evolution of increasingly sophisticated attack techniques, including multi-turn and multimodal injections [6]. In response, a wide range of defensive approaches has been proposed, ranging from input filtering and prompt structuring to multi-layered architectural frameworks [7]. However, existing research remains fragmented in several important respects. First, many studies focus on either attack development or individual defence mechanisms in isolation, rather than providing systematic, comparative evaluations across multiple models. Second, there is a lack of standardised experimental frameworks that assess both security effectiveness and associated trade-offs, particularly in terms of usability, false positives, and accuracy degradation. Third, although recent work emphasises the need for defence-in-depth strategies, there is limited empirical evidence examining how layered controls interact with different model architectures in practice [8]. Consequently, there remains a significant gap in understanding how defensive mechanisms influence both the security and operational performance of LLMs across diverse configurations. This study addresses this gap by providing a multi-model empirical evaluation that systematically analyses the impact of layered defences on detection effectiveness, attack success rates, usability, and performance.

To systematically investigate the effectiveness of prompt injection mitigation strategies, this study is guided by the following research questions:

RQ1: *To what extent do defensive mechanisms improve the detection of prompt injection attacks across different Large Language Models?*

RQ2: *How effective are these mechanisms in reducing the success rate of prompt injection attacks?*

RQ3: *What impact do defensive mechanisms have on model usability, particularly in terms of false positives and output accuracy?*

RQ4: *How do defensive mechanisms affect the computational performance of Large Language Models?*

These research questions enable a structured evaluation of the trade-offs between security, usability, and performance in LLM-based systems.

2. Related Work

The security of LLMs has attracted increasing attention in recent years, with prompt injection attacks being widely recognised as a fundamental vulnerability [3], [4]. These attacks arise from the inability of LLMs to consistently differentiate between trusted system-level instructions and untrusted user inputs, resulting in a susceptibility to manipulation through natural language prompts [5].

Existing literature has proposed various taxonomies of prompt injection techniques. Direct instruction override attacks explicitly instruct the model to disregard prior constraints, while persona-based attacks attempt to redefine the role or identity of the model to elicit unintended responses [9]. More advanced strategies include multi-turn attacks, which exploit conversational context over successive interactions [9], and obfuscation techniques designed to evade detection mechanisms [5]. Empirical studies have demonstrated that such approaches can achieve high success rates across a range of models, highlighting the persistent nature of the vulnerability [3], [5].

In response, several defensive strategies have been proposed. Input sanitisation techniques aim to detect and block malicious patterns within user prompts, often through heuristic or rule-based approaches [10]. Prompt delimiting seeks to enforce structural separation between user input and system instructions, thereby reducing the likelihood of instruction hijacking [4]. More sophisticated methods involve modifying the training process of LLMs, for instance through Reinforcement Learning from Human Feedback [11], to improve alignment and resilience to adversarial inputs.

However, these approaches exhibit notable limitations. Static filtering mechanisms are often brittle and can be circumvented through minor variations in phrasing or syntax [10]. Techniques requiring model retraining or white-box access are not always feasible in practical deployments, particularly in commercial contexts. Moreover, relatively few studies have conducted systematic evaluations that consider both attack effectiveness and defensive performance across multiple models. Crucially, the impact of such controls on usability, including false positives and degradation in response quality, remains underexplored.

This study builds upon existing research by addressing these gaps through a controlled experimental evaluation that integrates both attack and defence perspectives within a multi-model framework.

3. Methodology

This study adopts a controlled experimental design to systematically evaluate the effectiveness of prompt injection mitigation strategies in Large Language Models. The methodology is designed to ensure reproducibility, consistency, and alignment with established quantitative research practices.

3.1. Model Selection

Four open-source Large Language Models were selected for evaluation: Gemma 3, Llama 3, Mistral, and Phi-3 Mini. These models were chosen to reflect diversity in architectural design, parameter scale, and deployment context, consistent with recent research highlighting variability in model performance and robustness. The inclusion of both large-scale and lightweight models enables a more comprehensive assessment of how defensive mechanisms perform across different configurations.

3.2. Prompt Design Methodology

The prompt dataset was constructed to systematically represent both adversarial and benign interaction scenarios. A total of 100 prompts were developed, comprising 50 malicious prompts and 50 benign prompts.

Malicious prompts were designed based on established prompt injection techniques identified in the literature, including instruction override, role manipulation, multi-turn exploitation, and indirect prompt injection [1], [4], [11]. These prompts were carefully crafted to reflect realistic attack patterns rather than purely synthetic examples, ensuring ecological validity. Variations in phrasing, structure, and linguistic complexity were introduced to avoid bias toward specific detection patterns and to evaluate the robustness of defensive mechanisms against diverse adversarial strategies.

Benign prompts were constructed to represent typical user interactions, including informational queries, task-oriented instructions, and conversational inputs. Care was taken to ensure that benign prompts occasionally contained linguistic features similar to malicious prompts (e.g., imperative language), enabling a meaningful assessment of false positive behaviour.

3.3. Dataset Validation and Pilot Testing

To enhance the validity of the dataset, a pilot testing phase was conducted prior to the main experiment. During this phase, a subset of prompts was evaluated across selected models without defensive controls to verify that malicious prompts exhibited a reasonable success rate and that benign prompts were consistently interpreted as safe.

This validation process ensured that:

- Malicious prompts were sufficiently challenging and representative of real-world attack scenarios
- Benign prompts did not inadvertently trigger defensive mechanisms under baseline conditions
- The dataset provided a balanced and unbiased basis for evaluation

This approach aligns with established principles of experimental design, ensuring both construct validity and reliability of the evaluation process.

3.4. Defensive Mechanisms

Two primary defensive mechanisms were implemented: input filtering and prompt hardening. These approaches were selected based on their prevalence in existing LLM security research and their practical applicability in real-world systems [3], [10].

Input filtering involved the identification and blocking of potentially malicious patterns within user prompts [12], while prompt hardening focused on reinforcing system-level instructions to reduce susceptibility to manipulation [13]. The combination of these mechanisms reflects a layered defence strategy consistent with defence-in-depth principles.

3.5. Evaluation Metrics

The effectiveness of prompt injection defences was assessed using multiple evaluation metrics:

- Detection Rate: proportion of malicious prompts correctly identified
- Attack Success Rate (ASR): proportion of successful adversarial manipulations
- False Positive Rate: proportion of benign prompts incorrectly flagged
- Accuracy Degradation: reduction in response quality compared to baseline
- Computational Overhead: changes in CPU and memory utilisation

These metrics were selected to provide a multi-dimensional evaluation of both security effectiveness and usability, consistent with recent research emphasising the importance of comprehensive assessment frameworks [14], [15].

3.6. Justification of Sample Size

The sample size of 100 prompts was selected to balance experimental control with practical feasibility. This size is sufficient to enable meaningful comparative analysis across models and conditions while maintaining consistency in prompt evaluation.

Importantly, the dataset was evenly divided between malicious and benign prompts, ensuring balanced representation for both detection and false positive analysis. Similar sample sizes have been adopted in prior empirical studies evaluating adversarial robustness in LLMs, where controlled prompt sets are used to enable systematic comparison across experimental conditions [11], [13].

While larger datasets may provide additional statistical power, the structured design of the prompt set and the controlled experimental conditions in this study ensure that the results are both reliable and interpretable. The focus on depth of analysis rather than scale is consistent with exploratory empirical research in emerging domains such as LLM security.

4. Validity and Reliability

This study adopts several measures to ensure the validity and reliability of the experimental findings, while also acknowledging inherent limitations associated with controlled evaluations of Large Language Models.

4.1. Construct Validity

Construct validity refers to the extent to which the evaluation metrics accurately represent the concepts being measured. In this study, “attack success” is defined as any instance in which a malicious prompt results in the model deviating from its intended behaviour, including the generation of restricted content, the disclosure of sensitive information, or the overriding of system-level instructions. This definition is consistent with prior research on prompt injection and adversarial manipulation in LLMs [1], [10].

To ensure alignment between constructs and measurements, multiple metrics were employed, including detection rate, attack success rate, and false positive rate. These metrics collectively provide a comprehensive representation of both security effectiveness and system behaviour, reducing the risk of relying on a single operational definition.

4.2. Internal Validity

Internal validity concerns the extent to which observed outcomes can be attributed to the experimental variables rather than external factors [16]. This study was conducted in a controlled environment in which all models were evaluated using the same prompt dataset, defensive configurations, and evaluation procedures. This consistency minimises the influence of confounding variables and enables direct comparison across models and conditions.

However, it is important to acknowledge that LLM outputs may exhibit non-deterministic behaviour due to stochastic sampling processes [17]. To mitigate this, consistent parameter settings (e.g., temperature and sampling strategies) were applied across all experiments. While this does not eliminate variability entirely, it ensures that differences in outcomes are primarily attributable to the experimental conditions rather than uncontrolled randomness.

4.3. External Validity

External validity refers to the extent to which the findings can be generalised to real-world applications [18]. While this study employs realistic prompt injection scenarios derived from existing literature [4], [9], the experimental setting remains controlled and may not fully capture the complexity of real-world deployments.

In practice, LLMs are often integrated into dynamic systems involving user interaction, external data sources, and multi-turn conversations, which may introduce additional vulnerabilities not captured in this study. Furthermore, the selection of four models, while diverse, does not encompass the full range of available architectures.

Nevertheless, the use of multiple models, varied prompt types, and layered defensive mechanisms enhances the generalisability of the findings. The results therefore provide meaningful insights into likely system behaviour, while recognising that further validation in real-world environments is necessary.

4.4. Reliability

Reliability refers to the consistency and repeatability of the experimental results. This study ensures reliability through the use of a structured and reproducible methodology, including a fixed prompt dataset, clearly defined evaluation metrics, and standardised experimental procedures.

All prompts were applied consistently across models and conditions, and the evaluation process was designed to minimise subjective interpretation. In addition, the use of quantitative metrics allows results to be independently verified and replicated.

However, due to the inherent stochastic nature of LLMs, exact replication of outputs may vary slightly across runs. Despite this, the overall trends and comparative findings are expected to remain stable, supporting the reliability of the conclusions.

5. Results

The experimental evaluation revealed notable variation in both baseline vulnerability and the effectiveness of mitigation strategies across the four models. In order to provide a more granular understanding of model behaviour, the results are presented through a series of comparative tables, followed by interpretive analysis.

5.1. Detection Rate Analysis

The results presented in this section address **RQ1**, which examines the extent to which defensive mechanisms improve the detection of prompt injection attacks. Table I presents the detection rates for each model before and after the implementation of layered controls.

Table 1: Detection Rate (%) Across Models

Model	Without Controls (%)	With Controls (%)	Improvement (%)
Gemma 3	66	90	+24
Llama 3	96	100	+4
Mistral	56	80	+24
Phi-3 Mini	62	94	+32

DOI: https://doi.org/10.70715/jitcai.2026.v3.i3.071

The results indicate that all models benefit from the introduction of controls, although the degree of improvement varies considerably. Llama 3 exhibits the strongest baseline detection capability, achieving near-complete detection even prior to the application of controls. This suggests that its internal safety mechanisms are comparatively robust. In contrast, Mistral demonstrates the lowest baseline performance, indicating a higher susceptibility to prompt injection attacks.

Phi-3 Mini shows the greatest relative improvement following the introduction of controls, suggesting that external mitigation strategies can significantly enhance weaker baseline models. However, the dependence on such controls raises questions regarding stability and consistency in real-world deployment.

5.2. False Positive Analysis

The following results address **RQ3**, focusing on the impact of defensive mechanisms on usability, including false positive rates and output accuracy. Table II summarises the false positive rates observed across all models.

Table 2: False Positive Rate (%) Across Models

Model	Without Controls (%)	With Controls (%)
Gemma 3	0	10
Llama 3	0	12
Mistral	0	4
Phi-3 Mini	0	16

In the absence of controls, all models correctly classified benign prompts, resulting in zero false positives. However, the introduction of defensive mechanisms leads to a consistent increase in false positive rates across all models. This effect is most pronounced in Phi-3 Mini, which exhibits a false positive rate of 16%, suggesting that its filtering mechanisms may be overly restrictive.

Mistral, despite its weaker baseline security, demonstrates the lowest false positive rate following the application of controls. This may indicate a less aggressive filtering strategy, which, while preserving usability, may also contribute to its comparatively lower detection performance.

These findings highlight a critical trade-off, whereby improvements in security are achieved at the expense of usability and accessibility.

5.3. Attack Success Rate Reduction

This section addresses **RQ2** by evaluating the effectiveness of defensive mechanisms in reducing the success rate of prompt injection attacks. Table III presents the reduction in successful prompt injection attacks following the implementation of controls.

5.4. Table 3: Attack Success Rate Reduction (ASRR)

Model	ASRR (%)
Gemma 3	70.59
Llama 3	100
Mistral	54.55
Phi-3 Mini	84.20

The results demonstrate that layered controls are highly effective in reducing successful attacks, particularly for Llama 3, which achieves complete mitigation under the experimental conditions. Phi-3 Mini also demonstrates strong performance, although not to the same extent.

Mistral again exhibits comparatively weaker performance, with a lower reduction in attack success rates. This reinforces the observation that models with weaker baseline defences may not benefit equally from external controls.

Importantly, the absence of complete mitigation across all models indicates that prompt injection attacks remain a persistent threat, even in the presence of layered security mechanisms.

5.5. Accuracy Degradation

The following results address **RQ3**, focusing on the impact of defensive mechanisms on usability, including false positive rates and output accuracy. Table IV illustrates the impact of defensive controls on model accuracy, measured using cosine similarity.

Table 4: Accuracy Degradation (%) After Controls

Model	Accuracy Degradation (%)
Gemma 3	31.93
Llama 3	33.86
Mistral	16.98
Phi-3 Mini	56.45

The introduction of controls results in measurable degradation in output accuracy across all models. Phi-3 Mini experiences the most significant decline, suggesting that its responses are particularly sensitive to input modification and filtering. In contrast, Mistral demonstrates the lowest level of degradation, which may be attributable to its less aggressive filtering approach.

The results indicate that stronger defensive mechanisms can interfere with the generative capabilities of LLMs, thereby reducing the quality and relevance of outputs. This further reinforces the inherent trade-off between security and usability.

5.6. Performance Overhead

This section addresses **RQ4** by analysing the computational overhead associated with the implementation of defensive controls. Table V summarises the computational overhead associated with the implementation of controls.

Table 5: Performance Overhead

Model	CPU Increase (%)	RAM Increase (%)
Gemma 3	0.45	5.2
Llama 3	0.60	6.1
Mistral	0.30	4.8
Phi-3 Mini	0.55	5.9

The results indicate that the implementation of layered controls introduces only minimal computational overhead. Increases in CPU utilisation remain below 1% for all models, while memory usage shows modest increase. These findings suggest that performance cost is not a primary barrier to the adoption of such controls in practice.

Overall Interpretation

Taken together, the results demonstrate that while layered defensive strategies can substantially enhance the security posture of LLMs, they do not provide a complete solution. The variation in model performance highlights the importance of both model-level design and external controls. Furthermore, the observed trade-offs between detection, false positives, and accuracy underscore the complexity of deploying secure and usable LLM systems.

6. Discussion

The findings of this study provide a comprehensive empirical assessment of the effectiveness of prompt injection mitigation strategies in Large Language Models, while also offering important insights into their limitations. Consistent with prior research, the results confirm that prompt injection remains a persistent and difficult-to-mitigate vulnerability in LLM systems [19], [20]. However, this study extends the existing body of knowledge by demonstrating that the effectiveness of defensive mechanisms is not uniform, but rather varies significantly depending on both model architecture and the interaction between layered controls.

In relation to detection performance, the observed improvements across all models following the implementation of defensive mechanisms align with prior studies that highlight the effectiveness of guardrail-based approaches and input filtering techniques [21]. Nevertheless, the results also indicate that such improvements are highly dependent on the baseline capabilities of each model. Llama 3, for instance, exhibited near-complete detection even prior to the introduction of controls and achieved full detection under controlled conditions. This suggests that intrinsic model characteristics, including training data composition and alignment strategies, play a critical role in determining resilience to prompt injection. In contrast, models such as Mistral and Phi-3 Mini demonstrated lower baseline detection

rates and relied more heavily on external controls to achieve comparable improvements. This disparity highlights the limitations of applying uniform defensive strategies across heterogeneous model architectures.

The reduction in attack success rates observed in this study is broadly consistent with prior empirical findings indicating that layered defences can significantly mitigate adversarial inputs [22]. However, the persistence of residual vulnerabilities across several models indicates that such defences are not sufficient to eliminate the threat entirely. While some models achieved near-complete or complete mitigation, others continued to exhibit successful attacks despite the presence of controls. This finding contrasts with certain studies that report high levels of mitigation effectiveness under controlled conditions, suggesting that such results may not generalise across different models or experimental configurations. Consequently, this study reinforces the importance of multi-model evaluation frameworks, which remain relatively underexplored in current research [23].

A central contribution of this work lies in its empirical examination of the trade-offs between security and usability. Although previous research has acknowledged that defensive mechanisms may impact model performance [19], there has been limited quantitative analysis of these effects. The findings presented here demonstrate that improvements in detection and attack mitigation are consistently accompanied by increases in false positive rates and measurable degradation in output accuracy. This indicates that strengthening security controls imposes constraints on the model's ability to generate accurate and contextually appropriate responses. Such trade-offs are particularly significant in practical deployments, where excessive false positives may disrupt legitimate user interactions and reduce overall system reliability.

A particularly noteworthy and counterintuitive finding is that certain prompt injection attacks were observed to succeed only after the implementation of defensive controls. This suggests that the interaction between control mechanisms and model behaviour is not purely protective, but may in some cases introduce unintended vulnerabilities. One plausible explanation is that input filtering or prompt modification alters the semantic structure of the prompt in a manner that bypasses the model's intrinsic safety mechanisms. While prior studies have identified the brittleness of static filtering approaches [21], the present findings extend this understanding by demonstrating that such mechanisms can actively contribute to new attack pathways. This highlights a critical limitation of rule-based and static defences, which are often unable to account for the dynamic and context-dependent nature of language.

From a theoretical perspective, these findings reinforce the argument that prompt injection is a structural vulnerability inherent to the design of LLMs. As noted in prior research, the fundamental challenge lies in the inability of models to reliably distinguish between trusted system-level instructions and untrusted user inputs [22]. The probabilistic nature of language generation further complicates this issue, as models are designed to respond flexibly to input rather than enforce strict boundaries. The results of this study provide additional empirical support for this perspective by demonstrating that even with layered defensive controls, complete mitigation is not consistently achievable. This suggests that addressing prompt injection may require fundamental changes to model architecture and training methodologies, rather than reliance on external controls alone.

From a practical standpoint, the implications of these findings are significant for organisations deploying LLM-based systems. The results indicate that while layered defensive mechanisms can substantially enhance security, they must be carefully calibrated to avoid adverse effects on usability and performance. In particular, the observed increase in false positives and degradation in output accuracy may impact user trust and limit the effectiveness of LLM applications in real-world contexts. The minimal computational overhead associated with the implementation of controls suggests that performance constraints are unlikely to be a primary barrier; however, the balance between security and usability remains a critical consideration. Organisations must therefore adopt adaptive and context-aware defence strategies that can dynamically respond to evolving threats without imposing excessive restrictions on legitimate use.

In comparison with prior empirical studies, this work offers a more comprehensive and integrated evaluation of prompt injection attacks and defences. While much of the existing literature has focused on either attack development or individual defensive mechanisms, the present study examines their interaction within a multi-model experimental framework. This approach enables a more holistic understanding of the strengths and limitations of current mitigation strategies, particularly in relation to trade-offs between security, usability, and performance. As such, the findings contribute to a more nuanced understanding of LLM security and provide a foundation for future research in this area.

The findings of this study provide clear and structured answers to the research questions outlined in the Introduction.

In relation to **RQ1**, the results demonstrate that defensive mechanisms substantially improve the detection of prompt injection attacks across all evaluated models. Detection rates increased significantly following the introduction of

controls, with some models achieving near-complete or complete detection. However, the magnitude of improvement varied depending on the baseline capabilities of each model, indicating that detection performance is influenced by both intrinsic model properties and external controls.

With respect to **RQ2**, the implementation of layered defensive mechanisms proved effective in reducing the success rate of prompt injection attacks. In several cases, attacks were completely mitigated, while in others substantial reductions were observed. Nevertheless, the persistence of residual vulnerabilities in certain models indicates that these mechanisms do not provide comprehensive protection.

Regarding **RQ3**, the findings reveal a clear trade-off between security and usability. The introduction of defensive controls resulted in increased false positive rates and measurable degradation in output accuracy across all models. This suggests that improvements in security are accompanied by reductions in model usability and output fidelity, highlighting the need for balanced and carefully calibrated defence strategies.

Finally, in addressing **RQ4**, the results indicate that the computational overhead associated with defensive mechanisms is minimal. Increases in CPU and memory usage were modest across all models, suggesting that performance is not a significant constraint in the implementation of such controls.

Taken together, these findings emphasise that while defensive mechanisms can significantly enhance the security of LLMs, they do not fully resolve the underlying vulnerability to prompt injection. Instead, they introduce a set of trade-offs that must be carefully managed in practical deployments.

7. Implications

The implications of this research extend beyond the specific models evaluated in this study. From a theoretical perspective, the findings reinforce the notion that prompt injection attacks are deeply rooted in the architectural characteristics of LLMs. As such, addressing this vulnerability may require fundamental innovations in model design, rather than incremental improvements to existing defensive techniques.

From a practical standpoint, the results highlight the necessity for organisations to adopt a layered and adaptive approach to securing LLM deployments. While input filtering, prompt hardening, and output validation can provide meaningful risk reduction, these measures must be carefully calibrated to balance security with usability. Continuous monitoring and iterative refinement of defensive mechanisms are essential to address the evolving nature of adversarial threats.

8. Limitations and Future Work

While this study provides a structured and empirical evaluation of prompt injection defences in Large Language Models, several limitations should be acknowledged.

A primary limitation relates to the size of the dataset. The evaluation was conducted using a dataset of 100 prompts, comprising an equal distribution of malicious and benign inputs. Although this size enables controlled comparison and detailed analysis, it may not fully capture the diversity and complexity of real-world interactions. Larger-scale datasets could provide greater statistical robustness and enable the identification of more subtle behavioural patterns.

A second limitation concerns the controlled nature of the experimental environment. All models were evaluated under consistent and predefined conditions, including fixed prompts and defensive configurations. While this approach ensures internal validity, it does not fully reflect real-world deployment scenarios, where LLMs operate in dynamic environments involving multi-turn interactions, user variability, and integration with external systems. As a result, the observed effectiveness of defensive mechanisms may differ in practical applications.

Model-specific bias also represents an important limitation. The study evaluates four models selected to represent diversity in architecture and deployment characteristics; however, these models may not be representative of the full spectrum of available LLMs. Differences in training data, alignment strategies, and architectural design may influence both vulnerability to prompt injection and responsiveness to defensive mechanisms. Consequently, the findings should not be generalised to all models without further validation.

Additionally, the study is dependent on a predefined set of prompt injection techniques derived from existing literature. While these prompts were designed to reflect realistic and diverse attack patterns, they cannot encompass the full range of potential adversarial strategies. Prompt injection attacks continue to evolve rapidly, and novel techniques may exploit vulnerabilities not captured within the current dataset. This highlights the inherent challenge of evaluating security in a continuously changing threat landscape.

Future work should address these limitations by expanding the scale and diversity of prompt datasets, including the incorporation of real-world interaction logs where feasible. Further research should also explore the evaluation of defensive mechanisms in more dynamic and complex deployment environments, particularly those involving multi-turn conversations and external tool integration. In addition, extending the analysis to a broader range of models, including proprietary systems and domain-specific LLMs, would enhance the generalisability of findings.

Finally, there is a need for the development of more adaptive and context-aware defence strategies that move beyond static filtering approaches. Techniques such as dynamic prompt analysis, intent classification, and continuous monitoring of conversational context represent promising directions for improving the robustness of LLM systems against prompt injection attacks.

9. Conclusion

This paper has presented a rigorous empirical evaluation of prompt injection attacks and mitigation strategies across multiple Large Language Models. The findings demonstrate that while layered defensive controls can substantially reduce attack success rates, they introduce significant trade-offs in terms of usability and model performance. Moreover, the results reveal that certain defensive mechanisms may inadvertently introduce new vulnerabilities, highlighting the complexity of securing LLM systems.

Future research should focus on the development of adaptive, context-aware defence mechanisms that can dynamically respond to evolving attack strategies without compromising usability. Additionally, further investigation into the underlying causes of control-induced vulnerabilities may provide valuable insights for improving the robustness of LLM architectures. Addressing these challenges will be critical to ensuring the secure and reliable deployment of LLMs in increasingly sensitive and high-stakes applications.

10. References

[1] L. Banh and G. Strobel, “Generative Artificial Intelligence,” *Electronic Markets*, vol. 33, no. 1, pp. 1–17, 2023. doi: 10.1007/s12525-023-00680-1.

[2] R. Y. Choi, A. S. Coyner, J. Kalpathy-Cramer, M. F. Chiang and J. P. Campbell, “Introduction to Machine Learning, Neural Networks, and Deep Learning,” *Translational Vision Science & Technology*, vol. 9, no. 2, p. 14, 2020. doi: 10.1167/tvst.9.2.14.

[3] S. Abdelnabi, K. Greshake, 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,” 2023. doi: 10.1145/3605764.3623985.

[4] E. Derner et al., “A Security Risk Taxonomy for Prompt-Based Interaction with Large Language Models,” *IEEE Access*, vol. 12, pp. 126176–126187, 2024. doi: 10.1109/ACCESS.2024.3450388.

[5] V. Benjamin et al., “Systematically Analyzing Prompt Injection Vulnerabilities in Diverse LLM Architectures,” *arXiv preprint arXiv:2410.23308*, 2024. doi: 10.48550/arxiv.2410.23308.

[6] T. Geng, Z. Xu, Y. Qu, and W. E. Wong, “Prompt Injection Attacks on Large Language Models: A Survey of Attack Methods, Root Causes, and Defense Strategies,” *Computers, Materials & Continua*, vol. 0, no. 0, pp. 1–10, 2025, doi: https://doi.org/10.32604/cmc.2025.074081.

[7] A. Alzahrani, “PromptGuard a structured framework for injection resilient language models,” *Scientific Reports*, vol. 16, no. 1, pp. 1277–1277, Jan. 2026, doi: https://doi.org/10.1038/s41598-025-31086-y.

[8] S. Gulyamov *et al.*, “Prompt Injection Attacks in Large Language Models and AI Agent Systems: A Comprehensive Review of Vulnerabilities, Attack Vectors, and Defense Mechanisms,” *Information*, vol. 17, no. 1, p. 54, Jan. 2026, doi: https://doi.org/10.3390/info17010054.

[9] A. Alobaid, M. J. Roca, C. Castillo and J. Vendrell, "The Echo Chamber Multi-Turn LLM Jailbreak," *arXiv preprint arXiv:2601.05742*, 2026.

[10] S. A. Akheel, "Guardrails for Large Language Models: A Review of Techniques and Challenges," *Journal of Artificial Intelligence, Machine Learning and Data Science*, vol. 3, no. 1, pp. 2504–2512, 2025. doi: 10.51219/jaimld/syed-arham-akheel/536.

[11] J. Dai et al., "Safe RLHF: Safe Reinforcement Learning from Human Feedback," *arXiv preprint arXiv:2310.12773*, 2023. doi: 10.48550/arxiv.2310.12773.

[12] Raden Budiarto Hadiprakoso, Wiyar Wilujengning, and Amiruddin Amiruddin, "Adaptive Multi-Layer Framework for Detecting and Mitigating Prompt Injection Attacks in Large Language Models," *Journal of Information Systems Engineering and Business Intelligence*, vol. 11, no. 3, pp. 473–487, Oct. 2025, doi: https://doi.org/10.20473/jisebi.11.3.473-487.

[13] S. Gulyamov *et al.*, "Prompt Injection Attacks in Large Language Models and AI Agent Systems: A Comprehensive Review of Vulnerabilities, Attack Vectors, and Defense Mechanisms," *Information*, vol. 17, no. 1, p. 54, Jan. 2026, doi: https://doi.org/10.3390/info17010054.

[14] L. Chen and G. Varoquaux, "What is the Role of Small Models in the LLM Era: A Survey," *arXiv preprint arXiv:2409.06857*, 2024. doi: 10.48550/arxiv.2409.06857.

[15] L. Thode, U. Iftikhar, and D. Mendez, "Exploring the use of LLMs for the Selection phase in systematic literature studies," *Information and Software Technology*, p. 107757, May 2025, doi: https://doi.org/10.1016/j.infsof.2025.107757.

[16] T. A. Slocum, S. E. Pinkelman, P. R. Joslyn, and B. Nichols, "Threats to Internal Validity in Multiple-Baseline Design Variations," *Perspectives on Behavior Science*, vol. 45, no. 3, Jan. 2022, doi: https://doi.org/10.1007/s40614-022-00326-1.

[17] S. Beckers, "Large Language Models as Nondeterministic Causal Models," *arXiv.org*, 2025. https://arxiv.org/abs/2509.22297 (accessed Apr. 29, 2026).

[18] A. J. Averitt, P. B. Ryan, C. Weng, and A. Perotte, "A conceptual framework for external validity," *Journal of Biomedical Informatics*, vol. 121, p. 103870, Sep. 2021, doi: https://doi.org/10.1016/j.jbi.2021.103870.

[19] S. Abdelnabi, K. Greshake, 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," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2023. doi: 10.1145/3605764.3623985.

[20] E. Derner et al., "A Security Risk Taxonomy for Prompt-Based Interaction with Large Language Models," *IEEE Access*, vol. 12, pp. 126176–126187, 2024. doi: 10.1109/ACCESS.2024.3450388.

[21] S. A. Akheel, "Guardrails for Large Language Models: A Review of Techniques and Challenges," *Journal of Artificial Intelligence, Machine Learning and Data Science*, vol. 3, no. 1, pp. 2504–2512, 2025. doi: 10.51219/jaimld/syed-arham-akheel/536.

[22] D. Ayzenshteyn, R. Weiss and Y. Mirsky, "Cloak, Honey, Trap: Proactive Defenses Against LLM Agents," in *Proceedings of the 34th USENIX Security Symposium*, 2025, pp. 8095–8114.

[23] V. Benjamin et al., "Systematically Analyzing Prompt Injection Vulnerabilities in Diverse LLM Architectures," *arXiv preprint arXiv:2410.23308*, 2024. doi: 10.48550/arXiv.2410.23308.