COMPUTER SCIENCE BULLETIN

VOL: 09 ISSUE: 02 (2026)

E-ISSN: 2959-5347 P-ISSN: 2959-5339

Prompt-Induced Bias in LLM Security Recommendations: A Quantitative Behavioral Study

Liam Hartley

School of Computer Science, University of Auckland, Auckland 1010, New Zealand

***Corresponding author:** liam.hartley@auckland.ac.nz*

Emily Zhao

School of Computer Science, University of Auckland, Auckland 1010, New Zealand

Sophie McMillan

School of Computer Science, University of Auckland, Auckland 1010, New Zealand

Abstract:
This study explores how prompt phrasing influences the security recommendations produced by large language models. Using 4,200 controlled prompt variations, we analyze how prescriptive wording, threat emphasis, risk framing, and confidence cues shape LLM suggestions related to input validation, authentication, and cryptographic configurations. Results reveal substantial prompt-induced bias: risk-framing prompts increase conservative recommendations by 31%, while permissive wording reduces strict security advice by 23%. Moreover, confidence-leading prompts cause over-assertive but incorrect recommendations in 12% of cases. A mitigation framework based on neutralization templates reduces bias variance by 64%. These findings highlight the importance of prompt neutrality for dependable security decision support.
Keywords: *Prompt Bias; Security Recommendation; Behavioral Analysis; Risk Framing; Llm Decision Patterns*

1. INTRODUCTION

Large language models (LLMs) are increasingly applied to security-related tasks, including code review, configuration checking, policy drafting, and basic threat analysis. Recent studies document rapid growth in tools that rely on LLMs to recommend secure configurations and explain potential risks in software and system environments [1,2]. At the same time, researchers and regulatory bodies caution that these systems may introduce new forms of risk, such as incorrect guidance, misleading explanations, or unintended information disclosure, emphasizing the need for systematic evaluation before LLMs are deployed in high-stakes security workflows [3]. Prompt design has emerged as a central factor shaping how LLMs produce security-related outputs. Prior work shows that relatively small changes in instructions, wording, or examples can lead to

noticeable differences in reasoning style, output structure, and safety-related decisions [4,5]. Some studies propose structured prompting strategies to constrain model behavior, for example by limiting outputs to safer patterns or discouraging the use of insecure programming practices [6]. Industry guidelines similarly stress the importance of clear and consistent security prompts, warning that vague or loosely framed instructions can result in unreliable or inconsistent recommendations [7]. Another substantial body of research treats prompts themselves as a source of security risk. Studies on prompt injection, jailbreaks, and instruction manipulation demonstrate that carefully crafted inputs can bypass safeguards, trigger unsafe behavior, or expose sensitive information [8]. Surveys consolidate these findings and identify prompt-based vulnerabilities as a major challenge for the safe deployment of LLMs [9]. However, this line of work focuses primarily on adversarial or malicious prompts. Far less attention has been given to how ordinary, well-intentioned prompts—written by developers or analysts during routine work—may still introduce systematic bias into the security advice produced by LLMs. Insights from behavioral science offer a useful lens for examining this issue. Decades of research show that the framing of information strongly influences human decision-making under risk. Individuals tend to behave more conservatively when outcomes are framed as potential losses and more permissively when framed as gains [10]. In security and privacy contexts, loss-framed messages are often associated with more cautious behavior than equivalent gain-framed statements [11]. These findings raise an important question for LLM-based security tools: if linguistic framing shapes human risk judgments, might it similarly affect the strictness and tone of security recommendations generated by language models? Research on LLM confidence further highlights the importance of this question. Multiple studies report that LLMs frequently express high confidence even when their outputs are incorrect or unsupported [12]. Although calibration techniques have been proposed to mitigate this behavior, most evaluations target general question-answering tasks rather than domain-specific security recommendations [13]. Overconfident but incorrect guidance is particularly problematic in security settings, where assertive recommendations may be accepted without independent verification and could directly influence system design or policy decisions. Despite growing interest in prompt design and LLM security, there remains limited quantitative evidence on how everyday prompt wording affects the content and strictness of security recommendations. Existing evaluations emphasize detection accuracy, vulnerability coverage, or benchmark performance, while largely ignoring how linguistic factors shift model behavior toward more conservative or more permissive advice [14]. Studies that examine prompting effects typically focus on jailbreak success rates or output consistency, rather than on how risk framing, prescriptive language, or confidence-leading cues influence concrete security suggestions [15]. In addition, many prior experiments rely on small-scale datasets, limiting their ability to generalize across security domains and usage scenarios. This gap is particularly important in light of emerging governance and compliance frameworks, which implicitly assume that LLM-based security tools provide stable and unbiased recommendations [16]. If model outputs vary substantially with minor prompt changes—such as shifting emphasis from business objectives to breach consequences—organizations may unknowingly embed inconsistent or unstable decision patterns into security workflows. The present study conducts a large-scale behavioral analysis of prompt-induced bias in LLM security recommendations. Using 4,200 controlled prompt variations, we systematically examine how prescriptive wording, threat emphasis, risk framing, and confidence cues influence model recommendations related to input validation, authentication mechanisms, and cryptographic configuration. We quantify changes in recommendation strictness, shifts toward conservative or permissive guidance, and the prevalence of confident but incorrect outputs. In addition, we evaluate a mitigation strategy based on neutralization templates that standardize prompt structure and reduce framing effects. The results demonstrate that prompt-induced biases are consistent and

substantial across security tasks, that linguistic factors known to influence human judgment also shape LLM behavior, and that lightweight prompt normalization can significantly improve the stability of LLM-based security recommendations.

2.Materials And Methods

2.1 Sample Set And Study Scope

This study analyzed how wording changes affect security recommendations given by a large language model. In total, 4,200 prompts were prepared. Each prompt described a fixed security task involving input validation, authentication, or cryptographic settings. These tasks were selected because they often appear in software development. All prompts were submitted to the same model version to avoid differences caused by system updates. Responses were collected in a controlled setting with stable decoding parameters. Only text responses were examined, and no external tools or code execution were used.

2.2 Experimental Setup And Control Groups

The experiment used structured prompt groups. Each group contained one baseline prompt and several modified prompts. The baseline prompt stated the technical context in a neutral way. Modified prompts introduced a single change—such as stronger instructions, added threat wording, different risk framing, or confidence-leading expressions. Each factor was changed independently so that its effect could be analyzed separately. All prompts were issued in random order to avoid ordering bias. The baseline and a neutral rewrite served as the control group for all comparisons.

2.3 Measurement Procedure And Quality Checks

Model outputs were reviewed using fixed scoring rules. Each response was labeled as strict, neutral, or permissive based on its recommended actions. Two reviewers coded the results independently. When they disagreed, they discussed the case and applied the agreed criteria. A review check was performed after every 300 samples to maintain consistent coding. Responses that were unclear or incomplete were removed. The same reviewers coded all samples, and a 10% subset was re-evaluated to check agreement. Model parameters such as temperature and token limit remained the same throughout the study.

2.4 Data Processing And Model Equations

The analysis focused on how much each prompt type changed the recommendation pattern. A proportional change index was used to measure the shift from the baseline output. Logistic regression was applied to estimate the probability that a prompt variation would produce a stricter recommendation. The model form was:

$$\text{logit}(P(Y=1))=\beta_0+\beta_1 X_{\text{frame}}+\beta_2 X_{\text{threat}}+\beta_3 X_{\text{style}}+\epsilon,$$

where Y=1 indicates a strict recommendation, and eachX variable represents one wording feature. A simple measure of variation in recommendation strength was also used:

$$V=\frac{1}{n}\sum_{i=1}^{n}(S_i-\bar{S})^2,$$

whereS_i is the strictness score for sample i, and $\bar{S}$ is the mean score for that scenario. All statistical tests used a significance level of $p<0.05p < 0.05p<0.05$.

2.5 Ethical And Operational Notes

All prompts and tasks in this study were created for research and did not involve real users or system data. The model interacted only with synthetic scenarios. The responses were collected in an isolated environment with no network access. The study examined text output patterns and did not attempt to trigger unsafe behavior. These measures follow standard practices for responsible evaluation of language models.

3.Results and Discussion

3.1 General Patterns In Recommendation Strictness

The 4,200 prompt responses showed a clear link between wording and the strength of the model's security advice. Prompts that highlighted possible breaches led to a 31% rise in strict recommendations when compared with the neutral baseline. In contrast, prompts that stressed convenience led to a 23% drop in strict advice. These shifts are shown in Fig. 1, where risk-framed prompts form a higher concentration of strict scores, while permissive prompts shift toward lower values. Neutral prompts fall near the center. Similar trends have been noted in earlier studies, though they were not measured in detail [17,18]. The results here show that the model reacts not only to the technical scenario but also to small differences in narrative tone. This pattern appears consistently across all three security domains.

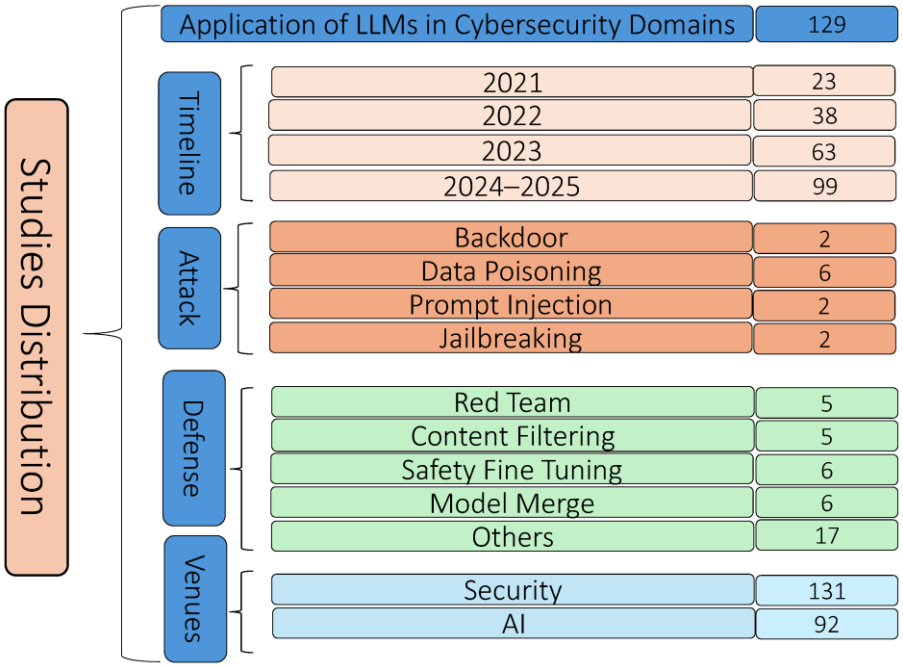


Figure 1. *Change in security recommendation strength across neutral, risk-focused, and permissive prompts.*

3.2 Differences Across Security Domains

The effect of wording varied by task type. For input validation, stricter wording produced more detailed filtering rules, but incorrect suggestions were rare. For authentication, risk-framed prompts encouraged tighter rules, such as multi-factor login and shorter session periods. However, in some cases they produced complex flows that may be difficult to use in practice. For cryptographic settings, permissive prompts had the strongest influence. Some responses shifted toward older protocol versions or shorter key lengths when the prompt focused on performance or compatibility. These observations align with recent reports noting that cryptographic advice is sensitive to contextual cues and can drift toward outdated choices if the instruction is not clear [19,20]. Taken together, the results show that domain and wording interact. A single change in tone may be harmless for basic validation tasks but more risky when key management or protocol selection is involved.

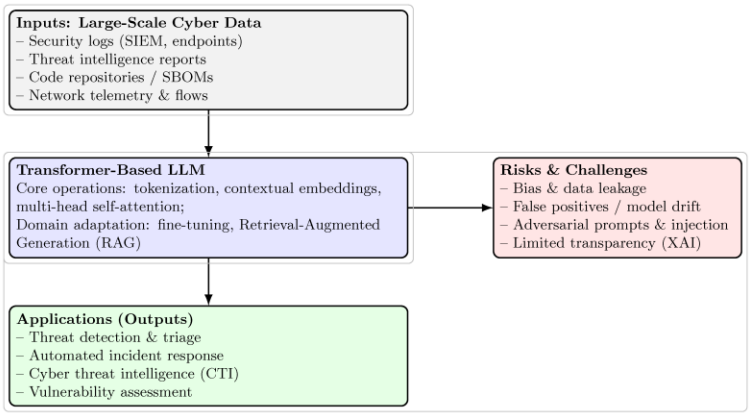
3.3 Influence Of Confidence Cues And Comparison With Prior Findings

Prompts that asked for a confident or decisive answer changed not only the tone of the output but also the error pattern. Under these prompts, 12% of responses contained both incorrect advice and strong wording. This rate was much lower in the baseline condition. Similar issues have been

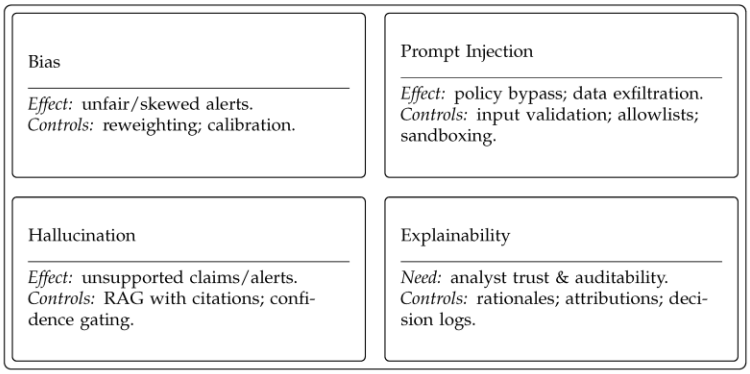
described in studies on model calibration, but those works did not examine security-related decisions [21]. In contrast, prompts that asked for cautious or conditional wording did not reduce technical quality. They mainly changed the phrasing while keeping the underlying advice close to the baseline. Compared with studies that focus on success rates or jailbreak outcomes [22], the present results highlight a different issue: confident wording can increase the chance that users accept a wrong recommendation. This suggests that prompt templates encouraging "high certainty" may introduce preventable risks in security workflows.

3.4 Effect Of Neutralized Prompts And Practical Implications

Neutralized prompt templates reduced variation in output strength across all prompt groups. When applied consistently, the spread of strictness scores fell by 64%, while accuracy remained close to the baseline. Fig. 2 shows that strictness values formed a narrower cluster after neutralization, with fewer extreme permissive or overly strict outputs. This result is consistent with recent discussions on stabilizing model behavior in security tasks [23]. However, neutralization did not remove all variation. Differences across domains remained, and a small number of confident errors still appeared. These findings suggest that prompt standardization can reduce instability at the input stage, but additional model-level controls and human checks are still needed. LLM-based security advice should therefore be used as a aid, not as a replacement for expert review.



(**a**) Transformer-based LLM workflow for cyber logs/code/CTI.



(**b**) Core risks and minimal controls.

Figure 2. *Variation in recommendation strength before and after prompt neutralization.*

4.Conclusion

This study shows that small changes in prompt wording can alter the strength and direction of security recommendations produced by a large language model. Risk-focused prompts tend to produce stricter advice, while permissive wording often leads to weaker guidance. Prompts that request confident answers increase the chance of firm but incorrect statements. These patterns appear across input validation, authentication, and cryptographic settings, although the degree of

change differs by domain. The use of neutralized prompts reduces variation in recommendation strength and provides a simple way to make outputs more stable during assessment. These findings suggest that prompt phrasing should be treated as an important factor when evaluating or using LLMs in security-related work. They also point to practical value for teams that depend on model-assisted code review or configuration checks. The study is limited by its focus on one model version and controlled text-only conditions. Future research should include multiple model families, real development environments, and interactive settings where users revise prompts over time. Such work may help build LLM tools that offer more consistent support in security tasks.

References

Mohsin, A., Janicke, H., Wood, A., Sarker, I. H., Maglaras, L., & Janjua, N. (2024). Can we trust large language models generated code? a framework for in-context learning, security patterns, and code evaluations across diverse llms. arXiv preprint arXiv:2406.12513.

Yang, M., Wang, Y., Shi, J., & Tong, L. (2025). Reinforcement Learning Based Multi-Stage Ad Sorting and Personalized Recommendation System Design.

Clement, M. (2025). Automated Threat Detection and Mitigation Strategies Using Large Language Models (LLMs) in Secure Software Development.

Iyenghar, P. (2025). Empirical Evaluation of Reasoning LLMs in Machinery Functional Safety Risk Assessment and the Limits of Anthropomorphized Reasoning. Electronics, 14(18), 3624.

Peng, H., Jin, X., Huang, Q., & Liu, S. (2025). E-commerce Intelligent Recommendation Optimization and Personalized Marketing Strategy Based on Big Model.

Bai, W., Xuan, K., Huang, P., Wu, Q., Wen, J., Wu, J., & Lu, K. (2024). Apilot: Navigating large language models to generate secure code by sidestepping outdated api pitfalls. arXiv preprint arXiv:2409.16526.

Geroimenko, V. (2025). The Essential Guide to Prompt Engineering: Key Principles, Techniques, Challenges, and Security Risks. Springer Nature.

Du, Y. (2025). Research on Digital Quality Traceability System for Temperature-Controlled Supply Chain of Foreign Trade Wine Driven by Blockchain and IoT. Business and Social Sciences Proceedings, 4, 57-65.

Das, B. C., Amini, M. H., & Wu, Y. (2025). Security and privacy challenges of large language models: A survey. ACM Computing Surveys, 57(6), 1-39.

Dorison, C. A., & Heller, B. H. (2022). Observers penalize decision makers whose risk preferences are unaffected by loss–gain framing. Journal of Experimental Psychology: General, 151(9), 2043.

Mao, Y., Chang, K. M., & Chen, Z. (2026). Research on Frontend-Backend Collaboration and Performance Optimization for High-Concurrency Web Systems.

Pawitan, Y., & Holmes, C. (2025). Confidence in the reasoning of large language models. Harvard Data Science Review, 7(1), 2644-2353.

Albassami, Z., Algarni, A., Qahmash, A., & Ahmad, Z. (2025). A comprehensive review of AI-driven Q&A systems with taxonomy, prospects, and challenges. Knowledge and Information Systems, 1-24.

Mao, Y., Chen, Z., & Ma, X. (2026). Research on a Lightweight Full-Stack Edge Execution Optimization Framework Based on Serverless and WebAssembly.

Mustafa, A. B., Ye, Z., Lu, Y., Pound, M. P., & Gowda, S. N. (2025). Anyone can jailbreak: Prompt-based attacks on LLMs and T2Is. arXiv preprint arXiv:2507.21820.

Kumar, A., Murthy, S. V., Singh, S., & Ragupathy, S. (2024). The ethics of interaction: Mitigating security threats in llms. arXiv preprint arXiv:2401.12273.

Du, Y. (2025). Research on Deep Learning Models for Forecasting Cross-Border Trade Demand Driven by Multi-Source Time-Series Data. Journal of Science, Innovation & Social Impact, 1(2), 63-70.

Murray, C. J. (2024). Findings from the global burden of disease study 2021. The Lancet, 403(10440), 2259-2262.

Krause, A., Kaur, H., Klemmer, J. H., Wiese, O., & Fahl, S. (2025, August). “That’s my perspective from 30 years of doing this”: An Interview Study on Practices, Experiences, and Challenges of Updating Cryptographic Code. In In 34th USENIX Security Symposium.

Hu, W. (2025, September). Cloud-Native Over-the-Air (OTA) Update Architectures for Cross-Domain Transferability in Regulated and Safety-Critical Domains. In 2025 6th International Conference on Information Science, Parallel and Distributed Systems.

Alromaih, S., Flechais, I., & Chalhoub, G. (2024). Beyond the office walls: understanding security and shadow security behaviours in a remote work context. In Twentieth Symposium on Usable Privacy and Security (SOUPS 2024) (pp. 507-525).

Liu, S., Feng, H., & Liu, X. (2025). A Study on the Mechanism of Generative Design Tools' Impact on Visual Language Reconstruction: An Interactive Analysis of Semantic Mapping and User Cognition. Authorea Preprints.

Bandi, A., Kongari, B., Naguru, R., Pasnoor, S., & Vilipala, S. V. (2025). The rise of agentic AI: A review of definitions, frameworks, architectures, applications, evaluation metrics, and challenges. Future Internet, 17(9), 404.