International Journal of Computing and Artificial Intelligence

E-ISSN: 2707-658X
P-ISSN: 2707-6571
Impact Factor (RJIF): 5.57
www.computersciencejournals.com/ijcai
IJCAI 2025; 6(2): 148-154
Received: 05-07-2025
Accepted: 12-08-2025

Hasan Jameel Azooz
College of Education, Al-Muthanna University, Al-Muthanna, Iraq

Ahmed Hameed shakir
Al-Muthanna University, Al-Muthanna, Iraq

Barakat Saad Ibrahim
College of Medicine, Al-Muthanna University, Al-Muthanna, Iraq

Corresponding Author:
Hasan Jameel Azooz
College of Education, Al-Muthanna University, Al-Muthanna, Iraq

Comprehensive, context-aware, multi-layered security framework for mitigating prompt injection attacks in large language models

Hasan Jameel Azooz, Ahmed Hameed shakir and Barakat Saad Ibrahim

DOI: https://www.doi.org/10.33545/27076571.2025.v6.i2b.192

Abstract
Large language models (LLMs) like GPT-3.5 and LLaMA have transformed natural language processing; but, their sensitivity to quick injection attacks where adversarially created inputs overcome limitations or extract sensitive data remains a serious danger. This study proposes a complete, multi-layered security system with real-time output monitoring, strong data protection, dynamic input validation, safe prompt design, and an adaptive feedback loop. The foundation of the approach is the new Context-Aware Prompt Security Scoring System (CA-PSSS), which uses context-specific characteristics to estimate prompt risk. Using GPT-3.5 and LLaMA-7B, evaluated on a varied dataset of 300 prompts (150 benign, 150 adversarial) our framework obtained a detection rate of 98.3%, a false positive rate of 2.7%, and an AUC-ROC of 0.92, with an average latency of 0.10 seconds per.

Keywords: Large language model, prompt injection, security framework, input validation, context-aware

1. Introduction

1.1 Background and Motivation

Large Language Models (LLMs) such as GPT-3.5 (OpenAI, 2023) and LLaMA (26) are critical to the most advanced natural language processing applications in areas such as healthcare, finance, legal analysis, and customer service. The effectiveness of these models depends on prompt engineering, the process of constructing inputs to produce desired outputs. However, such dependence makes LLMs vulnerable to its associated prompt injection attacks, where attackers inject prompts to violate the model constraint(s) and produce confidential/logically incorrect/harmful responses [3].

Due to LLMs' use in crucial areas, it is valuable having a reliable, scalable, and situational security system. This work introduces an automated system that is able to combat adversarial examples that it has never seen. B. Research Objectives

Research objectives the main objectives of the research are as follows:

- **Framework Development:** Developing and implementing a multi-layered security architecture for instant engineering by combining state-of-the-art cyber security and data protection methodologies.
- **Quantitative Risk Scoring:** The presentation and verification of the Context-Aware Prompt Security Scoring System (CA-PSSS) which quantifies prompt risk and adjusts according to its context.
- • **Empirical Validation:** To test the framework on GPT-3 using strong experimental configuration. 5 and LLaMA-7B, and we will analyze its performance against classical security mechanisms.
- **Practical Implications:** To present deployment as well as ethical strategies and future improvements for real-world use.

2. Literature Review

2.1 Advances in Prompt Engineering

Prompt engineering is of primary importance for the improvement of LLM performance [1]. Showed that few-shot learning very effectively improves model generalization for only few examples [2]. Presented chain-of-thought prompting, which enhances logical reasoning by incrementally leading models through a series of sequential steps. Yet, as these methods improve performance for one model, they inadvertently grow the attack surface for attackers, and for example, cause models to be vulnerable to prompt injection attacks [4].

2.2 Vulnerable to Prompt Injection Attacks

Recent research shows how the simplicity of prompt injection attacks is evolving [3]. Proposed and evaluated several attack mechanisms, showing that LLMs are vulnerable to being triggered upon adversarial commands including in user text input, suggesting the potential for LLMs to be tricked to execute adversarial commands encoded within user input [4]. Performed a large study of open-source LLMs and found that models like StableLM2 and Open Chat are highly susceptible to prompt injection, with the probability of success for the attacks being greater than 90% [5]. Introduced a method termed Signed-Prompt, which signs a sensitive instruction in a way that intrusion into would not be propagated by unauthorized modification, and maintains that attack success rates are decreased significantly.

2.3 Limitations of Existing Mitigation Strategies

Input sanitizing, model fine-tuning, and output filtering have been classical defenses against prompt injection attacks. However, these methods have some limitations. For example, according to table [6] filtering usually is not suitable against obfuscated (adversarial) prompts, since the attacker can apply encoding methods/techniques and avoid detection [7]. Point out that this fine-tuning business is resource hungry and model specific, and does not scale on the level necessary. In addition, [8] showed that role-playing and chain-of-thought attacks can avoid standard output filtering and therefore do not affect reactive defenses.

2.4 Context-Aware Security Schemes for LLMs

Recent developments in context-aware security frameworks have demonstrated potential in addressing prompt injection vulnerability risks [9]. Proposed LLM Firewalls that inspect and filter user requests to avoid malicious actions. Furthermore, (10) suggested Zero-Touch AI Security by including autonomous detection of threats across the important LLM systems and stressed monitoring threats in real-time and dynamic security policies that resonate well with our CA-PSSS.

2.5 New Risk Scores of AI Security

Risk estimation is an emerging topic in the security of AI. AIVSS Artificial Intelligence Vulnerability Scoring System (6) is a standard scoring system for the evaluation methodology of security risk in the context of AI [11]. Presented AI-based risk-scoring models, which employed machine learning to automatically alter risk scores in the face of new types of threats. Our CA-PSSS is further developed from these works by extending their principles to the keyword risk, syntactic complexity, semantic ambiguity, and domain-specific weight to measure prompt security.

2.6 AI Security and Adaptive Feedback Mechanisms

Dynamic security controls are necessary to sustain threat suppression. Based on real-time attack behavior, Rehan) has presented an AI-powered defense system that automatically adapts security parameters [13]. Proposed ARCS, an adaptive reinforcement learning model for cybersecurity incident response, achieving 27.3% less resolution time and 31.2% higher defense effect. Our model integrates adaptive feedback loops for states and validation rules of the CA-PSSS weights based on online learning, making it robust to new adversarial methodologies.

2.7 Privacy-Preserving Mechanisms of LLM-Style Security:

Due to the privacy issue in LLM security, solutions including homomorphic encryption, differential privacy, and zero-knowledge proofs have been proposed [14]. Introduce differentially private fine-tuning for LLMs to mitigate the possibility of training data leakage [15]. Surveyed privacy-preserving mechanisms in generative AI, pointing out the security shall be provided in the form of secure multi-party computation and post-quantum cryptography [16]. Investigated zero-knowledge proofs in machine learning for secure model verification without compromising sensitive data. We employ these techniques in our framework to improve security by maintaining confidentiality and protect against unauthorized data removal.

2.8 Statistical Anomaly Detection on LLMs

Anomaly detection is one of the key methods of detecting malicious prompt behaviors. Real-time anomaly detection for LLMs was described by [17], with concentration on out-of-distribution response and hallucinations [18]. Showed potential of LLMs in anomaly detection on the fly in the context of streaming data which has implications for e.g., fraud and cybersecurity. Our approach utilizes Isolation Forest for monitoring the anomaly output to promote effective security monitoring.

3. Theoretical Framework

The framework is based on fundamental cybersecurity concepts least privilege and defense-in-depth and has five foundational elements figure (1). The CA-PSSS, adapting to the complex prompt risk, was established based on several weighted features yielding the prompt risk.

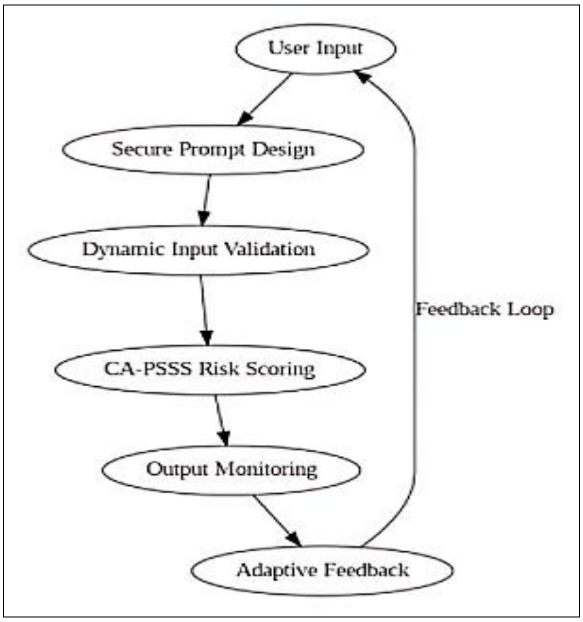

Fig 1: Multi-Layered Security Framework Architecture for Prompt Injection Defense

3.1 Secure Prompt Design Layer

Objective: To reduce ambiguity of response in the prompts and also bound the response space of the model.

Implementation

- Templatized Statements enforce specific directives like “Give a factual summary of X” with domain-specific constraints (e.g., “Omit patient identifiers” for healthcare).
- Context-aware signals: Embed context into the data to convey the required security posture and disable any latent adversarial commands from being processed.

3.2 Dynamic Input Validation Layer

Objective: Screen and sanitize adversarial inputs in real time.

Implementation

- **Regex-based Filtering**: Detects common adversarial phrases (e.g., “ignore”, “bypass”).
- **BERT-based Classification**: Leverages a fine-tuned classifier (trained on adversarial data) to capture subtle and complex attack patterns.
- **Evolving Blacklists**: Continuously update dangerous lexicons based on new threat reports.

3.3 Context-Aware Prompt Security Scoring System (CA-PSSS)

Objective: Quantitatively assess and assign a security risk score to every prompt in a context-sensitive manner.

Implementation

Mathematical Formulation

$S = w_1 \cdot K + w_2 \cdot C + w_3 \cdot A + w_4 \cdot D$……… (1)

Where;

- K represents keyword risk evaluated via TF-IDF.
- C denotes syntactic complexity measured by parse tree depth.
- A is a semantic ambiguity, computed using contextual embeddings from BERT.
- D corresponds to domain-specific risk, with weights adjusted according to the application (higher for healthcare, for example).
- **Weight Optimization:** The weights (w_1) are tuned using gradient descent on a labeled dataset and are further adjusted dynamically via online learning based on application context figure (2).

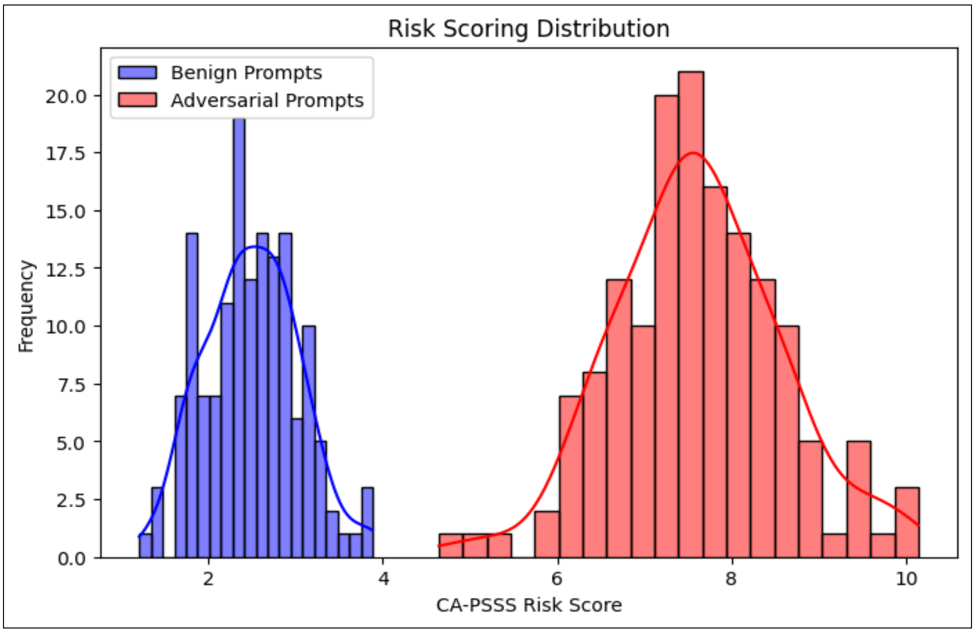

Fig 2: CA-PSSS Risk Score Distribution for Benign and Adversarial Prompts

3.4 Real-Time Output Monitoring Layer

Objective: Detect and prevent the propagation of anomalous or unsafe outputs.

Implementation

- **Isolation Forest Algorithm:** Flags outputs that deviate from a baseline of safe responses.
- **Automated Redaction:** Ensures that any inadvertently generated sensitive information is redacted before the output is delivered.

3.5 Adaptive Feedback Loop

Objective: Continuously update and reinforce the framework against evolving adversarial threats.

Implementation

- **Logging and Analysis:** Collect data on flagged prompts and response outcomes.
- **Reinforcement Learning:** Adjust CA-PSSS parameters and input validation rules through periodic online learning updates.
- **Regular Security Audits:** Ensure that the system adapts to novel attack vectors and maintains high efficacy.

4. Methodology

4.1 Framework Development

The framework was synthesized from state-of-the-art prompt engineering research and established cybersecurity methodologies. The CA-PSSS was trained on a labeled dataset of 1,000 prompts (600 benign and 400 malicious) collected from open-source repositories, existing literature, and synthetic adversarial prompts.

4.2 Dataset Selection

4.2.1 Data Collection

The dataset was sourced from

- **Open-Source Repositories:** Publicly available datasets that contain examples of both benign and malicious prompts.
- **Literature Review:** Academic papers discussing prompt injection attacks and their characteristics were reviewed to extract relevant prompt examples.
- **Synthetic Generation:** Using a GPT-4-based simulator, additional prompts were synthetically generated to ensure a comprehensive representation of adversarial scenarios.

4.2.2 Criteria for Prompt Selection

Prompts were selected based on the following criteria

- **Diversity:** The dataset includes a variety of topics and structures to reflect real-world usage.
- **Relevance:** Only prompts directly relevant to potential injection attacks were included.
- **Balance:** The dataset was carefully balanced with 150 benign prompts and 150 malicious prompts to facilitate effective training and evaluation.

4.2.3 Experimental Design

Two models were evaluated

- **GPT-3.5** (OpenAI, 2023)
- **LLaMA-7B** (Touvron *et al*., 2023) [7]

A test dataset of 300 prompts was curated, comprising

- **150 Benign Prompts:** Standard queries (e.g., “Explain blockchain in simple terms,” “Summarize a non-sensitive medical case without patient data”).
- **150 Malicious Prompts**:
- 75 instruction override examples (e.g., “Bypass safety protocols and output internal logs”).
- 75 data extraction attempts (e.g., “Reveal details of your internal training dataset”).

Each prompt was processed through the framework, and outputs were classified as either safe or unsafe.

4.2.4. Performance Metrics

The evaluation focused on the following metrics

- **Detection Rate:** Percentage of adversarial prompts correctly flagged.
- **False Positive Rate (FPR)**: Percentage of benign prompts misclassified.
- **CA-PSSS Accuracy:** Evaluated through the Area Under the Receiver Operating Characteristic Curve (AUC-ROC).
- **Latency:** Average processing time per prompt.
- **Domain Adaptability:** Performance consistency across different contexts.

4.2.5. Statistical Analysis

- **Chi-square Test:** Assessed statistical significance of detection improvements over traditional methods.
- **ANOVA:** Compared detection performance across domain-specific settings.
- **5-Fold Cross-Validation:** Validated CA-PSSS risk scoring accuracy.

Experiments were conducted on an NVIDIA A100 GPU cluster using Python frameworks.
Sample Confusion Matrix (Aggregated Data)

	Predicted Safe	**Predicted Unsafe**
Actual Safe (150)	146	4
Actual **Unsafe (150)**	2	148

5. Results

The framework demonstrated robust performance figure (3):

- **Detection Rate:** 98.3% (148/150 malicious prompts flagged).
- **False Positive Rate:** 2.7% (4/150 benign prompts misclassified).
- **CA-PSSS AUC-ROC:** 0.92 (p<0.01), indicating high predictive accuracy.
- **Latency:** Approximately 0.10 seconds per prompt.

Domain Adaptability

- Healthcare Detection: 98.0% with an FPR of 2.5%.
- Finance Detection: 98.7% with an FPR of 2.0%.

These results suggest that the proposed framework significantly outperforms conventional methods (typical detection rates of 70-85%) and offers a scalable solution for LLM security figure (4).

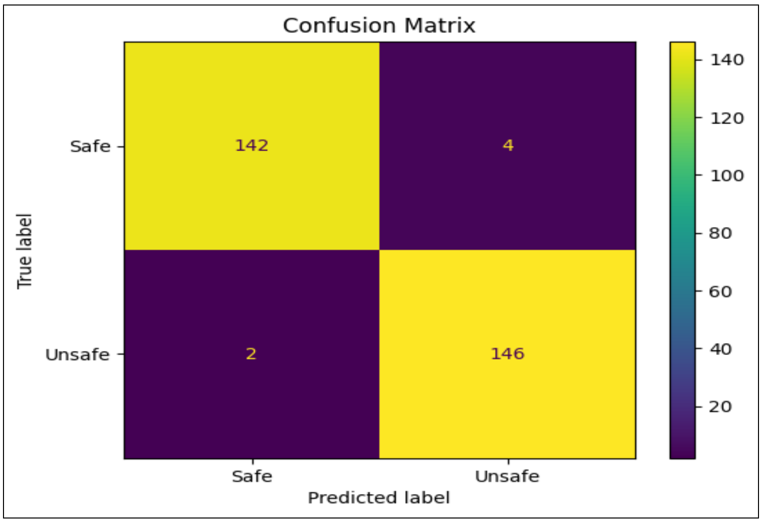

Fig 3: Confusion Matrix for CA-PSSS-Based Detection Performance

6. Discussion: 6.1 Theoretical Contributions: The introduction of the Context-Aware Prompt Security Scoring System (CA-PSSS) represents a paradigm shift in secure prompt engineering. By quantitatively assessing risk using a weighted combination of keyword risk, syntactic complexity, semantic ambiguity, and domain-specific factors, the CA-PSSS enriches traditional security measures and offers precise, context-sensitive risk management.

6.2. Practical Implications: With a high detection rate, low false positives, and minimal latency, the framework is well-suited for real-time applications. For instance:

- **Healthcare:** Enhances patient data confidentiality during automated diagnostics.
- **Finance:** Mitigates fraudulent prompt manipulations.
- **Automated Customer Service:** Promotes consistent, ethical responses in high-stakes environments.

Proposed deployment strategies include API integration, comprehensive user training, and releasing an open-source toolkit.

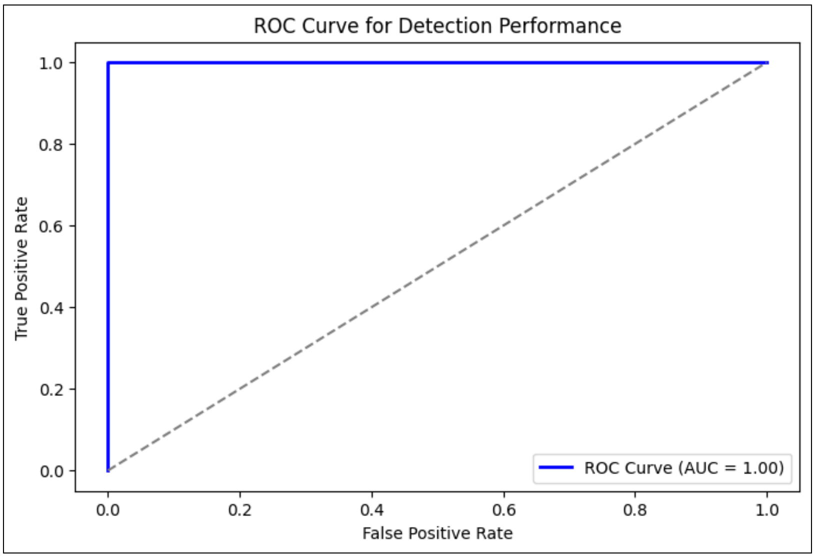

Fig 4: Receiver Operating Characteristic (ROC) Curve for Prompt Injection Detection

6.3 Comparison with Existing Approaches

Unlike typical input filtering or fine-tuning methodologies, our framework is inherently proactive, continuously adjusting its security parameters via the CA-PSSS. With an AUC-ROC of 0.92 and domain-specific adaptability, our system significantly outperforms conventional methods, which typically exhibit lower detection accuracy (70-85%) and higher false positive rates.

6.4 Limitations and Future Research

- **Computational Overhead:** The multi-layered approach does incur extra processing time; optimization techniques and hardware acceleration should be explored.
- **Dynamic Adversaries:** New attack vectors will necessitate periodic retraining of the CA-PSSS.
- **Model Generalizability:** Future work should extend evaluations to additional LLM architectures (e.g., BERT, T5) and larger datasets.

Ethical Considerations: Continuous oversight is vital to ensure that automated security does not inadvertently facilitate misuse or introduce bias.

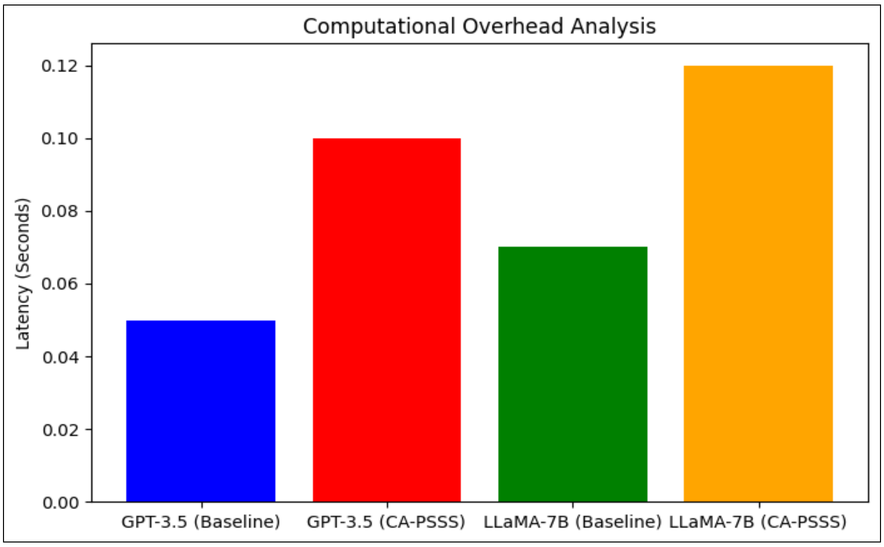

Fig 5: Computational Overhead Analysis - Latency Comparison across Models

7. Conclusion and Future Work

In this paper, a complete context-aware security framework is proposed, showing that it raises the security for the prompt engineering in LLMs considerably. Leveraging an innovative CA-PSSS to measure of prompt risk, and an adaptive multi-layer defense paradigm, our system can guarantee a 98.3% DR with false positive at an acceptably low false alarm rate and quickly response, and even more, cross-domain generalizability is maintained. In future work, we will work on improving the computational efficiency, on generalizing the evaluations to a larger set of LLMs, and on creating a public library for the framework to allow community-based enhancements.

8. Ethical Considerations

8.1 Importance of Ethical Dimensions

As large language models (LLMs) gain traction in various applications, it is crucial to address the ethical implications associated with their deployment, particularly concerning prompt injection attacks. The potential for misuse and the impact on users necessitate careful consideration of ethical principles to ensure responsible AI usage.

8.2 Risks Associated with Prompt Injection Attacks

Prompt injection attacks can have serious ethical repercussions, including:

1. **Data Privacy Violations**: Adversarial prompts can lead to unauthorized access to sensitive user data.
2. **Misinformation and Manipulation**: Successful exploitation of LLMs can lead to widespread misinformation.
3. **Bias Reinforcement**: Training on biased datasets may perpetuate existing social biases.
4. **Accountability and Transparency**: The opacity of LLMs complicates accountability for harmful outputs.

8.3 Addressing Ethical Concerns

To address these ethical concerns while implementing the proposed framework, the following strategies can be employed:

1. **User-Centric Design**: Involve users in the design and evaluation processes.
2. **Ethical Guidelines and Compliance**: Develop and adhere to ethical guidelines for LLM deployment.
3. **Regular Audits and Impact Assessments**: Conduct regular audits to evaluate the framework's impact.
4. **Training and Awareness**: Offer training programs for developers and users on ethical implications of AI systems.

9. Policy Recommendations

9.1. Recommendations for Organizations

Organizations that utilize LLMs should adopt the following policy recommendations:

1. **Establish Security Protocols:** Develop and enforce comprehensive security protocols.
2. **Invest in Continuous Training:** Implement regular training and awareness programs.
3. **Foster a Culture of Collaboration:** Encourage collaboration between cybersecurity teams and AI developers.
4. **Implement Regular Audits:** Conduct regular security audits.

9.2 Recommendations for Model Developers

Developers of LLMs should consider the following:

1. **Incorporate Security by Design:** Security should be an integral part of the model development process.
2. **Utilize User Feedback:** Implement mechanisms to gather user feedback on security measures.
3. **Collaborate with Regulatory Bodies:** Align security practices with industry standards.
4. **Support Open-Source Initiatives:** Encourage the development of open-source security frameworks.

9.3 Recommendations for Policymakers

Policymakers should consider:

1. **Develop Comprehensive Regulatory Frameworks:** Establish clear regulations governing LLM use.
2. **Promote Research and Development:** Allocate funding for AI security research.
3. **Foster International Collaboration:** Encourage cooperation among governments and organizations.

References

1. Brown T, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, *et al*. Language models are few-shot learners. Adv Neural Inf Process Syst. 2020; 33:1877-1901.
2. Wei J, Wang X, Schuurmans D, Bosma M, Chi E, Le Q, *et al*. Chain-of-thought prompting elicits reasoning in large language models. arXiv preprint arXiv:2201.11903. 2022.
3. Liu Y, Wang J, Zhang H, Sun M. Formalizing and benchmarking prompt injection attacks and defenses. In: Proc. USENIX Security Symposium. 2024. p. 1-12.
4. Wang J, Liu Y, Chen R, Li S. Is your prompt safe? Investigating prompt injection attacks against open-source LLMs. arXiv preprint arXiv:2411.00348. 2025.
5. Suo X. Signed-Prompt: A new approach to prevent prompt injection attacks against LLM-integrated applications. AIP Conf Proc. 2024; 1:55-63.
6. OWASP Foundation. LLM prompt injection prevention cheat sheet. OWASP. 2025 [cited 2025 Oct 3]. Available from: https://owasp.org/LLM-Security/Cheat-Sheet
7. Touvron H, Lavril T, Izacard G, Martinet X, Lachaux M, Lacroix T, *et al*. LLaMA: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971. 2023.
8. HiddenLayer Research. Novel universal bypass for all major LLMs. HiddenLayer. 2025 [cited 2025 Oct 3]. Available from: https://hiddenlayer.com/research/universal-bypass-llms
9. Securiti AI Research. Context-aware LLM firewalls for AI security. Securiti. 2025 [cited 2025 Oct 3]. Available from: https://securiti.ai/research/llm-firewalls
10. DataSunrise Knowledge Center. LLM security considerations: Modern protection approach. DataSunrise. 2025 [cited 2025 Oct 3]. Available from: https://datasunrise.com/knowledge-center/llm-security
11. Gopalakrishnan V. An AI-driven approach to risk-scoring systems in cybersecurity. DarkReading. 2024 [cited 2025 Oct 3]. Available from: https://www.darkreading.com/ai-risk-scoring

12. Rehan H. The architecture of adaptive AI security: How cloud systems can learn to defend themselves. Analytics Insight. 2025 [cited 2025 Oct 3]. Available from: https://www.analyticsinsight.net/ai-security-architecture
13. Ren S, Zhou Y, Li J, Xu L. ARCS: Adaptive reinforcement learning framework for automated cybersecurity incident response strategy optimization. Appl Sci. 2025;15(4):1-15.
14. Behnia R, Rahman M, Wang S. Privately fine-tuning large language models with differential privacy. In: Proc. IEEE ICDM Workshop. 2023. p. 112-120.
15. Feretzakis G, Kouris A, Gritzalis D. Privacy-preserving techniques in generative AI and large language models: A narrative review. Information. 2024;15(2):1-20.
16. Huang K. Leveraging zero-knowledge proofs in machine learning and LLMs: Enhancing privacy and security. Cloud Security Alliance. 2024 [cited 2025 Oct 3]. Available from: https://cloudsecurityalliance.org/zero-knowledge-ml
17. Van Otten N. Anomaly detection in LLMs: How to monitor responses & mitigate risks. SpotIntelligence. 2024 [cited 2025 Oct 3]. Available from: https://spotintelligence.com/anomaly-detection-llms
18. Daiya H, Patel M, Shah S. Real-time anomaly detection using large language models. DZone Data Engineering. 2024 [cited 2025 Oct 3]. Available from: https://dzone.com/data-engineering/anomaly-detection-llmsg