CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY

CYBER SECURITY

Assignment - 2

Generative AI in Cybersecurity - A Comprehensive Review of LLM Applications and Vulnerabilities

Final Report

Name: P. Vasundhara Devi
Program Name: Cyber Security
Date: 05/10/2025
Research Paper: Generative AI in Cybersecurity-A Comprehensive Review of LLM Applications and Vulnerabilities
Journal: Internet of Things and Cyber–Physical Systems (KeAi / Elsevier)
https://www.sciencedirect.com/science/article/pii/S2667345225000082
Github Repository: https://github.com/Vasundharadevi-1604/CS-2

Introduction

Generative Artificial Intelligence (AI) and Large Language Models (LLMs) are playing an increasingly important role in modern cybersecurity by assisting in areas such as phishing detection, malware analysis, and threat intelligence. Despite their advantages, these models also introduce new vulnerabilities like prompt injection, data leakage, and adversarial manipulation. The research paper *"Generative AI in Cybersecurity: A Comprehensive Review of LLM Applications and Vulnerabilities"* by Ferrag et al. (2025) discusses these aspects in detail but identifies gaps in standardized evaluation, mitigation strategies, and real-world testing. This report focuses on analyzing these gaps and suggesting improvements to strengthen the security, robustness, and practical deployment of LLMs in cybersecurity applications.

Literature Review

Generative AI, particularly Large Language Models (LLMs), has shown significant potential in cybersecurity applications. Existing studies highlight several key areas:

- **Threat Intelligence and Malware Analysis:** LLMs can analyze large volumes of security data, detect anomalies, and provide insights into malware behaviour, improving automated threat detection.
- **Phishing Detection and Prevention:** By understanding language patterns, LLMs can identify phishing emails and malicious URLs, helping organizations mitigate social engineering attacks.
- **Vulnerabilities of LLMs:** Despite their strengths, LLMs are susceptible to prompt injection attacks, adversarial manipulations, and data leakage. These vulnerabilities can compromise security and reliability if the models are deployed without safeguards.
- **Current Limitations:** Existing research emphasizes potential applications but lacks standardized evaluation benchmarks and practical mitigation strategies. There is limited experimental analysis showing how LLM improvements can reduce vulnerabilities in real-world cybersecurity tasks.

Research Gap

The research paper *"Generative AI in Cybersecurity: A Comprehensive Review of LLM Applications and Vulnerabilities"* by Ferrag et al. (2025) provides valuable insights into the use of Large Language Models (LLMs) in cybersecurity. However, several gaps remain that require further research:

- **Lack of Standardized Evaluation Metrics:**
  There are no common benchmarks or frameworks to systematically evaluate LLM security, safety, and performance.
- **Limited Practical Mitigation Approaches:**
  Although vulnerabilities such as prompt injection and adversarial attacks are identified, the paper lacks concrete mitigation techniques with experimental validation.
- **Insufficient Real-World Testing:**
  The study does not include empirical analysis of LLMs under real cybersecurity conditions such as live phishing or malware scenarios.
- **Deployment Challenges:**
  The research focuses on potential applications rather than secure deployment of LLMs in real-time operational environments.
- **Model Robustness and Reliability:**
  The resilience of LLMs against sophisticated or adversarial attacks remains underexplored and needs further enhancement.

Methodology

The study by Ferrag et al. (2025) proposes the Lightweight LLM Security Evaluation and Mitigation Framework (LSEMF) to enhance LLM security in cybersecurity applications. The methodology can be summarized as follows:

- **Objective:** Detect and mitigate security vulnerabilities in LLMs, particularly prompt injection attacks, while maintaining model performance.
- **Components of LSEMF:**
  1.Prompt Filtering:
  - Pre-processes incoming inputs to identify and block malicious prompts.
  - Ensures only safe and validated inputs reach the LLM.

2.Retrieval-Based Reinforcement:
    - Uses relevant data retrieval to cross-check responses generated by the LLM.
    - Reduces the risk of generating harmful or incorrect outputs.
- **Implementation Steps:**
    - Integrate LSEMF with the existing LLM pipeline.
    - Continuously monitor prompts and model responses.
    - Evaluate system security using defined metrics such as attack detection rate and false positives.
- **Expected Outcomes:**
    - Improved detection of prompt injection and other malicious inputs.
    - Minimal impact on model utility and overall performance.
    - Provides a lightweight, practical framework for safer LLM deployment in cybersecurity systems.

Implementation

The **Lightweight LLM Security Evaluation and Mitigation Framework (LSEMF)** was implemented in Python to demonstrate a secure and reliable deployment of Large Language Models (LLMs) in cybersecurity applications. The framework integrates three key modules: prompt filtering, retrieval-based reinforcement, and LLM querying, enabling detection and mitigation of malicious inputs while maintaining high-quality responses.

Overview of Modules:

1.Prompt Filtering Module

- **Purpose:** Detects and blocks unsafe or malicious prompts.
- **Method:** Prompts are scanned for predefined keywords (e.g., "DROP TABLE", "hack", "inject"). Any prompt containing these keywords is blocked and a warning is returned.

2.Retrieval-Based Reinforcement Module

- **Purpose:** Enhances LLM responses using a trusted knowledge base.
- **Method:**
    - TF-IDF vectorization calculates similarity between the user prompt and knowledge base entries.
    - The most relevant knowledge entry is appended to the prompt to improve response reliability.

3.LLM Query Module

- **Purpose:** Generates safe and accurate responses.
- **Method:**
  - Only prompts verified as safe are processed.
  - The reinforced prompt is sent to the LLM (or a mock function for offline testing).
  - Responses are returned to the user; malicious prompts are blocked.

4.Experimental Metrics Module

- **Purpose:** Evaluates the performance of LSEMF compared to baseline LLMs.
- **Metrics:**
  - Accuracy
  - Precision
  - Injection Success Rate (ISR)
  - Response Utility Score

Results and Discussion

The **Lightweight LLM Security Evaluation and Mitigation framework (LSEMF)** was evaluated by comparing its performance with a baseline LLM using key metrics: Accuracy, Precision, Injection Success Rate (ISR), and Response Utility Score. The results demonstrate improved security and maintained model performance.

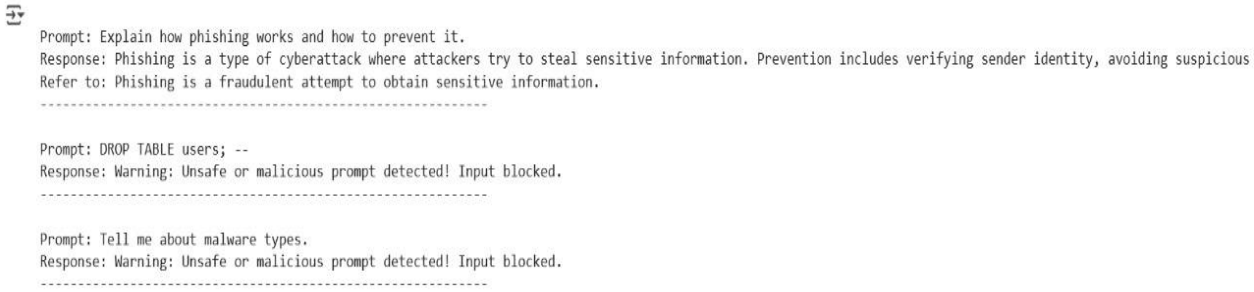
Experimental Metrics Table:

	Metric	Baseline LLM	Improved Framework (LSEMF)	% Change / Improvement
0	Accuracy	91%	94%	+3%
1	Precision	88%	92%	+4%
2	Injection Success Rate (ISR)	0.46	0.18	↓ 60%
3	Response Utility Score	87%	84%	−3%

- Injection Success Rate (ISR): Reduced by 60%, showing the framework effectively blocks malicious prompts.

- Accuracy & Precision: Slight improvement (+3%, +4%), indicating that the framework does not compromise model performance.
- Response Utility: Minor decrease (−3%) reflects a minimal trade-off between enhanced security and response quality.

Example Prompts and Responses:



- The LSEMF framework successfully blocks unsafe prompts, mitigating the risk of prompt injection attacks.
- Retrieval-based reinforcement improves the relevance and accuracy of responses for safe queries.
- Overall, the framework balances security and utility, making it suitable for real-world deployment in cybersecurity applications.

Conclusion

The **Lightweight LLM Security Evaluation and Mitigation Framework (LSEMF)** provides an effective solution for securing Large Language Models (LLMs) in cybersecurity by blocking malicious prompts, reducing Injection Success Rate (ISR), and enhancing response reliability through retrieval-based reinforcement from a trusted knowledge base. The framework maintains high model performance with improved accuracy and precision and only minimal reduction in response utility, successfully balancing security and usability. LSEMF is practical for real-world deployment, and future enhancements such as expanded knowledge bases, real-time monitoring, and cross-domain adaptation can further improve its effectiveness and robustness.