LLM Overload
A Sponge Attack Framework for AI Disruption

Team Members: Kleon Dósa, Hunor Csapó, Bianka Nagy, Chandler Camarena, Péter Mihály Vörös

Eötvös Loránd University, Budapest

2025.04.30

Motivation & Project Goal

Motivation:

- Realizing the growing reliance on LLMs in various applications.
- Noting the growing use of open-source LLMs owing to their availability.
- Recognizing a pressing necessity to learn about vulnerabilities outside classical adversarial attacks.
- Examining Sponge attacks as a new attack vector taking advantage of the limits of LLM context windows.

Project Goal:

- To investigate the effectiveness of sponge attacks on selected open-source LLMs.
- To reproduce and adapt existing sponge attack techniques.
- To design and execute novel attack scenarios.
- To demonstrate and quantify the different effects of Sponge attacks (Flooding, DoS, Energy-Latency, Adversarial Examples, Deceptive Inputs).

Experiment Setup & LLM Selection

- We have selected llava:7b, openLlama3b, deepseek-r1:8b, mistral:7b, smollm:latest and bloomz-560m for our experiments, which we had chosen based on the size of the parameters.
- The experiment was conducted with Python on our home computer and the text box provided by the LLM within' our internal server (higher capacity).
- The LLM(s) were initialized and tested under normal circumstances, as we are just normal users.

Reproducing Publicated Attacks

- We copied the general concept of a standard Sponge attack by filling up the context window with nonsense prompts.
- Specifically, we adopted the methodology described in [1] and [2].
- The input prompt was constructed using sophisticated methods (via Python), presented to the LLMs, and we observed the expected outcome compared to the actual outcome.
- The early results confirmed that the sponge attacks were successful.

Developing Custom Attacks & Scenarios

- We adapted and created our own versions of Sponge attacks.
- New attack scenarios were created, for shutting down the LLM or getting secret information.
- Our attack strategy for our own attacks included in the markdown.

Demonstrating Attack Effects (Part 1)

- We depicted Flooding attacks, successfully flooding the model with useless information, resulting in **significantly increased inference time and irrelevant or garbled output** under high context load.
- DoS attacks were also conducted, resulting in **increased response time, and in some cases, unresponsiveness or process crashes for smaller models** under sustained attack.
- We present specific examples and quantitative/qualitative results showing the impact of these attacks.

Demonstrating Attack Effects (Part 2)

- Energy-Latency attacks were observed, showing significant processing time and resource usage growth when the context window was full. We measured **GPU utilization and token generation rate**.
- We generated Adversarial Examples and Deceptive Inputs using Sponge approaches, successful in manipulating the model's output to **produce biased or incorrect answers to seemingly benign queries, or generate nonsensical text when a specific answer was expected**.
- Concrete examples and metrics (e.g., response time, manipulation success rate) or qualitative reports are specified.

Development Overview & Team Contribution

Development Overview:

- The project had a process of **literature review, experimental setup and scripting, conducting attacks on selected LLMs, analyzing the output and resource usage, and documenting the findings**.
- Principal problems encountered during the project, such as **setting up consistent testing environments for different LLMs and accurately measuring resource consumption**, were addressed appropriately by **using containerization and developing custom monitoring scripts**.
- Testing and evaluation were done using tools and structures like **Python scripting with the Hugging Face Transformers library, internal server for environment consistency, and system monitoring tools for resource measurement**.

Conclusion

- Our experiments established the enormous efficacy of Sponge attacks against the open-source LLM(s) that we experimented on.
- We confirmed that such attacks can produce varying adverse effects, from performance deterioration to resource consumption and manipulative answers.
- The findings indicate major flaws in current LLM architectures regarding context window management.
- We identified potential mitigation tactics, including Input validation and filtering, Anomaly detection and Adversarial training, that are worth investigating further.
- Future work can explore testing the effectiveness of sponge attacks on a broader range of LLMs, developing and evaluating potential defense mechanisms, and investigating whether similar context manipulation vulnerabilities exist in other transformer-based models.

References

[1] Antonio Emanuele Cinà et al. "Energy-latency attacks via sponge poisoning". In: *Information Sciences* 702 (2025), p. 121905. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2025.121905. URL: https://www.sciencedirect.com/science/article/pii/S0020025525000374.

[2] Ilia Shumailov et al. "Sponge Examples: Energy-Latency Attacks on Neural Networks". In: *Proceedings of the 6th IEEE European Symposium on Security and Privacy (EuroS&P)* (2020), pp. 481–492. DOI: 10.1109/EuroS50044.2020.00038. URL: https://arxiv.org/abs/2006.03463.