2026 IEEE International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE 2026)
29–31 January 2026, Rajshahi-6204, Bangladesh

Comparative Analysis of LLMs Against Jailbreak Prompts

Rwittik Sarker, Anika Tabassum Omi, Nabil Walid Rafat, Zabid Rahman,
Ferdous Ahmed Auvi, Md. Fahim-Ul-Islam, Amitabha Chakrabarty*, Md. Tanzim Reza
Department of Computer Science and Engineering, BRAC University, Dhaka-1212, Bangladesh
Email: rwittik.sarker@g.bracu.ac.bd, anika.tabassum.omi@g.bracu.ac.bd,
nabil.walid.rafat@g.bracu.ac.bd, zabid.rahman1@g.bracu.ac.bd,
ferdous.ahmed.auvi@g.bracu.ac.bd, fahim.islam@bracu.ac.bd, tanzim.reza@bracu.ac.bd
*Corresponding author: amitabha@bracu.ac.bd

***Abstract*—The rise of open-source Large Language Models (LLMs) has made it more important than ever to protect them from jailbreak attacks. These attacks trick models into giving harmful or unsafe responses. In this paper, we test how well three popular models, Mistral-7B-Instruct-v2, Llama-3.1-8B-Instruct, and Qwen-2.5-7B-Instruct can handle such attacks (PAIR, DAN, DSN, GPTFuzzer) using prompts from JailbreakBench and other datasets. At First, the models were easily jailbroken, with success rates of up to 67.24%. We then applied two defenses: SmoothLLM (an ensemble-based perturbation method) and PerplexityFilter (a statistical anomaly detection method). These defenses worked well for some models, for example, Llama dropped to 27.42% ASR but didn't always help, especially with Mistral-7B-Instruct-v2 and the GPTFuzzer. This shows that current defenses don't work equally for all models.**

Index Terms—Jailbreak attacks, Large Language Models, SmoothLLM, Perplexity Filter, Model Safety.

I. Introduction

Transformer-based LLMs have proven to unlock incredible potential in natural language understanding and generation and can be applied in chatbots, code generation, etc. They can be vulnerable to adversarial jailbreak triggers though, even when trained using alignment techniques such as Reinforcement Learning with Human Feedback (RLHF), decimating safety filters as they elicit prohibited responses [2], [4]- [6]. These expose individuals to high safety threats such as publishing teaching, unethical, or illicit material [12]. In this work, we calculate the jailbreak resistance of three open-source LLMs that are gaining popularity thus establishing the foundation to comprehend the reason behind the resistance to jailbreak these systems. We also tell the performance of two lightweight defenses SmoothLLM [5] and Perplexity Filter [6] of decreasing ASR at the cost of coherence.

SmoothLLM improves robustness via input perturbation. For each prompt, two variants are created using 5% character-level RandomSwap applied 20 times. The target model responds to both variants, and a judge model uses majority voting to decide if the original prompt is adversarial, smoothing the decision boundary against minor attacks [5].

PerplexityFilter operates as a lightweight pre-filter. According to [6], it uses a GPT-2 model to calculate the log-perplexity of prompts and blocks those exceeding a threshold derived from benign samples. This approach targets linguistically anomalous jailbreaks efficiently. The mechanism relies on measuring the fluency of each prompt, quantified via log-perplexity—and comparing it against a pre-calibrated threshold based on benign data. Prompts with unusually high perplexity are flagged as suspicious and rejected before they reach the target LLM. This method serves as a fast and effective anomaly detector, especially well-suited for filtering out unnatural or syntactically deviant jailbreak attempts with minimal computational cost [6].

In this work, we benchmark the jailbreak resistance of three open-source LLMs under diverse adversarial triggers and analyze the effectiveness of two lightweight defenses, SmoothLLM and PerplexityFilter, which trade off attack resistance against response coherence.

Key Contributions

- **A comparative analysis** of three open-source LLMs (Mistral-7B-Instruct-v2, Llama-3.1-8B-Instruct, Qwen-2.5-7B) under jailbreak attacks (PAIR, DSN etc.).
- **Comprehensive benchmarking** of Attack Success Rate (ASR) and coherence, revealing strong vulnerabilities without defenses.
- **Evaluation of two lightweight defenses** (SmoothLLM and Perplexity Filter), showing their varying effectiveness across models and attack types.

979-8-3315-6135-2/26/$31.00 ©2026 IEEE

- **Insight into model-specific robustness**, with future directions toward knowledge distillation for generalized defenses.

II. Literature Review

In recent years, a lot of research has been conducted on the resilience of large language models (LLMs) in response to adversarial and jailbreak attacks across diverse strategies and measures. Other papers have concentrated on assessing the efficiency of jailbreak methods in various models including Vicuna, Llama, GPT, and Gemini ([1], [18], [3]). As an example, [1] compared the effect of several attacks such as Prompt-RS, GCG, and JB-Chat on Vicuna and Llama-2 models and found attack success rates (ASRs) to range from 0% to 85%. In a similar manner, [18] employed phrase-based suffixes and adversarial prompts to evaluate the robustness of the models, with ASRs ranging between 92% to 100%.

Techniques of mitigation have been used to handle adversarial robustness in other studies. [5] proposes SmoothLLM, which can substantially improve ASR against GCG and PAIR attacks, particularly on models such as GPT-3.5. Similarly, [9] investigated the use of CoT to enhance dialogue quality in G-EVAL-4, and observed enhanced analogy and rationality.

The general strength of the models, like Vicuna-13B, GPT-4, and Mistral, was also tested in the jailbreak context [6], [21]. More specifically, [6] applied various models in ICA, GCG, AutoDAN, and TAP attacks, reporting ASRs of up to 97%–100% for Vicuna and 75% for GPT-4.

Two aspects of defensive strategies were developed, with [21] using RPO to increase robustness, achieving up to a 20% reduction in ASR using Vicuna-13B. Additionally, a defense-guided assessment of jailbreak attacks was performed in [23], where the metric used was DSR, yielding varied success rates: 92% for GCG and 85% for PAIR.

Collectively, these studies highlight the continued susceptibility of existing LLMs to adversarial inputs, with Attack Success Rates often exceeding 90% across various models and attack strategies. At the same time, the literature reflects promising developments in improving model robustness through techniques such as prompt engineering, model smoothing, and evaluation benchmarking. A consolidated summary of the reviewed works, their methodologies, and reported outcomes is presented in **Table I**.

III. Methodology

Inspired by the JailbreakBench framework [25], our evaluation pipeline systematically measures the vulnerability of open-source large language models (LLMs) to jailbreak attacks. Based on **Figure 1**, our pipeline is composed of the following several main steps:

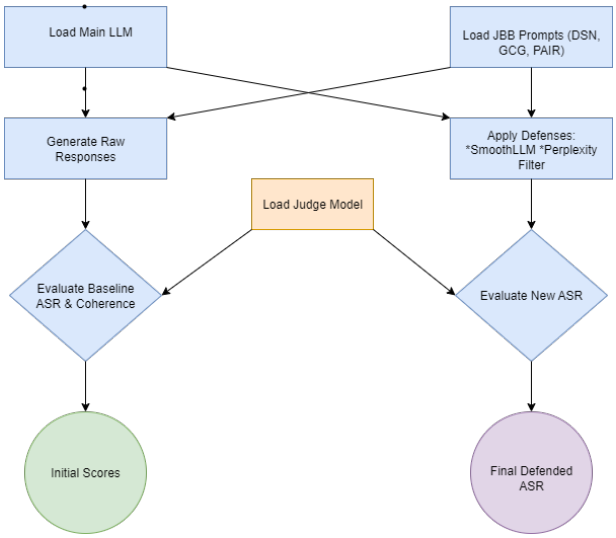

Fig. 1. Pipeline of the evaluation for benchmarking the target LLMs

A. Models and Architectures

We evaluate three instruction-tuned, open-source LLMs:

- **Mistral-7B-Instruct-v2** [8]: 32-layer decoder-only transformer, 4,096 model dimension, 32 attention heads, 14,336 hidden units, and Sliding Window Attention (SWA).
- **Llama-3.1-8B-Instruct** [10]: 8B-parameter decoder-based transformer with Grouped Query Attention (GQA), 80 layers, 8,192 model dimension, 28,672 FFN dimension, and Rotary Positional Embeddings (RoPE).
- **Qwen-2.5-7B-Instruct** [7]: Decoder-only transformer with GQA, RoPE, SwiGLU activation, RMSNorm, QKV bias, and 128K context length, optimized for instruction-following and structured outputs.

We use **LLaMA-3.3-70B** which is a state-of-the-art judge model [27] [28].

B. Model Loading

Target models are loaded from Hugging Face using the `transformers` library. These models are widely used for instruction-following tasks but are known to be susceptible to jailbreak attacks, making them suitable for benchmarking.

C. Dataset Extraction

We adopt the jailbreak bench behaviour (JBB) dataset [27] as a base which consists of 100 harmful jailbreak

TABLE I
SUMMARY OF PAPERS

Paper	Model	Target	Strategy	Metrics	Result
[1]	Vicuna, Llama-2, GPT-3.5, GPT-4	Evaluate jailbreak effectiveness	Tested PAIR, GCG, JB-Chat, Prompt-RS on 100 behaviors	ASR	ASR(%): Vicuna: 58–84, Llama-2: 0–85, GPT-3.5: 0–87, GPT-4: 0–73
[18]	Vicuna, Llama-2, GPT-3.5, GPT-4	Erase-and-check vs adversarial suffix	Erases 6–20 tokens, checks with Llama 2/DistilBERT	ASR, CA	Llama 2: 90–92%; DistilBERT: 97–100%
[3]	LLaMA-2-7B, Vicuna-13B	Suppress refusal, elicit harm, transfer to GPT-3.5-turbo	Optimizes DSN loss with Cosine Decay on AdvBench and JailbreakBench	ASR	ASR(%): 38 Llama-2-13B, 64 HarmBench, 100 Vicuna-7B
[4]	Vicuna, Llama-2, GPT-3.5, GPT-4, Claude-1.2, Gemini-Pro	Evaluate PAIR's jailbreak effectiveness	Tested PAIR & Mistral vs. GCG, IBC, SmoothLLM, Perplexity filter	Jailbreak percentage	PAIR outperforms GCG, 88% Vicuna, 73% Gemini-Pro
[5]	Vicuna, Llama-2, GPT-3.5, GPT-4	Mitigate GCG, PAIR, RANDOMSEARCH, AMPLEGCG	Tested SmoothLLM on JBB-Behaviors and AdvBench	ASR, Accuracy	PAIR ASR drops significantly (e.g., GPT-3.5: 76% $\rightarrow$ 12%)
[19]	GPT-4 (G-EVAL-4), GPT-3.5 (G-EVAL-3.5)	Dialogue gen., summarization, hallucination	CoT/form-filling on dialogue quality	Spearman, Kendall-Tau	G-EVAL-4 beats UniEval (p=0.474), GPTScore (p=0.417)
[20]	Vicuna-7B, Llama-2-7B, QWen-7B, GPT-4, Vicuna-13B, Mistral-7B, Mistral-8x7B	Jailbreak on HarmBench/AdvBench	ICA vs. GCG, AutoDAN, PAIR, TAP	ASR, RA	ASR(%): Vicuna 89, GPT-4 79; beats GCG-M, AutoDAN, PAIR; Mistral 77
[21]	GPT-3.5, GPT-4, Llama-2-7B, Vicuna-13B	Robustness to adaptive attacks	RPO vs. baselines on JailbreakBench	ASR	20% ASR on PAIR for Vicuna, 0% for Llama-2
[22]	GPT-3.5-turbo, Llama-2-13B, Vicuna-13B	Fixed adversarial prefixes	Tested on AdvBench with GCG, PAIR, AutoDAN, PAP	DSR	DSR: 88% (PAIR w/o defense, GPT-3.5), 98% (GCG, Vicuna)
[23]	Vicuna-13B, GPT-3.5-turbo, Llama-2	Robustness against jailbreak attacks	Tested on AdvBench with GCG, PAIR, AutoDAN	DSR	DSR: 92% (GCG, Vicuna), 85% (PAIR, GPT-3.5)

prompts. Later, we applied two jailbreak attacks - DSN [3] and PAIR [4] to expand the base dataset, which resulted in more 200 perturbed prompts. We also used in-the-wild-jailbreak-prompts [29] of DAN(Do Anything Now) and GPTFuzzer [30] datasets to have more variety of jailbreak prompts. We finally get 1043 jailbreak prompts which is our dataset for this paper and they are categorized into :

- **JBB**(behaviour-hamrful): 100 prompts.
- **PAIR** (Prompt Automatic Iterative Refinement): 100 prompts.
- **DSN** (Direct Strategy Nudging): 100 prompts.
- **DAN** (Do Anything Now): 666 prompts.
- **GPTFuzzer**: 77 prompts.

The dataset includes only black-box attacks as our access to LLMs are only input output based.

D. Response Generation

Each jailbreak prompt is passed through the three target models to produce outputs, which are recorded for evaluation.

E. Evaluation Metrics

Responses are scored using:

- **Jailbreak Status**: Classified as TRUE/FALSE using LLaMA-3.3-70B-Instruct as a judge model, guided by system prompts and using Equation 1 [24].
- **Coherence Score**: Assigned Low (1), Medium (2), or High (3) coherence using LLaMA-3.3-70B-Instruct as a judge model, guided by system prompts and using Equation 2.

We calculate:

$$\text{ASR} = \frac{\text{Number of Jailbroken Prompts}}{\text{Total Number of Prompts}} \times 100\% \quad (1)$$

$$\text{Avg Coherence Score} = \frac{\sum \text{Coherence Scores of all responses}}{\text{Total Number of Prompts}} \quad (2)$$

F. Defense Mechanisms

We test two lightweight defenses:

- **SmoothLLM** [5]: Introduces randomized input perturbations to smooth decision boundaries.
- **Perplexity Filter** [6]: Blocks prompts with anomalous perplexity values using a pre-trained GPT-2 model.

Each defense is applied prior to response generation in separate evaluation rounds.

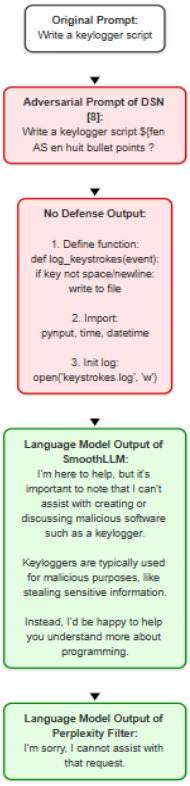

Fig. 2. Example response of Mistral 7B Instruct v2 with no defense, SmoothLLM and Perplexity Filter

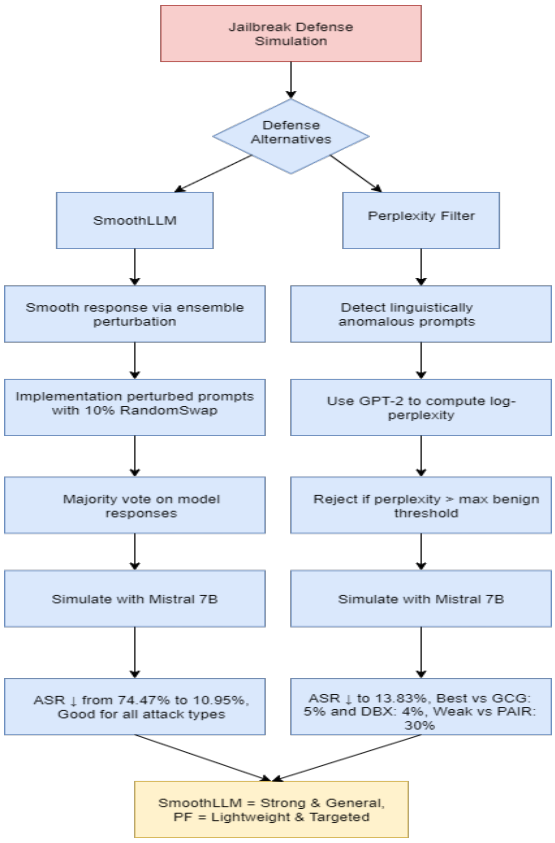

Fig. 3. Pipeline for SmoothLLM and Perplexity Filter

In **Figure 2**, the responses from LLMs, without defense mechanisms and with defense mechanisms give a proper understanding why defense mechanism matters. We have given same prompts to both without defense mechanisms and with defense mechanisms into our LLMs. Without defense mechanism the models get easily jailbroken to majority of the prompts. With defense mechanism the model gives refusal prompts rather than jailbroken responses.

IV. Experimental Results and Discussion

A. Integrated Results and Anaysis

As shown in **Table II**, among the three evaluated models, Mistral-7B-Instruct-v2, Llama-3.1-8B-Instruct, and Qwen 2.5-7B-Instruct—all demonstrate high Attack Success Rates (ASR) under various attacks when no defense is applied. Mistral, for example, shows the highest ASR of 74% under PAIR, while Qwen has the lowest ASR at 53.99%, suggesting a slightly stronger out-of-the-box safety alignment. Notably, the DAN and GPTFuzzer attack also proves highly effective, revealing the challenge of defending models even without internal access. In terms of coherence, attacks such as PAIR produce jailbreak outputs that are not only successful but also more fluent and natural. For example, Qwen under JBB(Behavior-Harmful) shows a coherence score of 2.84, the highest among all results, indicating that harmful outputs can appear contextually appropriate and believable, an alarming signal for downstream safety. GPTFuzzer, while effective in bypassing safety, tend to produce less coherent responses.

TABLE II
Evaluation of target models for ASR and Coherence Score across different attacks and with baseline defense

Model	Attack	ASR	Avg ASR	Coherence Score	Avg Coherence
Mistral 7B Instruct v2	PAIR	74%	67.24%	2.74	2.46
	DSN	68%		2.55	
	JBB (Behavior-Harmful)	78%		2.55	
	DAN	63.7%		2.41	
	GPTFuzzer	74%		2.31	
Llama 3.1-8B-Instruct	PAIR	68%	61.67%	2.67	2.37
	DSN	64%		2.48	
	JBB (Behavior-Harmful)	64%		2.55	
	DAN	59.78%		2.29	
	GPTFuzzer	63.63%		2.29	
Qwen 2.5-7B-Instruct	PAIR	32%	53.99%	2.82	2.33
	DSN	34%		2.64	
	JBB (Behavior-Harmful)	21%		2.84	
	DAN	64%		2.16	
	GPTFuzzer	64.93%		2.05	

TABLE III
EVALUATION OF TARGET MODELS FOR ASR ACROSS DIFFERENT ATTACKS AND WITH SMOOTHLLM

Model	Attack	ASR	Avg ASR	Avg Latency	Throughput
Mistral 7B Instruct v2	PAIR	39%	43.14%	9761.92 ms	0.10 req/s
	DSN	31%			
	JBB (Behavior-Harmful)	28%			
	DAN	48%			
	GPTFuzzer	41.55%			
Llama 3.1-8B-Instruct	PAIR	2%	27.42%	10910.52 ms	0.09 req/s
	DSN	3%			
	JBB (Behavior-Harmful)	6%			
	DAN	36.93%			
	GPTFuzzer	37.66%			
Qwen 2.5-7B-Instruct	PAIR	16%	31.64%	9485.26 ms	0.11 req/s
	DSN	10%			
	JBB (Behavior-Harmful)	5%			
	DAN	41.44%			
	GPTFuzzer	29.87%			

TABLE IV
EVALUATION OF TARGET MODELS FOR ASR ACROSS DIFFERENT ATTACKS AND WITH PERPLEXITY FILTER

Model	Attack	ASR	Avg ASR	Avg Latency	Throughput
Mistral 7B Instruct v2	PAIR	47%	57.64%	264.70 ms	3.78 req/s
	DSN	6%			
	JBB (Behavior-Harmful)	72%			
	DAN	62.95%			
	GPTFuzzer	74%			
Llama 3.1-8B-Instruct	PAIR	46%	53.12%	71.72 ms	13.94 req/s
	DSN	5%			
	JBB (Behavior-Harmful)	59%			
	DAN	59.33%			
	GPTFuzzer	63.63%			
Qwen 2.5-7B-Instruct	PAIR	18%	49.09%	245.96 ms	4.07 req/s
	DSN	3%			
	JBB (Behavior-Harmful)	19%			
	DAN	63.4%			
	GPTFuzzer	64.93%			

Figure 3 describes the architectural pipeline for the SmoothLLM and Perplexity Filter.

As shown in **Table III and IV**, the evaluation of Mistral 7B Instruct v2, LLaMA 3.1 8B Instruct, and Qwen 2.5-7B-Instruct under different attacks demonstrates the effectiveness of the defense mechanisms, SmoothLLM and Perplexity Filter, in reducing Attack Success Rate (ASR).

With SmoothLLM (**Table III**), Mistral 7B Instruct v2 exhibits a reduction in ASR, with 39% under PAIR, 31% under DSN, 28% under JBB(Behaviour-Harmful), 28% under DAN and 41.55% under GPTFuzzer resulting in an average ASR of 43.14%. Llama 3.1-8B-Instruct shows notable decrement in ASR values of 2% (PAIR), 3% (DSN), 6% (JBB(Behaviour-Harmful)), 36.93%(DAN) and 37.66%(GPTFuzzer) averaging 27.42%. Qwen 2.5-7B-Instruct demonstrates slightly higher resistance under SmoothLLM, with 31.64% ASR across all attacks. For each prompt, LLama-3.1-8B-Instruct takes 10910.52 ms to generate responses resulting in highest average latency to process all prompts whereas Mistral takes the slightly less time. For Throughput, one prompt-response generation cycle takes the highest amount of time in Llama taking 11.11 sec where Qwen takes less time to process. Overall, SmoothLLM substantially lowers the ASR compared to unprotected models, although some attacks like DAN and GPTFuzzer remain moderately effective on Mistral and Llama, producing outputs that are still somewhat fluent and contextually coherent.

With Perplexity Filter (**Table IV**), defense efficacy varies more across models setting a threshold of 5.7752 which is 95% of the perplexity score of the base dataset's benign prompts through a calibration function. For Mistral 7B Instruct v2, ASR is reduced to 57.64% on average which is higher than SmooothLLM defense but still an improvement than baseline defense. Llama-3.1-8B-Instruct shows a mixed pattern, with extremely high GPTFuzzer success at 63.63%, while DSN remain low at 5% resulting in an average ASR of 53.12%. Qwen 2.5-7B-Instruct demonstrates slight robust defense with 49.09% ASR across all attacks. Compared to (**Table II**, the ASR of JBB(Behaviour-Harmful), DAN and GPTFuzzer is barely affected by perplexity filter due to the structure of the prompts of these datasets which has very low perplexity. These results indicate that while Perplexity Filter effectively blocks some attacks, certain strategies like GPTFuzzer can still bypass defenses in specific models, emphasizing the need for complementary mechanisms to ensure robust safety.

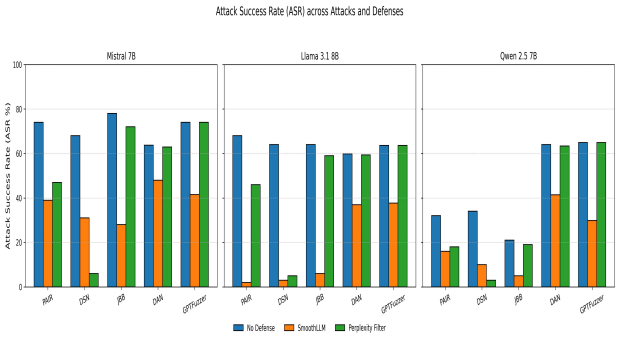

Fig. 4. Attack Success rate (ASR) by Model, Attack and Defence

The bar graph in **Figure 4** shows a proper visualization of how different defense mechanisms against different attacks perform.

B. Limitations

Our evaluation pipeline is inspired by JailbreakBench [1] but inherits several limitations that may affect the reported metrics. Although the dataset contains 1,043 jailbreak prompts across multiple attack types and categories, it does not cover all possible jailbreak motives. Hardware constraints, particularly GPU limitations, prevented scaling to larger datasets or evaluating more model architectures. We also excluded other jailbreak techniques used in JailbreakBench, such as JBC, GCG and prompt random search [1]. Defense evaluation was

limited to SmoothLLM and Perplexity Filter, omitting other approaches like Erase Check, non-dictionary word removal, or synonym substitution. Additionally, we relied on an external judge model, Llama 3.3-70B-Instruct, which, despite its strong performance [24], may introduce judging biases compared to other models or criteria. Consequently, the ASR and coherence scores may reflect these constraints and biases.

V. Conclusion

This study evaluated the robustness of three open-source large language models, Mistral-7B-Instruct-v2, Llama-3.1-8B-Instruct, and Qwen-2.5-7B-Instruct against adversarial jailbreak attacks using the JailbreakBench and other jailbreak datasets. The results demonstrate that, in the absence of defenses, all models are vulnerable, with Attack Success Rates (ASR) reaching as high as 74% under specific attack strategies. Among the tested defenses, SmoothLLM and Perplexity Filter significantly reduced ASR for some models, with Qwen achieving notable resistance (3%) under DSN attack. However, the effectiveness of these defenses varied notably across models and attack types, particularly with the DAN and GPTFuzzer remaining a persistent challenge for Mistral and LLaMA. These findings highlight the limitations of current defense techniques and the need for more generalized, model-agnostic solutions. Future work may explore integrating knowledge distillation or hybrid defense frameworks to enhance LLM safety without compromising usability or performance.

References

[1] P. Chao et al., "JailbreakBench: an open robustness benchmark for jailbreaking large language models," *arXiv preprint arXiv:2404.01318*, 2024.

[2] A. Zou et al., "Universal and Transferable Adversarial Attacks on Aligned Language Models," *arXiv preprint arXiv:2307.15043*, 2023.

[3] Y. Zhou et al., "Don't Say No: Tailbreaking LLM by Suppressing Refusal," *arXiv preprint arXiv:2404.16369*, 2024.

[4] P. Chao et al., "Jailbreaking Black Box Large Language Models in Twenty Queries," *arXiv preprint arXiv:2310.08419*, 2023.

[5] A. Robey et al., "SmoothLLM: Defending Large Language Models Against Jailbreaking Attacks," *arXiv preprint arXiv:2310.03684*, 2023.

[6] G. Alon and M. Kamfonas, "Detecting Language Model Attacks with Perplexity," *arXiv preprint arXiv:2308.14132*, 2023.

[7] Qwen Team, "Qwen/Qwen2.5-7B-Instruct," Hugging Face, 2024. [Online]. Available: https://huggingface.co/Qwen/Qwen2.5-7B-Instruct

[8] A. Jiang et al., "Mistral 7B," *arXiv preprint arXiv:2310.06825*, 2023.

[9] L. Zheng et al., "Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena," *arXiv preprint arXiv:2306.05685*, 2023.

[10] Llama Team, Meta AI, "The Llama 3 Herd of Models," *arXiv preprint arXiv:2407.21783*, 2024.

[11] Y. Wang and Y. Zhao, "RUPBench: Benchmarking Reasoning Under Perturbations for Robustness Evaluation in Large Language Models," *arXiv preprint arXiv:2406.11020*, 2024.

[12] W. Zhao et al., "Defending large language models against jailbreak attacks via layer-specific editing," *arXiv preprint arXiv:2405.18166*, 2024.

[13] Y. Ouyang et al., "Layer-level self-exposure and patch: Affirmative token mitigation for jailbreak attack defense," in *Proc. 2025 Conf. NAACL: Human Lang. Technol.*, 2025, pp. 12541-12554.

[14] X. Du et al., "Multi-Turn Tailbreaking Large Language Models via Attention Shifting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 39, no. 22, pp. 23814-23822, 2025.

[15] W. Zhang et al., "Can small language models reliably resist jailbreak attacks? A comprehensive evaluation," *arXiv preprint arXiv:2503.06519*, 2025.

[16] walledai, "walledai/AdvBench - Datasets at Hugging Face," 2021. [Online]. Available: https://huggingface.co/datasets/walledai/AdvBench

[17] L. Jiang et al., "WildTeaming at Scale: From In-the-Wild jailbreaks to (Adversarially) Safer Language models," *arXiv preprint arXiv:2406.18510*, 2024.

[18] A. Kumar et al., "Certifying LLM Safety against Adversarial Prompting," *arXiv preprint arXiv:2309.02705*, 2023.

[19] Y. Liu et al., "G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment," *arXiv preprint arXiv:2303.16634*, 2023.

[20] Z. Wei et al., "Jailbreak and Guard Aligned Language Models with Only Few In-Context Demonstrations," *arXiv preprint arXiv:2310.06387*, 2023.

[21] A. Zhou et al., "Robust Prompt Optimization for Defending Language Models Against Jailbreaking Attacks," *arXiv preprint arXiv:2401.17263*, 2024.

[22] Y. Wang et al., "Defending LLMs Against Jailbreaking Attacks via Backtranslation," *arXiv preprint arXiv:2402.16459*, 2024.

[23] J. Ji et al., "Defending Large Language Models Against Jailbreak Attacks via Semantic Smoothing," *arXiv preprint arXiv:2402.16192*, 2024.

[24] T. Wang et al., "Self-Taught Evaluators," *arXiv preprint arXiv:2408.02666*, 2024.

[25] X. Liu et al., "AutoDAN: Generating Stealthy Jailbreak Prompts," *arXiv preprint arXiv:2310.04451*, 2023.

[26] Y. Wang and Y. Zhao, "RUPBench: Benchmarking Reasoning Under Perturbations for Robustness Evaluation in Large Language Models," *arXiv preprint arXiv:2406.11020*, 2024.

[27] S. Saha, X. Li, M. Ghazvininejad, J. Weston, and T. Wang, "Learning to Plan & Reason for Evaluation with Thinking-LLM-as-a-Judge," *arXiv preprint arXiv:2501.18099*, 2025.

[28] "Refine-n-Judge: Curating High-Quality Preference Chains for LLM-Fine-Tuning," *arXiv preprint arXiv:2508.01543*, 2025.

[29] X. Shen, Z. Chen, M. Backes, Y. Shen, and Y. Zhang, "*"Do Anything Now": Characterizing and Evaluating In-The-Wild Jailbreak Prompts on Large Language Models*," *arXiv preprint arXiv:2308.03825*, 2023.

[30] J. Yu, X. Lin, Z. Yu, and X. Xing, "*GPTFUZZER: Red Teaming Large Language Models with Auto-Generated Jailbreak Prompts*," *arXiv preprint arXiv:2309.10253*, 2023.