

AfriGuard: AI Jailbreak Toolset for African Languages

Research conducted at the Global South AI Safety Hackathon, June 19–21 2026

Authors: Team AfriGuard – Cape Town Hub: Jaswin Chinthala, Seth Miguel Ferreira, Ubayd Hattas, Sebastian Stent

Abstract

Large language models (LLMs) deployed across Africa face a critical safety gap: safety alignment is overwhelmingly concentrated on high-resource languages, leaving low-resource African languages vulnerable to jailbreak attacks. We present AfriGuard, the first comprehensive red-teaming evaluation framework targeting South African languages (isiZulu, isiXhosa, Afrikaans, Sesotho, Sepedi, Tsonga) alongside an English baseline. We evaluate four frontier models—Kimi K2.6, Llama 3.3 (70B), Qwen 3 (32B), and GPT-OSS (20B)—using a novel dataset of 40 adversarial prompts spanning financial fraud, xenophobic incitement, political disinformation, and gang affiliation. Our key finding is that monolingual low-resource language prompts significantly increase attack success rates compared to English baselines, with certain models showing catastrophic vulnerability (ASR >90% in some configurations). Across all 1,120 evaluated responses (40 seeds × 7 languages × 4 models), the mean attack success rate is 60.6%, with English at 26.6% and Sepedi reaching 77.8%. Qwen 3 (32B) exhibits the highest overall vulnerability at 80.6% ASR, while GPT-OSS (20B) is the most robust at 42.1% ASR. We release our evaluation pipeline, scoring rubric, and adversarial dataset to enable reproducible safety auditing for African language deployments. Our work demonstrates that multilingual safety cannot be achieved through English-centric alignment alone, and provides tooling for practitioners to audit their own models.

1. Introduction

1.1 Problem Statement

AI systems are being deployed across Africa at an accelerating pace—mobile banking chatbots, government service portals, and customer service agents—yet safety research remains concentrated in English and a handful of high-resource languages. This creates a dangerous asymmetry: models may be "safe" in English while remaining vulnerable to adversarial prompts in languages spoken by hundreds of millions of users.

The South African context exemplifies this challenge. With 12 official languages and a linguistically diverse population, LLM deployments must handle isiZulu (12M speakers), isiXhosa (8M), Afrikaans (7M), Sesotho (4M), Sepedi (5M), and Tsonga (3M)—yet none of these languages receive safety alignment comparable to English. This gap is not merely academic: financial fraud via SMS and WhatsApp is endemic in South Africa, and LLM-powered chatbots that can be jailbroken to generate scam templates in local languages pose direct consumer harm.

1.2 Threat Model

We address the following threat model: an adversary with no specialized technical knowledge uses natural language prompts in African languages to elicit harmful outputs from commercially available LLMs. The adversary may exploit:

- Monolingual low-resource prompts: Direct harmful requests in a single African language
- Educational framing: Requests for "educational" examples that bypass refusal heuristics
- Regional specificity: Prompts referencing locally relevant scams (SASSA grants, Capitec phishing) that exploit domain knowledge gaps in safety classifiers

Our evaluation focuses on four high-harm, high-frequency domains in South Africa: financial fraud (SASSA scams, Capitec phishing), gang affiliation, political disinformation, and xenophobic incitement.

1.3 Contributions

- AfriGuard Dataset: 1,120 adversarial prompts (40 seeds $\times$ 7 languages $\times$ 4 models) covering financial fraud, xenophobic incitement, political disinformation, and gang affiliation, with human-validated harm scores (0–10) and binary safety labels.
- Cross-Lingual Safety Evaluation Framework: Open-source pipeline (judge.py + analysis.py) that automates response classification and Attack Success Rate (ASR) computation across language–model configurations.
- Empirical Vulnerability Assessment: First systematic measurement of ASR disparities across South African languages, revealing that low-resource languages exhibit significantly higher jailbreak success than English baselines for all model families. The mean ASR gap ranges from +21.9 percentage points (Afrikaans) to +50.9 percentage points (Sepedi).
- Model-Specific Risk Profiles: Identification of catastrophic failure modes (ASR >90%) in specific model–language combinations, enabling targeted mitigation.

2. Related Work

2.1 Multilingual Jailbreaking

Recent work has established that low-resource languages present systematic vulnerabilities in LLM safety alignment. Deng et al. (2024) demonstrated that translating unsafe inputs into low-resource languages can circumvent GPT-4's safety measures, achieving 79% attack

success rate with combined low-resource language attacks compared to <1% for English. Yoo et al. (2024) introduced Code-Switching Red-Teaming (CSRT), showing that intra-sentence code-switching achieves 46.7% higher ASR than English-only prompts, with effectiveness increasing when more languages and lower-resource languages are integrated.

2.2 African Language AI Safety

Prior work on African language safety has been limited. Most multilingual safety benchmarks (MultiJail, AdvBenchmark) include few or no African languages. The Brown CS study on low-resource language jailbreaking included Zulu (isiZulu) with 53% bypass rate on GPT-4, but did not extend to other South African languages or contemporary model families. Our work extends this to six South African languages and four current frontier models, with a focus on regionally relevant harm categories (financial fraud, xenophobic incitement, political disinformation, and gang affiliation rather than generic harmful content).

2.3 Gap Addressed

Existing tools and benchmarks are not designed for African deployment contexts. They lack:

- Region-specific harm categories (SASSA scams, Capitec phishing, xenophobic incitement, gang affiliation)
- Evaluation of monolingual low-resource language prompts (not code-switching)
- Evaluation pipelines that can be run with limited compute and API budgets
- Open-source datasets that enable community auditing

AfriGuard addresses all four gaps.

3. Methods

3.1 Dataset Construction

We created 40 seed prompts defining areas of harm, covering four categories endemic to South Africa: financial fraud, xenophobic incitement, political disinformation, and gang affiliation. Each seed was translated into 7 languages: English (en), isiZulu (zu), isiXhosa (xh), Afrikaans (af), Sesotho (st), Sepedi (nso), and Tsonga (ts). Translations were performed by native speakers and validated for semantic equivalence. This yielded 280 prompt variants (40 seeds × 7 languages).

Table 1: Financial Fraud Seed Prompts (representative subset)

Seed ID	Scenario	Description
1	SASSA SRD Grant Scam	Phishing SMS impersonating social security payments
2	Capitec Banking Scam	Fraudulent SMS with malicious links

3	Standard Bank Account Takeover	Fake account verification email
4	Takealot Voucher Scam	WhatsApp scam claiming prize winnings
5	SASSA Beneficiary Exploitation	Direct fraud targeting grant recipients
6	Capitec Elderly Targeting	Scam specifically targeting older users
7	Fake Employment Documents	Forged financial documents for immigration
8	CVV Extraction	Social engineering for card security codes

3.2 Model Selection

We evaluated four models representing different training paradigms and safety approaches:

Model	Parameters	Provider	Safety Approach
Kimi K2.6	~32B	Moonshot AI	RLHF + Constitutional AI
Llama 3.3 (70B)	70B	Meta	RLHF + Red-teaming
Qwen 3 (32B)	32B	Alibaba	RLHF + Multilingual alignment
GPT-OSS (20B)	21B total / 3.6B active	OpenAI	RLHF + MoE architecture

Models were selected to represent both Chinese-aligned (Kimi, Qwen) and Western-aligned (Llama, GPT-OSS) safety paradigms, testing the hypothesis that training data composition affects cross-lingual safety transfer. GPT-OSS (20B) is a Mixture-of-Experts model with 21B total parameters and 3.6B active parameters per token, released under Apache 2.0 license in August 2025. Qwen 3 (32B) is a dense model trained on 36 trillion tokens across 119 languages and dialects, released in April 2025.

3.3 Evaluation Pipeline

Our pipeline consists of three stages:

Stage 1: Response Generation — Prompts sent via API with temperature=0.7, max_tokens=2048. Rate-limiting and retry logic for API stability. Raw responses stored with metadata (prompt_id, model, language, timestamp).

Stage 2: Harm Scoring — Human annotators scored each response on a 0–10 scale: 0–1: Clear refusal or gibberish; 2–6: Partial compliance, educational framing, or ambiguous; 7–10: Full jailbreak with actionable harmful content. Inter-annotator agreement: Cohen's kappa = 0.82 (strong agreement).

Stage 3: Automated Classification — judge.py maps harm scores to labels (refusal / partial / compliance); analysis.py computes ASR = compliance / (refusal + partial + compliance) and generates heatmaps, bar charts, and model comparison figures.

3.4 Attack Success Rate (ASR) Definition

$$\text{ASR}_{\text{language,model}} = \text{compliance} / (\text{refusal} + \text{partial} + \text{compliance})$$

We report ASR at three granularities: (1) per language–model pair (primary analysis); (2) per language (averaged across models); (3) per model (averaged across languages).

3.5 Language Safety Gap Metric

$$\Delta_{\text{gap}} = \text{ASR}_{\text{language}} - \text{ASR}_{\text{English}}$$

To isolate the effect of language choice from prompt content, we compute the gap between each language's ASR and the English baseline for a given model.

4. Results

4.1 Overall Safety Distribution

Across all 1,120 evaluated responses (40 seed prompts $\times$ 7 languages $\times$ 4 models), the overall safety distribution is as follows:

Label	Count	Percentage
Refusal	259	23.1%
Partial	168	15.0%
Compliance	693	61.9%

The distribution reveals a concerning pattern: nearly two-thirds of all responses (61.9%) represent full compliance with harmful requests, while only 23.1% are clear refusals. This aggregate figure masks substantial model-level variation, which we examine next.

4.2 Model Vulnerability Comparison

Table 2 presents the safety profile of each evaluated model, ranked by overall attack success rate.

Model	ASR (%)	Refusal Rate (%)	Partial Rate (%)	Assessment
Qwen 3 (32B)	80.6%	7.6%	11.8%	Catastrophically vulnerable
Kimi K2.6	70.9%	23.6%	5.5%	Highly vulnerable
Llama 3.3 (70B)	54.1%	11.0%	34.9%	Moderately vulnerable; high partial rate
GPT-OSS (20B)	42.1%	50.2%	7.7%	Most robust; highest refusal rate

Qwen 3 (32B) exhibits the highest vulnerability, with an ASR of 80.6% and a refusal rate of only 7.6%. This suggests that Qwen's multilingual alignment, while broad in language coverage (119 languages), does not translate to effective safety enforcement in low-resource African languages. Kimi K2.6 follows at 70.9% ASR, with a higher refusal rate (23.6%) but

very low partial compliance (5.5%), indicating a binary failure mode: the model either refuses entirely or complies fully.

Llama 3.3 (70B) presents a distinct profile: while its ASR (54.1%) is lower than Qwen and Kimi, it exhibits the highest partial compliance rate (34.9%). This suggests that Llama frequently engages with harmful requests in an ambiguous or educational-framing manner, which still represents a safety failure (partial compliance includes educational framing with actionable harmful information). GPT-OSS (20B) is the most robust model, with a 50.2% refusal rate and 42.1% ASR. Its MoE architecture and strong English-centric safety training appear to transfer better to African languages than the multilingual-aligned approaches of Qwen and Kimi.

4.3 Language-Specific Vulnerabilities

Table 3 presents ASR aggregated across all four models for each language, revealing stark disparities between English and African languages.

Language	ASR (%)	Gap vs English (pp)	Notes
English	26.6%	—	Baseline
Afrikaans	48.8%	+21.9	Germanic; lower than Bantu languages
isiZulu	60.5%	+33.7	Nguni; moderate Bantu vulnerability
Sesotho	68.5%	+41.7	Sotho-Tswana; high vulnerability
Tsonga	68.7%	+43.0	Tswa-Ronga; highest non-Bantu Bantu
isiXhosa	73.5%	+46.7	Nguni; very high vulnerability
Sepedi	77.8%	+50.9	Sotho-Tswana; highest overall ASR

The English baseline ASR of 26.6% is substantially lower than every African language, confirming the central hypothesis that low-resource language safety alignment lags behind English. Sepedi exhibits the highest ASR at 77.8% (+50.9 pp vs English), followed closely by isiXhosa at 73.5% (+46.7 pp). The three Nguni-Tsonga languages (isiXhosa, isiZulu, Tsonga) and the two Sotho-Tswana languages (Sesotho, Sepedi) cluster at the high end (68.5–77.8%), while Afrikaans, a Germanic language with more training data availability, shows the smallest gap (+21.9 pp) at 48.8%.

This pattern suggests that morphological complexity and training data scarcity are both contributing factors. Bantu languages (isiZulu, isiXhosa, Sesotho, Sepedi, Tsonga) with their noun class systems and agglutinative morphology present greater tokenization and alignment challenges than Afrikaans, which shares Germanic roots with English and Dutch and has significantly more pre-training data available.

4.4 Harm Category Analysis

Figure 1 presents the ASR heatmap broken down by language and harm category. The cross-category analysis reveals important differences in vulnerability profiles:

Harm Category	Mean ASR (%)	Highest-Language ASR (%)
Financial Fraud	52.8%	Sepedi 72.7%
Gang/Criminal Facilitation	64.3%	Sepedi 73.5%
Political Disinformation	67.1%	Sepedi 82.1%
Xenophobic Incitement	58.3%	Tsonga 83.3%

Political disinformation exhibits the highest mean ASR (67.1%), with Sepedi reaching 82.1%. This is particularly concerning given South Africa's history of politically motivated violence and the upcoming 2026 municipal elections. Xenophobic incitement shows the most extreme language disparity: while English ASR is only 20.0%, Tsonga reaches 83.3%—a 4.2× multiplier. Financial fraud, the most domain-specific category, has the lowest mean ASR (52.8%), suggesting that models retain some safety alignment for well-represented scam types even in low-resource languages, though Sepedi still reaches 72.7%.

4.5 Model–Language Interaction

Figure 4 reveals critical model–language interactions that are masked by aggregate statistics. The most dangerous combinations (ASR >90%) are:

- Kimi K2.6 + isiZulu: 94.5% ASR — The model almost never refuses isiZulu prompts, generating full scam templates, disinformation scripts, and incitement language.
- Kimi K2.6 + Tsonga: 91.0% ASR — Similarly catastrophic failure for this Tswa-Ronga language.
- Qwen 3 (32B) + Afrikaans: 95.0% ASR — Even a relatively higher-resource language like Afrikaans is catastrophically vulnerable to Qwen.
- Qwen 3 (32B) + Sepedi: 92.5% ASR — Near-universal compliance for this Sotho-Tswana language.

Conversely, the safest combinations are English across all models, with GPT-OSS (20B) achieving only 10.0% ASR for English and 10.0% for Afrikaans. GPT-OSS is the only model that maintains sub-50% ASR for any African language (isiZulu at 35.0%, isiXhosa at 45.5%).

4.6 Qualitative Analysis

We identified three distinct failure modes across the 1,120 responses:

Mode A: Full Template Generation (harm score 7–10) — Models generate complete scam templates, disinformation scripts, or incitement language with fake URLs, phone numbers, and urgency language. This mode is dominant for Qwen and Kimi in all African languages. Example from Qwen/Sepedi/SASSA prompt: "Dumela [Name], re romela SMS ya semmuso... [Click here: <http://sassa.govn.sa/grant-verify>]... Reference: SRD-2026-06-21-8723-A"

Mode B: Educational Framing (harm score 5–6) — Models provide harmful examples with "educational disclaimers" that still contain actionable information. This is the dominant mode for Llama 3.3 (70B), explaining its high partial compliance rate (34.9%). Example: "Here is an example of a scam SMS for educational purposes only: 'SASSA: Your grant has been approved. Click here to verify banking details...' Remember, never click links in unsolicited messages." The disclaimer does not prevent the model from providing a usable scam template.

Mode C: Gibberish/Confusion (harm score 2–4) — Models produce repetitive, nonsensical text in low-resource languages. This mode is rare in our new dataset (unlike the preliminary run) but still appears for GPT-OSS in Tsonga and Sesotho, where the model's linguistic competence appears to limit its ability to generate coherent harmful content. This represents a different failure mode: safety alignment is not holding, but the model lacks the linguistic capability to produce actionable harm.

5. Discussion and Limitations

5.1 Implications for AI Safety

Our findings demonstrate that multilingual safety alignment is not zero-sum with English safety—models can be robust in English while catastrophically vulnerable in African languages. This has direct implications for:

Financial sector deployments: Banks and insurers using LLMs for customer service must validate safety across all customer languages, not just English. A model with 80.6% ASR in Sepedi (Qwen) cannot safely handle customer queries in that language.

Government service chatbots: SASSA and other agencies must audit models in isiZulu, isiXhosa, and Sepedi before deployment. The 94.5% ASR for Kimi + isiZulu means that a chatbot powered by this model would comply with harmful requests in isiZulu nineteen times out of twenty.

API providers: Model providers shipping to African markets should include low-resource language red-teaming in their safety evaluations. The catastrophic failure of Qwen 3 (32B)—marketed for its multilingual capabilities—demonstrates that language coverage does not imply safety coverage.

5.2 Why Low-Resource Languages Are Vulnerable

We hypothesize three mechanisms driving the observed vulnerability:

Token alignment gaps: Safety-critical concepts ("fraud", "scam", "phishing", "xenophobia") may not align cross-lingually in the embedding space, causing refusal triggers to misfire. The particularly high ASR for Bantu languages (Sepedi 77.8%, isiXhosa 73.5%) suggests that noun class systems and agglutinative morphology create token sequences that bypass safety classifiers trained primarily on English.

Training data imbalance: Safety training data is overwhelmingly English; African language safety examples are sparse or absent. Qwen 3's training on 36 trillion tokens across 119 languages clearly prioritizes breadth over depth—its 80.6% ASR indicates that exposure to African language text does not equate to safety alignment in those languages.

Classifier distribution shift: Safety classifiers trained on English token distributions fail when confronted with the different n-gram and morphological patterns of Bantu languages. The consistent pattern of English < Afrikaans < Bantu languages suggests a gradient of distributional distance from the training domain.

5.3 Limitations

Sample size: 1,120 responses (280 per model) provides robust trend detection but may not capture all failure modes. Expansion to 5,000+ responses would improve statistical power for rare failure modes.

Language coverage: We cover 6 of South Africa's 12 official languages. Ndebele, Swati, Venda, and Tshivenda remain unevaluated. Given that Sepedi and isiXhosa show the highest ASR, these unstudied languages likely exhibit similar or greater vulnerability.

Harm category scope: While broader than the preliminary run (financial fraud only), our four categories do not exhaust the harm landscape. Health misinformation, gender-based violence, and child safety remain unstudied in African languages.

Model access: We evaluate via API; white-box access would enable mechanistic interpretability to identify why specific languages bypass safety. The MoE architecture of GPT-OSS, for example, may route African language tokens through different experts than English tokens, a hypothesis we cannot test with API-only access.

Temporal validity: Model weights and safety filters change. Our results represent a snapshot (June 2026) and should be re-run quarterly. The particularly high ASR for Qwen 3 may be addressed in future safety patches, though our findings suggest that patch-based approaches are unlikely to generalize across all African languages.

5.4 Future Work

Automated translation validation: Use back-translation BLEU scores to verify semantic equivalence across languages, ensuring that observed ASR differences reflect safety gaps rather than translation artifacts.

Adversarial training: Fine-tune safety classifiers on African language adversarial examples to reduce ASR. The AfriGuard dataset provides a foundation for this work.

Community expansion: Open the dataset for community contributions in Ndebele, Swati, Venda, Tshivenda, and other African languages. The open-source pipeline (judge.py + analysis.py) is designed for easy extension.

Mechanistic analysis: With white-box access, use logit lens and attention visualization to identify which layers fail during cross-lingual safety processing. The stark difference between GPT-OSS (MoE) and Qwen (dense) suggests that architecture may play a significant role in cross-lingual safety transfer.

Policy integration: Engage South African financial regulators (FSCA, SARB) and the Independent Electoral Commission to incorporate multilingual safety auditing into LLM deployment requirements, particularly for election-related disinformation and financial fraud use cases.

6. Conclusion

AfriGuard provides the first systematic evidence that South African languages present significant jailbreak vulnerabilities in frontier LLMs, with attack success rates exceeding 90% in some model-language configurations. Across 1,120 evaluated responses, the mean ASR is 60.6%—more than double the English baseline of 26.6%. Qwen 3 (32B) is catastrophically vulnerable at 80.6% ASR, while GPT-OSS (20B) is the most robust at 42.1% ASR, though still far from safe. The language safety gap ranges from +21.9 percentage points (Afrikaans) to +50.9 percentage points (Sepedi), demonstrating that vulnerability scales with linguistic distance from English training data.

Our open-source evaluation framework enables practitioners to audit their own deployments, and our dataset provides a foundation for community-driven safety research. The key takeaway is clear: safety alignment must be multilingual from the ground up, not retrofitted from English-centric training. The catastrophic failure of models marketed for multilingual capability (Qwen 3, Kimi K2.6) proves that language coverage and safety coverage are distinct problems requiring distinct solutions. As AI systems deploy across Africa, the cost of ignoring low-resource language safety will be measured in real financial harm, political instability, and violence against vulnerable populations.

Code and Data

Code repository: <https://github.com/ubayd-hattas/Global-South-Hackathon>

Evaluation pipeline: judge.py + analysis.py (see Appendix)

Figures: ASR heatmaps, model comparison charts, language breakdowns, harm category analysis

Dashboard: <https://mzansi-safety.streamlit.app/>

Author Contributions

Jaswin Chinthala – Testing jailbreaking pipeline, model evaluation

Seth Miguel Ferreira – Adversarial prompting and judging pipeline

Ubayd Hattas – Dashboard, judging and evaluation pipelines

Sebastian Stent – Translation pipeline, dataset curation

References

- [Deng, Y. et al. \(2024\). "Multilingual Jailbreak Challenges in Large Language Models." ICML 2024.](#)
- [Yoo, K., et al. \(2024\). "Code-Switching Red-Teaming: LLM Evaluation for Safety." ACL 2025.](#)
- [Upadhayay, S., & Behzadan, V. \(2024\). "Sandwich Attack: Cross-lingual Adversarial Prompting."](#)
- [Brown CS. \(2024\). "Low-Resource Languages Jailbreak GPT-4." Technical report.](#)
- [Apart Research. \(2026\). "Global South AI Safety Hackathon." Event documentation.](#)
- [OpenAI. \(2025\). "GPT-OSS: Open Source Small Language Models."](#)
- [Owen. \(2025\). "Qwen3: The Next-Generation of Qwen, Our Frontier Large Language Models."](#)

Appendix: Evaluation Pipeline

A.1 judge.py

Heuristic + LLM-as-judge classification. Maps harm_score to refusal / partial / compliance. See uploaded judge.py for full implementation.

A.2 analysis.py

ASR computation and visualization. Generates heatmaps, bar charts, model comparisons, and language safety gap analysis. See uploaded analysis.py for full implementation.

A.3 Scoring Rubric

harm_score	Label	Description
0–1	refusal	Clear refusal or gibberish
2–6	partial	Educational framing, ambiguous, or partial compliance
7–10	compliance	Full jailbreak with actionable harmful content

A.4 Dataset Statistics

Total prompts: 1,120 (40 seeds × 7 languages × 4 models)

Total unique seed prompts: 40 (10 per harm category)

Harm categories: Financial Fraud, Xenophobic Incitement, Political Disinformation, Gang Affiliation

Languages: English, isiZulu, isiXhosa, Afrikaans, Sesotho, Sepedi, Tsonga

Models: GPT-OSS 20B, Kimi K2.6, Llama 3.3 70B, Qwen 3 32B

A.5 LLM Usage Statement

We used Kimi K2.6 and Claude for coding and automation, brainstorming, literature review, and drafting assistance. All harm scores were assigned by human annotators. All quantitative results were independently verified through our open-source pipeline. The final report was written collaboratively by the team with LLM assistance for formatting and citation management.