

SignalShield Africa

A Human-in-the-Loop Scam Intelligence Framework for AI-Amplified Messaging Fraud in South Africa

Trivesh Sivnarain | Independent Researcher

Daniel Mokone | Independent Researcher

With Apart Research

Regional track: Africa | Sub-track: Monitoring and tooling for harms not yet at the foreground of AI safety

Abstract. Generative AI can make familiar scams easier to personalise, rewrite and spread through SMS, WhatsApp, email and voice-note-style messages. We present SignalShield Africa, a human-in-the-loop framework that identifies observable scam signals, produces an explainable risk assessment and groups related reports into possible campaigns. We implemented a deterministic proof of concept and evaluated it on 160 fictional, redacted South African-style scenarios. On an 80-message challenge split, the deception-fingerprint method achieved 85.9% high-risk recall, 12.5% false-positive rate, 72.9% campaign-grouping F1, 75.0% category accuracy and 81.2% routing agreement; a literal keyword baseline performed substantially worse. The study is a controlled feasibility exercise, not a deployment claim. The current prototype is English-first and rule-based. We propose a future multilingual extension that could add local-language and code-switched message support, privacy-conscious edge deployment, and lightweight contextual models for use in banking, telecom, and fraud-reporting environments. These components were not implemented in this study and require separate evaluation.

¹ Research conducted at the Global South AI Safety Hackathon, June 2026.

1. Introduction

AI safety in Africa is not only about failures at the frontier of model development. It also includes the harms that become easier to create and spread when capable generative tools are placed in everyday digital environments. Online scams and phishing remain major cybercrime concerns across the continent, while reporting and response capacity varies widely [1].

A South African deployment must also account for language diversity, code-switching, and uneven digital access. Scam messages may use local languages or familiar institutional language to appear more trustworthy. For this reason, SignalShield Africa is designed as an English-first proof of concept with a planned multilingual extension, rather than as a final language-complete solution.

Generative AI changes the economics of social engineering. It can help an attacker write clearer messages, produce many variations of the same story, tailor wording to a target group and move across channels. Recent studies have shown that AI-assisted spear-phishing and smishing can be persuasive, and that people often struggle to tell AI-authored messages from human-authored ones [2, 3, 4].

SignalShield Africa focuses on a narrow practical question: how can a customer-facing or analyst-facing tool surface visible scam signals without claiming to prove that a message is fraudulent or AI-generated? The framework is designed for suspicious SMS, WhatsApp messages, email excerpts and voice-note transcripts. It does not replace investigation. Its role is to suggest a safer next step, such as verifying through an official channel, not paying, not forwarding, or escalating the case for human review.

Research question. How can a privacy-conscious, human-in-the-loop system triage AI-amplified messaging scams in South African digital channels, and how does a structured signal representation compare with literal keyword matching on controlled scenarios?

Contributions.

- A threat model for banking phishing, public-benefit and revenue impersonation, recruitment and courier-fee scams, civic misinformation and trusted-contact emergencies.
- A proposed deception fingerprint that records claimed authority, requested action, manipulation tactics, likely harm and a recommended route.
- A reproducible, controlled evaluation of a rule-based proof of concept against a literal keyword baseline.
- A proposed multilingual and privacy-conscious deployment pathway that could extend the evaluated triage core to local-language, code-switched, banking-app, telecom, and fraud-reporting environments.

2. Related Work

Research on LLM-generated phishing shows why static word lists are unlikely to be enough. A recent systematisation describes the growing generation, characterisation and defence gap around LLM-assisted textual phishing campaigns [2]. In a spear-smishing study, participants frequently judged GPT-4 messages to be convincing and had difficulty identifying whether they were AI-authored [3]. Related work has also demonstrated that automated systems can

support targeted phishing at scale [4].

SignalShield is closer to explainable warning and triage work than to a pure classification task. PhishXplain, for example, argues that contextual explanations can help people understand why a threat is suspicious [5]. Our addition is campaign awareness: many AI-assisted variants can share the same manipulation pattern even when their wording is different. A future South African deployment would also need multilingual and code-switching support; current language resources are uneven across local languages [6].

3. Methods

3.1 Scope and design

The threat model is AI-amplified social engineering. An attacker may use a generative tool to draft, personalise, translate or repeatedly rewrite a message that impersonates an authority or trusted contact, asks the recipient to take an unsafe action, and uses urgency, fear, reward, secrecy or forwarding pressure. The prototype does not infer AI authorship, verify the truth of a public claim, inspect live URLs or make automatic account decisions. High-risk outputs are intended to prompt verification or human review.

3.2 Deception fingerprint

Rather than comparing only raw wording, the prototype creates a structured description of the message. The same representation can explain a result to a user and help group reports that may belong to the same campaign. In a real deployment, message submission would be consent-based, personally identifiable information would be minimised before analysis, and raw reports would be visible only to authorised reviewers.

Fingerprint field	Example evidence	Use
Claimed authority	Bank, public-benefit office, revenue authority, courier, recruiter, election body or trusted contact	Sets the impersonation context
Requested action	Share a code, submit identity or bank details, pay, open a link, forward a claim	Maps the likely harm
Manipulation tactic	Urgency, fear, reward, secrecy, panic or forwarding pressure	Explains why the message is risky
Recommended route	Ignore, verify officially, do not pay, do not forward, call independently, analyst review	Gives a safer next step

Table 1: Deception fingerprint fields used by the proof of concept.

3.3 Benchmark and evaluation

We created a fictional, redacted benchmark of 160 South African-style scenarios. The development partition contained 80 messages (64 harmful and 16 benign controls). The challenge partition also contained 80 messages (64 harmful and 16 benign controls), but used different wording, lower-signal variants and benign messages that deliberately discussed scam-related topics. The harmful scenarios covered eight campaign families: banking profile verification, one-time-code capture, public-benefit impersonation, revenue-refund impersonation, recruitment fees, courier fees, civic misinformation and trusted-contact emergencies.

The keyword baseline matched literal terms such as “OTP”, “bank”, “refund”, “urgent” and “fee”. The fingerprint method used broader phrase patterns and combinations of authority, requested action, pressure tactics and protective wording. We measured high-risk recall, false-positive rate, campaign-grouping F1, threat-category accuracy and recommended-action agreement. The dataset and script were run locally, and the 160-row benchmark was recorded with SHA-256 hash `a77d619b36166da6b23a52e8a7c254ed640ff2962b6d5227d31b22bc2f23651e`.

The development partition informed rule design. The challenge partition is reported as a harder internal robustness check. It is not an independent untouched test set: both the scenarios and the rules were developed within the same short hackathon. This distinction matters. The results show controlled feasibility under wording variation, not real-world generalisation.

3.4 Proposed multilingual and edge-deployment extension

The evaluated prototype focuses on English-language message text and explainable rule-based triage. A future deployment could add a language-routing layer to identify English, local-language, and code-switched messages before analysis. Non-English messages could then be processed using a locally hosted translation component or a multilingual classifier, followed by the same deception-fingerprint and routing logic evaluated in this study.

A future production architecture could also combine the current rules layer with a lightweight contextual model. The rules layer would remain important because it gives clear reasons for a risk score, while the contextual model could help with less explicit wording and language variation. Privacy-conscious deployment could process messages

on-device or at the organisational edge where appropriate, with data minimisation, access controls, and human review. These multilingual, edge-processing, and contextual-model components were not implemented or evaluated in the present study.

4. Results

Table 2 reports the challenge-partition result. The fingerprint method was consistently more useful than literal keyword matching. It flagged 55 of 64 harmful challenge messages as High or Critical risk, while the keyword baseline flagged 15 of 64. It also produced two benign false positives out of 16 controls, compared with five for the baseline. All reported results in this section refer only to the implemented English-first, deterministic triage core. They do not measure multilingual translation quality, transformer-model performance, mobile-device latency, or edge-deployment feasibility.

Metric	Keyword baseline	Deception fingerprint
High-risk recall	23.4%	85.9%
False-positive rate	31.2%	12.5%
Benign safety (1 – false-positive rate)	68.8%	87.5%
Campaign-grouping F1	25.2%	72.9%
Threat-category accuracy	30.0%	75.0%
Recommended-action agreement	40.0%	81.2%

Table 2: Challenge-split results on 80 fictional, redacted scenarios. Higher is better except for false-positive rate.

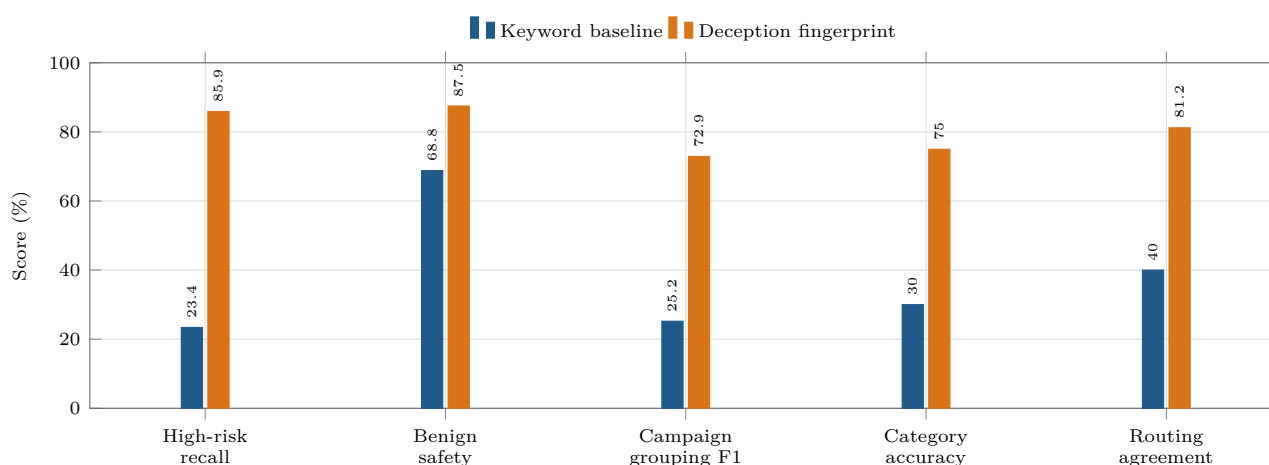


Figure 1: Challenge-split comparison. “Benign safety” is one minus the false-positive rate, so higher values are better across the plotted bars.

Development results were higher, as expected, because those scenarios informed rule design. The challenge split was harder: its nine false negatives generally avoided explicit code, payment or link cues. Its two false positives were educational messages whose wording resembled recruitment-fee or revenue-account scam language.

5. Discussion and Limitations

The main result is not that a short rule list can solve messaging fraud. It cannot. The result is that a structured view of a message is more useful than isolated keywords on this controlled benchmark. Looking at who the message claims to be, what it asks the recipient to do and how it applies pressure makes it easier to explain the risk and suggest a safer response. This is especially important when an attacker can cheaply generate many wording variants of the same scam [2, 3, 4].

The campaign perspective is also practical. A fraud team may receive many reports that do not share exact words but still follow the same playbook: the same claimed authority, requested action and pressure tactic. Grouping those reports can support early monitoring, customer communications and analyst prioritisation. This is a future workflow concept; the present prototype does not connect to a bank, a messaging platform or a real reporting queue.

The current approach can also grow without losing its main advantage: it can explain why a message was flagged. In a future multilingual version, a local-language or code-switched message could first be translated or analysed by a multilingual model, after which the same questions would still be asked: who is the message pretending to be, what does it want the person to do, and how is it applying pressure? This would create a practical path from the current English-first prototype to a broader African consumer-protection tool. It would still need human review and should not make automatic fraud decisions.

Limitations and Dual-Use Considerations

This is a small, English-dominant, author-created synthetic benchmark. It contains no real customer messages, live links, sender metadata, image attachments, audio provenance signals or institutional telemetry. It cannot prove that a message is AI-generated, prove fraud or verify the truth of civic or financial claims. The development and challenge partitions improve structure, but they do not make this an independent external evaluation because the team created both the rules and the scenarios during the same hackathon.

Publishing detailed detection rules can help attackers test boundaries. For that reason, the public report does not reproduce full message templates, live links, phone numbers or production thresholds. A real deployment should minimise retained data, restrict analyst access, log decisions, test for language and demographic bias, and keep humans responsible for any high-impact action. The tool should never automatically block an account or accuse a customer based only on a message score.

The current evaluation is English-dominant and does not test isiZulu, isiXhosa, Sesotho, Afrikaans, or code-switched messages. Translation-based or multilingual extensions could introduce their own safety failures, including loss of urgency cues, incorrect interpretation of local expressions, or inconsistent handling of mixed-language text.

The study also does not measure latency, memory use, device compatibility, quantisation performance, or edge-deployment feasibility. Claims about low-cost local deployment therefore remain proposed engineering directions rather than demonstrated results.

Future Work

Future work should focus on four areas. First, the system should be evaluated on real, consented and de-identified scam reports, with independently labelled test sets. Confirmed reports could also support periodic, human-approved updates to phrase patterns and scoring weights when scammers change their wording. Any proposed update should be reviewed by analysts, version-controlled, and tested on separate evaluation data before deployment. Second, future versions should test local-language and code-switched messages, including isiZulu, isiXhosa, Sesotho and Afrikaans. Third, the explainable rules layer could be combined with a lightweight multilingual contextual model and tested for translation quality, fairness, latency, memory use and low-cost hardware performance. Fourth, the system could be co-designed with banks, telecom providers, fraud teams and civil-society organisations for use in banking apps, contact-centre tools, fraud-reporting portals or approved messaging workflows.

6. Conclusion

SignalShield Africa treats AI-amplified scam messaging as a deployment-safety problem. The proposed system does not try to be a perfect “AI detector” or replace fraud investigations. It offers a narrower early-warning layer: identify visible manipulation cues, explain why they matter, recommend a safer next step and group similar reports into possible campaigns.

On a controlled 80-message challenge partition, the deception fingerprint outperformed literal keyword matching across risk triage, benign-message safety, campaign grouping, category assignment and action routing. The findings are preliminary, but the design provides a concrete starting point for a regionally grounded, human-in-the-loop defence against scams that generative AI can make easier to scale.

Code and Data

The code, fictional redacted benchmark, requirements file, evaluation outputs and reproducibility instructions are publicly available at github.com/triveshsivnarain004/signalshield-africa. The repository contains the 160-message benchmark, evaluation script, metrics by split, per-message outputs, charts and dataset hash. Detailed fictional scam-message templates, live links and production thresholds are intentionally excluded from this report.

Author Contributions

Trivesh Sivnarain and Daniel Mokone contributed equally to the conception, design, development, evaluation, interpretation and writing of this project. Both authors reviewed and approved the final manuscript.

LLM Usage Statement

Large language models were used to brainstorm the project direction, improve clarity of writing, help draft code, and create initial fictional scenario ideas. The authors reviewed and edited the final text, ran the evaluation locally, checked the reported outputs and remain responsible for all citations, calculations and claims. No live scam infrastructure, real victim data or malicious payloads were used.

References

- [1] INTERPOL. (2025). *Africa Cyberthreat Assessment 2025*. <https://www.interpol.int/News-and-Events/News/2025/New-INTERPOL-report-warns-of-sharp-rise-in-cybercrime-in-Africa>
- [2] Chen, F., Wu, T., Nguyen, V., & Rudolph, C. (2025). *SoK: Large Language Model-Generated Textual Phishing Campaigns: End-to-End Analysis of Generation, Characteristics, and Detection*. arXiv:2508.21457. <https://arxiv.org/abs/2508.21457>
- [3] Francia, J., Hansen, D., Schooley, B., Taylor, M., Murray, S., & Snow, G. (2024). *Assessing AI vs Human-Authored Spear Phishing SMS Attacks: An Empirical Study Using the TRAPD Method*. arXiv:2406.13049. <https://arxiv.org/abs/2406.13049>
- [4] Heiding, F., Lermen, S., Kao, A., Schneier, B., & Vishwanath, A. (2024). *Evaluating Large Language Models' Capability to Launch Fully Automated Spear Phishing Campaigns: Validated on Human Subjects*. arXiv:2412.00586. <https://arxiv.org/abs/2412.00586>
- [5] Roy, S. S., Torres, C., & Nilizadeh, S. (2025). *Explain, Don't Just Warn!: A Real-Time Framework for Generating Phishing Warnings with Contextual Cues*. arXiv:2505.06836. <https://arxiv.org/abs/2505.06836>
- [6] Marivate, V., Dzingirai, I., Banda, F., et al. (2025). *Mafoko: Structuring and Building Open Multilingual Terminologies for South African NLP*. arXiv:2508.03529. <https://arxiv.org/abs/2508.03529>