Poster: A Few-Shot Learning Method for SMS Phishing Detection and Explanation Using SLMs

Seyed Mohammad Sanjari
Department of Computer Science
Tennessee Tech University
ssanjarip42@tntech.edu

Jessee Roberts
Department of Computer Science
Tennessee Tech University
jtroberts@tntech.edu

Mir Mehedi A. Pritom
Department of Computer Science
Tennessee Tech University
mpritom@tntech.edu

I. Introduction and Motivation

SMS Phishing (*aka* Smishing) is a rising cybersecurity threat that uses deceptive mobile messages to lure users into revealing sensitive information or clicking on malicious links attached in SMS messages. These messages often imitate trusted entities and rely on urgency, fear, or curiosity to manipulate recipients and thus remain highly effective [1].

Traditional smishing detection techniques either rely on rule-based or supervised ML-based models, both of which suffer from scalability, adaptability, dynamicity, and deployability issues. Many state-of-the-art models require training and fine-tuning on large datasets, which is often time not present publicly to train an effective model. Moreover, some existing Large Language Model (LLM) based approaches may require significant compute resources, which limits their practical use, especially for mobile or edge deployment scenarios where we have limited computation capacity [2].

In this work, we aim towards developing a in-device smishing detection that satisfies the following core requirements: **(i)** Lightweight privacy-preserving, suitable for mobile devices and offline inference to ensure user data does not leave user device; **(ii)** Robust against text evasion and adversarial perturbations; **(iii)** Explainable, providing users with reasoning on why a message is flagged as smishing.

To meet these goals, our research explore the potential use of pre-trained light-weight small language models (SLMs) such as *Gemma-2B* [3], *Phi-3 mini* [4], and *Qwen2.5-3B* [5] with few-shot learning for smishing detection within the mobile environment while also enabling the explanation. The in-mobile computation ensures user data privacy as the SMS may include personal information which should not leave the user device. Additionally, the models are prompted with a handful of labeled examples and makes predictions without any fine-tuning or retraining, enabling low-overhead deployment.

Additionally, since language models are trained on diverse and noisy data, they naturally exhibit robustness to textual perturbations like misspellings, inserted tokens, or paraphrased attack content, which makes them well-suited for adversarially robust detection. We further enhance explainability by analyzing which few-shot examples influenced each classification decision, allowing the model's behavior to be interpreted directly from the prompt structure. Our preliminary study results show the efficacy of the proposed method for correctly identifying smishing messages while ensure user data privacy and explaining the detection in a user-friendly manner.

Challenges. SLMs do offer efficiency and on-device deployment potential, but there are challenges– a higher tendency to produce hallucinated outputs, and the risk of reinforcing biases present in their training data. Additionally, ensuring privacy during inference and preventing user data leakage remains critical. Finally, aggressive compression techniques like pruning or quantization can degrade model accuracy, requiring careful tuning to maintain reliable performance.

II. Methodology

We use instruction-tuned smaller language models such as *Gemma-2B*, *Qwen2.5-3B*, and *Phi-3 mini* to classify SMS messages as either 'smishing' or 'benign' using a few-shot learning approach. We guide the model by embedding a small set of labeled examples directly in the prompt so that it can infer the decision boundary by contextualization. Our proposed 3-step smish detection and explanation process is discussed below–

[S1] Prompt Construction. We begin by constructing prompts for few-shot classification. To do this, we first encode both the labeled dataset and the target SMS message using the Sentence-BERT (SBERT) model [6]. To guide the language model's behavior, we define an explicit instructional role within the prompt. The model is instructed to act as "an expert in identifying SMS phishing attempts (smishing)" and to classify each message as either "smishing" or "benign." This role specification helps align the model's responses with the intended task and ensures consistent, focused outputs by limiting the response to the label only. Based on the semantic similarity, we retrieve the top 3 most similar examples for each class (*'smishing'* and *'benign'*) from the labeled dataset corresponding to the target message. Each retrieved examples are then provided to the SLM as prompts in following format: "<SMS Text>, Classification: <Label>". These examples guide the small language models (SLMs) on how to classify new unseen messages.

[S2] Message Classification. Here we use the SLMs to classify target messages using both zero-shot and few-shot prompting. In the zero-shot setting, the model receives a general instruction and the target message. In the few-shot setting, the prompt includes six contextually similar labeled

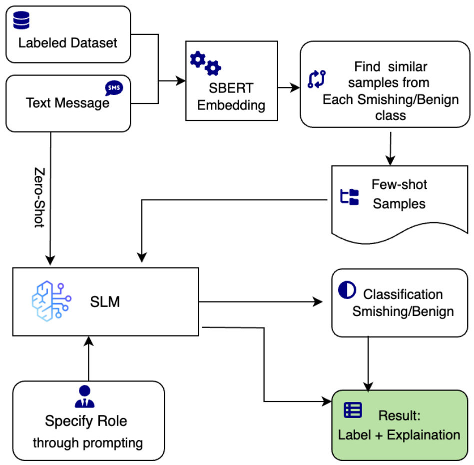

Fig. 1: Few-shot learning and SLM-based smish detection approach

examples (3 per class) followed by the test SMS. We also provide a role specification prompt instructing the model to act as an expert in smishing detection. The model is prompted to output a classification label—either smishing or benign. The final prediction is parsed using rule-based pattern matching for consistency and reproducibility. This approach avoids full fine-tuning and satisfies privacy-preserving for on-device deployment.

[S3] Explainability. Our explainability method benefits from the same Sentence-BERT (SBERT) semantic embeddings used during few-shot sampling. After classification, we analyze which set of few-shot examples (*'smishing'* or *'benign'*) is more semantically aligned with the target message. The explanation is generated based on the distribution of similarities. This method provides a lightweight, transparent justification without requiring attention-level analysis.

Finally, we evaluate with a real-world dataset comprising both smishing and benign messages using metrics such as accuracy, precision, recall, and F1-score.

III. Preliminary Results

A. Dataset

For this research, we used the Super SMS Dataset [7], a large and up-to-date collection of over 53,000 real-world messages. It provides a realistic mix of smishing and benign texts, allowing us to test our model's accuracy and robustness effectively. we chose 150 random messages as target message (with 90 benign and 60 smishing), and a set of $1,000$ messages for few-shot samples (with 618 benign and 382 smishing).

B. Evaluation Results

To assess performance, we evaluate our method on 150 real-world target messages using both zero-shot and few-shot prompting strategies where few-shot is showing better classification accuracy for both SLM models.

Example Use Case Scenario. In figure 2 we showed how the model not only classifies messages correctly but also explains its reasoning in a human-understandable way. This can help users to convince about the detection or correct misclassification.

TABLE I: Few-shot vs Zero-shot Smishing Classification Using *Phi-3-mini*, *Gemma-2B*, and *Qwen-2.5-3B* (full and 4bit) SLMs

Setting	Class	Phi-3-mini	Gemma-2B	Qwen-2.5-3B	Qwen-2.5-3B(4bit)	Support
Few-shot	Benign (0)	Precision: 0.85	0.79	0.88	0.90	90
		Recall: 0.89	0.94	0.77	0.51	
		F1-score: 0.87	0.86	0.82	0.65	
	Smishing (1)	Precision: 0.82	0.88	0.70	0.55	59
		Recall: 0.76	0.61	0.85	0.92	
		F1-score: 0.79	0.72	0.77	0.69	
	Overall	**Accuracy: 0.84**	**0.81**	**0.80**	**0.67**	**149**
Zero-shot	Benign (0)	Precision: 0.96	0.95	0.94	0.98	90
		Recall: 0.26	0.66	0.88	0.46	
		F1-score: 0.40	0.78	0.91	0.62	
	Smishing (1)	Precision: 0.46	0.64	0.83	0.54	59
		Recall: 0.98	0.95	0.92	0.98	
		F1-score: 0.63	0.77	0.87	0.70	
	Overall	**Accuracy: 0.54**	**0.77**	**0.89**	**0.66**	**149**

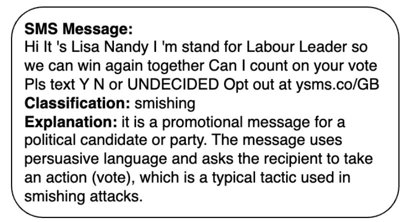

Fig. 2: An example of 'smishing' classification and explanation

IV. Conclusion and Future Directions

This study demonstrates the feasibility of using instruction-tuned small language models (SLMs) for smishing detection with few-shot prompting. This method offers an interpretable and lightweight alternative to fully fine-tuned LLMs. Our results show that carefully selected few-shot examples can improve classification accuracy and provide meaningful semantic context for generating natural-language explanations. By extracting themes from semantically similar few-shot samples, we enable the model to justify its decision in a concise, user-friendly form without exposing the actual samples or requiring manual rule engineering. This approach enhances transparency and user trust, which are critical for deployment in resource-constrained or privacy-sensitive environments, such as mobile devices. However, we acknowledge that SLMs can be vulnerable to various attacks as well, which we want to further explore in the future and adopt a secure use of SLMs for in-device deployment and computation.

References

[1] S. Mishra and D. Soni, "Smishing Detector: A security model to detect smishing through SMS content analysis and URL behavior analysis," *Future Generation Computer Systems*, vol. 108, pp. 803–815, 7 2020.
[2] M. Salman, M. Ikram, N. Basta, and M. A. Kaafar, "Spallm-guard: Pairing sms spam detection using open-source and commercial llms," 2025. [Online]. Available: https://arxiv.org/abs/2501.04985
[3] G. Team, "Gemma," 2024. [Online]. Available: https://www.kaggle.com/m/3301
[4] M. Abdin, J. Aneja, H. Awadalla, A. Awadallah, A. A. Awan, N. Bach, A. Bahree, A. Bakhtiari, J. Bao, H. Behl *et al.*, "Phi-3 technical report: A highly capable language model locally on your phone," *arXiv preprint arXiv:2404.14219*, 2024.
[5] Q. Team, "Qwen2.5: A party of foundation models," September 2024. [Online]. Available: https://qwenlm.github.io/blog/qwen2.5/
[6] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," *arXiv preprint arXiv:1908.10084*, 2019.
[7] M. Salman, M. Ikram, and M. A. Kaafar, "An empirical analysis of sms scam detection systems," *arXiv preprint arXiv:2210.10451*, 2022.

A Few-Shot Learning Method for SMS Phishing Detection and Explanation Using SLMs

Seyed Mohammad Sanjari[1,2] Jesse Roberts[1] Mir Mehedi A. Pritom[1,2]
[1]Department of Computer Science, Tennessee Tech University
[2]Resilient AI-enabled Cybersecurity and Trustworthiness (ReACT) Lab, Tennessee Tech University

CEROC
Cybersecurity Education, Research and Outreach Center

1. Introduction

What is SMISHING ?

SMS phishing (smishing) is one of the most prevalent attack vectors targeting mobile users. Losses from scam texts jumped from $86M in 2020 to $470M in 2024, a nearly 5.5x increase [1].

Traditional ML or LLM-based detection methods are:

- Opaque (lack explainability),
- Large-scale (difficult to deploy on-device),
- Hard to adapt (require re-training for new attack variants).

Smishing Example

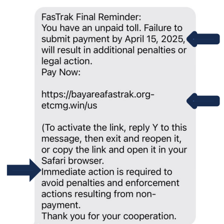

Fig. 1: A financial service type smish message (asking to use a link to submit a payment)

2. Objectives

To address the limitations of traditional smishing detectors, we propose, a lightweight and explainable detection framework designed for on-device deployment.

Research Objectives

- **Develop a lightweight smishing detection framework**
  → Design a system suitable for on-device and offline inference using Small Language Models (SLMs).
- **Use few-shot learning for classification**
  → Classify SMS messages as *smishing* or *benign* using only a few labeled examples, without model fine-tuning.
- **Enable explainable smishing detection**
  → Provide users with interpretable explanations by linking test messages to semantically similar examples.
- **Facilitate privacy-preserving deployment**
  → Avoid sending messages to cloud servers by ensuring local-only inference on mobile or edge devices.

3. Methodology

A. Prompt Construction

- Embed all labeled SMS messages using SBERT.
- Compute cosine similarity to the test message.
- Select top-3 similar smishing and benign messages as few-shot examples.
- Specify role to SLM through prompts [2]

B. Message Classification

- Use instruction-tuned SLMs.
- Perform classification using few-shot prompts [3].
- Pattern-match output to extract reliable labels.

C. Explainability

- Justify predictions based on similarity distribution.
- Use similarity scores to identify the closest supporting example.

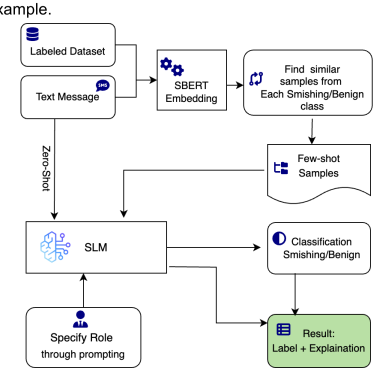

Fig. 2: Detection method overview

4. Case Study

- We visualize how our system works on real examples.
- Messages with a dark background indicate smishing, while those with a light background indicate benign content.
- Each classification is accompanied by a brief explanation to help users understand the reasoning behind the decision.

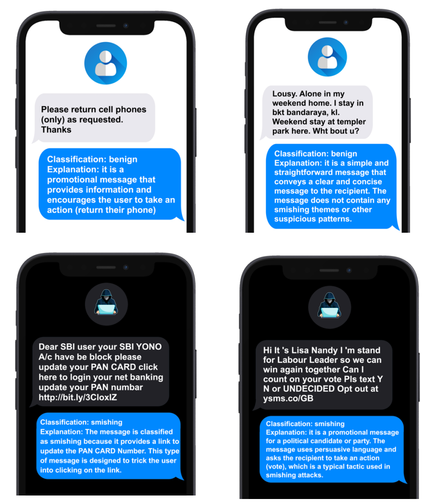

Fig. 3 Examples of smishing (dark background) and benign (light background) message detection

5. Dataset and Evaluation

We evaluated our method using real-world *smishing* and *benign* messages from the Super SMS Dataset [4]. A set of 150 test messages (90 benign, 59 smishing) was used for classification, and 1,000 labeled messages were used for few-shot support sample selection. We tested ***Phi-3-mini***, ***Gemma-2B***, and ***Qwen 2.5-3B*** (standard and quantized) using zero-shot and few-shot prompting.

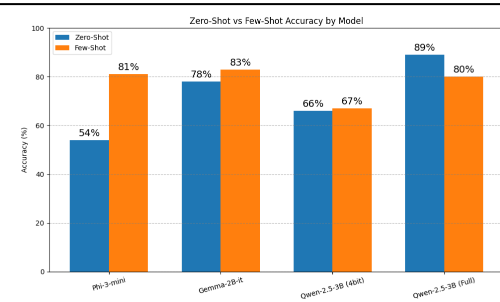

Fig. 4 Few-shot vs Zero-shot smishing classification performances using SLMs

6. Takeaways

- Few-shot prompting enables SLMs to detect smishing effectively without model fine-tuning.
- Gemma-2B-it achieves the highest few-shot accuracy (83%), while Qwen-2.5-3B (Full) leads in zero-shot (89%).
- Phi-3-mini shows a significant boost from 54% (zero-shot) to 81% (few-shot), highlighting the value of contextual examples.
- Semantic explanations using similar messages improve user trust and system transparency.
- Challenges remain around hallucinations, model bias, and accuracy loss due to compression in quantized models.

7. Future Works

- Enhance dataset variety
- Evaluate additional SLMs to compare efficiency and accuracy
- Investigate techniques to ensure safety and security of SLMs deployed on mobile devices
- Analyze model resilience to prompt injection, evasion, and hallucination attacks
- Compare against fine-tuned SLMs and LLM baselines to contextualize performance trade-offs

8. References

1. *New FTC data show top text message scams of 2024; overall losses to text scams hit $470 million.* (2025, April 16). Federal Trade Commission. https://www.ftc.gov/news-events/news/press-releases/2025/04/new-ftc-data-show-top-text-message-scams-2024-overall-losses-text-scams-hit-470-million
2. Wang, Z. M., Peng, Z., Que, H., Liu, J., Zhou, W., Wu, Y., ... & Peng, J. (2023). Rolellm: Benchmarking, eliciting, and enhancing role-playing abilities of large language models. *arXiv preprint arXiv:2310.00746.*
3. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *Advances in neural information processing systems, 33*, 1877-1901
4. Salman, M., Ikram, M., & Kaafar, M. A. (2022). An empirical analysis of SMS scam detection systems. arXiv preprint arXiv:2210.10451.

9. Acknowledgement

We thank the Cybersecurity Education, Research, and Outreach Center (CEROC) and the Department of Computer Science at Tennessee Tech University for generously supporting this research.