OPEN ACCESS

Volume: 5

Issue: 3

Month: July

Year: 2026

ISSN: 2583-7117

Published: 20.07.2026

Citation:

Krithika Mani "Small Language Models for Phishing Website Detection: A Review of Cost, Performance, and Privacy Trade-Offs" International Journal of Innovations in Science Engineering and Management, vol. 5, no. 3, 2026, pp. 120-123

DOI:

10.69968/ijisem.2026v5i3120-123

Small Language Models for Phishing Website Detection: A Review of Cost, Performance, and Privacy Trade-Offs

Krithika Mani[1]

[1]*MSc, Computing and Technology, University: Northumbria university, Mail id : keerthimohano71992@gmail.com*

Abstract

This review investigates Goldenits et al.'s (2025) empirical comparison of fifteen open small language models (SLMs) for phishing website detection, which aimed to evaluate whether locally hosted models can achieve the same accuracy as proprietary large language models (LLMs), without the prohibitive costs or privacy concerns. The authors test each model with a stratified sample of 1,000 labelled websites from a pool of 10,395 websites and measure the accuracy, precision, recall and F1 score of the results. The best local model, llama3.3:70b, achieves an F1 score of 0.893 and recall of 0.948, which is close to, but still lower than, the F1 scores above 0.95 achieved by the largest proprietary systems (Goldenits et al. 2025). This review restates the three research questions posed in this paper, assesses the evidence provided for each of the questions and situates the evidence in the context of the existing literature on cost-aware deployment (Irugalbandara et al. 2024; Kavya & Sumathi, 2024) and LLM-based phishing detection (Koide et al. 2024). It concludes that, besides the number of parameters, the deployability of an SLM depends on its architecture and on the reliability of its output format, which does not depend only on its classification skill.

Keywords; Phishing Website Detection, Small Language Models (SLMs), Large Language Models (LLMs), Cybersecurity, Cost-Aware AI Deployment.

INTRODUCTION

Phishing continues to be a cybersecurity threat that is found everywhere. APWG reported a total of over 1.1 million unique phishing pages in the second quarter of 2025 alone, a 13% rise from the previous quarter (APWG, 2025). This scale is important because it reveals the weaknesses of what is currently in use as defence. Classical machine learning classifiers, URL blacklists, and visual-similarity models based on CNNs work decently in controlled scenarios, but their effectiveness is reduced in the presence of attackers who modify their tactics and need continuous re-training to stay one step ahead. (Kavya & Sumathi, 2024) Large language models (LLMs) have been suggested as a more flexible alternative, as they can ingest raw HTML as a natural language text and reason about its structure without manually engineered features. The commercial systems currently have good detection performance (Koide et al. 2024) using screenshots, OCR-extracted text and HTML. However, this comes at a practical cost: using the API from a vendor means being charged per call, relying on external infrastructure, having to wait for external responses and sending potentially sensitive HTML and URLs to a third party for processing (Irugalbandara et al. 2024). Goldenits et al. (2025) explore whether there is a possibility to escape this trade-off using small language models (SLMs) - open models with 70 billion parameters or less that can be deployed locally-and still achieve detection performance comparable to state-of-the-art.

RESEARCH QUESTIONS AND SCOPE

The study is organised around three research questions, each targeting a distinct dimension of deployability rather than accuracy alone:

- **RQ1:** Does an SLM produce coherent and reliable phishing classifications, together with an interpretable rationale for each decision?

- **RQ2:** How do the cost and performance of locally hosted SLMs compare with state-of-the-art proprietary models such as GPT-4?
- **RQ3:** How does the length of the HTML input affect runtime and detection accuracy?

Fifteen publicly available SLMs are compared, drawn from the Gemma, DeepSeek, Qwen, Llama, Mistral, gpt-oss, Phi, and Dolphin families, ranging from 1 billion to 70 billion parameters (Goldenits et al. 2025). The comparison is restricted to raw HTML and URL input, without screenshots or other modalities. This narrows the scope of the claims the study can support, since visual and OCR-based signals are excluded by design, but it also keeps the experiment reproducible and leaves an explicit opening for multimodal extension in future work.

DATA UNDER EVALUATION

The evaluation is based on an open database of 10,395 labelled websites, half of which are genuine websites and the rest are phishing pages (Goldenits et al. 2025). From this pool, they take a stratified sample of 1,000 sites and create two versions of each site's HTML: one that retains 5% of the most informative tags (links, images and metatags, but not necessarily those sorted alphabetically), and another that retains 50% of these. This design decision is not accidental; truncation is used to reduce the number of tokens that a model needs to process, thus impacting runtime and cost, and to remove content that is not relevant to phishing detection. The two levels of truncation also allow isolating the length of input as a variable, a requirement for RQ3. The downside is that the 5% extract may not contain context, like if there is prose around the form or other HTML structure, a human parser might take that into account, but the 5% truncation approach is an assumption about which HTML elements are the most diagnostic.

METHODOLOGY

There are two phases in the evaluation: In Experiment 1, the feasibility screen, 40 websites are tested using all the models, and each website is tested five times, using deterministic decoding, in order to check output stability, formatting compliance and runtime behaviour. At this point, the models that fail to reliably generate a short justification, a binary decision and a phishing score are discarded before large-scale testing starts. In Experiment 2, the models that survived the procedure are tested on the entire 1,000-site sample, and the accuracy, precision, recall and F1-score (Goldenits et al. 2025) are reported. The two-stage design is methodologically sound since it decouples the practical engineering question of whether the output of a model can be automatically parsed from the substantive question of whether the model's judgment is correct. By folding both in a single pass, it would have merged two failure modes, which would have made results harder to interpret.

KEY FINDINGS

RQ1: Coherence and reliability of classification. The answer is mixed in the results. Multiple models, such as llama3.1:8b, llama3.2:1b, gpt-oss:20b, qwen3:4b, qwen3:30b, and deepseek-r1:1.5b, were excluded because they often generated invalid and unparsable JSON responses before the large-scale evaluation (Goldenits et al. 2025). This is a worthwhile academic result in itself - a model which correctly classifies phishing sites in theory is useless in an automatic pipeline if there is no reliable way to output it. The models that made it through this filter, just as with most of the others, fell into two categories: reasoning-related “thinking” models like the DeepSeek, Qwen and gpt-oss series, and architecture-based non-reasoning models like Gemma and Llama, with the former clearly outperforming the latter. Another argument for reliability comes from the correlation between the risk scores and the binary decision: all instances of a confirmed phishing page were judged as having a risk score of 0 (min), and a few of the prompt examples were assigned a risk score of 10 (max), indicating that the risk score acts as a more constrained calibration of the models' confidence in the judgement, rather than a true measure of risk. This undermines the interpretability answer to RQ1, as the risk score can never hit its signal ceiling when the binary decision is correct, which makes it not entirely reliable as a continuous signal. The authors consider this a desirable bias because the costs of a false alarm and a missed attack are very different (recall vs precision), in nearly all models. RQ2 - cost and performance compared to proprietary models. The large but not dominant llama3.3:70b model performed best overall with an F1-score of 0.893, accuracy of 0.887 and recall of 0.948 (Goldenits et al. 2025), while the much smaller mistral-nemo:12b model performed at 0.858, demonstrating that competitive performance does not necessarily depend on size. The smallest models or weaker variants in otherwise high-

performing families, such as gemma3:4b, dolphin3:8b, and phi3:14b, achieved F1-scores ranging from 0.46 to 0.69, which are not usable for deployment in production. When compared to proprietary benchmarks, the best SLM has achieved is still below the F1 scores above 0.95 that were previously reported for the largest proprietary models, although the difference is a lot less than it was a few years ago, when the open models achieved about 74% accuracy. The cost of running the entire experiment locally is estimated to have been €41 in renter A100 GPU time, while the cost of equivalent-scale runs of GPT-4o and GPT-4-turbo are estimated to have been around €31 and €109, respectively (Goldenits et al. 2025). A sustained high-traffic deployment would get into the break-even range within a few months, as the authors calculate it to be at 625 million to 3.6 billion processed tokens, which is cheaper than paying for the API call. This directly addresses RQ2: local SLMs are not yet the more accurate alternative; however, they become the cheaper alternative at realistic production quantities, depending on the tolerance for inaccuracy of the use case. Input length vs runtime and accuracy (RQ3). Scaling by prompt length instead of just the number of parameters in the model: Goldenits et al. (2025) found a correlation between prompt length and the total runtime of the test set across model families, with approximately a 51% increase in runtime when HTML truncation was increased from 5% to 50%. Overall, the two 70-billion-parameter models were the slowest, while others with smaller numbers of parameters, like llama3.2:1b and dolphin3:8b, did the analyses in a fraction of the time, further proving that runtime is not just about the number of parameters. This is a direct and well-supported response to RQ3: longer HTML inputs will be more expensive in runtime, and the cost of truncation will vary depending on the architecture, meaning that there are practical implications as to the aggressiveness of truncation in a production pipeline.

Table 1 Summary of reported classification performance by top- and bottom-performing models

Model	Parameters	Accuracy	Precision	Recall	F1-Score
llama3.3	70B	0.887	n/r	0.948	0.893
deepseek-r1	70B	n/r	n/r	n/r	0.873
mistral-nemo	12B	n/r	n/r	n/r	0.858
gemma3	12B / 27B	n/r	n/r	n/r	0.80–0.84
gemma3 / dolphin3 / phi3 (small variants)	4B–14B	n/r	n/r	n/r	0.46–0.69

n/r = not individually reported in the source paper. Source: Goldenits et al. (2025).

COST AND DEPLOYMENT CONSIDERATIONS

In addition to raw accuracy, the paper considers three ways of deployment: through a single proprietary API, rented GPU capabilities on-demand, and outright hardware ownership (Goldenits et al. 2025). A comparison is useful because it doesn't assume that costs are simply a side benefit of the decision – they're a direct consequence of it. Economies of scale are only achieved at volumes larger than 625M tokens-with GPT-4o costing €31 for the full experiment compared to GPT-4-turbo's cost of €109-so local deployment is not necessarily cheaper on a small scale. One restriction that the authors themselves acknowledge is that it takes several seconds to analyse per-site; if this is done for production traffic, then either parallelised infrastructure could be used, or a tiered pipeline could be implemented to offload obviously benign traffic to a cheaper classical machine learning model before invoking an SLM. This is important because it modifies the cost advantage – because the break-even is based on throughput that a naive single-instance deployment might not have been able to maintain.

DISCUSSION AND IMPLICATIONS

The review's key finding is that there is no absolute superiority of proprietary versus locally-hosted models; the model that is best is dependent on the volume of use, sensitivity of the data to be accessed, and the ceiling accuracy that is acceptable. There are three practical advantages to using locally hosted SLMs: there is no need for sensitive HTML or URL data to leave the organisation's infrastructure; the SLMs are independent of vendor pricing and uptime; and none of the tested SLMs had been adapted on the organisation's phishing data, which can in principle be done on the underlying model (Goldenits et al. 2025). The flip side is that there was no evidence of fine-tuning in this case, so it is possible but not assessed, and likely to be just as likely to reduce the remaining gap with the proprietary F1 scores over 0.95 as it would be to widen it; it would be fair to offer that as a possibility, though there is no evidence either way. The authors suggest using proprietary APIs for infrequent applications where precise results are important, and locally hosted SLMs, such as llama3.3:70b, deepseek-r1:70b and mistral-nemo:12b, for ongoing high-volume applications and cost management, along with data privacy.

CONCLUSION

One of the first systematic and cost-driven comparisons of open SRLMs for detecting phishing via HTML is given by Goldenits et al. (2025). The main contribution here is empirical evidence that mid-sized open models, in the 10- to 20-billion-parameter class, can now match the performance of much larger proprietary models and can be operated entirely by local infrastructure - directly addressing the motivating question behind RQ2. The study is, however, limited to a single-modality approach (HTML and URL) and lacks fine-tuning experiments, as both of these are natural directions for future work, which the authors point out. The limitations do not detract from the central conclusions of the paper, but rather are indicative of a lower bound rather than an upper bound for the fine-tuned, multimodal SLMs that are ultimately possible. The methodology and the open data and open code provide a yardstick for other open models to test against once they are developed.

REFERENCES

[1] Anti-Phishing Working Group (APWG). (2025). Phishing Activity Trends Report: 2nd Quarter 2025. APWG. https://docs.apwg.org/reports/apwg_trends_report_q2_2025.pdf

[2] Chataut, R. Gyawali, P. K. & Usman, Y. (2024). Can AI Keep You Safe? A Study of Large Language Models for Phishing Detection. 2024 IEEE 14th Annual Computing and Communication Workshop and Conference (CCWC), 0548-0554. https://doi.org/10.1109/CCWC60891.2024.10427626

[3] Goldenits, G. König, P. Raubitzek, S. & Ekelhart, A. (2025). Small Language Models for Phishing Website Detection: Cost, Performance, and Privacy Trade-Offs. arXiv:2511.15434.

[4] Irugalbandara, C. Mahendra, A. Daynauth, R. Arachchige, T. K. Dantanarayana, J. L. Flautner, K. Tang, L. Kang, Y. & Mars, J. (2024). Scaling Down to Scale Up: A Cost-Benefit Analysis of Replacing OpenAI's LLM with Open Source SLMs in Production. 2024 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), 280-291. https://doi.org/10.1109/ISPASS61541.2024.00034

[5] Kavya, S. & Sumathi, D. (2024). Staying ahead of phishers: a review of recent advances and emerging methodologies in phishing detection. Artificial Intelligence Review, 58(2), 50. https://doi.org/10.1007/s10462-024-11055-z

[6] Koide, T. Nakano, H. & Chiba, D. (2024a). ChatPhishDetector: Detecting Phishing Sites Using Large Language Models. IEEE Access, 12, 154381-154400. https://doi.org/10.1109/ACCESS.2024.3483905

[7] Koide, T. Fukushi, N. Nakano, H. & Chiba, D. (2024b). ChatSpamDetector: Leveraging Large Language Models for Effective Phishing Email Detection. arXiv:2402.18093.

[8] Lee, J. Lim, P. Hooi, B. & Divakaran, D. M. (2024). Multimodal Large Language Models for Phishing Webpage Detection and Identification. 2024 APWG Symposium on Electronic Crime Research (eCrime), 1-13. https://doi.org/10.1109/eCrime66200.2024.00007

[9] Nasution, A. H. Monika, W. Onan, A. & Murakami, Y. (2025). Benchmarking 21 open-source large language models for phishing link detection with prompt engineering. Information, 16(5).