Benchmarking Large Language Models for Zero-shot and Few-shot Phishing URL Detection

Najmul Hasan
University of North Carolina at Pembroke
nh0033@bravemail.uncp.edu

Prashanth BusiReddyGari
University of North Carolina at Pembroke
Prashanth.BusiReddyGari@uncp.edu

Abstract

The Uniform Resource Locator (URL), introduced in a connectivity-first era to define access and locate resources, remains historically limited, lacking future-proof mechanisms for security, trust, or resilience against fraud and abuse, despite the introduction of reactive protections like HTTPS during the cybersecurity era. In the current AI-first threatscape, deceptive URLs have reached unprecedented sophistication due to the widespread use of generative AI by cybercriminals and the AI-vs-AI arms race to produce context-aware phishing websites and URLs that are virtually indistinguishable to both users and traditional detection tools. Although AI-generated phishing accounted for a small fraction of filter-bypassing attacks in 2024, phishing volume has escalated over 4,000% since 2022, with nearly 50% more attacks evading detection. At the rate the threatscape is escalating, and phishing tactics are emerging faster than labeled data can be produced, zero-shot and few-shot learning with large language models (LLMs) offers a timely and adaptable solution, enabling generalization with minimal supervision. Given the critical importance of phishing URL detection in large-scale cybersecurity defense systems, we present a comprehensive benchmark of LLMs under a unified zero-shot and few-shot prompting framework and reveal operational trade-offs. Our evaluation uses a balanced dataset with consistent prompts, offering detailed analysis of performance, generalization, and model efficacy, quantified by accuracy, precision, recall, F1 score, AUROC, and AUPRC, to reflect both classification quality and practical utility in threat detection settings. We conclude few-shot prompting improves performance across multiple LLMs.

1 Introduction

Phishing remains a prevalent and evolving cybersecurity threat, with malicious URLs serving as a primary tool to deceive users into visiting fraudulent websites. These attacks often lead to credential theft, financial fraud, or data breaches. Traditional detection systems, particularly blacklist-based methods, fall short when faced with newly generated or obfuscated phishing URLs, as they are highly dependent on prior knowledge Tian et al. [2025]. Recent threat intelligence reports highlight the growing scale and sophistication of phishing attacks across industries IBM Security [2024], Hoxhunt [2024]. This limitation has fueled the need for learning-based approaches that generalize better to unseen or adversarial examples.

Early research on phishing URL detection leveraged machine learning (ML) and deep learning (DL) by extracting the lexical and structural characteristics of URLs. These approaches commonly use classifiers such as decision trees, support vector machines, and neural networks Omari and Oukhatar [2025], Kocyigit et al. [2024], Ghalechyan et al. [2024]. With the increasing complexity of phishing tactics, newer models have adopted deep neural architectures such as convolutional and recurrent networks that capture sequential patterns directly from raw URLs without manual feature engineering

LAW 2025 Workshop, 39th Conference on Neural Information Processing Systems (NeurIPS 2025).

Zara et al. [2024], Barik et al. [2025]. Several works also employ optimization strategies like genetic algorithms and ensemble models to handle class imbalance and enhance model robustness Remya et al. [2024], Rafsanjani et al. [2024].

Recently, LLMs have gained traction for cybersecurity tasks, including malicious URL detection. Pre-trained transformers such as BERTDevlin et al. [2019] and GPTRadford et al. [2018] can be adapted to URL-based tasks through fine-tuning or prompt-based classification Mahdaouy et al. [2024], Liu et al. [2025]. These models can capture contextual cues from input text, even when the data is minimal or unstructured. PhishURLDetect Ali and Subba [2025], for example, shows that fine-tuning LLMs using parameter-efficient methods such as LoRAHu et al. [2022] can produce competitive performance while significantly reducing computational overhead.

Despite these advances, comparative evaluation of proprietary LLMs under standardized zero-shot and few-shot prompting settings remains underexplored. Prior studies either benchmark a single model or evaluate performance under varied conditions without consistent metrics or datasets Nasution et al. [2025]. Moreover, while real-world phishing data is often imbalanced, balanced datasets are commonly used in evaluation to enable fair assessment of model generalization and efficiency. Some recent models designed for malicious URL classification Zhou et al. [2025], Senanayake et al. [2025] and phishing webpage detection Lee et al. [2024] highlight the potential of neural and transformer-based systems, but few investigate prompt-only inference in an instruction-tuned context.

In this work, we fill this gap by evaluating a range of proprietary LLMs on a balanced phishing URL dataset using prompt-based classification. Our experiments cover both zero-shot and few-shot scenariosBrown et al. [2020] across models such as GPT-4o (OpenAI), Claude-3-7-sonnet-20250219 (Anthropic), Grok-3-Beta(xAI). We report key evaluation metrics including accuracy, precision, recall, F1 score, AUROC, and AUPRC to assess model effectiveness and practical applicability in real-world phishing detection scenarios. In summary, our contributions to the community are as follows:

1. We present a unified benchmark of instruction-tuned proprietary LLMs for phishing URL detection under standardized zero-shot and few-shot prompting settings.
2. We evaluate all models on a publicly available phishing URL dataset, which we balanced by sampling equal numbers of phishing and legitimate URLs, using a consistent and model-agnostic prompt design.
3. We offer a comparative analysis of model performance and generalizability, and release our codebase to promote reproducibility and support further research in the community.

2 Related Work

Phishing URL detection has been widely investigated across a variety of ML paradigms. Early approaches primarily relied on handcrafted lexical and host-based features to build classifiers such as decision trees, support vector machines, or ensemble methods Omari and Oukhatar [2025], Kocyigit et al. [2024]. These models often struggled with generalization, especially in the presence of evolving phishing tactics and class imbalance. To address this, optimization strategies such as genetic algorithms, synthetic resampling (e.g., SMOTETomekOmari and Oukhatar [2025]), and feature weighting schemes have been proposed to enhance detection performance Remya et al. [2024], Rafsanjani et al. [2024].

Deep learning-based methods have introduced greater flexibility by automatically learning hierarchical representations from raw URLs. Studies have applied convolutional and recurrent neural networks to extract local and sequential patterns Zara et al. [2024], Barik et al. [2025], while recent designs incorporate multilayer perceptrons and attention mechanisms Ghalechyan et al. [2024]. Several efforts have focused on enhancing feature selection for neural architectures through empirical analysis and comparative evaluations Ghalechyan et al. [2024], Kocyigit et al. [2024]. Despite their promise, many of these models are limited to static supervised training and are typically benchmarked on datasets that lack zero-shot capabilities.

LLMs have recently been explored for phishing detection, leveraging their ability to capture semantic cues in textual inputs. DomURLs_BERT proposed a fine-tuned BERT-based classifier to detect both domains and URLs Mahdaouy et al. [2024], while PMANet introduced a post-trained transformer-based attention framework that incorporates multilevel semantic and lexical features Liu et al. [2025].

These works show the adaptability of language models to security tasks, although they primarily focus on traditional fine-tuning, and do not evaluate zero- or few-shot settings.

The parameter-efficient PhishURLDetect framework applied LoRA to adapt large models to phishing detection with reduced computational cost, but the evaluation was limited to a single instruction-tuned model and did not compare against LLMs Ali and Subba [2025]. Meanwhile, MADONNA employed neural networks with browser-level telemetry to detect malicious domains in real-time but was not designed to operate in prompt-based configurations Senanayake et al. [2025].

A broader benchmarking effort by Nasution et al. [2025] evaluated twenty-one open-source LLMs using prompt engineering for phishing link detection. However, the study excluded commercial models and was restricted to English-language prompts. Similarly, Zhou et al. [2025] proposed a CSPPC-BiLSTM framework that fused handcrafted features with sequence modeling but did not address inference efficiency or few-shot generalization. Few works have investigated prompt-only inference for phishing detection, although Lee et al. [2024] introduced a multimodal vision-language framework for phishing webpage identification.

Finally, the comprehensive survey by Tian et al. [2025] provides an overview of malicious URL detection, available datasets, and public codebases. Highlights the lack of unified benchmarking among emerging LLM-based methods and underscores the need to evaluate models under consistent experimental setups. This motivates the need for our work, which systematically compares zero-shot and few-shot prompting across a representative set of LLMs on a balanced dataset of phishing and legitimate URLs.

3 Methodology

We evaluate three large language models (GPT-4o, Claude-3.7-sonnet-20250219, and Grok-3-Beta) on phishing URL classification under zero-shot and few-shot prompting. Experiments are conducted on both balanced and imbalanced datasets to assess performance across different class distributions. Figure 1 illustrates the complete pipeline.

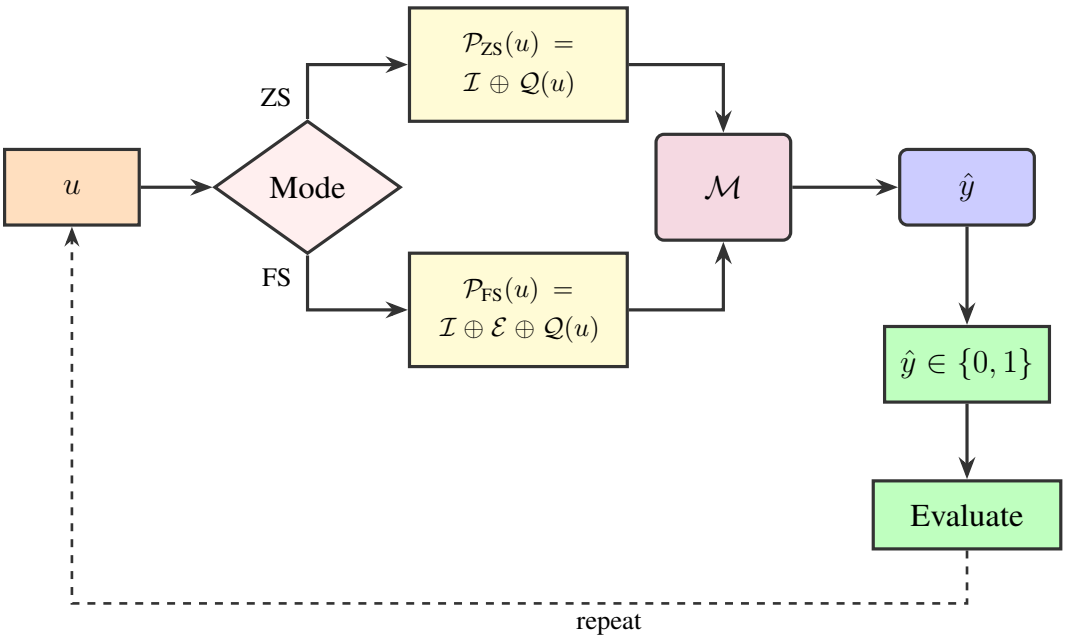


Figure 1: Phishing URL classification methodology. Each URL u is processed via zero-shot ($\mathcal{P}_{\text{ZS}}$) or few-shot ($\mathcal{P}_{\text{FS}}$) prompting. Zero-shot prompts contain task instruction $\mathcal{I}$ and query $\mathcal{Q}(u)$; few-shot prompts additionally include examples $\mathcal{E}$. The LLM $\mathcal{M}$ generates responses parsed to binary predictions $\hat{y} \in \{0, 1\}$ (0=phishing, 1=legitimate). Predictions are evaluated against ground truth y_i using macro-averaged accuracy, precision, recall, F1-score, AUROC, and AUPRC across all test URLs.

3.1 Experimental Setup

We use the PhiUSIIL Phishing URL dataset Prasad and Chandra [2024], preprocessing to retain URL strings and binary labels (0=phishing, 1=legitimate). We conduct two sets of experiments to evaluate model performance under different class distributions.

For balanced evaluation, we construct a corpus of 10,000 URLs by randomly sampling 5,000 phishing and 5,000 legitimate URLs using seed 42. In zero-shot experiments, all 10,000 samples are used for evaluation. In few-shot experiments, we sample 6 examples (3 phishing, 3 legitimate) as $\mathcal{E}$ and evaluate on the remaining 9,994 samples, ensuring examples are disjoint from the evaluation set.

For imbalanced evaluation, we construct test sets of 1,000 URLs with phishing ratios of 1% and 10%. Each ratio is tested with two random seeds (S123, S456) to verify stability across different samples. In zero-shot experiments, all 1,000 samples are used for evaluation. In few-shot experiments under 10% imbalance with seed S123, we vary the number of examples in $\mathcal{E}$ using 1, 3, or 9 examples and evaluate on the remaining samples.

3.2 Prompt Construction

Each prompt consists of three components: a task instruction $\mathcal{I}$, a query $\mathcal{Q}(u)$ for the target URL, and optionally, a set of examples $\mathcal{E}$ for few-shot learning. The task instruction is:

> You are a cybersecurity expert. Respond only with 0 for phishing or 1 for legitimate.

The query for each URL u is:

> URL: `{u}`
> Is this URL phishing or legitimate? Respond with 0 or 1.

For zero-shot prompting, we construct $\mathcal{P}_{\text{ZS}}(u) = \mathcal{I} \oplus \mathcal{Q}(u)$, where $\oplus$ denotes concatenation. For few-shot prompting, we construct $\mathcal{P}_{\text{FS}}(u) = \mathcal{I} \oplus \mathcal{E} \oplus \mathcal{Q}(u)$, where each example in $\mathcal{E}$ is formatted as:

> URL: `{u'}`
> Answer: `{y'}` (*label*)

with $y' \in \{0, 1\}$ and label text "phishing" for $y' = 0$ or "legitimate" for $y' = 1$.

For GPT-4o and Grok-3-Beta, zero-shot prompts consist of a system message containing the task instruction, followed by a user message with the URL query. Few-shot prompts use the same system message, followed by one user message per example, then a final user message with the target URL query. For Claude-3.7-sonnet, zero-shot prompts concatenate the task instruction and query separated by double newlines. Few-shot prompts concatenate the instruction, each example (also separated by double newlines), then the query.

3.3 Evaluation Protocol

We access the three models via their official APIs: GPT-4o (OpenAI), Claude-3.7-sonnet-20250219 (Anthropic), and Grok-3-Beta (xAI). All inference is performed with temperature set to 0 and maximum output tokens limited to 10. Model responses are parsed to extract binary predictions $\hat{y} \in \{0, 1\}$. Responses that cannot be parsed are excluded from evaluation.

We compute six metrics: accuracy, macro-averaged precision, recall, F1-score, AUROC, and AUPRC. Macro-averaging computes per-class metrics and averages them, treating both classes equally. Let $\mathcal{C} = \{0, 1\}$ denote the class set. For class c, true positives (TP_c), false positives (FP_c), and false negatives (FN_c) are used to compute precision $\text{TP}_c/(\text{TP}_c + \text{FP}_c)$, recall $\text{TP}_c/(\text{TP}_c + \text{FN}_c)$, and F1-score $2 \cdot \text{Precision}_c \cdot \text{Recall}_c/(\text{Precision}_c + \text{Recall}_c)$. Macro-averaged metrics average these values across both classes. All metrics are implemented using scikit-learn Pedregosa et al. [2011].

4 Experimental Results

We evaluate three large language models (GPT-4o, Claude-3.7-sonnet-20250219, and Grok-3-Beta) on phishing URL classification under zero-shot and few-shot settings. Experiments are conducted on both

balanced and imbalanced datasets to assess model performance across different class distributions. We report Accuracy, Precision, Recall, and F1 Score, with all metrics macro-averaged unless otherwise stated.

4.1 Balanced Dataset Evaluation

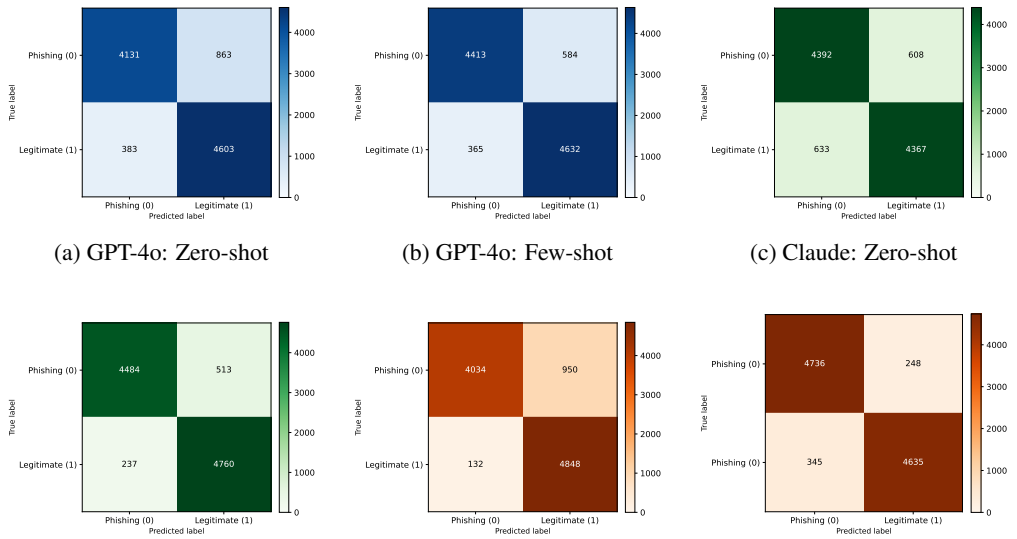
On a balanced test set of 10,000 URLs, few-shot prompting with six examples substantially improves performance across all models (Table 1). Grok-3-Beta achieves the strongest few-shot performance, leading in five of six metrics: Accuracy (0.9405), Precision (0.9492), F1 (0.9399), AUROC (0.9405), and AUPRC (0.9573). Claude-3.7 attains the highest Recall (0.9526) but lower Precision (0.9027) compared to Grok-3-Beta. GPT-4o demonstrates consistent gains across metrics but trails the other models in overall performance.

Grok-3-Beta exhibits a notable precision-recall trade-off when transitioning from zero-shot to few-shot settings: Recall decreases from 0.9735 to 0.9307 while Precision increases from 0.8361 to 0.9492, indicating stricter classification thresholds in few-shot mode.

Table 1: Performance comparison on balanced phishing URL detection (10,000 URLs). All metrics are macro-averaged.

Model	Setting	Accuracy	Precision	Recall	F1	AUROC	AUPRC
GPT-4o	Zero-shot	0.8752	0.8421	0.9232	0.8808	0.8752	0.9018
GPT-4o	Few-shot	0.9050	0.8880	0.9270	0.9071	0.9050	0.9258
Claude-3.7	Zero-shot	0.8759	0.8778	0.8734	0.8756	0.8759	0.9072
Claude-3.7	Few-shot	0.9250	0.9027	0.9526	0.9270	0.9250	0.9395
Grok-3-Beta	Zero-shot	0.8914	0.8361	0.9735	0.8996	0.8914	0.9114
Grok-3-Beta	Few-shot	0.9405	0.9492	0.9307	0.9399	0.9405	0.9573

4.1.1 Per-Class Evaluation



(a) GPT-4o: Zero-shot (b) GPT-4o: Few-shot (c) Claude: Zero-shot

(d) Claude: Few-shot (e) Grok-3: Zero-shot (f) Grok-3: Few-shot

Figure 2: Confusion matrices for all models under zero-shot and few-shot prompting.

Figure 2 shows confusion matrices for all models. Few-shot prompting with six examples reduces false negatives across all models. Grok-3-Beta demonstrates the largest improvement, with false negatives dropping from 950 to 248. GPT-4o and Claude-3.7 reduce false negatives from 863 to 584 and 608 to 513, respectively. Grok-3-Beta few-shot achieves the fewest false negatives (248), while

Grok-3-Beta zero-shot achieves the fewest false positives (132). Claude-3.7 few-shot reduces false positives to 237. These results reflect the precision-recall trade-off in Table 1.

4.1.2 AUROC and AUPRC Curves

ROC and Precision-Recall curves for all models under zero-shot and few-shot prompting are presented in Figures 3 to 5. Few-shot prompting with six examples consistently improves AUROC and AUPRC across models.

Grok-3-Beta achieves the highest few-shot performance (AUROC: 0.9405, AUPRC: 0.9573) with a steep ROC rise and stable PR curve, indicating strong discriminative ability. Claude-3.7 shows substantial improvements from zero-shot (AUROC: 0.8759, AUPRC: 0.9072) to few-shot (AUROC: 0.9250, AUPRC: 0.9395), demonstrating effective learning from examples. GPT-4o exhibits consistent gains, with AUROC improving from 0.8752 to 0.9050 and AUPRC from 0.9018 to 0.9258. These curves illustrate model behavior across classification thresholds, complementing the scalar metrics presented earlier.

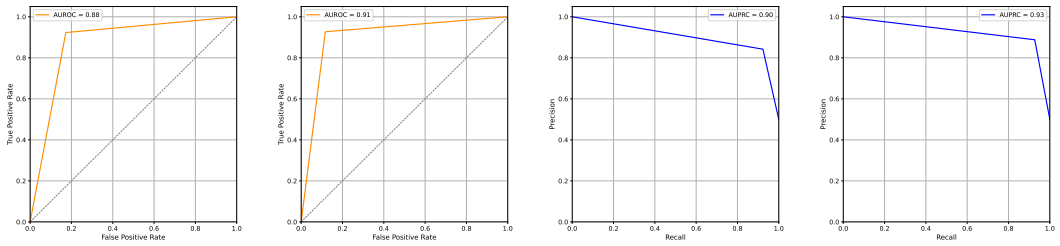


Figure 3: GPT-4o ROC (left) and PR (right) curves under zero-shot and few-shot prompting.

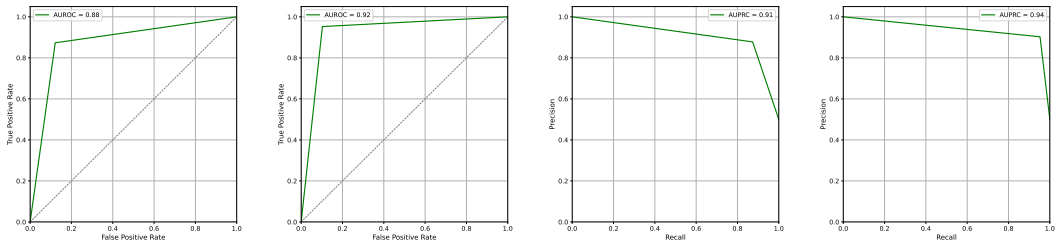


Figure 4: Claude-3.7 ROC (left) and PR (right) curves under zero-shot and few-shot prompting.

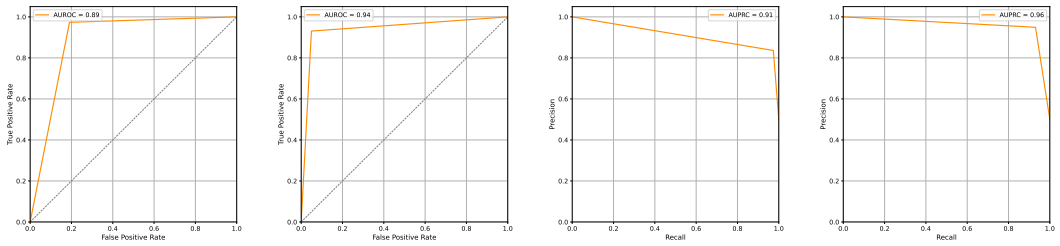


Figure 5: Grok-3-Beta ROC (left) and PR (right) curves under zero-shot and few-shot prompting.

4.2 Imbalanced Dataset Evaluation

Increasing phishing URL proportion from 1% to 10% improves F1 scores across all models (Table 2). GPT-4o improves from 0.559 to 0.785, Claude-3.7 from 0.534 to 0.761, and Grok-3-Beta from 0.657

to 0.854. Models demonstrate robust performance across random seeds S123 and S456, with F1 differences below 0.05 at 10% imbalance.

Few-shot learning with varying numbers of examples (1, 3, or 9) exhibits distinct patterns per model. Grok-3-Beta peaks at 1 example (F1: 0.906, Precision: 0.949) then degrades with 3 examples (F1: 0.821) and 9 examples (F1: 0.831). Claude-3.7 improves overall from 1 example (F1: 0.857) to 9 examples (F1: 0.876), with intermediate performance at 3 examples (F1: 0.842). GPT-4o gains consistently from 1 example (F1: 0.709) through 3 examples (F1: 0.801) to 9 examples (F1: 0.861). Grok-3-Beta dominates zero-shot settings with accuracy 0.976 and recall 0.938.

Table 2: Performance comparison of LLMs on phishing URL detection across zero-shot and few-shot settings. All experiments use 1,000 test samples. All metrics except Accuracy are macro-averaged across phishing and legitimate classes. S123 and S456 denote random seeds 123 and 456, respectively. Imbalance ratios (1% and 10%) indicate the proportion of phishing URLs in the test set. Few-shot experiments use seed 123 with 10% imbalance and vary the number of examples ($\mathcal{E} = 1, 3, 9$).

Model	**Metric**	**Zero-Shot**				**Few-Shot**		
		S123-1%	**S123-10%**	**S456-1%**	**S456-10%**	**$\mathcal{E}$=1**	**$\mathcal{E}$=3**	**$\mathcal{E}$=9**
GPT-4o	Accuracy	0.917	0.902	0.935	0.927	0.833	0.908	0.942
	Precision	0.544	0.742	0.561	0.789	0.676	0.754	0.821
	Recall	0.859	0.866	0.918	0.888	0.863	0.891	0.919
	F1-Score	0.559	0.785	0.591	0.828	0.709	0.801	0.861
Claude-3.7	Accuracy	0.881	0.879	0.903	0.903	0.945	0.933	0.951
	Precision	0.535	0.716	0.542	0.749	0.837	0.801	0.846
	Recall	0.890	0.879	0.902	0.911	0.881	0.905	0.915
	F1-Score	0.534	0.761	0.553	0.799	0.857	0.842	0.876
Grok-3-Beta	Accuracy	0.964	0.945	0.976	0.962	0.969	0.915	0.924
	Precision	0.602	0.839	0.640	0.881	0.949	0.768	0.783
	Recall	0.932	0.872	0.938	0.921	0.872	0.931	0.918
	F1-Score	0.657	0.854	0.708	0.900	0.906	0.821	0.831

5 Conclusion

Phishing attacks continue to evolve in sophistication, challenging traditional detection methods that depend on labeled datasets and manual feature engineering. Zero-shot and few-shot learning with large language models offer a practical alternative when labeled data is scarce or phishing tactics evolve rapidly. We benchmarked three large language models (GPT-4o, Claude-3.7-sonnet-20250219, and Grok-3-Beta) on phishing URL detection using prompt-based classification under both zero-shot and few-shot settings.

Our experiments on a balanced dataset of 10,000 URLs demonstrate that few-shot prompting with six examples substantially improves performance. Grok-3-Beta achieves the highest few-shot accuracy (0.9405) and F1 score (0.9399), while Claude-3.7-sonnet yields the best recall (0.9526). Under imbalanced conditions with 1% and 10% phishing ratios, models maintain robust performance, with Grok-3-Beta showing strong zero-shot capabilities. Few-shot learning consistently improves F1 scores, with Grok-3-Beta reaching 0.906 at $\mathcal{E}$=1 under 10% imbalance. These results demonstrate that prompt-based approaches can effectively detect phishing URLs with minimal examples.

Future work should validate these findings across diverse phishing datasets, explore prompt optimization strategies, and investigate the trade-offs between prompt-based inference and fine-tuned models in operational settings.

References

Irshad Ali and Basant Subba. Phishurldetect: A parameter efficient fine-tuning of llms using lora for detection of phishing urls. In *Proceedings of the 26th International Conference on Distributed Computing and Networking*, pages 278–279, 2025.

Kousik Barik, Sanjay Misra, and Raghini Mohan. Web-based phishing url detection model using deep learning optimization techniques. *International Journal of Data Science and Analytics*, pages 1–23, 2025.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

Hayk Ghalechyan, Elina Israyelyan, Avag Arakelyan, Gerasim Hovhannisyan, and Arman Davtyan. Phishing url detection with neural networks: an empirical study. *Scientific Reports*, 14(1):25134, 2024.

Hoxhunt. Phishing trends report 2024. `https://hoxhunt.com/guide/phishing-trends-report`, 2024. Accessed May 10, 2025.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

IBM Security. X-force threat intelligence index 2024. `https://www.ibm.com/downloads/documents/us-en/107a02e94948f4ec`, 2024. Accessed May 10, 2025.

Emre Kocyigit, Mehmet Korkmaz, Ozgur Koray Sahingoz, and Banu Diri. Enhanced feature selection using genetic algorithm for machine-learning-based phishing url detection. *Applied sciences*, 14 (14):6081, 2024.

Jehyun Lee, Peiyuan Lim, Bryan Hooi, and Dinil Mon Divakaran. Multimodal large language models for phishing webpage detection and identification. *arXiv preprint arXiv:2408.05941*, 2024.

Ruitong Liu, Yanbin Wang, Haitao Xu, Zhan Qin, Fan Zhang, Yiwei Liu, and Zheng Cao. Pmanet: Malicious url detection via post-trained language model guided multi-level feature attention network. *Information Fusion*, 113:102638, 2025.

Abdelkader El Mahdaouy, Salima Lamsiyah, Meryem Janati Idrissi, Hamza Alami, Zakaria Yartaoui, and Ismail Berrada. Domurls_bert: Pre-trained bert-based model for malicious domains and urls detection and classification. *arXiv preprint arXiv:2409.09143*, 2024.

Arbi Haza Nasution, Winda Monika, Aytug Onan, and Yohei Murakami. Benchmarking 21 open-source large language models for phishing link detection with prompt engineering. *Information*, 2025.

Kamal Omari and Ayoub Oukhatar. Advanced phishing website detection with smotetomek-xgb: Addressing class imbalance for optimal results. *Procedia Computer Science*, 252:289–295, 2025.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

Arvind Prasad and Shalini Chandra. PhiUSIIL Phishing URL (Website). UCI Machine Learning Repository, 2024. DOI: https://doi.org/10.1016/j.cose.2023.103545.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Ahmad Sahban Rafsanjani, Norshaliza Binti Kamaruddin, Mehran Behjati, Saad Aslam, Aaliya Sarfaraz, and Angela Amphawan. Enhancing malicious url detection: A novel framework leveraging priority coefficient and feature evaluation. *IEEE Access*, 2024.

S Remya, Manu J Pillai, Kajal K Nair, Somula Rama Subbareddy, and Yong Yun Cho. An effective detection approach for phishing url using resmlp. *IEEE Access*, 2024.

Janaka Senanayake, Sampath Rajapaksha, Naoto Yanai, Harsha Kalutarage, and Chika Komiya. Madonna: Browser-based malicious domain detection using optimized neural network by leveraging ai and feature analysis. *Computers & Security*, 152:104371, 2025.

Ye Tian, Yanqiu Yu, Jianguo Sun, and Yanbin Wang. From past to present: A survey of malicious url detection techniques, datasets and code repositories. *arXiv preprint arXiv:2504.16449*, 2025.

Ume Zara, Kashif Ayub, Hikmat Ullah Khan, Ali Daud, Tariq Alsahfi, and Saima Gulzar. Phishing website detection using deep learning models. *IEEE Access*, 2024.

Jinyang Zhou, Kun Zhang, Anas Bilal, Yu Zhou, Yukang Fan, Wenting Pan, Xin Xie, and Qi Peng. An integrated csppc and bilstm framework for malicious url detection. *Scientific Reports*, 15(1): 6659, 2025.