

Comparative Evaluation of Prompt Engineering Configurations and Open-Source Large Language Models for Phishing Email Detection

Naavin A/L Rajeswaren, Agus Hartoyo*, Farhad Nadi

Faculty of Artificial Intelligence and Frontier Technologies, UNITAR International University,
Malaysia

*Corresponding Author

DOI: <https://doi.org/10.47772/IJRISS.2026.100800666>

Received: 24 August 2026; Accepted: 29 August 2026; Published: 15 September 2026

ABSTRACT

Large Language Models (LLMs) offer a flexible approach to phishing-email classification, but their performance can vary with both prompt design and model selection. This study evaluates these factors through a two-stage experimental design using a balanced dataset of 1,000 emails comprising 500 phishing and 500 legitimate messages. In Stage 1, five complete prompt-engineering configurations—Zero-Shot, Few-Shot Demonstration, Expert Role-Based, Structured Criteria-Guided, and CoT-Style Structured Reasoning Prompting—were compared using Mistral. The best-performing configuration was then held constant in Stage 2 to compare five locally deployed open-source LLMs: Mistral, Phi-3, Qwen2.5, Llama 3.1, and Gemma 2. Performance was assessed using accuracy, precision, recall, F1-score, valid-output coverage, and processing time. Structured Criteria-Guided Prompting achieved the strongest prompt-level performance, with 86.33% accuracy and an F1-score of 85.43%. In the model comparison, Gemma 2 achieved the highest accuracy (87.00%), recall (90.00%), and F1-score (87.38%), while maintaining 100% valid-output coverage. Qwen2.5 achieved the highest precision (95.73%) and the shortest recorded processing time of 1 h 7 min 46 s. Valid-output coverage also varied across prompt configurations, with Expert Role-Based Prompting producing the lowest coverage at 88.2%. Overall, the results show that prompt configuration and model choice jointly influence phishing detection performance, false-alarm rates, output reliability, and local processing efficiency.

Keywords: phishing email detection, large language models, prompt engineering, open-source LLMs, cybersecurity, Ollama

INTRODUCTION

Phishing email remains a persistent cybersecurity threat because it combines technical deception with social-engineering tactics such as urgency, impersonation, credential requests, deceptive links, and contextual manipulation [1], [2]. As phishing messages become more polished and plausible, effective detection increasingly depends on interpreting message intent and context rather than relying solely on fixed indicators.

Advances in machine learning, deep learning, transformers, and Large Language Models (LLMs) have substantially expanded the techniques available for phishing-email detection [3]–[8]. LLMs are particularly relevant because they can perform text-classification tasks through natural-language instructions, while prompt engineering allows their behaviour to be adapted without modifying model parameters [9], [10]. However, reported results vary across models, prompts, datasets, and deployment settings, making it difficult to determine how prompt formulation and model choice interact under controlled conditions.

This study addresses this issue through a two-stage comparative evaluation. First, five prompt-engineering configurations are evaluated using Mistral as a common baseline model. The strongest-performing configuration is then applied consistently across five locally deployed open-source LLMs: Mistral, Phi-3, Qwen2.5, Llama 3.1, and Gemma 2. Performance is assessed using accuracy, precision, recall, F1-score, valid-output coverage, and processing time. Accordingly, the study addresses the following research questions:

RQ1: How do different prompt-engineering configurations affect phishing-email classification performance using a locally deployed LLM?

RQ2: Under a common structured prompt, how do selected open-source LLMs compare in accuracy, precision, recall, and F1-score?

RQ3: What trade-offs are observed between classification performance, valid-output coverage, and processing efficiency?

RELATED WORK

From Conventional Detection to LLM-Based Email Analysis

Phishing-email detection has progressed from rules, blacklists, URL indicators, and manually engineered features toward increasingly data-driven approaches. NLP-based methods enabled greater use of lexical and semantic information [3], while deep-learning models further reduced reliance on manual feature engineering by learning representations directly from email data [6]. Recent comparative studies demonstrate that strong supervised baselines already exist: Altwaijry et al. [7] reported high performance from deep-learning architectures, while Meléndez et al. [8] found transformer-based models competitive with or superior to conventional machine-learning classifiers.

LLMs extend this progression by supporting classification directly from natural-language instructions. Jamal et al. [5] applied transformer-based language modelling to phishing, spam, and legitimate-email classification, while Koide et al. [4] used LLM prompting for phishing-email detection and explanation. These studies demonstrate the feasibility of contextual language models for email-threat analysis, but their results also depend on factors such as model family, dataset, task formulation, and evaluation setting [4]–[6]. This variability motivates comparisons in which major experimental conditions are held constant.

Prompt Engineering for Security Classification

Prompt engineering specifies an LLM task through combinations of instructions, examples, roles, contextual information, reasoning cues, and output requirements without modifying model parameters [10]. Zero-shot and few-shot prompting are established in-context settings [11], role/persona prompting is also widely recognized [12], and Chain-of-Thought (CoT) and Zero-Shot-CoT represent established reasoning-oriented approaches [13], [14]. Recent reviews and taxonomies emphasize that these elements are composable rather than mutually exclusive; a single prompt may combine several of them [10], [15], [16].

Prompt engineering has also been examined in phishing-related security tasks. Trad and Chehab [9] compared prompting with fine-tuning for phishing URL detection and showed that task-specific prompting can support classification without parameter-level adaptation. Their study provides methodological support for evaluating prompt formulation, although its URL-classification task differs from the phishing-email classification considered here.

Accordingly, the five configurations evaluated in this study are defined by their operational characteristics rather than treated as five independent canonical techniques. **Zero-Shot Prompting** provides only the classification instruction, while **Few-Shot Demonstration Prompting** supplies labelled examples. **Expert Role-Based Prompting** assigns the model a cybersecurity-analyst role together with task-specific guidance. **Structured Criteria-Guided Prompting** provides explicit indicators for both phishing and legitimate emails together with a decision rule and constrained output. **CoT-Style Structured Reasoning Prompting** directs the model to consider several indicators sequentially before returning the final label. Because intermediate reasoning was not recorded, the latter is treated as a reasoning-oriented configuration rather than as an observed chain of thought.

Research Positioning

Existing literature therefore establishes three important points: advanced supervised models already provide strong phishing-email baselines [6]–[8]; LLMs can perform phishing and malicious-email classification [4], [5];

and prompt formulation can materially influence LLM behaviour [9], [10]. What remains less clear is how different complete prompt configurations and open-source LLMs behave when evaluated under the same dataset and execution conditions. This study addresses that narrower problem by examining prompt configuration first and model choice second, while considering not only conventional classification metrics but also valid-output coverage and local processing efficiency.

METHODOLOGY

Research Design

This study employed a two-stage experimental design, as illustrated in Figure 1. In Stage 1, five prompt-engineering configurations were evaluated using Mistral while the dataset and evaluation procedure were held constant. The strongest-performing configuration was then carried forward to Stage 2, in which five locally deployed open-source LLMs—Mistral, Phi-3, Qwen2.5, Llama 3.1, and Gemma 2—were evaluated using the same dataset and prompt. This design allowed prompt configuration to be examined first and model choice second under consistent experimental conditions.

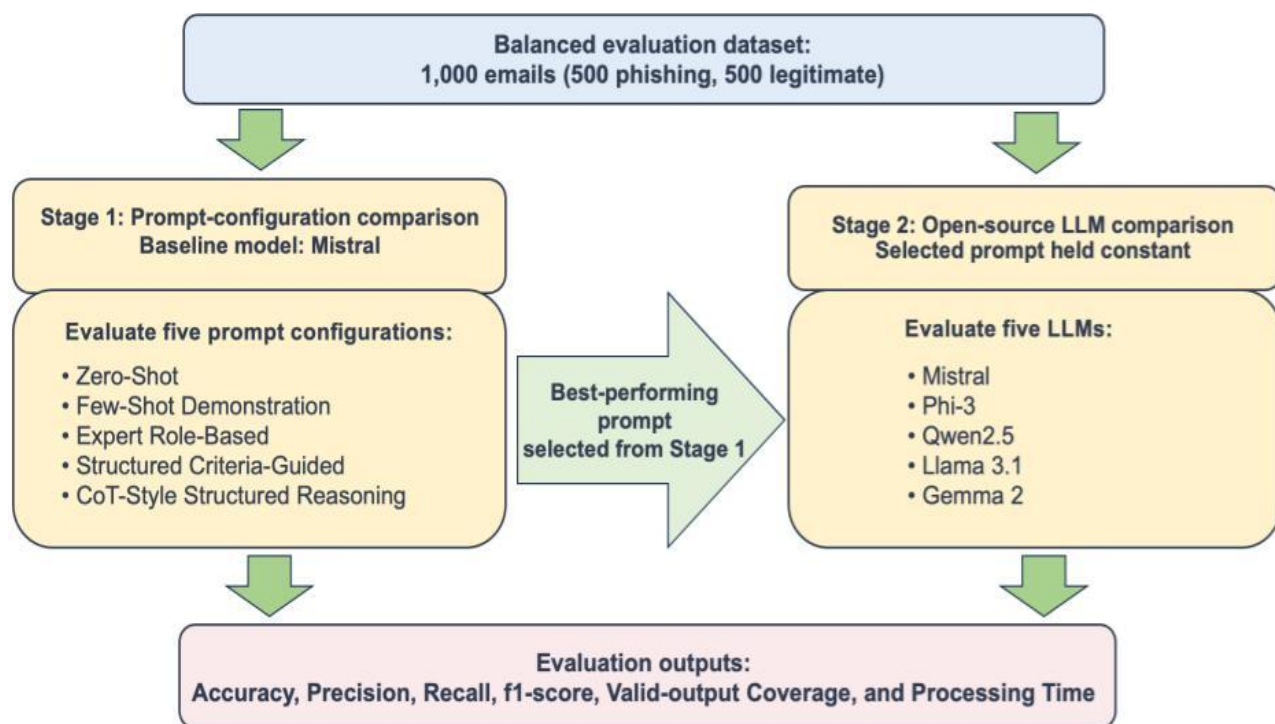


Figure 1. Two-stage experimental workflow for evaluating prompt-engineering configurations and open-source LLMs for phishing-email classification.

Dataset and Preprocessing

The experiments used the *Phishing Email Dataset* obtained from Kaggle. The consolidated source file, `phishing_email.csv`, contains 82,486 email records, comprising 42,891 phishing and 39,595 legitimate emails. The dataset combines email samples from several publicly available sources, including Enron, Ling, CEAS, Nazario, Nigerian Fraud, and SpamAssassin.

Relevant email attributes were combined into a single text representation for LLM input, and incomplete or invalid records were excluded where necessary. A balanced experimental sample of 1,000 emails was then prepared, consisting of 500 phishing and 500 legitimate messages. This same set of 1,000 labelled emails was used throughout the prompt and model comparisons. No model training or fine-tuning was performed; therefore, no training, validation, or test split was required. The same fixed 1,000-email evaluation set was used across all prompt and model conditions.

Prompt-Engineering Configurations

Five prompt-engineering configurations were evaluated in Stage 1 using Mistral and the same 1,000-email dataset. Their experimental implementations are summarized in Table 1. Each prompt required the model to classify an email as **Phishing** or **Legitimate** to support automated evaluation. The complete prompt templates used in the experiments are provided in the Appendix for reproducibility.

Table 1: Prompt-engineering configurations evaluated in Stage 1

Configuration	Experimental implementation
Zero-Shot Prompting	Direct classification instruction with no examples or additional reasoning guidance.
Few-Shot Demonstration Prompting	Included three labelled email examples before the target email.
Expert Role-Based Prompting	Assigned the model a cybersecurity-analyst role and supplied task-specific phishing cues.
Structured Criteria-Guided Prompting	Provided explicit indicators for both phishing and legitimate emails, together with a decision rule and constrained output.
CoT-Style Structured Reasoning Prompting	Directed the model to consider sender intent, urgency, requests for sensitive information, social-engineering cues, and legitimate indicators sequentially before returning the final label.

The same Mistral model, email dataset, hardware environment, and evaluation procedure were retained across the five configurations. The comparison therefore concerns the performance of the complete prompt formulations rather than isolated prompt-engineering mechanisms. The highest-performing configuration was subsequently used unchanged in Stage 2 for the comparison of the five open-source LLMs.

LLM Comparison and Local Execution

Following Stage 1, the prompt configuration achieving the highest accuracy and F1-score was selected for the model-comparison stage. Using this selected configuration, Mistral, Phi-3, Qwen2.5, Llama 3.1, and Gemma 2 were evaluated under the same experimental conditions. All models were executed locally through the Ollama framework, providing a common runtime for the comparison.

The experiments were conducted on an Acer Nitro AN515-45 laptop equipped with an AMD Ryzen 5 6600H processor with six physical cores and 12 logical processors, 8 GB RAM, NVMe solid-state storage, and Microsoft Windows 11 Home Single Language (64-bit). Python was used to read the dataset, construct prompts, communicate with the locally running LLMs, store predictions, and calculate evaluation measures.

Emails were processed sequentially, one at a time, with each classification submitted through a separate Ollama call and without retaining conversation history between emails. No custom decoding parameters were specified in the experimental script; generation therefore followed the defaults of the installed Ollama model instances.

For each email, the experimental script generated the relevant prompt, submitted it to the selected model through Ollama, and stored the returned prediction together with the corresponding ground-truth label. The same dataset, prompt configuration, evaluation procedure, and local execution environment were retained across the five-model comparison.

Evaluation Measures and Raw-Output Handling

Phishing was treated as the positive class. Classification performance was evaluated using accuracy, precision, recall, and F1-score, derived from true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) [17], [18]:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4)$$

Accuracy measures overall classification correctness, precision indicates the proportion of predicted phishing emails that were actually phishing, recall measures the proportion of phishing emails successfully detected, and F1-score summarizes the balance between precision and recall.

Although the prompts requested a binary classification of Phishing or Legitimate, the generated responses were subsequently normalized using keyword-based output parsing. A response containing the term phishing was mapped to Phishing, while a response containing the term legitimate was mapped to Legitimate. Responses containing neither term were recorded as Unknown. For performance evaluation, valid outputs were defined as responses successfully mapped to either of the two target classes. Accuracy, precision, recall, and F1-score were calculated using these valid binary predictions, while valid-output coverage was additionally reported to indicate the proportion of the 1,000 submitted emails that produced a valid mapped classification. Valid-output coverage was calculated as:

$$Coverage = \frac{N_{valid}}{N_{total}} \times 100\% \quad (5)$$

Processing time was measured as the total wall-clock duration of the sequential 1,000-email processing loop under the same local hardware and software environment. The reported values therefore represent end-to-end experimental processing time rather than isolated model inference latency. Because all models were executed on the same local hardware and software environment, processing time was used as an additional indicator of computational efficiency.

RESULTS

Stage 1: Prompt-Configuration Comparison

Table 2 shows that Structured Criteria-Guided Prompting achieved the strongest overall performance, with the highest accuracy (86.33%) and F1-score (85.43%), together with 90.91% precision. Its comparatively balanced performance across precision and recall supported its selection for the subsequent model-comparison stage.

The other configurations showed stronger trade-offs between phishing detection and false positives. Zero-Shot Prompting achieved 93.40% recall and an F1-score of 82.87%, but produced 160 false positives. CoT-Style Structured Reasoning Prompting reached 100.00% recall but generated 364 false positives, while Few-Shot Demonstration Prompting achieved 99.80% recall with 438 false positives. These high-recall configurations therefore identified nearly all phishing emails at the cost of substantially more legitimate emails being classified as phishing.

Table 2: Prompt-configuration confusion counts and performance

Prompt	TP	FP	FN	TN	Unknown	Coverage %	Accuracy %	Precision %	Recall %	F1 %
Zero-Shot Prompting	467	160	33	331	9	99.1	80.52	74.48	93.40	82.87
Few-Shot Demonstration Prompting	499	438	1	62	0	100.0	56.10	53.26	99.80	69.45
Expert Role-Based Prompting	320	68	109	385	118	88.2	79.93	82.47	74.59	78.34
Structured Criteria-Guided Prompting	390	39	94	450	27	97.3	86.33	90.91	80.58	85.43
CoT-Style Structured Reasoning Prompting	500	364	0	133	3	99.7	63.49	57.87	100.00	73.31

Valid-output coverage also varied across configurations. Expert Role-Based Prompting recorded 118 Unknown outputs, giving the lowest coverage at 88.2%, whereas Structured Criteria-Guided Prompting recorded 27 Unknown outputs and achieved 97.3% coverage. Zero-Shot, CoT-Style Structured Reasoning, and Few-Shot Prompting achieved coverage rates of 99.1%, 99.7%, and 100.0%, respectively. Under the implemented output-processing procedure, responses that could not be mapped to either target class were recorded as Unknown and excluded from the conventional classification metrics.

Stage 2: Open-Source LLM Comparison

As shown in Table 3, Gemma 2 achieved the strongest overall classification performance, with the highest accuracy (87.00%) and F1-score (87.38%), together with 90.00% recall. It correctly identified 450 phishing emails and 420 legitimate emails, with 80 false positives and 50 false negatives. Gemma 2 also produced no Unknown outputs, resulting in 100.0% valid-output coverage. Its overall performance was therefore the most balanced among the five evaluated models.

Mistral achieved an F1-score of 85.43% and precision of 90.91%, with 97.3% valid-output coverage. Qwen2.5 obtained the highest precision at 95.73% and produced only 15 false positives, indicating strong accuracy when assigning the phishing label. However, it also produced 164 false negatives, reducing recall to 67.20% and F1-score to 78.97%.

Llama 3.1 similarly showed high precision (91.94%) but lower recall (66.20%), with 169 phishing emails missed as false negatives and an F1-score of 76.98%. Phi-3 recorded the weakest overall classification performance, with 55.06% accuracy and an F1-score of 59.95%. Its error profile included 286 false positives and 163 false negatives, indicating substantially poorer separation between phishing and legitimate emails than the other evaluated models.

Table 3: Model-level confusion counts and performance using the Structured Criteria-Guided Prompt

Model	TP	FP	FN	TN	Unknown	Coverage %	Accuracy %	Precision %	Recall %	F1 %
Mistral	390	39	94	450	27	97.3	86.33	90.91	80.58	85.43

Model	TP	FP	FN	TN	Unknown	Coverage %	Accuracy %	Precision %	Recall %	F1 %
Phi-3	336	286	163	214	1	99.9	55.06	54.02	67.33	59.95
Qwen2.5	336	15	164	485	0	100.0	82.10	95.73	67.20	78.97
Llama 3.1	331	29	169	471	0	100.0	80.20	91.94	66.20	76.98
Gemma 2	450	80	50	420	0	100.0	87.00	84.91	90.00	87.38

Table 4 shows clear differences in recorded local processing time across the five models. Qwen2.5 completed the 1,000-email evaluation in the shortest time (1 h 7 min 46 s; approximately 4.07 s/email) and also achieved the highest precision (95.73%). In contrast, Gemma 2 required 2 h 22 min 31 s (8.55 s/email) but delivered the strongest overall classification profile, with the highest accuracy (87.00%), recall (90.00%), and F1-score (87.38%). Phi-3 recorded the longest processing time (2 h 26 min 51 s; 8.81 s/email) while also producing the lowest accuracy and F1-score, indicating that longer processing time did not correspond to better classification performance under the evaluated local setup.

Table 4: Local processing time for the model comparison

Model	Total time	Approx. s/email (1,000 submitted)	Speed rank
Mistral	1 h 13 min 53 s	4.43	2
Phi-3	2 h 26 min 51 s	8.81	5
Qwen2.5	1 h 7 min 46 s	4.07	1
Llama 3.1	1 h 13 min 53 s	4.43	2
Gemma 2	2 h 22 min 31 s	8.55	4

DISCUSSION

Structured Criteria-Guided Prompting achieved the best overall prompt-level performance. This configuration provided indicators for both phishing and legitimate emails, together with a decision rule and a constrained output format. The results suggest that giving the model clear criteria for both classes helped produce a more balanced classification. However, the advantage should be understood as the performance of the complete configuration tested here, rather than evidence that one prompting approach is universally superior. This is consistent with previous work showing that LLM performance can vary with prompt structure, examples, role instructions, and reasoning guidance [9]–[12], [14]–[16].

CoT-Style Structured Reasoning and Few-Shot Demonstration Prompting achieved very high recall, but this came with many false positives. Both configurations detected almost all phishing emails, yet they also classified many legitimate emails as phishing. This reflects a clear precision–recall trade-off. In practice, such behaviour may reduce missed phishing attempts but could also increase false alarms and analyst workload. The results therefore show the behaviour of these specific prompt configurations rather than indicating that CoT-style or Few-Shot prompting is generally unsuitable. The operational cost of these errors also depends on deployment. In systems where phishing predictions trigger automatic blocking or quarantine, false positives may disrupt legitimate communication, whereas false negatives leave malicious emails undetected. The preferred precision–recall balance therefore depends on how the detector is used.

The model comparison also showed different error patterns. Gemma 2 provided the strongest overall balance, with the highest accuracy, recall, and F1-score. Qwen2.5 and Llama 3.1 achieved high precision but lower recall, meaning that they produced relatively few false positives but missed more phishing emails. Phi-3 performed less

effectively in both areas, producing comparatively high numbers of false positives and false negatives. These differences show that model choice affects not only overall performance but also the types of errors produced.

Valid-output coverage provided another useful measure of reliability. Expert Role-Based Prompting produced 118 Unknown outputs and the lowest coverage at 88.2%. This indicates that classification performance alone does not fully describe practical reliability. A model or prompt configuration must also produce outputs that can be consistently mapped to the required classes. In a deployed system, frequent unresolved outputs would require additional handling even when the valid predictions themselves perform reasonably well.

The findings are generally consistent with earlier research showing that contextual language modelling can support phishing and malicious-email detection [4]–[6], [8]. They also support earlier observations that prompt formulation can influence LLM classification behaviour [9], [10]. Technical detection also forms only one layer of phishing defence. Firdousi et al. [19] showed through simulated phishing exercises that user susceptibility can persist even after cybersecurity-awareness training, reinforcing the value of combining technical detection with human-centred security measures. The results also show practical trade-offs between the evaluated models. Gemma 2 may be preferable when reducing missed phishing emails is a higher priority because it achieved the strongest recall and overall F1-score. Qwen2.5 may be more suitable when reducing false alarms and shortening local processing time are more important, since it achieved the highest precision and the shortest recorded processing time. These choices reflect the trade-offs observed under the evaluated conditions rather than general recommendations for all deployment settings.

Limitations

Two limitations mainly concern the scope of the evaluation. First, the experiments used a balanced subset of 1,000 emails drawn from a single phishing-email dataset. The balanced design provided equal representation of phishing and legitimate messages and supported consistent comparison across prompt configurations and models, but it does not reflect the class distribution or diversity of all real-world email environments. Second, the model comparison was limited to five locally deployed open-source LLMs evaluated on a single hardware environment. This provided a consistent basis for comparing classification performance and processing time, but the observed results, particularly processing efficiency, should be interpreted within that deployment setting rather than generalized to all models or hardware platforms.

A further consideration concerns the use of a common prompt in Stage 2. The prompt selected using Mistral was deliberately held constant across all five models to support a controlled cross-model comparison. However, this design does not establish that the selected configuration is individually optimal for Phi-3, Qwen2.5, Llama 3.1, or Gemma 2.

Future Work

Future work could extend the evaluation to additional phishing-email datasets, including naturally imbalanced and multilingual collections, to examine whether the observed prompt and model behaviours remain consistent across different data distributions. A broader range of open-source LLMs and hardware configurations could also be evaluated, together with repeated runs to examine performance stability. Future implementations could additionally retain complete model responses and apply stricter structured-output mechanisms, allowing more detailed analysis of response behaviour and output-format reliability. Future evaluations could therefore compare two complementary settings: a common prompt across models for controlled comparison and model-specific prompt optimization to estimate the best achievable performance of each model.

CONCLUSION

The evaluation followed a two-stage design to examine the effects of prompt configuration and model choice on phishing-email classification. Five complete prompt configurations were first compared using Mistral. The strongest configuration was then held constant across five locally deployed open-source LLMs to enable a consistent model comparison.

Structured Criteria-Guided Prompting achieved the highest prompt-level accuracy and F1-score. In the model comparison, Gemma 2 achieved the highest accuracy, recall, and F1-score, while Qwen2.5 achieved the highest precision and the shortest recorded local processing time. Valid-output coverage also differed across conditions, with several configurations producing outputs recorded as Unknown, particularly Expert Role-Based Prompting.

Overall, the results show that prompt configuration and model choice should be considered together when developing LLM-based phishing-email detection. Different combinations produced different trade-offs in missed phishing emails, false alarms, valid-output coverage, and processing efficiency. A suitable deployment strategy therefore depends not only on overall classification performance, but also on which types of errors and operational costs are most important in the intended security setting.

Conflict Of Interest

The authors declare that they have no conflicts of interest related to this study.

Data Availability Statement

The source dataset used in this study, the Phishing Email Dataset, is publicly available on Kaggle. The experiments used the consolidated `phishing_email.csv` file from this dataset, from which a balanced evaluation subset of 1,000 emails was prepared. The processed evaluation dataset, experimental result files, and supporting source code are available from the corresponding author upon reasonable request.

REFERENCES

1. Alkhalil, Z., Hewage, C., Nawaf, L., & Khan, I. (2021). Phishing attacks: A recent comprehensive study and a new anatomy. *Frontiers in Computer Science*, 3, 563060. <https://doi.org/10.3389/fcomp.2021.563060>
2. Kavva, S., & Sumathi, D. (2025). Staying ahead of phishers: A review of recent advances and emerging methodologies in phishing detection. *Artificial Intelligence Review*, 58. <https://doi.org/10.1007/s10462-024-11055-z>
3. Salloum, S., Gaber, T., Vadera, S., & Shaalan, K. (2022). A systematic literature review on phishing email detection using natural language processing techniques. *IEEE Access*, 10, 65703–65727. <https://doi.org/10.1109/ACCESS.2022.3183083>
4. Koide, T., Fukushi, N., Nakano, H., & Chiba, D. (2024). ChatSpamDetector: Leveraging large language models for effective phishing email detection. In *International Conference on Security and Privacy in Communication Systems* (pp. 297–319). Springer Nature Switzerland.
5. Jamal, S., Wimmer, H., & Sarker, I. H. (2024). An improved transformer-based model for detecting phishing, spam and ham emails: A large language model approach. *Security and Privacy*, 7(5), e402. <https://doi.org/10.1002/spy2.402>
6. Kyaw, P. H., Gutierrez, J., & Ghobakhlou, A. (2024). A systematic review of deep learning techniques for phishing email detection. *Electronics*, 13(19), 3823. <https://doi.org/10.3390/electronics13193823>
7. Altwaijry, N., Al-Turaiki, I., Alotaibi, R., & Alakeel, F. (2024). Advancing phishing email detection: A comparative study of deep learning models. *Sensors*, 24(7), 2077. <https://doi.org/10.3390/s24072077>
8. Meléndez, R., Ptaszynski, M., & Masui, F. (2024). Comparative investigation of traditional machine-learning models and transformer models for phishing email detection. *Electronics*, 13(24), 4877. <https://doi.org/10.3390/electronics13244877>
9. Trad, F., & Chehab, A. (2024). Prompt engineering or fine-tuning? A case study on phishing detection with large language models. *Machine Learning and Knowledge Extraction*, 6(1), 367–384. <https://doi.org/10.3390/make6010018>
10. Sahoo, P., Singh, A. K., Saha, S., Jain, V., Mondal, S., & Chadha, A. (2024). A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv*. <https://arxiv.org/abs/2402.07927>
11. Brown, T. B., Mann, B., Ryder, N., et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 1877–1901.

- https://proceedings.neurips.cc/paper_files/paper/2020/hash/1457c0d6bfc4967418bfb8ac142f64a-Abstract.html
12. White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., et al. (2023). A prompt pattern catalog to enhance prompt engineering with ChatGPT. arXiv:2302.11382. <https://arxiv.org/abs/2302.11382>
 13. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35, 24824–24837. https://proceedings.neurips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract.html
 14. Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2022). Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems*, 35. https://proceedings.neurips.cc/paper_files/paper/2022/hash/8bb0d291acd4acf06ef112099c16f326-Abstract-Conference.html
 15. Naser, M. Z. (2025). A review of prompt engineering techniques for large language models. *International Journal of Human–Computer Interaction*, 1–35.
 16. Liu, Y. Y., Zheng, Z., Zhang, F., Feng, J. C., Fu, Y. Y., Zhai, J. D., et al. (2026). A comprehensive taxonomy of prompt engineering techniques for large language models. *Frontiers of Computer Science*, 20(3), 2003601.
 17. Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4), 427–437. <https://doi.org/10.1016/j.ipm.2009.03.002>
 18. Powers, D. M. W. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *Journal of Machine Learning Technologies*, 2(1), 37–63. <https://arxiv.org/abs/2010.16061>
 19. Firdousi, A. R., Nadi, F., Daud, P., & Ismail, N. A. (2025). Raising Cybersecurity Awareness Among Departmental Employees: Implementation of Trend Micro’s Phish Insight Tool. *Proceedings of the Future Technologies Conference*. Springer. https://doi.org/10.1007/978-3-032-07989-3_18

APPENDIX

Complete Prompt Templates

The five prompt templates used in Stage 1 are reproduced below. The placeholder [EMAIL TEXT] represents the target email content inserted into the prompt at runtime. The templates are reproduced as implemented in the experimental source code. The Structured Criteria-Guided prompt was subsequently used unchanged for the Stage 2 model comparison.

Prompt 1: Zero-Shot Prompting

Classify the following email as either Phishing or Legitimate.

Email:

[EMAIL TEXT]

Return only one word:

Phishing or Legitimate

Prompt 2: Few-Shot Demonstration Prompting

You are an email classification system.

Classify emails as Phishing or Legitimate.

Example 1:

Email:

Your account has been suspended. Click the link below and verify your password immediately.

Classification:

Phishing

Example 2:

Email:

Hello team, please find attached the meeting notes from today's discussion.

Classification:

Legitimate

Example 3:

Email:

Congratulations, you won a prize. Provide your banking details to claim.

Classification:

Phishing

Now classify this email:

Email:

[EMAIL TEXT]

Return only one word:

Phishing

or

Legitimate

Prompt 3: Expert Role-Based Prompting

You are a cybersecurity analyst specializing in email classification.

Analyze whether the email is phishing or legitimate.

Consider:

- normal business communication patterns
- suspicious urgency
- requests for credentials

- impersonation attempts
- malicious intent

Do not assume every suspicious email is phishing.

Classify based only on the evidence.

Email:

[EMAIL TEXT]

Return only one word:

Phishing or Legitimate

Prompt 4: Structured Criteria-Guided Prompting

You are an email security classification system.

Your task is to determine whether an email is Phishing or Legitimate.

Analyze BOTH phishing indicators and legitimate email indicators.

Consider phishing indicators:

- Requests for passwords or sensitive information
- Fake login or verification requests
- Threats, urgency, or pressure tactics
- Impersonation attempts
- Suspicious instructions

Consider legitimate indicators:

- Normal business communication
- Regular updates or notifications
- Work-related discussions
- No request for confidential information
- No harmful intention

Do not assume an email is phishing only because it looks unusual.

Classify based on the overall evidence.

Email:

[EMAIL TEXT]

Return only one word:

Phishing or Legitimate

Prompt 5: CoT-Style Structured Reasoning Prompting

You are a cybersecurity email analysis system.

Analyze the email step by step internally by considering:

1. Sender intention
2. Suspicious language or urgency
3. Requests for sensitive information
4. Social engineering techniques
5. Legitimate communication indicators

After completing the analysis, provide only the final classification.

Email:

[EMAIL TEXT]

Your response must be exactly one word:

Phishing or Legitimate