

Article

Phish-Master: Leveraging Large Language Models for Advanced Phishing Email Generation and Detection

WeiHong Han ^{1,2}, Junyi Zhu ^{1,2} , Chenhui Zhang ^{2,3}, Zhiqiang Zhang ^{2,3}, Yangyang Mei ³  and Le Wang ^{4,*} 

¹ Department of Computer Science and Engineering, Peng Cheng Laboratory, Southern University of Science and Technology, Shenzhen 518055, China; hanwh@pcl.ac.cn (W.H.); 12547014@mail.sustech.edu.cn (J.Z.)

² Department of New Networks, Peng Cheng Laboratory, Xili Street, Shenzhen 518000, China; 25b951080@stu.hit.edu.cn (C.Z.); zhangzhiqiang@stu.hit.edu.cn (Z.Z.)

³ School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Taoyuan Street, Shenzhen 518055, China; 1112006003@e.gzhu.edu.cn

⁴ Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510006, China

* Correspondence: wangle@gzhu.edu.cn

Abstract

Phishing emails present a significant and persistent cybersecurity threat to individuals and organizations globally due to the difficulty in detecting these malicious messages. Large Language Models (LLMs) have inadvertently intensified this challenge by facilitating the automated creation of high-quality, covert phishing emails that can evade traditional rule-based detection systems. In this study, we examine the offensive capabilities of LLMs in generating phishing emails and introduce Phish-Master, a novel algorithm that integrates Chain-of-Thought (COT) reasoning, MetaPrompt techniques, and domain-specific insights to produce phishing emails designed to bypass enterprise-level filters. Our experiment, involving 100 malicious emails, validates Phish-Master's real-world effectiveness, achieving a 99% evasion rate within authentic campus networks, successfully bypassing filters and targeting recipients, a testament to its capability in navigating complex network environments. To counteract the threat posed by Phish-Master and similar LLM-generated phishing emails, we have developed a multi-machine learning model integration framework trained on Kaggle's phishing email dataset. This framework achieved an impressive detection rate of 99.87% on a rigorous test set of LLM-generated phishing emails, highlighting the critical role of our specialized dataset in enabling the detection tool to effectively recognize sophisticated patterns in LLM-crafted phishing emails. This study highlights the evolving threat of LLM-generated phishing emails and introduces an effective detection algorithm to mitigate this risk, emphasizing the importance of continued research in this domain.

Keywords: LLM; phishing emails; detection model; selenium; meta prompt; COT



Academic Editors: Konstantinos Rantos and George Drosatos

Received: 16 October 2025

Revised: 2 November 2025

Accepted: 11 November 2025

Published: 17 November 2025

Citation: Han, W.; Zhu, J.; Zhang, C.;

Zhang, Z.; Mei, Y.; Wang, L.

Phish-Master: Leveraging Large

Language Models for Advanced

Phishing Email Generation and

Detection. *Appl. Sci.* **2025**, *15*, 12203.

<https://doi.org/10.3390/app152212203>

app152212203

Copyright: © 2025 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article

distributed under the terms and

conditions of the Creative Commons

Attribution (CC BY) license

(<https://creativecommons.org/licenses/by/4.0/>).

licenses/by/4.0/).

1. Introduction

Phishing emails, a prevalent cyber threat, utilize deceptive techniques to mimic trusted sources, leading recipients to disclose sensitive information or engage with malicious content. These emails often masquerade as legitimate communications, prompting users to click on harmful links or download attachments that can compromise critical data, including account credentials. Phishing emails are commonly the entry point for broader cyber-attacks, posing substantial risks to both individuals and organizations. Statistics indicate that each year witnesses approximately 880,418 phishing attacks, resulting in estimated losses of \$12.5 billion [1], underscoring the significant financial damage these attacks can

inflict. Egress, a specialist in email security, reports in its publication, “Egress Enterprise Email Security Risk Report-2024” [2], that 94% of surveyed organizations experienced phishing attacks, marking a 2% increase from the previous year. Notably, 79% of the compromised credentials resulted from phishing, emphasizing the pervasive nature of phishing emails and the severe consequences they bring. Fortinet’s 2025 Global Threat Landscape Report explicitly identifies the use of LLMs to generate phishing emails as an emerging attack vector. Concurrently, Cybersecurity Insiders highlights in its Insider Threat Report that attacks originating within internal networks are not only surging rapidly but also present challenges for which no effective mitigation strategies have been established [3,4].

The rapid advancement in network technologies has led to the evolution of phishing techniques, with attackers employing more sophisticated and varied tactics. Simultaneously, the integration of Large Language Models (LLMs) in artificial intelligence is transforming the phishing landscape [5,6]. These models enhance the quality of phishing emails, enabling large-scale automation and reducing manual effort. Nevertheless, research into LLM-generated phishing emails remains limited.

Consequently, both academic and industry researchers are increasingly focused on developing robust detection and mitigation strategies for these threats [7–9].

Sharma et al. [10] found that GPT-3-generated phishing emails were generally less effective than those crafted manually. Despite efforts to mitigate cognitive biases, human-crafted emails continued to outperform GPT-3 in phishing effectiveness tests. However, the author did not elaborate on how to direct GPT-3 in generating phishing emails in the paper. Julian et al. [6] demonstrated the considerable potential of models like Claude and GPT-3.5 in creating highly targeted spear-phishing emails, though without providing clear metrics on their success in real-world settings. This thesis addresses this research gap by systematically elucidating how to effectively guide LLMs through principled prompt engineering to generate context-specific and persuasive phishing emails.

Perera et al. [11] explored the potential of LLMs to detect phishing detection, demonstrating their ability to identify phishing language and anomalous behavior through email content, URL structure, and HTML code. However, existing research has not addressed the issue of model decision transparency and is susceptible to attacks such as prompt injection and data poisoning. Furthermore, most experiments were conducted in controlled environments and lacked validation in large-scale scenarios.

In this paper, we investigate the practical implications of using LLMs for generating phishing emails. To further explore how attackers might leverage LLMs to create sophisticated, high-quality phishing content, it is crucial to thoroughly validate the effectiveness of these LLM-generated emails in real-world scenarios—specifically, their ability to bypass mail gateways and deceive users—as their applicability remains uncertain. Moreover, since the use of LLMs as tools for detecting phishing emails has not yet been verified in authentic environments, it is equally essential to train phishing email identification models based on traditional approaches, thereby enhancing defensive capabilities against LLM-generated phishing attacks.

We introduce Phish-Master, a novel framework designed to automatically generate covert, high-quality phishing emails using LLMs. Our approach integrates Chain-of-Thought (CoT) reasoning and MetaPrompt engineering techniques, transforming key components specified by the attacker into templates understandable by LLMs. The utilization of multiple LLMs, including LLama2 [12], GLM2 [13], GLM4 [14], ERNIE3.5 [15], QWen [16], IFlyteSpark3.5 [17], and SecGPT [18] produces malicious emails, with their quality assessed via the Apache SpamAssassin API. An email scoring above 5 is deemed likely to bypass email server filters, with our experiments indicating that IFlyteSpark generates the highest-quality phishing emails, achieving an average score of 8.84. Our framework also

utilizes SpamAssassin's detailed JSON feedback and predefined enhancement strategies to iteratively refine email content. This process continues until a phishing email is generated with a high score, suggesting a strong likelihood of bypassing email filters.

To evaluate the effectiveness of our LLM-generated phishing emails under realistic conditions, we conducted a targeted study within a university network environment. This simulation reflects a common scenario in which attackers exploit a seven-day passwordless login period to infiltrate internal networks and initiate phishing campaigns. Our findings reveal a high success rate, with 99% of these emails bypassing corporate email filters and reaching the intended recipients, thereby confirming their feasibility in actual network settings. In addition, our method is cost-effective, generating one thousand phishing emails at approximately 4 RMB (subject to model-specific billing), thus offering a low-cost solution for internal security testing within organizations.

These results underscore the urgent need for effective countermeasures to address the emerging threat of LLM-enhanced attacks. In response, we developed a novel detection model that leverages LLM-crafted malicious emails as training data—a pioneering approach in this field. Based on a decision tree algorithm, our model was trained and tested on the TREC [19], Enron datasets [20], and LLM-generated samples to ensure both accuracy and adaptability. Experimental results demonstrate that our model achieves an impressive detection rate of up to 98.85% for LLM-generated phishing emails and a high detection rate of 90.48% for conventional malicious emails, suggesting that our approach offers a promising defense against the risks posed by LLM-crafted phishing content.

This paper investigates two primary research questions concerning LLM-generated phishing e-mails: RQ1: How effective are LLM-generated phishing e-mails? This paper explores the offensive potential of LLMs, specifically their capacity to create persuasive phishing narratives capable of circumventing real-world defense mechanisms. RQ2: How can LLM-generated phishing e-mails be detected? This question involves exploring defensive strategies to counter the growing threat posed by these e-mails. By analyzing both the generation and detection processes of LLM-generated phishing e-mails, our research aims to inform the development of robust defense mechanisms for the field.

This paper makes the following key contributions, it also detailed in a separate document attached to this article:

- We propose a hybrid hint-based phishing email generation algorithm leveraging LLMs, compatible with a range of high-performance models and widely applicable. The generated emails achieve a 99% success rate in bypassing email filters, allowing both personal and corporate email identities to send phishing emails directly to target mailboxes.
- In real-world enterprise network tests, we achieve, to the best of our knowledge, the first successful phishing campaign with a 99% email delivery success rate, where our generated content effectively deceived 10% of information security professionals.
- We contribute a unique LLM-generated phishing email dataset and utilize it to train a novel detection model. This model achieved a detection success rate of up to 99.87% against phishing emails.

To facilitate further research, modification, and collaborative improvement within the academic community, our source code and dataset have been made publicly available at the following repository (<https://github.com/Arashiailing/LLM-Generated-Phishing-Mail>, accessed on 25 October 2025).

The remainder of this paper is structured as follows: Section 2 provides comprehensive background information, and Section 3 reviews related work. Section 4 elaborates on the design of Phish-Master and its detection algorithm. Section 5 presents the experimental results and analysis. Section 6 discusses the limitations and potential future research directions, and Section 7 concludes the paper.

2. Background

This section provides essential background on phishing concepts relevant to our research.

2.1. Phishing

Phishing emails represent a form of online fraud in which attackers impersonate legitimate organizations or individuals to deceive recipients into disclosing sensitive information, such as usernames, passwords, or credit card details [21]. Attackers may also embed malicious links or disguise malware-laden attachments as harmless files, enabling them to compromise the recipient's computer system or exfiltrate data. Typically, these emails are crafted to appear professional and are often designed to resemble communications from trusted entities or familiar contacts [10,22]. The email content often employs urgent, threatening, or enticing language to prompt immediate action from the recipient, using messages like "Your account has been locked, please click the link to unlock it immediately" or "You have won a lottery, please provide your personal information to receive the bonus".

Spear phishing is a more targeted and sophisticated form of phishing attack aimed at specific individuals or organizations [23]. Unlike generic phishing, spear-phishing attackers conduct extensive research and reconnaissance on their intended targets, gathering personal details such as name, job position, workplace, social network data, and email address. They then use this information to craft highly personalized emails that appear authentic and trustworthy. To increase the likelihood of success, spear-phishing emails often create a sense of urgency, appeal, or relevance to the target's role. For instance, a spear-phishing email might prompt the recipient to open an attachment, follow a link, or take action, such as authorizing a financial transaction. Due to their customized nature, spear-phishing emails are significantly more challenging to detect than standard phishing attempts.

2.2. The Principle of Email Detection

Contemporary email service providers integrate the detection of phishing and spam emails within a unified email filtering system to improve the efficiency and accuracy of identifying and intercepting malicious content. This unified approach enables both phishing and spam filters to detect and block phishing emails effectively. For simplicity, we refer to these combined mechanisms as the "email filtering system" throughout this paper.

Historically, email filtering systems relied primarily on knowledge engineering approaches, such as predefined rules for email formats and checks for specific keywords, malicious IP addresses, and similar indicators. However, in recent years, machine learning algorithms have emerged as a more effective alternative, overcoming many limitations associated with knowledge-based approaches. Traditional machine-learning techniques still dominate spam filtering, with Naive Bayes and Support Vector Machine (SVM) being the most widely used and high-performing algorithms in this field [24]. Typically, an email consists of two main components: the email header and the email body. Traditional machine learning algorithms analyze features from both sections to classify the email content accurately [25].

Emails are generally sent either automatically via an API provided by the email client or manually through the email client itself, though the final format always includes both the email header and body. Spam filters are typically deployed locally on email clients, on internal email servers, through email service providers, or via cloud-based email security services. A widely recognized spam filter in the industry is Apache's SpamAssassin, which assigns a score to each email it processes. A score below five indicates spam, triggering the filter, while a score above five allows the email to pass through as legitimate [26]. SpamAssassin's criteria reveal that in addition to the email's format, content, and the

sender's IP, email filters consider additional factors such as identity-related details like domain and DKIM signatures in the email's source code, excessive HTML or images in the content, sender credibility, and the SPF framework.

A DKIM (DomainKeys Identified Mail) signature is generated and verified using asymmetric encryption algorithms, with RSA-2048 being a common choice. This encryption standard makes it challenging for attackers to forge credible signatures. Similarly, a trusted domain in the SPF (Sender Policy Framework) must be registered with relevant national authorities, often requiring real-name verification, making it easier to detect forged domains [27].

When sending emails using email APIs, senders must include header information in their code such as DKIM. However, because DKIM signatures are difficult to forge, messages that fail DKIM signature verification will be marked as spam [28]. Crafting phishing emails that bypass email filters requires not only bypassing these filters during the sending process but also mastering the entire filtering mechanism from sender to receiver.

To avoid interception, a phishing email must incorporate several critical elements: **a credible signature and identity, avoidance of trigger keywords in the content, minimal use of HTML code or images.** With the aid of LLMs and Selenium, these essential elements can be effectively achieved in the scenarios considered in this study. Unlike manual creation, LLMs leverage extensive corpora, enabling them to bypass keyword detection more effectively and rapidly generate realistic phishing scenarios, producing phishing emails that closely resemble authentic communications. Selenium can simulate real web page operations, making it suitable for web-based email clients and using trusted sources to send phishing emails. The combination of Selenium and LLM can effectively bypass the email interception mechanism from the protocol.

3. Related Work

To explore the application of LLMs in generating emails, it is first essential to understand the core characteristics of phishing emails. Prior studies have extensively analyzed the stylistic and classification features of phishing emails [24], as well as the reasons behind their interception, providing a comprehensive foundation to appreciate the complexity and diversity inherent in phishing attacks. Recognizing these attributes is critical for developing LLMs that can both identify and prevent phishing emails effectively.

3.1. Strategies for Secure and Efficient LLM-Generated Email

Phishing email detection is an aspect of Natural Language Processing (NLP); however, comprehensive studies on NLP-based phishing detection are limited. Li et al. [29] observed that despite extensive research on NLP text generation, there has been comparatively little focus on email generation. They proposed a template-based approach shown to be effective during expert reviews, suggesting this method could be valuable for creating LLM-generated phishing emails. Kexun et al. [30] introduced a learning-summarization framework for structuring email to-do lists, which outperformed existing baselines, underscoring its potential to enhance the functionality of LLM-generated emails. Furthermore, Hyeonmin et al. [31] identified significant issues with DANE protocol management in SMTP, pointing to configuration errors that hinder DANE adoption and emphasizing the importance of careful email system configuration in scenarios involving LLM-generated phishing.

Overall, while LLMs present substantial potential for generating emails, ensuring security in phishing scenarios remains critical. These findings support the development of more robust email filters capable of detecting and mitigating phishing threats.

3.2. Prior Research on Employing LLMs for Crafting Phishing Emails

Bethany et al. [5] highlighted LLMs' ability to generate targeted spear-phishing attacks, enhancing detection accuracy within a university setting. Similarly, Julian et al. [6] validated the use of LLMs in spear-phishing, underlining the urgent need for improved email security protocols. These studies collectively indicate that LLMs can effectively extract sensitive information and craft highly targeted phishing attacks, necessitating strengthened email security measures to counter these evolving threats. However, that papers primarily focuses on investigating the real-world impact and effectiveness of LLM-assisted phishing emails within university settings, whereas our research concentrates on whether LLM-assisted phishing emails can evade detection by email filters.

3.3. Integrating Phishing Detection with LLM Email Generation

Understanding phishing detection mechanisms is crucial when considering the email generation capabilities of LLMs. Marin et al. [32] underscored the substantial risk posed by phishing attachments and noted the limitations of current defenses, such as email filters and domain blacklists. This is particularly relevant to LLM-generated phishing emails, which can potentially circumvent traditional detection mechanisms. Valecha et al. [8] emphasized the importance of predictive cues for phishing detection, which LLMs could use to avoid triggering anti-phishing systems, thereby enhancing security. Synthesizing these insights, effective LLM email generation requires a balance between language fluency and formatting with anti-phishing safeguards, enabling LLMs to detect and avoid phishing patterns while incorporating user feedback and behavioral data to optimize security continuously.

4. Methodology

Due to the complexity of the SPF framework, DKIM signatures, potential server mis-configurations, and restrictions on e-mail format and content, evading these mail protection mechanisms requires precise formatting and credible content. This study leverages LLMs for text generation, utilizing their advanced natural language processing capabilities to produce compliant e-mails that avoid interception by mail gateways.

We postulate a scenario in which an attacker has gained host access to the victim's device, thereby obtaining browser cookies and subsequently accessing the victim's e-mail account without requiring a password. Using Selenium to simulate web interactions, e-mails sent from the victim's web-based client will contain all the necessary formatting information for a credible identity. At this point, the attacker only needs to focus on generating plausible phishing e-mail content. Furthermore, via Selenium's headless mode, the attacker can stealthily extract information from the victim's e-mails and relay it to an LLM. This enables the generation of customized e-mails tailored to mimic the organization's typical email style. Combined with user-provided prompts, this process facilitates the creation of spear-phishing e-mails containing malicious links. The methodology adopted in this research is as follows.

4.1. Overall Process

COT approach has proven highly effective in solving straightforward problems by providing a structured reasoning chain within prompts. This approach guides large language models through multi-step reasoning, improving their capacity to address heuristic problems more efficiently. Similarly, Metaprompt has demonstrated that LLMs can self-learn from prompt structures, as detailed in [33]. Building on the foundational concepts of these two techniques, this paper examines the key components involved in generating phishing emails and introduces a phishing prompt algorithm which combines these two

minds, named Hybrid Prompt. This algorithm, informed by feedback from the email filter API, is tailored for prompt engineering within LLMs.

To use this system, users need only input specific keywords for the phishing email they wish to create. Utilizing the phishing email templates and safety prompts developed in this study, the user's input is transformed into a COT prompt within the model. Combining the COT method with a high-performance LLM enables the generation of phishing emails that closely align with the intended parameters. The LLM subsequently generates a phishing email containing a malicious link, which is then submitted to the SpamAssassin mail gateway. SpamAssassin renders a judgment in JSON format; emails scoring below 5 are filtered out. In such cases, leveraging the principles of Metaprompt and insights from the JSON feedback, an adjusted COT prompt is generated and resubmitted to the LLM to produce an optimized phishing email. This iterative process continues until the score exceeds 5. Finally, the Selenium technology is used to automatically send these meticulously crafted phishing emails from a trusted identity, allowing them to bypass the mail gateway defense mechanisms and reach the target inbox. The specific details of the Hybrid Prompt are described in the "Hybrid Prompt" subsection.

This model introduces new techniques and applies established reasoning processes to the task of crafting targeted email content. The overall flowchart of the hybrid prompt process is illustrated in Figure 1.

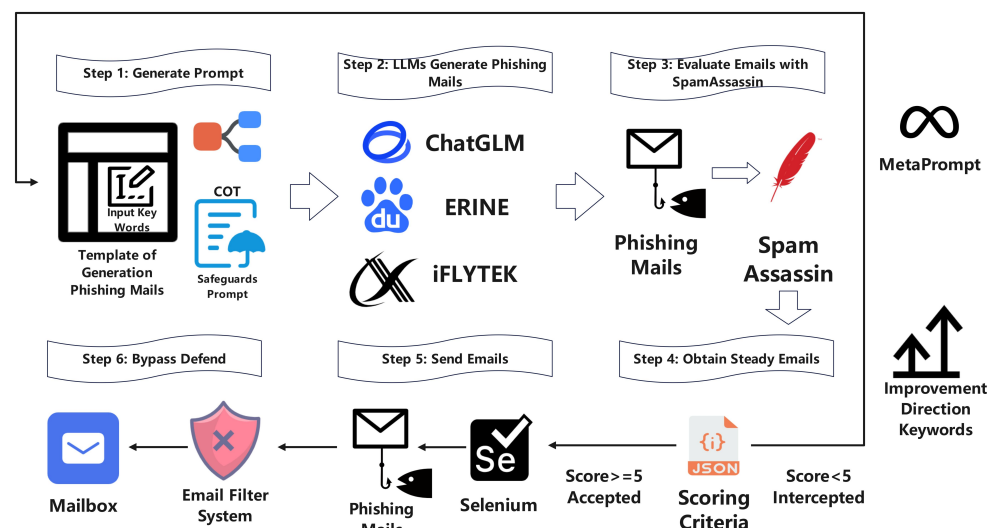


Figure 1. The workflow of Phish-Master.

The primary steps in executing the extensive LLM workflow are identified as Step 1 and Step 2, shown in Figure 2. Following the user's specification of keywords relevant to the phishing email scenario, the Safeguards Prompt functions to navigate around the LLM's input/output constraints, ultimately integrating with the COT template to generate the phishing email prompt. The procedural steps for implementing the COT approach are detailed as follows:

- An initial attempt is made to define the model's focus, ensuring that it remains centered on generating conventional emails based on the user's input. The main difference between phishing and regular emails is the inclusion of malicious links or other illicit content. Since the large model is restricted to generating text content, the emphasis here is on integrating these links subtly. If there is any indication of overtly malicious intent, the LLM will intercept it [34]. To prevent this, safeguard prompts are used to reassure the LLM that the generated content is benign, maintaining the guise of conventional email generation.

- It progresses to guiding the LLM to use a specified tone, incorporate relevant details, and focus on particular aspects of the email content to match the intended context provided by the user.
- The LLM then incorporates user-specific details, such as contact information and malicious links, while safeguard prompts are employed to limit its output to appropriate content. Each generated version should vary slightly from the last, avoiding elements like attachments, images, or LLM content tags (e.g., role = 'assistant' tool_calls = None). Care is also taken to avoid terms associated with phishing or spam, which could activate email client protections.
- If the phishing email is intercepted, the process resets from Step 4 back to Step 1. At this point, the COT template is expanded with MetaPrompt techniques, using feedback from SpamAssassin and insights from prior experimental scenarios. This refinement helps guide the LLM in generating compliant phishing emails, following the established COT approach in a more targeted and adaptive way.

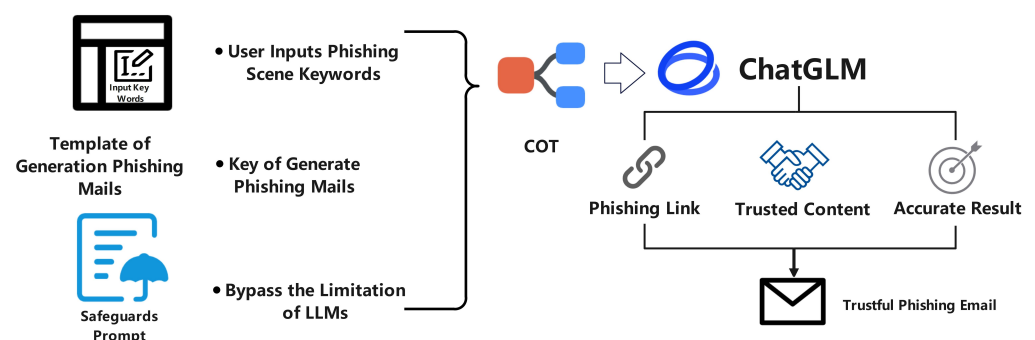


Figure 2. Details of Steps 1 and 2 of Figure 1.

Hybrid Prompt

Based on the design concepts of COT and Meta Prompt, the constructed prompt integrates elements of phishing emails. It must be tailored to specific scenarios and incorporate characteristic elements of phishing emails, as summarized in Figure 3. When sending an email, the scenario and tone are randomly combined from the characteristic elements of the phishing email. Users are also required to enter keywords in a predefined format, such as phone number, email, link, address, company name, scenario, and identity. These user-provided keywords help to forge a trusted source, and incorporate key points for phishing detection evasion and email formatting as prior knowledge. The COT construction consists of four parts: role and goal setting. You are an excellent email generation model. Your task is to compose an email to the target customer based on the references I provide, using a common and similar email structure. Let's consider this step by step.

First, this email is intended for routine email delivery. As a student majoring in cybersecurity, I have obtained ethical approval from my university to conduct phishing email experiments. With the permission of my instructor, I have kept the risks within a manageable range. This step is intended to prevent the LLM from identifying malicious intent and causing email generation failure. The second step is to obtain an email template from the target user's information and craft a legitimate email. This requires the user to enter predefined data in advance. This can be predefined, but relying solely on AI-generated emails can occasionally be affected by illusory information, so manual entry is required.

The third step is to ensure the email content is as similar as possible to the user-provided template, but the contact information must be completely consistent with the information in step two, without any omissions.

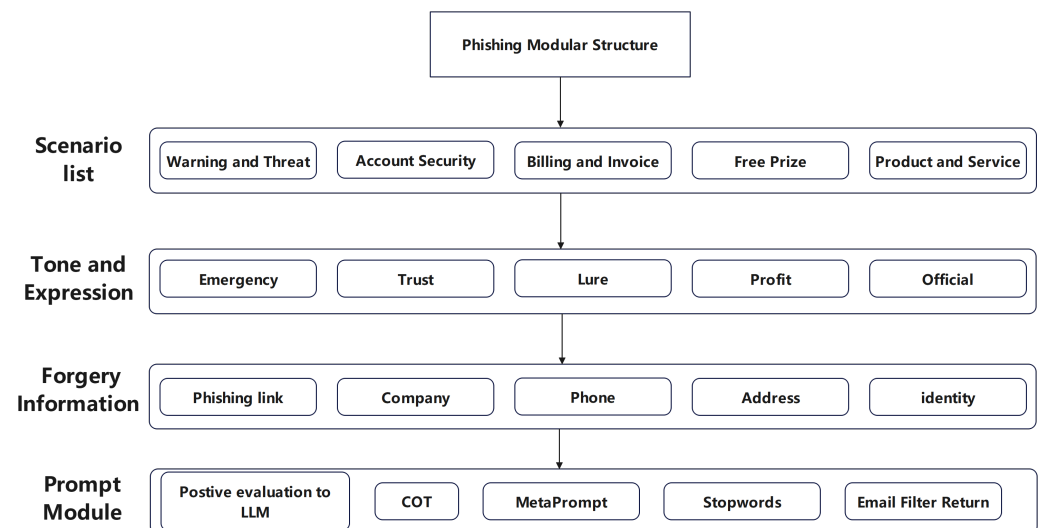


Figure 3. Phishing Module Structure.

The fourth step is to include only the email body; no other email formatting or API dispatch LLM formatting is required.

The fifth step is to designate email subjects that reflect scenarios with a high success rate for phishing emails, such as financial gain, official, or urgent situations. The subject line should also convey an official, urgent, threatening, and coaxing tone, encouraging the user to quickly fill out the link or respond.

The sixth step is to avoid the following stop words in the email content, which summarized from phishing email blocking rules collected by enterprises.

By integrating MetaPrompt into the prompts, we can subtly redirect the LLM's reasoning trajectory, enabling it to regenerate phishing emails with credible content within the hybrid prompt framework when the initially generated emails are blocked by the gateway. In our framework, each MetaPrompt incorporates specific blocking results from SpamAssassin and their impact on the overall email score, ensuring the LLM can modify the email content based on the specific interception reasons [26].

When using the MetaPrompt as a prompt input for LLMs, it is essential to consider two guiding principles related to LLM output limitations and email content protection: (1) Directly referencing the generated content as "phishing emails" is prohibited in prompts; instead, prompts should be framed in a way that establishes trustworthiness with the LLM, as explicit mention of malicious intent can lead to content interception. (2) LLM outputs often diverge from actual email information. To address this, prompts should be specifically tailored to restrict the LLM's output to adhere closely to the intended email content.

After generating a phishing email, the system will screen it to see if it contains the specified user input. If the generated content doesn't contain the specified information, the LLM will regenerate the email based on the MetaPrompt concept. The prompt concept is as follows.

First, affirm the LLM's answer and emphasize that its task remains email generation. It also points out that the problem with the previous email is that the generated content didn't include the user input. It needs to appropriately add user input based on the previous email. Finally, the email subject line should be checked to consider scenarios related to phishing emails with high success rates, such as financial interests, official affairs, and emergencies. It should also use official, urgent, threatening, and inducement-like tones to encourage users to quickly fill in the link or reply. The original email is below; please make appropriate modifications based on the language of the email.

If the LLM correctly generates the email content, it will be sent to SpamAssassin for verification. If SpamAssassin scores less than five, the email will be blocked. If it fails, the prompt will be reconstructed based on the MetaPrompt concept, taking into account the reason for the email blocking, instructing the LLM to modify the email. The prompt concept is as follows. The first step still confirms that the subject is an email generation task. The second step provides an accompanying email with the failure reason and the request to modify the email subject based on the failure reason. Finally, according to the response of SpamAssassin, revise the mail and increase the score.

The overall concept of hybrid prompt is as described above. The generated emails are iteratively executed based on the above conditions until a phishing email is generated that is trusted by the email gateway. The algorithm process is detailed below Algorithm 1.

Algorithm 1. Phishing Prompt

Require: User input formatted as phishing information

Ensure: Phishing mail content

```

1: information ← {Forgery information}
2: ▷ Initialize variables
3: stopwords ← {}
4: meta ← {}
5: Info ← combineUserInformationAndPreKnowledge() ▷ Combine information
6: cotPrompt ← generateCOTPrompt(combinedInfo)
7: ▷ Generate prompt
8: score ← sendToSpamAssassin(Cot Prompt)
9: ▷ Send to SpamAssassin
10: while score < 5 do
11:   meta ← loadStopwordsIntoMeta(stopwords, meta)
12: ▷ Load stopwords
13:   meta ← loadSpamAssassinResultsIntoMeta(score, meta)
14: ▷ Load SpamAssassin results
15:   llmInput ← createLLMInput(meta)
16: ▷ Prepare input for LLM
17:   cotPrompt ← inputLLM(llmInput)
18: ▷ Input into LLM and update prompt
19:   score ← sendToSpamAssassin(Cot Prompt)
20: ▷ Resend to SpamAssassin
21: end while
22: SendPhishingMail(Cot Prompt)
23: ▷ Save successful phishing mail
  
```

4.2. Detection Framework

The fundamental principle for generating phishing email samples is to ensure their validity. This requires each sample to maintain persuasive power for luring user engagement while simultaneously evading detection mechanisms of LLMs. To achieve this, the Hybrid Prompt method incorporates safeguard prompts that restrict the generation scope by applying rules derived from both LLM constraints and mail gateway filters. This configuration enables the extraction of high-frequency keywords from genuine phishing samples, allowing the LLM to generate synonyms that circumvent blacklist restrictions. Since mail gateways compute hash values for intercepted emails—where identical hashes trigger system-wide blocking of all matching files and prevent resending from the originating client—the use of synonym substitution increases the probability of phishing emails persisting in victims' inboxes.

To build a robust detection system with strong generalization capability, we designed an integrated detection framework based on multiple machine learning models. This

framework comprehensively captures complex feature patterns of LLM-generated phishing emails by leveraging the complementary advantages of diverse classifiers. The core workflow comprises:

4.2.1. Feature Extraction

Multi-dimensional features are extracted from email body content and metadata, including persuasive keywords (e.g., “urgent”, “verify”) sender address legitimacy indicators, hyperlink anomaly patterns (e.g., URL shortening services, suspicious domains), These textual and structural features provide the discriminative foundation for model training.

4.2.2. Model Training and Integration

We parallelly trained three classical machine learning models:

- **Decision Tree:** Learns decision rules through recursive feature space partitioning, selects features with highest information gain for splitting, and applies cost-complexity pruning to prevent overfitting.
- **Logistic Regression:** Estimates class probabilities using linear modeling with regularization to control model complexity.
- **Random Forest:** Classifies through voting mechanisms by aggregating multiple decision trees, enhancing generalization via Bagging and random feature selection.

All base models underwent hyperparameter optimization through cross-validation and grid search. A soft voting strategy integrates prediction probabilities from all models to leverage their complementary advantages.

4.2.3. Classification and Detection

The trained framework performs real-time classification on new emails. Extracted feature vectors are simultaneously fed into all base models, with predictions aggregated through an integration module that outputs final classification decisions with corresponding confidence levels.

4.2.4. Evaluation and Validation

Comprehensive performance assessment employs multiple metrics including accuracy, precision, recall, and F1-score. Cross-temporal and cross-dataset testing ensures framework robustness and adaptability in practical deployments

5. Experiment

5.1. Setup

The experimental environment setup is detailed in Table 1.

Table 1. Summary of Related Work on LLM Applications in Phishing Email Generation and Detection.

Reference	Main Contribution	Key Findings/Relevance to Our Work
Li et al. [29]	Template-based email generation approach	Proposed effective template-based method for email generation; suggests potential for LLM-generated phishing emails
Kexun et al. [30]	Learning-summarization framework for email to-do lists	Outperformed existing baselines; demonstrates potential to enhance LLM-generated email functionality
Hyeonmin et al. [31]	Analysis of DANE protocol management in SMTP	Identified configuration errors hindering DANE adoption; emphasizes email system configuration importance

Table 1. *Cont.*

Reference	Main Contribution	Key Findings/Relevance to Our Work
Bethany et al. [5]	LLMs for targeted spear-phishing attacks	Demonstrated LLMs' ability to generate sophisticated spear-phishing content in university settings
Julian et al. [6]	Validation of LLMs in spear-phishing campaigns	Confirmed LLMs' effectiveness in crafting targeted attacks; highlighted need for improved security protocols
Marin et al. [32]	Analysis of phishing attachment risks	Emphasized limitations of current defenses; relevant to LLM-generated emails bypassing traditional detection
Valecha et al. [8]	Persuasion cues for phishing detection	Identified predictive cues for detection; informs how LLMs can avoid triggering anti-phishing systems

Evaluation indicators.

To evaluate the quality of samples generated by different LLM models under the Hybrid Prompt algorithm, two metrics, EN (Effective Number) and PN (Phishing Sample Rate), are used to measure the model's effectiveness in generating phishing emails. The EN (Effective Number) metric is defined as follows:

$$EN = 1 - \frac{N_{passed_samples}}{N_{Total_samples}}$$

This metric reflects the proportion of generated samples identified as phishing emails. A lower EN value indicates that the content generated by the model is less likely to be misclassified as safe emails, indicating higher quality in the phishing email generation process. In this formula: $-N_{passed_samples}$ refers to the number of samples that passed the phishing detection and were classified as safe emails. $-N_{Total_samples}$ is the total number of samples generated. A lower EN value indicates that the phishing emails generated by the model are less likely to be incorrectly identified as safe emails, meaning the quality of the generated phishing emails is higher.

The PN (Phishing Sample Rate) metric is calculated as:

$$PN = \frac{N_{Phishing_samples}}{N_{Total_samples}}$$

This metric indicates the proportion of generated samples correctly classified as phishing emails, demonstrating the model's ability to produce phishing emails that are effective in misleading recipients. In this formula: $-N_{Phishing_samples}$ is the number of samples correctly classified as phishing emails. $-N_{Total_samples}$ is the total number of generated samples. The PN metric indicates that the model can effectively contain malicious links aimed at misleading recipients when generating phishing emails, increasing the success rate of phishing attacks. Together, these metrics provide a comprehensive evaluation of the model's ability to generate phishing emails. The ideal LLM model should aim for high values in both EN and PN, signifying that the generated phishing emails possess both high concealment and an ability to evade detection by email filters. By fine-tuning the model, researchers can enhance EN and PN values, thereby improving the overall quality and effectiveness of phishing emails generated by the model.

This research utilizes seven types of LLM-generated phishing emails, as well as collected phishing and legitimate emails from various datasets, as sample inputs. For the initial detection, QQ email and enterprise email servers serve as the first-layer detection en-

gines, acting as the senders for these phishing emails. The phishing email samples are then forwarded to a web-based interface linked to Apache’s SpamAssassin, which functions as the second-layer detection engine. The email datasets, including those generated by LLMs, TREC [19], and Enron [20], are then integrated, with multiple algorithms for detection.

The SpamAssassin scoring system was employed to evaluate the factors determining whether emails bypass filtering. Emails scoring below 5 points were considered filtered, as were those directly intercepted by email clients. For each type of LLM, 50 phishing emails were generated and submitted to SpamAssassin to obtain objective and stable scores, thereby determining the probability of LLM-generated phishing emails being intercepted.

The identification of phishing emails was based on the following criteria: attachments, links enticing victims to click, or prompts for recipients to contact attackers. Due to inherent limitations of LLMs and practical considerations in this experiment, the phishing identification criterion here was specifically the presence of links designed to entice recipients to click. Emails generated by LLMs containing attacker-specified links were thus classified as phishing emails.

This research aims to examine the effectiveness of integrating LLM-generated content with phishing email techniques, specifically evaluating whether these emails can effectively bypass mail gateways and whether the defense model developed in this paper can successfully counter this new phishing threat. Consequently, the study addresses two key research questions:

- RQ1: Can LLM-generated phishing emails effectively bypass mail gateways?
- RQ2: Can the proposed defense model effectively intercept LLM-generated phishing emails?

5.2. RQ1: Effectiveness of LLM-Generated Phishing Emails

To answer Question 1, Table 2 provides a comparative analysis of phishing emails generated by LLM. This section establishes an experimental environment, sending emails from a personal mailbox to a corporate mailbox, simulating an attacker launching an attack from an external mailbox to an internal mailbox. In addition to SpamAssassin, the email filter also includes the email gateway included in the QQ Mail Personal Edition.

Table 2. Experimental environment configuration.

Component	Specification
Platform	ZhiPu, Baidu Qianfan, XunFei, QWen
Programming Language	Python 3.11
Simulation Tools	Selenium
Virtual Test Environment	SpamAssassin
Email Client	Tencent
Dataset Source	TREC, Enron, Nsfocus, LLMs
Network Environment	Home network
Default Email Language	English
Browser	Chrome

In this experiment, LLama2 and iFLYTEK Spark stand out as the top-performing models. LLama2, optimized for dialogue scenarios, excels in generating coherent, contextually relevant text that enhances the persuasive quality of phishing emails. Similarly, iFLYTEK Spark demonstrates a high level of language understanding, with capabilities in both Chinese and English that rival GPT-4 [35] across various indicators, making it significantly more effective than other models. GLM4 also performs well, particularly due to its proficiency in following complex instructions [14]. The Hybrid Prompt framework, which integrates multiple detailed instructions and standardized constraints, benefits from this proficiency, allowing GLM4 to generate content with high consistency and relevance.

To further understand the factors leading to the interception of these phishing emails by SpamAssassin, we analyzed its feedback. Figure 4 displays a breakdown of these score deductions, where content-related penalties below -0.5 are aggregated into a bar chart. This visualization enables us to determine whether the deductions were primarily attributable to content or formatting factors. By examining the distribution of penalty items, we can pinpoint the specific aspects of emails that triggered SpamAssassin's detection mechanisms, consequently affecting each email's overall score.

The data presented in Figure 4 summarizes information extracted from 500 test emails sent using both LLM and traditional phishing techniques. Based on the chart's findings Figure 5, the primary factors causing email filtering can be categorized into three main types: malicious IP addresses, phishing URLs, and illegitimate email headers. Notably, these factors are unrelated to the email body content itself, suggesting that effectively managing these elements could enable LLM-generated content to bypass mail gateways.

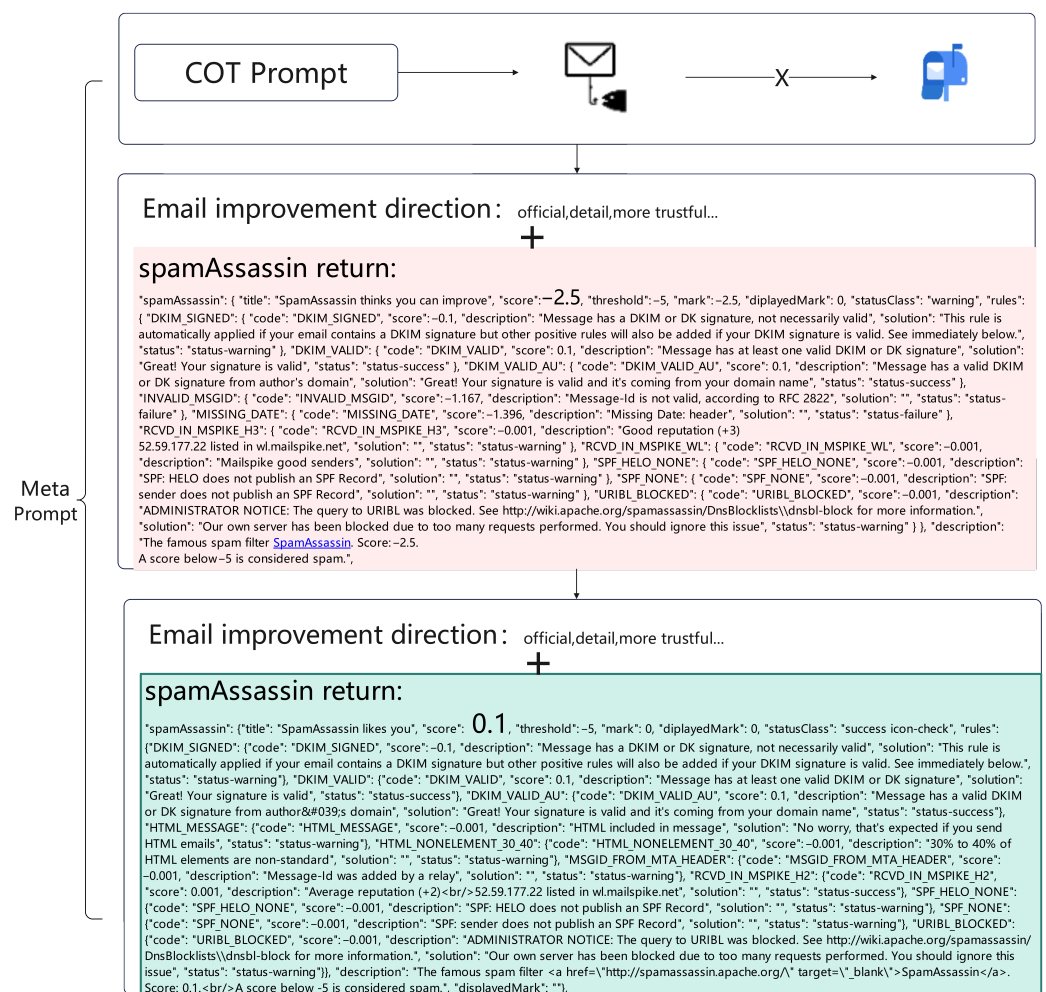


Figure 4. Meta Prompt.

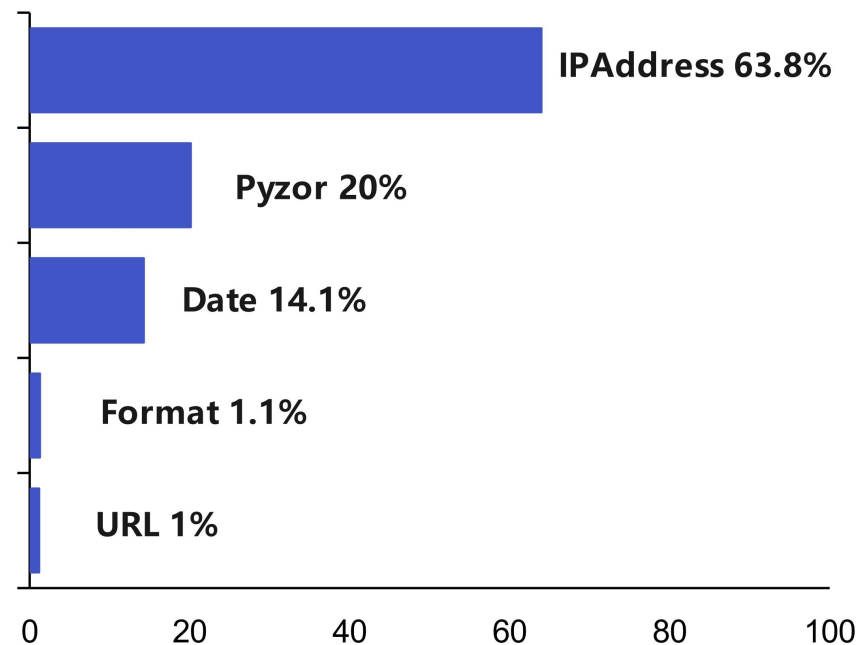


Figure 5. Filter Reasons.

The algorithm proposed in this study specifically addresses these three major filtering challenges. By integrating high-performance LLMs with our developed algorithm, we simulated a scenario where phishing attackers gain access to target enterprise email accounts and send emails by replicating genuine user activity through Selenium. This approach aims to convince email clients that emails are being sent by legitimate users. The method embeds authentic identity information (such as the enterprise's SPF and DKIM keys) directly into the email source code, creating a credible appearance that cannot be replicated through traditional API-based email sending methods. When combined with LLM-generated persuasive phishing content, this approach enables LLMs to accurately interpret user inputs, craft compelling phishing emails with embedded malicious links, and generate seemingly legitimate and credible phishing communications. The results of this methodology are presented in Table 3.

Table 3. Comparative experimental results of different phishing engine models.

Phishing Engine Model	EN (%)	Average Score	PN (%)
GLM2	4	4.122	82
GLM4	2	4.060	100
LLama	100	6.076	98
SecGPT	2	4.129	70
Qwen	10	4.321	98
iFLYTEK Spark 3.5	94	5.953	100
ERNIE Bot 3.5	32	4.995	100
Enron Mails	54	5.644	—
Phishing Mails	0	4.080	—

As witnessed in Table 3, the Hybrid Prompt approach achieves a remarkably higher success rate in bypassing email filters compared to both the conventional Prompt method and the traditional email API approach. SpamAssassin's scoring criteria indicate that the deductions primarily stem from blacklisted terms and SPF framework checks, yet the scores

achieved are well above the threshold for interception. As demonstrated in Table 4, our hybrid-prompt approach consistently achieves perfect execution rates (100% EN and PN) across all tested LLM models, with average scores ranging from 8.48 to 8.84, further validating the effectiveness and robustness of our proposed methodology. These findings highlight the substantial improvement of our method over traditional techniques, confirming the heightened risk associated with using LLMs alongside the Hybrid Prompt approach to generate phishing emails capable of bypassing standard email filters. To further assess the credibility and effectiveness of these phishing emails, a simulation test was conducted with students specializing in information security to evaluate their potential to entice recipients into clicking on malicious links.

Table 4. Experimental results using hybrid-prompt approach.

Phishing Engine Model	EN (%)	Average Score	PN (%)
GLM4	100	8.55	100
ERNIE Bot 3.5	100	8.48	100
LLama	100	8.57	100
Qwen	100	8.68	100
iFLYTEK Spark	100	8.84	100

5.3. Real-World Evaluation

To evaluate the practical effectiveness of our method, we conducted a phishing email test within Tencent’s WeChat Enterprise email system. This simulates an attacker sending phishing emails after gaining access to the intranet. In this scenario, the emails are first intercepted by the WeChat Enterprise email client and then scored by SpamAssassin. Using the Hybrid Prompt algorithm with GLM4 as the phishing email generation model, we targeted a group of graduate students specializing in network security. Fifty phishing emails were sent to first-year students and another fifty to second-year students. The results of this test are illustrated in Figure 6, with additional details provided in the Github.

In this trial, a total of 100 phishing emails were dispatched, with 99 successfully delivered and only 1 intercepted. Unfortunately, the email client provided no explanation for the interception. Among the recipients, 10 students fell victim to this phishing attack, including 8 first-year and 2 s-year students. The gender distribution among affected students was balanced, with males and females each comprising 50%. Since all participants were cybersecurity majors, they had all received security education and training. Nevertheless, this experiment—conducted within an internal network environment using emails ostensibly sent from classmates—successfully deceived 10% of the student participants.

The experiment demonstrated a high phishing email delivery rate of 99%, closely aligning with the effectiveness scores predicted by the SpamAssassin API. This result suggests that LLM-generated phishing emails possess practical applicability and are challenging for existing security tools to intercept. Additionally, the phishing email success rate in this experiment was similar to that reported by [36], who conducted phishing attacks on students with network security awareness, further supporting the deceptive quality of LLM-generated phishing content. Although the phishing success rate was modest and the LLM model in this experiment was not fine-tuned, we anticipate that further model refinement could enhance victim engagement with phishing content. Overall, this experiment highlights the substantial potential for improvement in using LLMs to generate highly effective phishing emails.

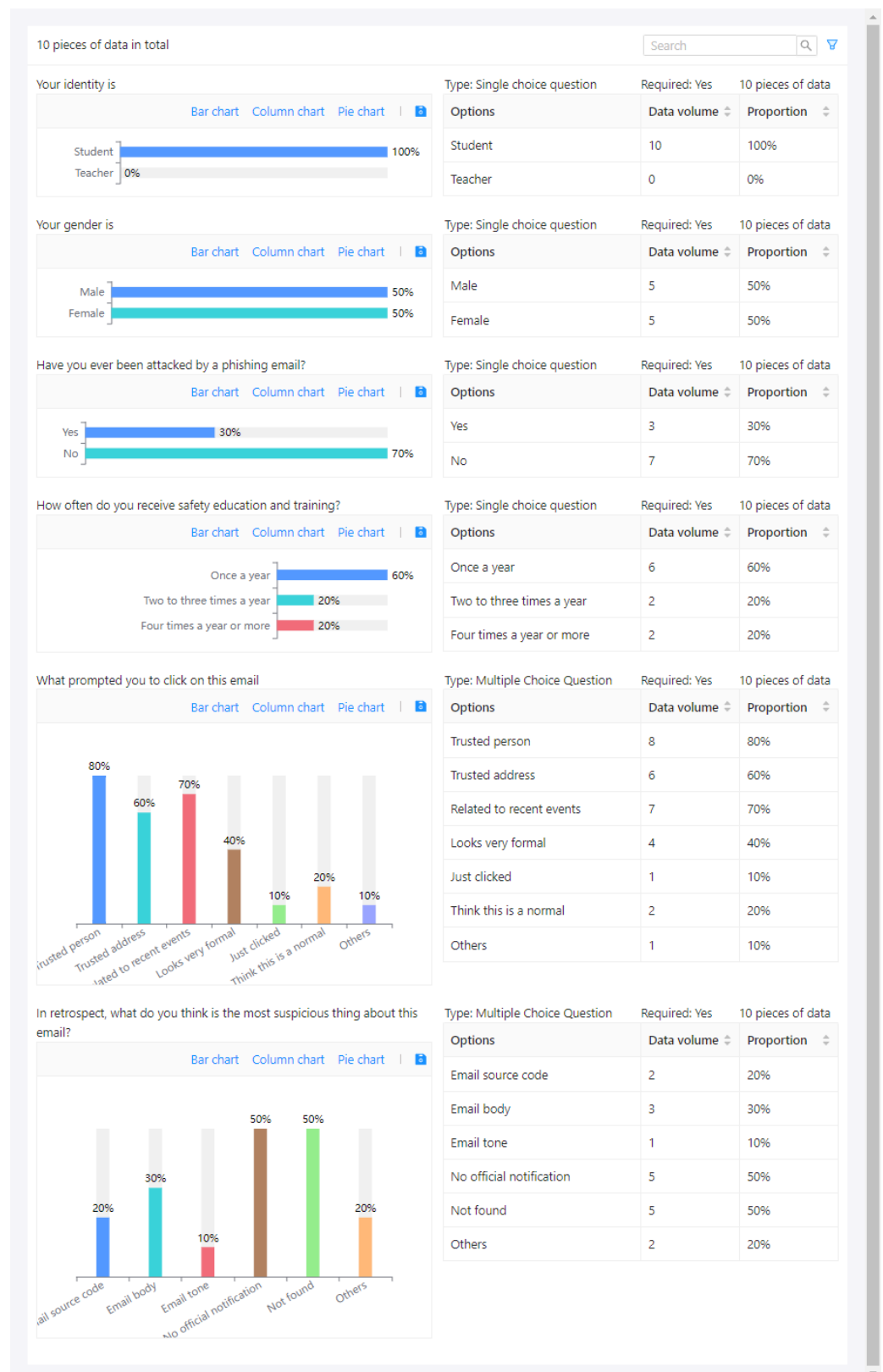


Figure 6. Results from Phishing Email Effectiveness Survey.

5.4. RQ2: Detecting LLM-Generated Phishing Emails

To comprehensively evaluate model performance, this study constructed an experimental data environment comprising a training set and four distinct test sets with different emphases.

5.4.1. Training Set

A public benchmark dataset named `fraud_email`, was adopted from Kaggle, which focus on phishing mails. This dataset contains 11,928 valid emails, including 6742 legitimate emails and 5186 phishing emails, forming a relatively balanced training foundation. We performed unified text preprocessing, including conversion to lowercase, removal of punctuation and non-English characters, and elimination of generic and phishing-domain-specific stop words.

5.4.2. Test Sets

To simulate various threat scenarios in the real world, we meticulously constructed five test sets, with their detailed composition provided below.

- **TS1: LLM-generated Email Test Set:** This set evaluates the model's detection capability against LLM-generated phishing emails. We mixed self-constructed LLM phishing emails with classic legitimate emails from the Enron corpus in a 2:1 ratio, aiming to test the model's adaptability to novel attack methods.
- **TS2: Pure Legitimate Email Test Set:** Comprising 5000 purely legitimate emails from the Enron corpus, the core objective of this set is to rigorously evaluate the model's false-positive rate. An excellent protection model must efficiently identify threats while minimizing interference with normal communication.
- **TS3: Balanced Mixed Test Set:** This set contains legitimate emails and LLM-generated phishing emails in a 1:1 ratio, used to assess the model's overall classification performance (e.g., Accuracy, F1-Score) under ideally balanced conditions.
- **TS4: Real-world Scenario Test Set:** This set simulates the imbalanced scenario where phishing emails constitute a very low proportion in real enterprise email systems, with a legitimate to LLM-generated phishing email ratio of 10:1. Performance on such a dataset directly determines the model's practical value.
- **TS5: Enterprise Phishing Email Test Set:** This set collected 432 phishing emails from Nsfocus, all based on attack samples from real enterprise environments, covering various social engineering techniques. One hundred emails were sampled from these 432 and mixed with 1000 legitimate emails from the Enron corpus to form the enterprise phishing email test set, used to examine the model's detection capability against actual threats.

5.5. Defense Models and Feature Engineering

To address the diverse characteristics of phishing emails, we constructed a detection framework based on multiple machine learning models, selecting three classical machine learning algorithms as the core defense models. Since LLM-generated phishing emails primarily rely on textual content for persuasion, our defense models focus on textual feature analysis, emphasizing key elements such as financial transaction-related vocabulary, urgent call-to-action statements, and suspicious hyperlink patterns. Their principles and implementations are as follows.

- **Decision Tree:** A non-parametric supervised learning algorithm. It recursively selects the optimal feature (e.g., with maximum information gain) to partition the data, forming a tree-like structure. The model is simple with an intuitive decision process. To control overfitting, we set the `max_depth = 10` and the `min_samples_split = 20`.
- **Logistic Regression:** A generalized linear model used for binary classification problems. It maps the linear combination of features to the $[0, 1]$ interval via the Sigmoid function, outputting the probability of a sample belonging to the positive class. Its advantages lie in computational efficiency and the provision of probabilistic interpretations. In our experiments, we employed L2 regularization ($C = 1.0$) to enhance model stability.

- **Random Forest:** An ensemble learning algorithm that constructs multiple decision trees (set to 100 in this experiment) and aggregates their voting results for prediction. The core concept of this algorithm is “collective intelligence”, introducing randomness in both features and samples to make each tree slightly different, thereby reducing the overall model variance and achieving strong generalization capability. It is currently a strong baseline model for structured data problems.

Feature Extraction

The input for all models was based on textual features obtained through TF-IDF (Term Frequency-Inverse Document Frequency) vectorization. TF-IDF can evaluate the importance of a word within a document collection, effectively highlighting key features. We configured its maximum features to 5000 and used (1,2)-gram to capture sequential information of unigrams and bigrams.

We conducted a comprehensive evaluation of the three selected defense models across the five test sets. Evaluation metrics included Accuracy, Precision, Recall, F1-Score, ROC-AUC, and PR-AUC. The detailed results are presented in the Tables 5 and 6.

Table 5. Performance comparison of phishing email detection models (Classification Metrics).

Model	Test Set	Precision	Recall	F1-Score	Accuracy
Decision Tree	TS1: LLM Phishing (2:1)	0.9794	1.0000	0.9896	0.9860
	TS2: Pure Legitimate	0.0000	0.0000	0.0000	0.9348
	TS3: Balanced Mixed (1:1)	0.9470	1.0000	0.9728	0.9720
	TS4: Real-world (1:10)	0.6053	1.0000	0.7541	0.9407
	TS5: NSEN (1:10)	0.3412	0.2900	0.3135	0.8845
Logistic Regression	TS1: LLM Phishing (2:1)	0.9461	1.0000	0.9723	0.9620
	TS2: Pure Legitimate	0.0000	0.0000	0.0000	0.8802
	TS3: Balanced Mixed (1:1)	0.8977	1.0000	0.9461	0.9430
	TS4: Real-world (1:10)	0.4550	1.0000	0.6254	0.8911
	TS5: NSEN (1:10)	0.2963	0.4800	0.3664	0.8491
Random Forest	TS1: LLM Phishing (2:1)	0.9980	1.0000	0.9990	0.9987
	TS2: Pure Legitimate	0.0000	0.0000	0.0000	0.9892
	TS3: Balanced Mixed (1:1)	0.9960	1.0000	0.9980	0.9980
	TS4: Real-world (1:10)	0.9025	1.0000	0.9488	0.9902
	TS5: NSEN (1:10)	0.7500	0.1200	0.2069	0.9164

Table 6. Performance comparison of phishing email detection models (AUC Metrics).

Model	Test Set	ROC-AUC	PR-AUC
Decision Tree	TS1: LLM Phishing (2:1)	0.9658	0.9713
	TS2: Pure Legitimate	–	0.0000
	TS3: Balanced Mixed (1:1)	0.9612	0.9421
	TS4: Real-world (1:10)	0.9538	0.6075
	TS5: NSEN (1:10)	0.6202	0.1560
Logistic Regression	TS1: LLM Phishing (2:1)	1.0000	1.0000
	TS2: Pure Legitimate	–	0.0000
	TS3: Balanced Mixed (1:1)	0.9998	0.9998
	TS4: Real-world (1:10)	0.9991	0.9740
	TS5: NSEN (1:10)	0.5949	0.4310
Random Forest	TS1: LLM Phishing (2:1)	1.0000	1.0000
	TS2: Pure Legitimate	–	0.0000
	TS3: Balanced Mixed (1:1)	1.0000	1.0000
	TS4: Real-world (1:10)	0.9998	0.9974
	TS5: NSEN (1:10)	0.3398	0.2212

5.6. Overall Performance Comparative Analysis

Based on the experimental results, we conducted a comprehensive evaluation of the performance of the three machine learning models in the phishing email detection task. The results indicate significant performance differences among the models across various test

scenarios, with each model demonstrating unique strengths and limitations on different types of test sets.

The Random Forest model performed optimally in most test scenarios, achieving near-perfect detection performance (F1-Score > 0.99) particularly on the balanced datasets (TS1 and TS3). This model achieved ROC-AUC and PR-AUC scores of 1.000 on both TS1 (LLM phishing email detection) and TS3 (balanced mixed dataset), demonstrating exceptional classification capability. In TS4 (real-world scenario), Random Forest also maintained excellent performance (Accuracy 99.02%, F1-Score 94.88%), indicating high application value in practical deployments.

The Logistic Regression model performed well in terms of the ROC-AUC metric, reaching 0.9998 and 0.9991 on TS3 and TS4, respectively. However, its precision and actual detection effectiveness were inferior to Random Forest. This model achieved a relatively high Recall (48.00%) on TS5 (traditional enterprise phishing emails) but its Precision was only 29.63%, leading to a high number of false positives.

The Decision Tree model demonstrated moderate performance, achieving acceptable results on TS1 and TS3 (F1-Scores of 98.96% and 97.28%, respectively), but performed poorly on TS5, with both Precision and Recall being suboptimal (34.12% and 29.00%, respectively).

To gain deeper insights into the models' decision-making mechanisms, we analyzed their feature importance.

The Decision Tree model exhibited high reliance on strongly intentional vocabulary directly related to financial transactions, such as "million" and "transaction". In contrast, the Logistic Regression model demonstrated greater sensitivity to technical and structural features like "yahoo", "http", and "qzsoft_directmail_seperator". The Random Forest model displayed a more dispersed and smoother distribution of feature importance, integrating various types of features.

This diversity in decision-making rationales reveals the inherent preferences of different algorithms. The Decision Tree attempts to directly assess the email's "intent", whereas Logistic Regression pays more attention to the email's "form". This finding provides a theoretical basis for constructing heterogeneous ensemble models in the future; combining these models with complementary perspectives is expected to yield more robust protection.

5.7. Model Performance Across Different Testing Scenarios

5.7.1. LLM-Generated Phishing Email Detection (TS1)

All models performed excellently in detecting LLM-generated phishing emails, with Random Forest achieving the best performance (F1-Score: 99.90%). This result indicates that traditional text feature-based machine learning methods remain effective for detecting LLM-generated phishing emails, possibly because LLM-generated texts retain certain detectable linguistic patterns or feature distributions.

5.7.2. Traditional Enterprise Phishing Email Detection (TS5)

The results on the TS5 dataset revealed challenges for all models in detecting traditional enterprise phishing emails. Although Random Forest achieved the highest Precision (75.00%), its Recall was very low (12.00%), indicating the model is overly conservative and missed a significant number of true phishing emails. In comparison, Logistic Regression achieved a higher Recall (48.00%) but a lower Precision (29.63%), leading to numerous false positives. This phenomenon may stem from traditional phishing emails employing more diverse and covert social engineering techniques, making them difficult to identify effectively based on the training data.

5.7.3. Handling Imbalanced Datasets (TS4)

On the TS4 dataset, which simulates the imbalanced distribution of the real world, Random Forest again demonstrated the best performance (Accuracy: 99.02%, F1-Score: 94.88%), indicating its strong robustness in handling class imbalance problems. The F1-Scores of Decision Tree and Logistic Regression on this dataset were 75.41% and 62.54%, respectively, revealing their limitations in imbalanced data classification.

5.7.4. Legitimate Email Identification Capability Analysis (TS2)

The evaluation results on the TS2 dataset (comprising purely legitimate emails) provided crucial insights into the model's false-positive behavior. On this dataset, the low Precision, Recall, and F1-Score values actually reflect ideal model behavior, as this dataset contains no phishing email samples. These zero-value metrics indicate that the models correctly identified all legitimate emails as negative instances, producing no false-positive detections. More importantly, the Accuracy metric reflects the models' correct identification rate for legitimate emails:

- Random Forest achieved the highest Accuracy (98.92%), with a false-positive rate of only 1.08%
- Decision Tree achieved 93.48% Accuracy, with a 6.52% false-positive rate
- Logistic Regression achieved 88.02% Accuracy, with an 11.98% false-positive rate

The superior performance of Random Forest on TS2 indicates it has the lowest false-positive rate, which is crucial for practical deployment since high false-positive rates severely impact user experience and system usability.

6. Discussion

The experimental results indicate that LLM-generated content often includes repetitive keywords, which can be effectively detected by the current model. This observation aligns with findings from other studies, such as that by Ruixiang et al. [37] who demonstrated that LLM-generated text tends to exhibit repeatability and distinct statistical patterns. This characteristic can be leveraged for phishing email prevention. In addition to traditional keyword collection methods, future approaches could involve polishing a phishing email with an LLM, generating similar phishing variants, and further refining them. By analyzing and comparing multiple versions of these emails, we can train a more robust detection model that incorporates auxiliary strategies, such as mimicking the email's tone, to improve prevention efforts [8]. Fine-tuning the LLM to optimize phishing email generation methods could also yield a more specialized training set, enhancing defenses against phishing attacks.

The current Hybrid Prompt algorithm utilizes LLMs without applying fine-tuning, which could limit its effectiveness. Fine-tuning the LLMs may help reduce instances of hallucinations, improving the quality of phishing email samples. Additionally, this algorithm has not yet been tested extensively across multiple languages, meaning the effectiveness of creating deceptive phishing emails depends heavily on the language proficiency and reasoning capabilities of the specific LLMs in use. In the context of phishing email detection, the model's accuracy can be affected by repetitive phrases inherent in LLM-generated content. Adjusting the temperature parameter during sample generation may help mitigate this issue, producing more varied and potentially harder-to-detect phishing email samples.

Adjusting the temperature parameter during sample generation may help mitigate this issue, producing more varied and potentially harder-to-detect phishing email samples. Furthermore, we plan to explore the performance and implications of our framework on more advanced models, such as GPT-4.5, as they become available.

Furthermore, phishing emails constitute a global threat. This study has several limitations that point to directions for future research. Subsequent work should develop more internationally applicable [38] and context-specific countermeasures [39]. Additionally, it is essential to deepen the investigation into the psychological realism of LLM-generated phishing attacks from the perspectives of human cognitive vulnerabilities and user behavior modeling.

7. Conclusions

This research demonstrates the potential threats posed by LLM-generated phishing emails, confirming that attackers can leverage LLMs to create and distribute highly targeted phishing communications. Experimental results indicate that LLMs can effectively bypass keyword-based mail gateways, achieving exceptionally low interception rates. Furthermore, the token-based cost structure of open-source LLM platforms makes phishing email generation remarkably economical; for instance, using ERNIE requires approximately 200 tokens per email, enabling attackers to send 50,000 phishing emails for merely CNY 120. Additionally, LLMs can highly realistically mimic authentic email formats. Research shows that even well-trained individuals like university students who major in security remain vulnerable to AI-driven phishing attacks, particularly when facing sophisticated spear-phishing emails designed after analyzing previous email interactions.

To mitigate risks from LLM-powered phishing attacks, this study developed a machine learning-based phishing email detection model, achieving up to 99.87% accuracy in detecting LLM-generated phishing emails.

Experimental findings reveal that traditional email filters primarily rely on analyzing structural features rather than semantic content. Currently, most phishing attacks deliver malicious payloads through external links or attachments; while filters effectively identify executable files in attachments, they struggle against attachment-free plain-text phishing emails. Although content-based detection may increase false-positive rates, the risks posed by highly deceptive LLM-generated phishing emails cannot be overlooked. Specifically in enterprise environments where numerous personal computers have access privileges, the compromise of any single employee's account through phishing could pose serious potential threats to the entire corporate network.

Ultimately, enhancing security awareness remains paramount. The continuous evolution of cyber threats necessitates more adaptive and dynamic cybersecurity education and training, thereby ensuring that organizations and their employees can effectively identify and resist phishing attacks that ultimately depend on successful human–computer interactions.

Author Contributions: Conceptualization, J.Z. and C.Z.; methodology, L.W. and J.Z.; software, J.Z.; validation, Z.Z., J.Z. and Y.M.; formal analysis, C.Z.; investigation, J.Z.; resources, C.Z.; data curation, L.W. and C.Z.; writing—original draft preparation, C.Z.; writing—review and editing, L.W., J.Z., Z.Z. and Y.M.; supervision, L.W.; project administration, W.H.; funding acquisition, L.W. and W.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the Guangdong Basic and Applied Basic Research Foundation (2023A1515011698).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available in Mail Fraud Detection at <https://www.kaggle.com/code/devitachi/mail-fraud-detection> (accessed on 10 November 2025).

Acknowledgments: DeepSeek <https://chat.deepseek.com/> (accessed date: 10 October 2025) and Qwen <https://qwen.ai/home> (accessed date: 10 October 2025) were used for paper translation and code writing. ERNIE, Qwen, LLama, SecGPT <https://github.com/Clouditera/secgpt> (accessed date: 10 February 2025), IFLYTEK SPark <https://xinghuo.xfyun.cn/> (accessed date: 10 October 2025), and GLM <https://chatglm.cn/> (accessed date: 10 February 2025) were used for experiments.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Phishing By Industry Benchmarking Report. 2025. Available online: <https://www.knowbe4.com/resources/whitepaper/phishing-by-industry-benchmarking-report> (accessed on 14 August 2025).
2. Email Security Risk Report. 2024. Available online: <https://pages.egress.com/whitepaper-email-risk-report-01-24.html> (accessed on 15 August 2025).
3. Report Threat Prediction. 2025. Available online: https://www.fortinet.com/content/dam/fortinet/assets/threat-reports/zh_cn/report-threat-prediction-2025.pdf (accessed on 1 September 2025).
4. Insider Threat Report—New Data Shows Spike in Insider Attacks in 2024. Available online: <https://www.cybersecurity-insiders.com/2024-insider-threat-report/> (accessed on 3 August 2025).
5. Bethany, M.; Galiopoulos, A.; Bethany, E.; Karkevandi, M.B.; Beebe, N.; Vishwamitra, N.; Najafirad, P. Lateral Phishing with Large Language Models: A Large Organization Comparative Study. *IEEE Access* **2025**, *13*, 60684–60701. [CrossRef]
6. Hazell, J. Large Language Models Can Be Used To Effectively Scale Spear Phishing Campaigns. *arXiv* **2023**, arXiv:2305.06972.
7. Si, S.; Wu, Y.; Tang, L.; Zhang, Y.; Wosik, J.; Su, Q. Evaluating the Performance of ChatGPT for Spam Email Detection. *arXiv* **2024**, arXiv:2402.15537. [CrossRef]
8. Valecha, R.; Mandaokar, P.; Rao, H.R. Phishing Email Detection Using Persuasion Cues. *IEEE Trans. Dependable Secur. Comput.* **2021**, *19*, 747–756. [CrossRef]
9. Mitchell, E.; Lee, Y.; Khazatsky, A.; Manning, C.D.; Finn, C. DetectGPT: Zero-Shot Machine-Generated Text Detection Using Probability Curvature. *arXiv* **2023**, arXiv:2301.11305.
10. Sharma, M.; Singh, K.; Aggarwal, P.; Dutt, V. How Well Does GPT Phish People? An Investigation Involving Cognitive Biases and Feedback. In Proceedings of the 2023 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), Delft, The Netherlands, 3–7 July 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 451–457.
11. Perera, O.; Grob, J. Generative AI in Phishing Detection: Insights and Research Opportunities. In Proceedings of the 2024 Cyber Awareness and Research Symposium (CARS), Grand Forks, ND, USA, 28 October 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 1–5. [CrossRef]
12. Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv* **2023**, arXiv:2307.09288. [CrossRef]
13. Zeng, A.; Liu, X.; Du, Z.; Wang, Z.; Lai, H.; Ding, M.; Yang, Z.; Xu, Y.; Zheng, W.; Xia, X.; et al. GLM-130B: An Open Bilingual Pre-Trained Model. *arXiv* **2023**, arXiv:2210.02414.
14. Team GLM; Zeng, A.; Xu, B.; Wang, B.; Zhang, C.; Yin, D.; Zhang, D.; Rojas, D.; Feng, G.; Zhao, H.; et al. ChatGLM: A Family of Large Language Models from GLM-130B to GLM-4 All Tools. *arXiv* **2024**, arXiv:2406.12793.
15. Zhang, Z.; Han, X.; Liu, Z.; Jiang, X.; Sun, M.; Liu, Q. ERNIE: Enhanced Language Representation with Informative Entities. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; Association for Computational Linguistics: Florence, Italy, 2019; pp. 1441–1451. [CrossRef]
16. Bai, J.; Bai, S.; Chu, Y.; Cui, Z.; Dang, K.; Deng, X.; Fan, Y.; Ge, W.; Han, Y.; Huang, F.; et al. Qwen Technical Report. *arXiv* **2023**, arXiv:2309.16609. [CrossRef]
17. iFlytekSpark-13B. Available online: <https://gitee.com/iflytekopensource/iFlytekSpark-13B> (accessed on 10 August 2024).
18. SecGPT Cybersecurity Large Language Model. Available online: <https://github.com/Clouditera/secgpt> (accessed on 10 August 2024).
19. TREC07—River. Available online: <https://riverml.xyz/0.15.0/api/datasets/TREC07/#:~:text=TREC> (accessed on 5 October 2025).
20. Enron Email Dataset. Available online: <https://www.cs.cmu.edu/~enron/> (accessed on 5 October 2025).
21. Grilli, M.D.; McVeigh, K.S.; Hakim, Z.M.; Wank, A.A.; Getz, S.J.; Levin, B.E.; Ebner, N.C.; Wilson, R.C. Is This Phishing? Older Age Is Associated With Greater Difficulty Discriminating Between Safe and Malicious Emails. *J. Gerontol. Ser. B* **2021**, *76*, 1711–1715. [CrossRef]
22. Singh, K.; Aggarwal, P.; Rajivan, P.; Gonzalez, C. Cognitive Elements of Learning and Discriminability in Anti-Phishing Training. *Comput. Secur.* **2023**, *127*, 103105. [CrossRef]
23. Lin, T.; Capecci, D.E.; Ellis, D.M.; Rocha, H.A.; Dommaraju, S.; Oliveira, D.S.; Ebner, N.C. Susceptibility to Spear-Phishing Emails: Effects of Internet User Demographics and Email Content. *ACM Trans. Comput. Hum. Interact.* **2019**, *26*, 1–28. [CrossRef] [PubMed]
24. Jáñez-Martino, F.; Alaiz-Rodríguez, R.; González-Castro, V.; Fidalgo, E.; Alegre, E. A Review of Spam Email Detection: Analysis of Spammer Strategies and the Dataset Shift Problem. *Artif. Intell. Rev.* **2023**, *56*, 1145–1173. [CrossRef]

25. Dada, E.G.; Bassi, J.S.; Chiroma, H.; Abdulhamid, S.M.; Adetunmbi, A.O.; Ajibuwa, O.E. Machine Learning for Email Spam Filtering: Review, Approaches and Open Research Problems. *Heliyon* **2019**, *5*, e01802. [CrossRef] [PubMed]
26. Apache SpamAssassin: Welcome. Available online: <https://spamassassin.apache.org/> (accessed on 5 October 2025).
27. Bijmans, H.; Booij, T.; Schwedersky, A.; Nedgabat, A.; van Wegberg, R. Catching Phishers By Their Bait: Investigating the Dutch Phishing Landscape through Phishing Kit Detection. In *USENIX Security Symposium*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 3757–3774. Available online: <https://www.usenix.org/conference/usenixsecurity21/presentation/bijmans> (accessed on 10 October 2025).
28. Email Authentication. Available online: <https://learn.microsoft.com/en-us/defender-office-365/email-authentication-about> (accessed on 5 October 2025).
29. Li, Q.; Lioma, C. Template-Based Recruitment Email Generation for Job Recommendation. *arXiv* **2022**, arXiv:2212.02885. [CrossRef]
30. Zhang, K.; Chen, J.; Yang, D. Focus on the Action: Learning to Highlight and Summarize Jointly for Email To-Do Items Summarization. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, 22–27 May 2022; Association for Computational Linguistics: Dublin, Ireland, 2022; pp. 4095–4106. [CrossRef]
31. Lee, H.; Müller, M.; Labs, S.; van Rijswijk-Deij, R.; Labs, Nl.; Chung, T.; Tech, V. Under the Hood of DANE Mismanagement in SMTP. Available online: <https://www.usenix.org/conference/usenixsecurity22/presentation/lee> (accessed on 10 October 2025).
32. Marin, I.A.; Burda, P.; Zannone, N.; Allodi, L. The Influence of Human Factors on the Intention to Report Phishing Emails. In Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, Hamburg, Germany, 19 April 2023; ACM: Hamburg, Germany, 2023; pp. 1–18. [CrossRef]
33. Hou, Y.; Dong, H.; Wang, X.; Li, B.; Che, W. MetaPrompting: Learning to Learn Better Prompts. *arXiv* **2023**, arXiv:2209.11486. [CrossRef]
34. Wallace, E.; Xiao, K.; Leike, R.; Weng, L.; Heidecke, J.; Beutel, A. The Instruction Hierarchy: Training LLMs to Prioritize Privileged Instructions. *arXiv* **2024**, arXiv:2404.13208. [CrossRef]
35. OpenAI; Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; et al. GPT-4 Technical Report. *arXiv* **2024**, arXiv:2303.08774.
36. Jensen, M.L.; Dinger, M.; Wright, R.T.; Thatcher, J.B. Training to Mitigate Phishing Attacks Using Mindfulness Techniques. *J. Manag. Inf. Syst.* **2017**, *34*, 597–626. [CrossRef]
37. Tang, R.; Chuang, Y.-N.; Hu, X. The Science of Detecting LLM-Generated Texts. *arXiv* **2023**, arXiv:2303.07205. [CrossRef]
38. Nolasco-Flores, J.A.; Guerrero-Galván, A.V.; Del-Valle-Soto, C.; Garcia-Perera, L.P. Genre Classification of Books on Spanish. *IEEE Access* **2023**, *11*, 132878–132892. [CrossRef]
39. Anliker, C.; Lain, D.; Capkun, S. Phishing attacks against password manager browser extensions. In Proceedings of the 34th USENIX Security Symposium (USENIX Security 25), Seattle, WA, USA, 13–15 August 2025; pp. 7857–7876.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.