

Evaluating spam filters and Stylometric Detection of AI-generated phishing emails

Chidimma Opara^{ID*}, Paolo Modesti^{ID}, Lewis Golightly^{ID}

Department of Computing and Games, Teesside University, TS1 3BX, Middlesbrough, UK

ARTICLE INFO

Keywords:

AI-generated phishing email
Phishing detection
Stylometric analysis
Large Language Models (LLMs)
Machine learning
Cybersecurity

ABSTRACT

The advanced architecture of Large Language Models (LLMs) has revolutionised natural language processing, enabling the creation of text that convincingly mimics legitimate human communication, including phishing emails. As AI-generated phishing emails become increasingly sophisticated, a critical question arises: How effectively can current email systems and detection mechanisms identify these threats? This study addresses this issue by analysing 63 AI-generated phishing emails created using GPT-4o. It evaluates the effectiveness of major email services, Gmail, Outlook, and Yahoo, in filtering these malicious communications. The findings reveal that Gmail and Outlook allowed more AI-generated phishing emails to bypass their filters compared to Yahoo, highlighting vulnerabilities in existing email filtering systems. To mitigate these challenges, we applied 60 stylometric features across four machine learning models: Logistic Regression, Support Vector Machine, Random Forest, and XGBoost. Among these, XGBoost demonstrated superior performance, achieving 96% accuracy and an AUC score of 99%. Key features such as imperative verb count, clause density, and first-person pronoun usage were instrumental to the model's success. The dataset of AI-generated phishing emails is publicly available on Kaggle to foster further research.

1. Introduction

Large Language Models (LLMs) have revolutionised natural language processing by generating text that closely mimics human writing across diverse applications. These advances bring significant benefits, such as automating content creation (Kirova, Ku, Laracy, & Marlowe, 2024), enhancing customer service (Chew, Lin, Chen, Fan, & Lee, 2024), and improving technical support (Raman, Calyam, & Achuthan, 2024). However, alongside these benefits, LLMs introduce serious risks by creating new attack vectors for cybercriminals. One prominent threat is their misuse to craft highly convincing phishing emails, posing substantial risks to individuals and organisations (Chataut, Gyawali, & Usman, 2024; Heiding, Schneier, Vishwanath, Bernstein, & Park, 2024; Roy, Thota, Naragam, & Nilizadeh, 2023). The widespread availability of LLMs like ChatGPT,¹ Claude,² and Gemini³ encourages this threat, enabling adversaries to produce customised, credible messages with minimal effort (Wang et al., 2023). By simply providing prompts, malicious actors can quickly generate persuasive phishing emails without needing explicit training examples, making it easier to bypass spam filters and exploit vulnerabilities in email-based security systems.

Phishing is a well-established social engineering strategy in which attackers impersonate trusted entities, such as banks or government agencies, to deceive users into divulging personal or financial information (Drake, Oliver, & Koontz, 2004). Traditional detection methods, which rely on external signals like suspicious URLs, domain reputation, or explicit brand impersonation, have been effective against conventional phishing attacks. However, these state-of-the-art methods often rely on outdated datasets, such as SpamAssassin or Enron, which predate the advent of LLM-driven attacks and therefore lack realistic examples of AI-generated phishing content (Alhogail & Alsabih, 2021; Gallo, Maiello, Botta, & Ventre, 2021).

The emergence of AI-generated phishing emails presents a growing challenge for detection systems. While recent studies have explored using LLMs to identify such threats (Chataut et al., 2024; Heiding et al., 2024), these approaches often face “black-box” challenges inherent to deep learning models, complicating transparency and trust in their solutions. Additionally, some research has investigated the effectiveness of AI-generated phishing emails in bypassing spam filters, but these studies frequently focus on a single email provider, offering limited

* Corresponding author.

E-mail addresses: c.opara@tees.ac.uk (C. Opara), p.modesti@tees.ac.uk (P. Modesti), l.golightly@tees.ac.uk (L. Golightly).

¹ <https://chatgpt.com/>

² <https://claude.ai/>

³ <https://gemini.google.com/>

insights into the broader spam-filtering ecosystem (Bethany et al., 2024).

To address these gaps, this paper investigates two key research questions: *First*, how effective are the spam filters of major email providers in detecting AI-generated phishing emails? *Second*, what stylistic and linguistic features can distinguish AI-generated phishing emails from legitimate ones when applied to machine learning classifiers?

By systematically generating 63 AI-based phishing emails using simple text prompts and testing them across major webmail providers, this study demonstrates the vulnerabilities of existing detection mechanisms. The findings reveal significant disparities: Yahoo flagged 90% of these emails as spam, Gmail showed moderate efficiency, and Microsoft Outlook⁴ flagged only 4%. Tests with AI-generated legitimate emails further exposed provider-specific inconsistencies. Yahoo frequently over-blocked messages, Gmail flagged legitimate emails from Outlook, and Outlook remained highly permissive overall. These results underscore the pressing need for enhancements in spam-filtering systems to effectively address the growing threat of AI-generated phishing content.

To bridge this gap, this study introduces a novel detection approach that leverages stylistic and linguistic cues, moving beyond the traditional reliance on suspicious URLs, domain reputation, or explicit brand impersonation. By curating 60 stylistic features, 47 of which are newly applied to phishing detection, this method captures nuanced stylistic patterns, such as sentence structure, punctuation usage, tone markers, and syntactic complexity, offering a more advanced detection framework than conventional keyword-based approaches.

The effectiveness of these features was evaluated using four popular shallow machine learning classifiers, with XGBoost achieving an impressive 96% accuracy. Key predictive features included the count of imperative verbs (e.g., “click”, “verify”) and clause density, which measures sentence complexity. These results reveal that AI-generated phishing emails often employ urgent prompts and sophisticated sentence structures to enhance persuasiveness.

This methodology improves the interpretability and effectiveness of text-based detection strategies and provides a transparent understanding of the linguistic traits most critical for identifying AI-generated phishing emails. By emphasising these features, this study offers a practical solution for detecting zero-day phishing attacks that might evade conventional filters while establishing a new benchmark for applying stylistic analysis in cybersecurity.

1.1. Research contributions

This research makes the following contributions:

- 1. Empirical Analysis of Spam Filters:** We empirically evaluate the spam filters employed by Gmail, Yahoo, and Microsoft Outlook against 63 AI-generated phishing emails. Our tests reveal the strengths and vulnerabilities of these filters and demonstrate how easily LLM-generated phishing emails evade traditional detection. Additionally, by conducting a counter-experiment with an equal number of legitimate AI-generated emails, we uncover significant false-positive rates, further underscoring the inconsistencies in current spam filter performance.
- 2. Introduction of 47 New Stylistic Features for AI-Generated Phishing Detection:** This study applies 60 stylistic and linguistic features, including 47 that are new to the phishing detection domain. Unlike traditional detection methods that depend on external signals such as suspicious links or domain reputation, our approach focuses on intrinsic textual properties like sentence complexity, imperative verb usage, and

function word patterns. These features provide a complementary layer of security, particularly for zero-day phishing attacks where conventional external indicators may be absent.

- 3. Interpretable Machine Learning Approach:** By applying these stylistic features to four interpretable shallow classifiers, we achieve a classification accuracy of 96%, with XGBoost emerging as the best-performing model. This approach highlights the linguistic elements, such as imperative verbs and clause density, that most effectively distinguish AI-generated phishing emails from legitimate messages. By prioritising interpretability, our method offers a transparent alternative to black-box LLM-based detectors, shedding light on the stylistic markers driving classification.
- 4. Publicly Available Dataset.** To support further research, we release our dataset of AI-generated phishing and legitimate emails on [Kaggle](#), offering a current, high-quality dataset for ongoing and future studies in LLM-based social engineering attacks.

The remainder of this paper is structured as follows: Section 2 reviews the current state-of-the-art research on phishing email detection and examines the role of Large Language Models (LLMs) in this evolving threat landscape. Section 3 defines the threat model underpinning this study, highlighting how attackers leverage LLMs to create sophisticated phishing campaigns. Section 4 describes the methods for generating phishing emails with GPT-4o, including extracting stylistic features for machine learning classification. Section 5 presents the experimental results, evaluating the effectiveness of spam filters and machine learning classifiers. Finally, Section 6 summarises the findings, discusses their implications, and outlines potential directions for future research.

2. Related work

Traditionally, phishing detection has relied on shallow machine learning models like Naïve Bayes and Support Vector Machines (SVM). These models classify emails by analysing features such as HTML tags and network parameters (Alhogail & Alsabih, 2021; Gallo, Gentile, Ruggiero, Botta, & Ventre, 2024; Toolan & Carthy, 2010). While these methods have shown high accuracy, attackers have become adept at bypassing detection by using URL shorteners, disguising hyperlinks with legitimate-looking text, or registering new domains. Additionally, relying on HTML and network-based features (like domain age or JavaScript detection) raises privacy and security concerns, mainly when external resources or third-party servers are involved.

As phishing tactics evolve, there has been a shift towards using deep learning models to better capture human language's nuances. Transformer-based models, such as BERT (Bidirectional Encoder Representations from Transformers) (Devlin, Chang, Lee, & Toutanova, 2018), have significantly advanced phishing detection by excelling at tasks that require deep linguistic understanding. For instance, Wang et al. (2023) developed PhishBERT, a model fine-tuned on a large dataset of phishing URLs, which improved detection accuracy. Similarly, recent research has explored multimodal LLMs that analyse both text and other forms of data, offering a more comprehensive approach to identifying phishing attempts. Maneriker et al. (2021) introduced URLTran, a transformer model designed specifically for phishing URL detection, further demonstrating the potential of transformer models in cybersecurity. Despite their strong performance, these models operate as “black boxes”, meaning their decision-making processes are not transparent. This lack of interpretability makes it challenging to understand why an email is classified as phishing or legitimate, which is a common challenge in AI research. The growing field of explainable AI aims to address this issue by making model outputs more interpretable.

To improve phishing detection, researchers have also explored the use of stylistic features. For example, Dewan, Kashyap, and Kumaraguru (2014) analysed spear phishing emails using stylistic

⁴ Here referring to the email provider, not the email client with the same name.

features extracted from email content combined with social features from LinkedIn profiles. Their approach, tested on a spear phishing, spam, and non-phishing emails dataset, achieved a 97.76% accuracy rate in detecting spear phishing. Similarly, Duman, Kalkan-Cakmakci, Egele, Robertson, and Kirda (2016) introduced EmailProfiler, a tool that identifies spear phishing by analysing both email metadata and 199 stylometric features from the email body. By building behavioural profiles for email senders based on header information, the system detected inconsistencies in new emails by comparing them to these profiles. EmailProfiler's detection accuracy ranged from 67% to 100% across each profile, depending on the complexity of the spear phishing attack.

While these models have shown high accuracy, they have yet to be tested on datasets that include AI-generated emails, and the stylometric features used were not exclusively focused on the textual content of the emails.

2.1. The role of LLMs in phishing detection

Large Language Models (LLMs) have become central to phishing detection as phishing tactics evolve. Models like GPT-3 and GPT-4 present a unique challenge because they can both generate and detect phishing content, a dual role explored in recent research. For example, Chataut et al. (2024) demonstrated GPT-4's ability to detect phishing emails, showcasing its skill in understanding and generating complex text structures. Similarly, Heiding et al. (2024) fine-tuned GPT-4 for phishing detection, leveraging its advanced text generation capabilities. Patel, Rehman, and Iqbal (2024) extended this analysis by evaluating various LLMs, including GPT-3.5 and GPT-4, across different phishing datasets. Their findings showed that while these models are highly effective, performance can vary depending on the dataset and the specific phishing tactics, underscoring the need for continuous refinement and domain-specific training data.

While LLMs have made significant strides in phishing detection, one major challenge is the proprietary nature of these models, which limits transparency into how they work. Additionally, with commercial webmail services being frequent targets of phishing attacks (APWG, 2024), it is essential to analyse how well spam filters handle AI-generated phishing emails on these platforms.

This paper addresses key gaps in phishing detection by analysing the bypass rates of AI-generated phishing emails across major email providers and identifying distinct text-based features that differentiate AI-generated phishing emails from legitimate ones.

3. Threat model

This study analyses a threat model where attackers exploit Large Language Models (LLMs) to generate advanced phishing emails that bypass spam filters and deceive recipients. Attackers utilise LLMs, such as GPT-4, by crafting specific prompts that guide the model to produce realistic phishing emails. These emails mimic legitimate communication styles while embedding psychological tactics commonly used in phishing attacks, such as urgency, authority, and curiosity.

The workflow of generating phishing emails using LLMs is illustrated in Fig. 1.

The process involves the following steps:

1. **Prompt Engineering:** Attackers design prompts tailored to specific themes (e.g., financial notifications, job opportunities) and levels of sophistication (basic, intermediate, advanced). These prompts direct the LLM to generate text that aligns with the intended phishing objective.
2. **Content Generation:** The LLM produces phishing emails based on the prompts, incorporating elements such as directive language (e.g., "click", "verify"), spoofed sender details, and urgent calls to action. The generated emails often exclude overt malicious markers (e.g., suspicious URLs), making them more difficult for traditional detection systems to identify.

3. **Email Distribution:** Attackers distribute phishing emails through common webmail providers (e.g., Gmail, Outlook, Yahoo) to reach a broad audience. By staggering dispatch schedules and avoiding bulk sending, they aim to evade basic spam detection heuristics.
4. **Victims Response:** Recipients are deceived into taking actions such as disclosing personal information or downloading malicious content, driven by the high authenticity and contextual relevance of the phishing emails.

Adopting Large Language Models (LLMs) to create phishing scams provides attackers with significant advantages. LLMs enable the rapid and large-scale generation of phishing content while their user-friendly interfaces ensure accessibility to a broad spectrum of attackers, regardless of their technical expertise. This inclusivity allows even less tech-savvy individuals to implement sophisticated evasion techniques, such as text encoding, browser fingerprinting, and clickjacking, further enhancing the effectiveness of these phishing campaigns.

4. Methodology

This section outlines the end-to-end process employed in this study to generate emails, extract stylometric features, and apply machine learning classification. Fig. 2 provides a visual overview of each stage in this research process.

The next subsection details the email generation process.

4.1. AI-generated email creation

This subsection describes the steps to generate phishing and legitimate emails using GPT-4o. To ensure realism, the emails were crafted with varying levels of sophistication (basic, intermediate, advanced) and diverse themes. These variations simulate the complexity and context often observed in real-world communications (see Sections 4.1.1 and 4.1.2).

Using GPT-4o, we generated a total of 63 phishing emails and 63 legitimate emails between August and September 2024. Subsequently, the generated emails were used to test the robustness of spam filters across different webmail providers.

The decision to use AI-generated legitimate emails instead of human-written emails from older datasets was based on both theoretical and practical considerations. First, stylistic consistency plays a critical role in evaluating phishing detection methods. Research indicates that AI-generated content exhibits distinct stylistic and semantic characteristics compared to human-written text (An, Yang, Yang, & Wang, 2023; Giray, Sevnarayan, & Ranjbaran Madiseh, 2024; Opara, 2024). This observation applies equally to AI-generated emails. Studies by Heiding et al. (2024) and Chataut et al. (2024) underscore the necessity of aligning writing styles between phishing and legitimate email samples to avoid introducing unintended biases. This study ensures that the two classes share stylistic and linguistic consistency by generating both phishing and legitimate emails using the same LLM. This approach minimises potential classification biases that could otherwise result from discrepancies in tone, structure, or linguistic patterns, thereby enhancing the robustness and validity of the results.

Secondly, using AI-generated legitimate emails introduces a unique challenge for the classifier. This approach creates an adversarial scenario where the model must distinguish between two closely resembling classes: AI-generated phishing emails and AI-generated legitimate emails. This setup tests the classifier's ability to leverage stylometric and linguistic features to detect phishing independent of overt markers like malicious URLs or attachments.

Fig. 3 visually represents each component of the email generation process.

The email generation process consists of the following key steps:

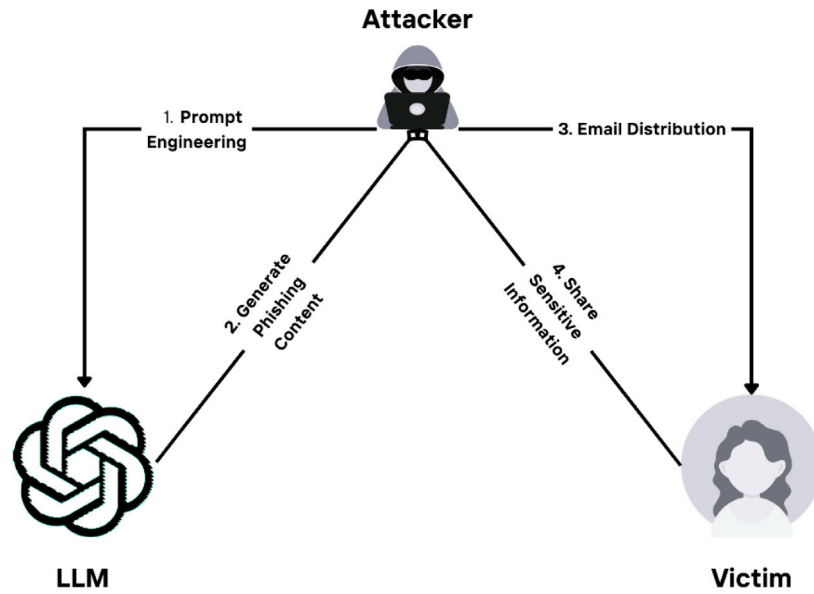


Fig. 1. Threat model to generate phishing emails using LLMs.

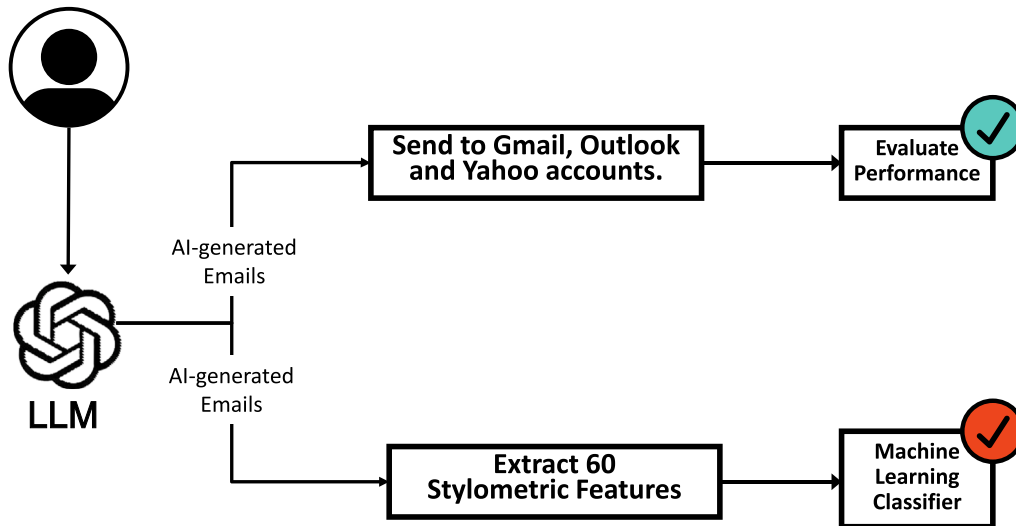


Fig. 2. Overview of this study.

Class Initialisation: The script initialises essential components, including the API key, the Faker Library,⁵ and predefined email themes. These elements form the foundation for generating diverse and realistic email content.

Prompt Engineering: A random email theme is selected, and the Faker library generates realistic fake data, such as names, phone numbers, and links. For phishing emails, the prompts are crafted to mimic legitimate communication while embedding urgency and a call to action.

For instance, a phishing prompt under the *promotional offers* theme might read:

‘Generate a phishing email that appears to be from a reputable organisation, informing the recipient about an exclusive promotional offer or deal and urging them to take immediate action by [call to action]’.

For legitimate emails, the prompt ensures clarity and trustworthiness:

‘Generate a professional email from a reputable or informing the recipient about a promotional offer or deal they might find valuable. Encourage them to learn more by [call to action]. The email should be clear, friendly, and trustworthy’.

By varying the themes, we ensured that each email appeared unique, reducing the likelihood of being flagged by spam filters.

Content Generation: During this stage, the script assembles the email content by integrating the selected theme, the sophistication level (ranging from basic to advanced for phishing emails), the call-to-action statements, and the generated fake data. For phishing emails, the call-to-action statements were designed to create urgency and compel recipients to act, including Downloading the attached invoice and reviewing it or replying with your account details to verify your identity. In contrast, legitimate emails featured professional and trustworthy call-to-action statements, such as clicking the link to view the full details, replying to this

⁵ <https://faker.readthedocs.io/en/master/>

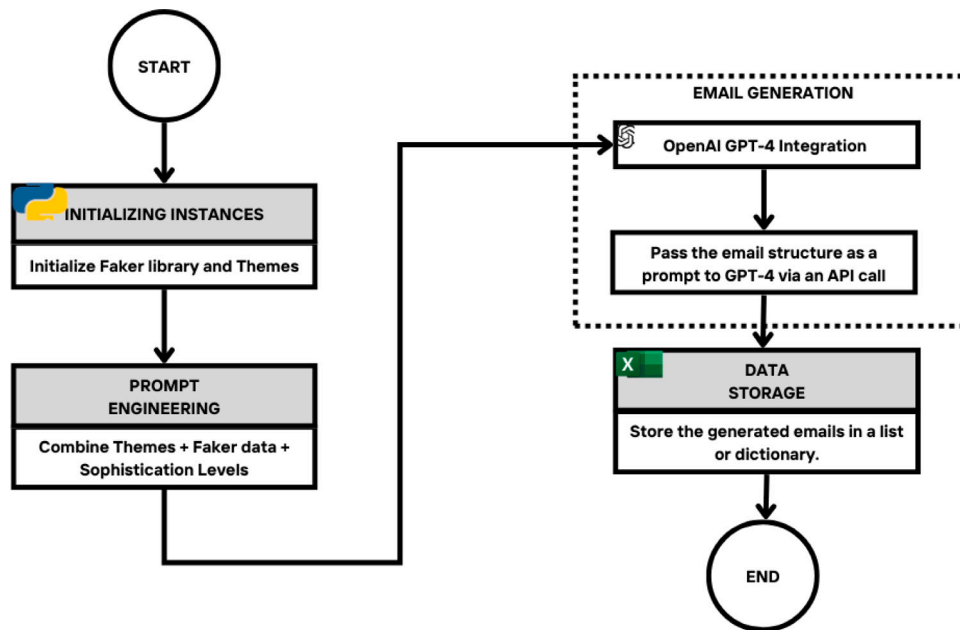


Fig. 3. Components of the email generation process.

email with your questions or interest or calling us at [phone number] for more information. These call-to-action statements were important in tailoring the generated emails to align with their intended purpose, whether deceptive phishing or legitimate communication.

Export and Validation: The generated emails are stored in a CSV file, ensuring they are systematically organised and readily available for subsequent analysis.

Note: We chose GPT-4o for this research due to its suitability for most tasks, as recommended by OpenAI. Other models, such as GPT-4o-mini, GPT-1o-mini, and the legacy model GPT-4, are available on the OpenAI platform and are noted to be faster at reasoning, according to OpenAI reports.⁶ However, GPT-4o was selected as it aligns well with the goals of this research.

Next, we explore in detail the sophistication levels used in the prompts provided to GPT-4o for generating phishing emails.

4.1.1. Phishing email sophistication levels

The prompt fed into GPT-4o was designed to generate 63 phishing emails, categorised into three levels of sophistication: basic, intermediate, and advanced.

Basic Prompt:

‘Generate a simple phishing email asking the user to [call to action].’

Translation into Phishing Email: GPT produces a straightforward phishing email with minimal formatting and a simple message. The language will likely be informal and to the point, without excessive polish or complexity. The call-to-action will be direct, encouraging the recipient to perform an action (like clicking a link or downloading an attachment).

Intermediate Prompt:

‘Create a phishing email with proper formatting and a company logo, requesting the user to [call to action].’

Translation into Phishing Email: GPT creates a phishing email that mimics the appearance of an official company communication. This email will include basic branding elements, such as a fake company logo, and will have a more professional tone. The formatting will be more structured, resembling a standard corporate email.

Advanced Prompt:

‘Generate a highly sophisticated phishing email with urgent language, a spoofed sender address, and professional formatting asking the user to [call to action].’

Translation into Phishing Email: GPT produces a highly convincing phishing email with a professional layout, an urgent tone, and more refined language. The email will appear to come from an official, and it may simulate authentic company branding and formatting to trick the recipient into believing it is legitimate.

The rationale behind generating phishing emails at different sophistication levels is based on research showing that attackers often weigh the potential return on investment (ROI) when crafting phishing emails (Heiding et al., 2024), Parsons, McCormac, Pattinson, Butavicius, and Jerram (2013), Vishwanath, Herath, Chen, Wang, and Rao (2011). According to the studies by Heiding et al. (2024), human-written phishing emails take about 15 min to create, with an opportunity cost of approximately \$8.64 per attempt. Given the best-case success rate of 19% considered in the study, attackers must earn at least \$0.41 per successful attack to break even when phishing emails are sent to 112 users.

For high-value targets, such as corporate executives or government officials, a sophisticated attack's return on investment (ROI) can often justify the extra effort involved. However, when targeting a broader audience, the ROI may not support the expense of using advanced techniques. Instead, attackers may rely on basic or intermediate attacks, which can be distributed to more potential victims at a lower cost, improving success rates through sheer volume. Additionally, research in social engineering also suggests that simplicity and directness can sometimes be more effective than complexity (Jakobsson & Myers, 2006), Jagatic, Johnson, Jakobsson, and Menczer (2007). Basic attacks

⁶ <https://platform.openai.com/docs/models>

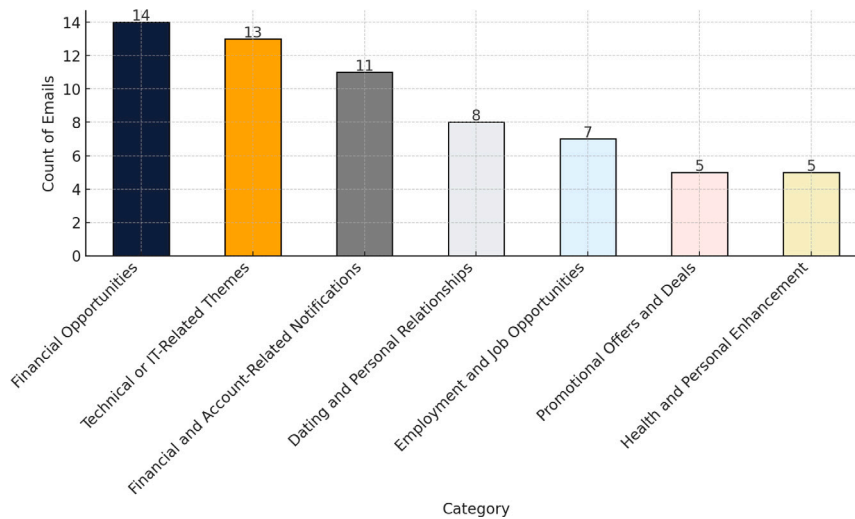


Fig. 4. AI-generated emails by themes.

often exploit common psychological triggers such as fear, urgency, or greed, which are potent motivators for action. These triggers can be effective even without sophisticated techniques, particularly when targeting individuals with low cybersecurity literacy.

In total, the curated dataset contains a balanced distribution of phishing emails across all sophistication levels. Specifically, 1/3 of the emails are classified as *Basic*, representing straightforward and unsophisticated tactics. Another 1/3 fall under the *Intermediate* category, where attackers employ moderately advanced techniques. The remaining 1/3 are categorised as *Advanced*, indicating highly sophisticated methods likely targeting specific individuals.

Furthermore, we calculated the cost of generating 17,750 words of 63 phishing emails using GPT-4o based on OpenAI's API pricing structure.⁷ As of the time of this research, GPT-4o charges roughly \$0.03 per 1,000 tokens, where one token equals about four characters or 0.75 words. Creating this volume of phishing emails at the time of this publication would cost approximately \$0.71.

4.1.2. Email themes

The themes for the AI-generated emails were carefully selected to cover a broad range of common phishing tactics. These themes were chosen based on their prevalence in reported phishing incidents (APWG, 2024) and their effectiveness in deceiving recipients (Cofense, 2023; Verizon, 2024). We identified seven key themes: Promotional Offers and Deals, Financial Opportunities, Employment and Job Opportunities, Technical or IT-related Topics, Health and Personal Enhancement, Financial and Account-Related Notifications, and Dating and Personal Relationships.

Including various themes in phishing email generation is essential, as it mirrors the diverse and evolving nature of real-world phishing attempts.

Fig. 4 shows that the largest category, **Financial Opportunities**, represents 22.2% of the dataset created by GPT-4o. Close behind, **Technical or IT-related Themes** make up 20.6%, typically involving fake security alerts or software updates. **Financial and Account-Related Notifications** account for 17.5%, exploiting fears over financial security.

Other themes include **Dating and Personal Relationships** (12.7%), **Employment and Job Opportunities** (11.1%), and **Promotional Offers and Deals** (7.9%). Finally, **Health and Personal Enhancement** rounds out the dataset with 7.9%. This range of themes mirrors the diverse strategies used to target various victim profiles.

4.2. Sending procedure and mitigating non-textual biases

According to the (APWG, 2024) report for Q2 2024, 72% of phishing attacks were launched using free webmail domains, a slight decrease from 73% in the previous quarter. Gmail was the most frequently used platform, accounting for 72.4% of phishing-related webmail accounts, followed by Microsoft's webmail services at 16.3%.

To evaluate the effectiveness of email service providers in preventing AI-generated phishing emails, two sets of email accounts were created across Gmail, Outlook, and Yahoo. One set was designated for sending the 63 AI-generated emails, while the other received them. This controlled setup enabled a consistent analysis of each platform's spam filter performance for phishing and legitimate AI-generated emails.

The emails were sent in two phases between August 1–4, 2024, and September 13–15, 2024, at staggered intervals ranging from 10 to 60 min. This scheduling avoided triggering bulk detection systems, which often flag large volumes of emails sent in quick succession. By varying the timing and sending emails at different times of day, the experiment aimed to simulate natural human behaviour and assess whether time-related factors influenced spam filter performance.

Default spam filter settings were used throughout the experiment to replicate the typical user experience (Dutta & Bansal, 2016). Additionally, no malicious links or attachments were included in the emails. This focus on textual content was significant for two reasons:

- Phishing links are often unique to recipients ("zero-day") and frequently bypass existing blacklist-based detection systems (Cidon et al., 2019; Jain & Gupta, 2018).
- The study aimed to determine whether spam filters could identify phishing attempts based solely on AI-generated text, thereby isolating the role of stylometric features.

Note. The commercial webmail services evaluated in this study, which include Gmail, Outlook, and Yahoo, utilise proprietary spam detection systems incorporating machine learning, heuristic analysis, and user feedback. These systems are constantly evolving to counter new threats. More details on their mechanisms can be found in resources such as Google's Transparency Reports,⁸ Microsoft Defender⁹ documentation, and Yahoo's help pages.¹⁰

⁸ <https://transparencyreport.google.com/>

⁹ <https://learn.microsoft.com/en-us/microsoft-365/security/office-365-security/anti-spam-protection>

¹⁰ <https://help.yahoo.com/kb/mail>

⁷ <https://openai.com/api/pricing/>

Despite the controlled methodology, it is important to note that the experiment was one-shot, meaning the filters did not have repeated exposure to any particular sender or content. Consequently, the spam filters may not have had the opportunity to “learn” to reject emails from specific senders or patterns, as would occur with persistent phishing campaigns. This ensures that the results reflect the performance of spam filters in a one-time interaction scenario, focusing solely on the intrinsic properties of the email content.

4.3. Stylometric feature extraction

This section discusses extracting 60 stylometric and linguistic features from each email's text. These features range from word category distributions to readability scores, and they play a critical role in detecting AI-generated phishing attempts.

The primary objective of the stylometric feature extraction process is to uncover distinctive linguistic and structural patterns in both phishing and legitimate emails. These patterns serve as reliable markers that enable robust machine learning models to identify phishing attempts based solely on textual characteristics. This step is crucial for advancing phishing detection beyond traditional methods, often relying on external indicators.

While existing research predominantly emphasises network and HTML-based features such as clickable hyperlinks, JavaScript, and domain age, our study focuses exclusively on the textual content of email subjects and bodies. Although network and HTML-based features remain valuable for phishing detection, stylometric analysis complements these approaches by focusing on the language and writing style of email content. Stylometric features are particularly effective in addressing zero-day attacks, where conventional indicators such as suspicious links or known domains may not be present (Dewan et al., 2014). By analysing linguistic and structural patterns that are harder for attackers to manipulate consistently, stylometric techniques enhance detection accuracy and provide a robust layer of defence (Duman et al., 2016), Gallo et al. (2024).

Consequently, a total of 60 stylometric and linguistic features were extracted from each AI-generated phishing and legitimate email, drawing inspiration from prior studies (Dewan et al., 2014), Duman et al. (2016), Gallo et al. (2024), and Toolan and Carthy (2010). Among these, 47 features are novel in the context of phishing email detection. To ensure the relevance and validity of the stylometric features, specific word choices in features such as the Imperative Verbs Count were guided by prior research on commonly observed patterns in phishing and legitimate emails. For example, imperative verbs such as “click”, “verify”, “submit”, “download”, and “update” were included, as they are frequently employed in phishing emails to evoke a sense of urgency and prompt immediate action from recipients.

The features are organised into several key categories, as shown in Fig. 5.

- **Lexical Features (16.7%)**

These features examine basic text properties such as Word Count, Character Count, and Average Word Length. They help determine how much content is in an email and whether it follows normal patterns. For example, emails with short or unusually long word counts might indicate suspicious behaviour. Another measure, Lexical Diversity (the ratio of unique words to total words), shows how varied or repetitive the text is, which can hint at its legitimacy.

- **Syntactic Features (18.3%)**

These focus on the structure and grammar of an email. Metrics like Sentence Count, Clause Density, and Pronoun Density analyse how well-organised the email is. Phishing emails often overuse pronouns like “you” or “we” to make them feel personal, while legitimate ones have a more balanced structure. Additionally, Function Word Density measures the use of filler or connector words, which may be overused in spam to sound natural.

- **Punctuation Features (16.7%)**

This category examines punctuation patterns, including how often periods, commas, exclamation marks, or colons are used. Phishing emails might overuse exclamation marks to create urgency, while unusual punctuation patterns – like too many colons in short messages – can signal suspicious formatting.

- **Readability Scores (16.7%)**

These scores assess how easy or difficult the email is to read using metrics like the Flesch Reading Ease or SMOG Index. Phishing emails often use overly simple or complex language, while legitimate emails typically have a moderate, natural readability level. These scores help identify emails that may have been artificially created or poorly written.

- **Word Category Features (11.7%)**

This group focuses on the types of words used. For example, phishing emails often include a high number of Imperative Verbs (like “click”, “verify”), Technical Jargon (to appear official), or Promotional Words (to push offers). These word choices can reveal the intent behind the email.

- **Email-Specific Features (6.7%)**

These features analyse email-related details like multiple “@” symbols, mentions of attachments, or links. Phishing emails may include an abnormal number of clickable elements or repeated requests to verify personal details, while legitimate emails follow consistent patterns.

- **Complexity Features (5.0%)**

These measure the structural or semantic complexity of the text. For example, Bigram Count and Trigram Count (how often 2- or 3-word combinations appear) show how advanced the text is. Word Length Variation (differences in word lengths) helps distinguish between natural and artificially generated text, as spam often uses repetitive or unusually varied word lengths.

- **Stylistic Features (8.3%)**

This category captures the tone of the email. Features like Politeness Markers (e.g., “please”, “thank you”) or Urgency Markers (e.g., “immediately”, “urgent”) highlight whether the email is trying to pressure the recipient. Phishing emails often use pushy or aggressive language, while legitimate emails maintain a more balanced tone.

The comprehensive set of features is detailed in Table 1.

4.4. Machine learning application

The task of detecting AI-generated phishing emails using stylometric features was formulated as a binary classification problem. AI-generated phishing emails were treated as negative samples, and AI-generated legitimate emails were treated as positive samples. Specifically, the 60 stylometric features extracted in Section 4.3 were applied to machine learning models, classifying each email as either phishing or legitimate (See Fig. 6).

Four machine learning models were used in this study. Logistic Regression, Kernel SVM, Random Forest, and XGBoost. These models were chosen for their widespread use in supervised classification tasks and their ability to handle complex data patterns effectively. Each model was evaluated to determine its effectiveness in identifying AI-generated phishing emails, providing a comparative analysis of performance metrics such as accuracy, precision, and recall.

The proposed phishing detection algorithm leverages these models to classify emails based on stylometric features. It follows a structured process that includes data preprocessing, feature extraction, classifier training, and evaluation, as outlined in Algorithm 1. This algorithm ensures a systematic approach to extracting meaningful patterns from email text and applying machine learning for classification.

Among the models used, *Logistic Regression* (LR) is a statistical model used for binary classification. It models the probability that an input

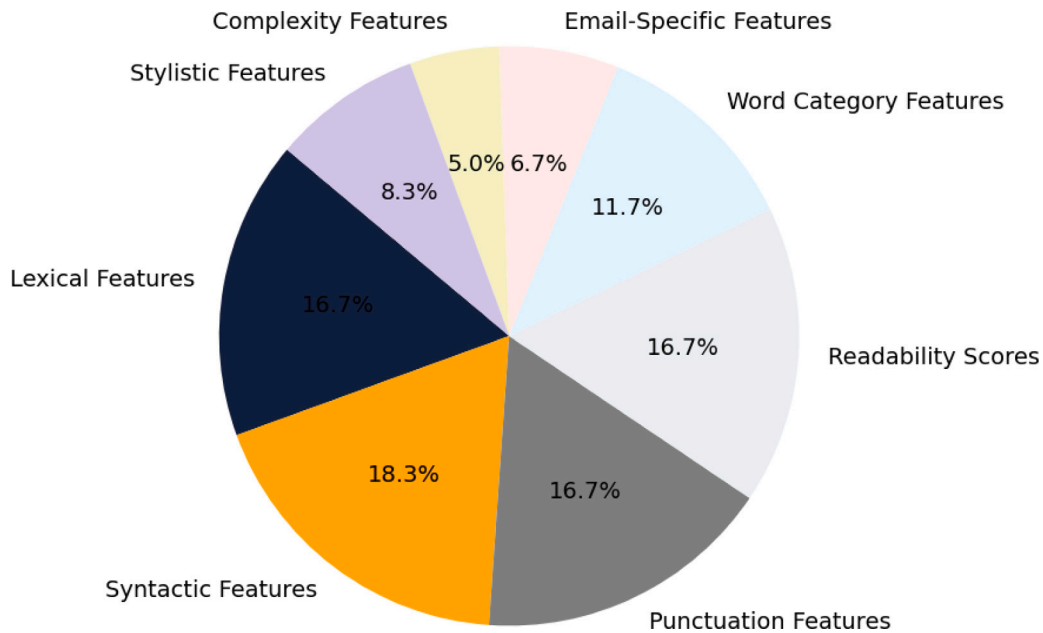


Fig. 5. A pie chart of the categorisation of the extracted stylometric features.

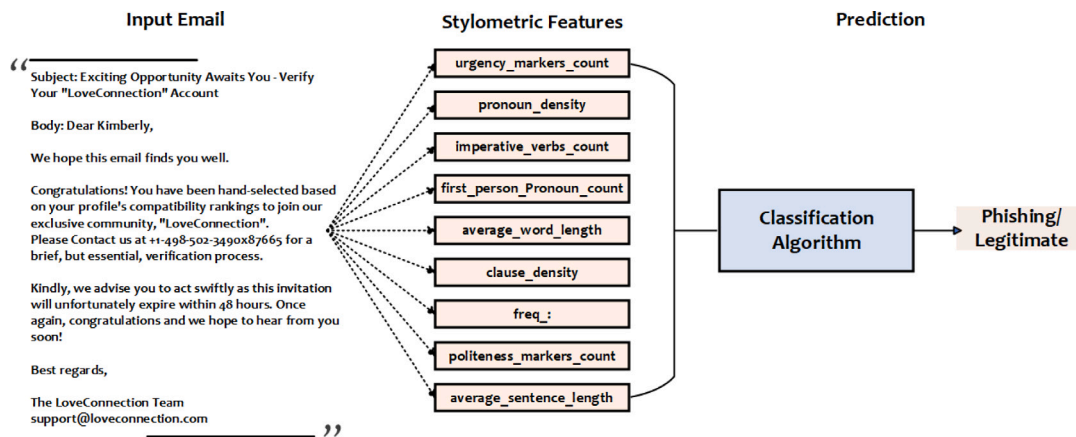


Fig. 6. Machine learning implementation: First, textual characteristics are extracted with a feature-engineering approach. Then, a classification algorithm is used to build the model.

belongs to a particular class using the logistic function, which maps real-valued numbers to the $[0, 1]$ interval. LR is particularly effective when the relationship between features and the log odds of the outcome is linear (Hosmer, Lemeshow, & Sturdivant, 2013). Its simplicity, efficiency, and interpretability make it widely used in many domains.

SVM is a supervised learning model for classification tasks. It works by finding an optimal hyperplane that maximises the margin between two classes in the feature space. SVM uses the kernel trick for non-linear boundaries, which maps input features into a higher-dimensional space to enable linear separation (Cortes, 1995). This makes SVM effective for both linear and non-linear classification problems.

Random Forest is an ensemble learning method that builds multiple decision trees during training and combines their outputs to make a final prediction. It is a bagging technique that reduces variance and prevents overfitting. Each tree is built from a random subset of the training data and features, selected with replacement, and the final prediction is based on the majority vote for classification (Breiman, 2001). Random Forests are robust against overfitting, handle high-dimensional data, and provide feature importance estimates.

XGBoost implements gradient-boosted decision trees designed for speed and performance. Gradient boosting is an iterative technique

where new models are trained to correct the errors of previous models, typically using decision trees as base learners. XGBoost's objective function includes a regularisation term to control model complexity (Friedman, 2001). It optimises predictions using gradient descent. XGBoost is highly scalable and efficient with large datasets. It also supports advanced features like regularisation, parallel processing, and out-of-core computation, making it a powerful tool for supervised learning.

4.5. Experimental setup and metrics

All experiments were conducted on the Google Colab platform. The default settings from the Scikit-Learn library¹¹ were used for Logistic Regression, Support Vector Machine (SVM), and XGBoost models, while the Random Forest model was trained with 100 trees. The dataset, which includes 60 stylometric features, was split into training and test sets using an 80/20 split to ensure reliable results. This split allows the models to train on 80% of the data and be evaluated on the remaining

¹¹ <https://scikit-learn.org/stable/>

Table 1
Stylometric features extracted from the emails.

Lexical features		
Feature	Description	Reference
Word count	This feature counts how many words appear in the email by splitting the text into separate tokens based on spaces or punctuation and then adding up the total number of these tokens.	Gallo et al. (2024), Duman et al. (2016), Dewan et al. (2014), Toolan and Carthy (2010)
Character Count	This feature counts the total number of characters in the email's text, which includes letters, digits, punctuation, and whitespace.	Gallo et al. (2024), Duman et al. (2016), Dewan et al. (2014), Toolan and Carthy (2010)
Average Word Length	This feature computes the mean number of characters per word. It adds up the length of each word (in characters) and then divides it by the total number of words in the email.	New
Sentence Count	This feature counts the total number of sentences in the email by looking for specific sentence-ending punctuation marks, such as periods ("."), exclamation marks ("!"), or question marks ("?").	Duman et al. (2016)
Average Sentence Length	This feature calculates how many words there are, on average, in each sentence. It divides the total number of words in the email by the total number of sentences.	New
Unique Word Count	This feature counts how many distinct words appear in the email by taking the set of all words (i.e., removing duplicates) and tallying it up.	Dewan et al. (2014), Toolan and Carthy (2010)
Lexical Diversity	This feature assesses the proportion of unique words to total words, indicating how varied the vocabulary is. It divides the number of unique words by the total number of words.	New
Number of Emails	This feature checks for the presence of the "@" symbol, indicating email addresses, which can show if the text references multiple email addresses.	Toolan and Carthy (2010)
Uppercase Word Count	This feature counts how many words are entirely uppercase.	Duman et al. (2016)
Uppercase Word Count Ratio	This feature counts how many words are entirely uppercase (which can convey emphasis or shouting) and compares that count against the total word count.	New
Complex Words Count	This feature counts how many words exceed a certain length threshold—often, words over six characters are considered "complex."	New
Average Syllables per Word	This feature estimates the average number of syllables in each word. A function calculates the total number of syllables in the entire text, and then that value is divided by the total number of words.	New
Syntactic Features		
Feature	Description	Reference
Comma, Semicolon, and Colon Counts	These features count how many times commas (","), semicolons (";"), and colons (":") appear in the text, giving a sense of the writer's syntactic style.	New
Exclamation Count, Quotation Count, Dash Count	Similarly, these features record how many exclamation marks ("!"), quotation marks ("\""), or dashes ("—") appear in the text, which can reflect the tone or structure of the writing.	New
Sentence Complexity Ratio	This feature measures how often certain conjunctions or linking words appear per sentence, indicating how complex or compound the sentences tend to be. For this study, the specific conjunctions considered are "and," "but," "or," and "because"	New
Clause Density	This feature takes the sentence complexity ratio (the count of linking words per sentence). It divides it by the total number of sentences again, further measuring how many subordinate clauses may appear across the entire text.	New
Pronoun Density	This feature divides the total pronoun count by the total number of words, showing how heavily the email relies on pronouns compared to other words.	New
Preposition Density	This feature calculates how often prepositions appear relative to the number of words. For this study, the specific prepositions considered are "in," "on," "at," "by," and "with".	New
Function Word Density	This feature measures the usage rate of common function words compared to the total word count. For this study, the specific common function words considered are "the," "is," "at," "which," and "on".	New

(continued on next page)

Table 1 (continued).

Punctuation Features		
Feature	Description	Reference
Frequency of the following punctuations: '.', '!', '(', ')', ' '_', '“”, '”’, '(', ')', ' '/', '‘’, '’’,	Each of these frequencies calculates how often a punctuation symbol appears relative to the total number of characters in the email.	New
Punctuation Variety	This feature counts the number of different punctuation symbols that appear at least once in the email, indicating how varied the punctuation usage is.	New
Readability Scores		
Feature	Description	Reference
Flesch Reading Ease	This feature implemented with text-analysis libraries <code>textstat</code> Uses average sentence length and average syllables per word to generate a readability score. $\text{Flesch Reading Ease} = 206.835 - 1.015 \times \left(\frac{\text{total words}}{\text{total sentences}} \right) - 84.6 \times \left(\frac{\text{total syllables}}{\text{total words}} \right)$	Gallo et al. (2024)
SMOG Index	Uses the count of words with at least three syllables (“polysyllables”), scaled by the total number of sentences, to approximate a U.S. grade level. $\text{SMOG Index} = 1.0430 \times \sqrt{30 \times \frac{\text{number of polysyllables}}{\text{number of sentences}}} + 3.1291.$	New
Dale--Chall Readability Score	Incorporates the proportion of “difficult words” (words not on a specific familiar-word list) and average sentence length to estimate reading difficulty. $\text{Dale} - \text{Chall Readability Score} = 0.1579 \times (\% \text{ of difficult words}) + 0.0496 \times \left(\frac{\text{total words}}{\text{total sentences}} \right) + 3.6365.$	Gallo et al. (2024)
Coleman--Liau Index (Liau, Bassin, Martin, & Coleman, 1976)	It relies on the average number of letters per 100 words and the average number of sentences per 100 words rather than counting syllables directly. $\text{Coleman} - \text{Liau Index} = 0.0588 \times L - 0.296 \times S - 15.8,$ where L = average number of letters per 100 words, S = average number of sentences per 100 words.	Gallo et al. (2024)
Gunning Fog Index (Bogert, 1985)	Measures the complexity of text based on average sentence length and percentage of complex words (often, those with three or more syllables). $\text{Gunning Fog Index} = 0.4 \times \left(\frac{\text{total words}}{\text{total sentences}} + 100 \times \frac{\text{complex words}}{\text{total words}} \right).$	New
Word Category Features		
Feature	Description	Reference
Number of Pronouns	This feature counts how often common personal pronouns occur in the email. For this study, the specific pronouns considered are “I,” “you,” “he,” “she,” “it,” “we,” and “they”.	New
First-Person and Second-Person Pronoun Counts	These features measure how many times the email uses first-person, specifically “I” and “we”, or second-person, specifically “you” references, which can suggest a more personal or direct tone.	New
Imperative Verbs Count	This feature counts the presence of directive or commanding verbs in the email. For this study, the specific verbs considered are “click,” “verify,” “submit”, “download”, and “update”.	New
Modal Verbs Count	This feature checks how frequently modal verbs occur in each email, indicating possibility or necessity. For this study, the specific modal verbs considered are “can,” “could,” and “should”.	New
Uncertainty Adverbs Count	This feature identifies how often “maybe,” “possibly,” and “perhaps” appear in each email, reflecting uncertainty or tentativeness.	New
Technical Jargon Count	This feature checks for words associated with technical or specialised domains. For this study, the specific technical jargon considered are “security,” “account,” “update”, and “technical”.	Dewan et al. (2014)
Promotional Words Count	This feature counts how often the promotional words below “offer,” “deal,” and “free” appear in the email, which can signal marketing or promotional content.	Dewan et al. (2014)

(continued on next page)

Table 1 (continued).

Email-Specific Features		
Feature	Description	Reference
Number of Emails	This feature checks for the presence of the “@” symbol, indicating email addresses, which can show if the email contains multiple email addresses.	Toolan and Carthy (2010)
Uppercase Word Count/Ratio	These features track how many words are entirely uppercase (which can convey emphasis or shouting) and compare that count against the total word count.	New
Number of Attachment Mentions	This feature simply counts how many times the word “attachment” (or a similar term) appears, which may indicate an email that references files.	Duman et al. (2016)
Complexity Features		
Feature	Description	Reference
Bigram and Trigram Counts	These features implemented using the <i>CountVectorizer</i> measure the total number of 2-word and 3-word sequences in the text (as identified by n-gram vectorisers). A higher count could indicate more unique combinations of words.	Duman et al. (2016)
Word Length Variation	This feature calculates the standard deviation (statistical variability) of the lengths of all words. If it is high, it means the email contains a mix of very short and very long words.	New
Stylistic Features		
Feature	Description	Reference
Politeness Markers Count	This feature counts how frequently the polite words “please,” “thank,” and “appreciate” occur in each email.	New
Aggressiveness Markers Count	This feature counts how often words implying urgency or aggression appear. For this study, the specific aggressive markers are “must,” “now,” and “immediately”.	New
Urgency Markers Count	This feature measures the presence of words or phrases indicating urgency. For this study, the specific urgency markers are “urgent,” “asap,” and “immediately”.	Gallo et al. (2024)
Conditional Phrases Count	This feature captures how many times “if” or “unless” occur in each email, which may indicate conditional statements or requests.	New
Personalisation Markers Count	This feature measures the number of words aimed at personalising the email. For this study, the specific Personalisation markers are “[Recipient’s Name],” “you,” and “your”.	New

20%. Feature standardisation was performed using the *StandardScaler*, which scales the features to have zero mean and unit variance. This step is essential for models like Logistic Regression and SVM, which are sensitive to the scale of input data (see Table 2).

4.5.1. Evaluation metrics

The performance of the machine learning models was evaluated using *recall*, *precision*, and *accuracy*. Recall is calculated as:

$$\text{Recall} = \frac{TP}{TP + FN}$$

Where TP represents True Positives and FN represents False Negatives. Precision is computed as:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Where FP represents False Positives. Accuracy was determined using:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Where TN represents True Negatives.

The *Area Under the Curve* (AUC) was also calculated to assess the model’s ability to distinguish between two classes (legitimate vs. phishing emails). A higher AUC value indicates better model performance.

Section 5.3 provides a detailed analysis of the results from the four machine learning algorithms.

5. Results

To guide the discussion of the results conducted in this study, we look at the following research questions:

- How do email providers’ spam filters (Gmail, Outlook, Yahoo) perform against AI-generated phishing emails compared to legitimate ones?
- What are the semantic and lexical differences between AI-generated phishing emails and existing human-written phishing emails (SpamAssassin or CEAS datasets)?
- What are the performance differences among the machine learning classifiers when applied to stylometric features extracted from phishing and legitimate emails?
- How does removing specific features (via ablation studies) impact the performance of machine learning classifiers in detecting phishing emails?

To answer RQ1, in Section 5.2, we evaluated the performance of 3 email providers (Gmail, Outlook, and Yahoo) in detecting AI-generated phishing emails compared to legitimate ones. To address RQ2, in Section 5.1, we conducted semantic and lexical content analysis, comparing AI-generated phishing emails with the SpamAssassin and CEAS datasets to identify common phishing patterns such as authority, urgency, and emotional appeals. For RQ3, in Section 5.3, we extracted 60 stylometric features from the emails and applied them to four machine learning classifiers (Logistic Regression, SVM, Random Forest, and XGBoost) to evaluate their effectiveness in distinguishing phishing emails. Finally, to address RQ4, in Section 5.3.2, we performed an ablation study to assess the impact of removing key features on the XGBoost model’s performance, demonstrating the relative importance of stylometric features in phishing detection.

Algorithm 1: Phishing Email Detection Using Stylometric Features**Input:**

- $\mathcal{E} = \{e_1, e_2, \dots, e_N\}$: Dataset of N emails, where each email e_i is a string.
- $\mathcal{F} = \{f_1, f_2, \dots, f_m\}$: Set of m stylometric features.
- C : A machine learning classifier (e.g., XGBoost).

Output: $\hat{y} = \{\hat{y}_1, \hat{y}_2, \dots, \hat{y}_t\}$: Predicted labels for t test emails.

Step 1: Data Preprocessing

foreach $e_i \in \mathcal{E}$ **do**

- Tokenise e_i into words and sentences.
- Compute basic text metrics (e.g., word count, sentence count).
- Normalise e_i by removing stopwords and punctuation.

Step 2: Feature Extraction

foreach $e_i \in \mathcal{E}$ **do**

foreach $f_j \in \mathcal{F}$ **do**

Compute $f_j(e_i)$ as:

$$f_j(e_i) = \frac{1}{|e_i|} \sum_{k=1}^{|e_i|} g(e_{i,k}),$$

Where $g(e_{i,k})$ is the feature-specific computation (e.g., checking for imperative verbs).

Store feature vector $F(e_i) = [f_1(e_i), f_2(e_i), \dots, f_m(e_i)]$.

Step 3: Classifier Training

Split \mathcal{E} into training set $\mathcal{E}_{\text{train}}$ and test set $\mathcal{E}_{\text{test}}$.

Train the classifier C using:

$$C_{\text{train}} = \text{train}(F(\mathcal{E}_{\text{train}}), y_{\text{train}}),$$

where y_{train} is the set of ground truth labels for $\mathcal{E}_{\text{train}}$.

Step 4: Classification and Evaluation

foreach $e_i \in \mathcal{E}_{\text{test}}$ **do**

Predict label \hat{y}_i using:

$$\hat{y}_i = C_{\text{train}}(F(e_i)).$$

Compute evaluation metrics (e.g., accuracy, precision, recall) over \hat{y} and y_{test} .

5.1. Phishing email evaluation (content analysis and comparison with spamassassin/CEAS)

To assess the authenticity of the AI-generated phishing emails, we conducted some content analysis to evaluate whether the emails follow patterns found in actual phishing attempts, such as using authority, urgency, or emotional appeals. Furthermore, using qualitative metrics to show similarities in terms of structure and language, we compared the GPT-4o-generated phishing emails with publicly available phishing datasets [SpamAssassin](#) and [CEAS_08 phishing email datasets](#). We chose these datasets as they are the most widely used in phishing email analysis ([Toolan & Carthy, 2010](#)), [Moradpoor, Clavie, and Buchanan \(2017\)](#) and [Karim, Azam, Shanmugam, Kannoopatti, and Alazab \(2019\)](#). The SpamAssassin dataset contained 1642 phishing emails, while the CEAS_08 contained 21,803 phishing emails.

5.1.1. Semantic content analysis of the AI-generated phishing emails

In the content analysis of the AI-generated emails, we specifically examined the presence of *authority cues*, *urgency*, and *emotional appeals*—patterns that are commonly used in phishing attempts ([Abdelhamid, Ayesha, & Thabtah, 2014](#)), [Barracough, Hossain, Tahir, Sexton, and Aslam \(2013\)](#), and [Form et al. \(2015\)](#). These elements are important in persuading recipients to engage with phishing emails, as they create a sense of legitimacy and pressure to act. Regarding *authority*, the AI-generated emails frequently contained words such as “official”,

“verify”, and “account”. These terms add an authoritative tone to the emails, making them appear as if they are coming from legitimate organisations. This strategic use of authority-related language helps establish trust, a key tactic in phishing.

For *urgency*, terms like “urgent”, “immediately”, and “deadline” were prominent in the AI-generated emails. Incorporating these words creates a sense of time pressure, compelling the recipient to act quickly without carefully considering the content of the email. This is a well-documented tactic in phishing ([Alhogail & Alsabih, 2021](#); [Gallo et al., 2024, 2021](#)), as it prevents recipients from scrutinising the email too closely. In terms of *emotional appeals*, phrases such as “exclusive”, “opportunity”, and “reward” were frequently used. These emotional triggers are designed to entice recipients by offering something valuable or time-sensitive.

The strong presence of these three elements – authority, urgency, and emotional appeals – demonstrates that AI-generated phishing emails effectively mimic human-written phishing tactics.

5.1.2. Lexical comparison of text length and word length with existing human-written phishing emails

To further demonstrate the authenticity of the AI-generated phishing emails, we compared their *average text length* and *average word length* with human-written phishing emails from the Spamassassin and CEAS_08 datasets ([Arifa Islam, 2023](#)). The following table provides a comparison of these metrics:

[Table 3](#) demonstrates that the AI-generated emails are shorter than human-written Spamassassin emails but longer than CEAS emails. This suggests that AI-generated phishing emails are designed to be concise yet informative, striking a balance between length and content. In contrast, human-written Spamassassin Emails exhibit the longest average text and word length, which may be attributed to the inclusion of more detailed technical information or longer, complex content. The human-written CEAS_08 emails, on the other hand, are shorter but feature more complex words, likely reflecting their focus on promotional or product-related spam rather than phishing attempts designed to deceive.

Furthermore, we conducted a word frequency analysis to identify the most commonly used words in AI-generated and human-written phishing emails. The top 10 most frequent words for each group are as follows:

- **AI-Generated Emails:** The most common words in AI-generated emails include “cnn”, “html”, “http”, “index”, and “video”. This suggests that AI-generated emails rely heavily on embedding links and media elements to enhance legitimacy or entice recipients to click on external content.
- **Spamassassin:** Similar to AI-generated emails, common words such as “html”, “video”, “index”, and “cnn” were frequently used. These terms indicate the use of multimedia content, a tactic often employed in phishing to make emails appear more sophisticated and professional.
- **CEAS_08:** Words such as “cnn”, “com”, “http”, and “www” were among the most frequently used in CEAS emails. This reflects an emphasis on external links and trusted domains, which is common in promotional emails or phishing attacks seeking to build trust through familiarity.

The word frequency analysis highlights that both AI-generated and human-written emails rely on embedding URLs and multimedia elements to enhance credibility and engagement.

Overall observations. Structural Similarities: The analysis reveals that AI-generated phishing emails closely follow the structural patterns of human-written phishing emails. They frequently embed URLs, use authority cues, and employ urgency and emotional appeals, much like their human-written counterparts.

Table 2
Hyperparameter summary for all models

Model	Hyperparameters	Values
Logistic Regression	random_state	42
SVM	random_state, probability	42, True
Random Forest	n_estimators, random_state	100, 42
XGBoost	use_label_encoder, eval_metric, random_state	False, 'logloss', 42

Table 3
Comparison of text length and word length for AI and human-written emails.

Category	Avg text length	Avg word length
AI Emails	1814.06	5.60
Spamassassin	2557.66	8.97
CEAS_08	845.42	9.36

Linguistic Complexity: The human-written emails from the Spamassassin and CEAS datasets used more complex language and technical jargon than AI-generated emails. However, despite being less complex, AI-generated phishing emails are still highly effective at mimicking the fundamental patterns of phishing attacks, using simpler but equally persuasive language.

5.2. Spam filter performance

This section details the performance of major email service providers (Gmail, Outlook, Yahoo) when handling AI-generated phishing and legitimate emails.

As shown in Table 4, from the 63 phishing emails sent from the Gmail account, 4 were flagged as spam before delivery. These included 2 emails of intermediate sophistication, 1 basic and 1 advanced. Out of the 63 emails sent from the Gmail account, 59 were successfully delivered to the controlled accounts. Specifically, Gmail had 51 successful deliveries, resulting in a bypass rate of 86.44% across various themes. Yahoo's spam filters proved the most stringent, allowing only 10.17% of emails through, compared to Outlook's 96.61% bypass rate.

Out of the 63 phishing emails sent from the Yahoo account, all were delivered except one whose subject contained Unicode characters. For the 62 emails successfully sent to the controlled accounts, Microsoft Outlook delivered 61 emails, resulting in a bypass rate of 98% across various themes. Gmail showed even greater permissibility, with all emails bypassing its filters, leading to a bypass rate of 100%. In contrast, Yahoo demonstrated the most stringent filtering, allowing 45 emails to pass through, resulting in a bypass rate of just 27.86%.

As a counter-experiment, we also tested what happens when receiving legitimate emails. The results in Table 5 highlight significant inconsistencies in how email providers handle AI-generated legitimate emails. Yahoo, in particular, shows a high rate of false positives, flagging 66.7% and 58.7% of emails from both Gmail and Outlook. In contrast, Outlook performs flawlessly, allowing all emails to go through without issue. Gmail, while better than Yahoo, still flagged 19% of emails from Outlook.

It is important to consider that the flagging of emails, particularly the legitimate ones, as spam may be influenced by factors beyond their content, such as how email providers perceive the volume and pattern of the emails being sent. In this experiment, the high false-positive rate for the AI-generated legitimate emails observed in platforms like Yahoo may result from the emails being classified as potential nuisance or bulk emails, which often trigger spam filters regardless of content accuracy.

These findings indicate that current filtering mechanisms struggle to accurately process AI-generated emails, suggesting the need for improvements to better address new, AI-driven challenges in email communication.

From a security perspective, results suggest that providers such as Gmail, Outlook and Yahoo, to a lesser extent, may not currently utilise the full range of tools available for detecting phishing. To enhance their

systems, they might consider incorporating or improving stylometric and linguistic analysis features as an additional layer of defence. While they currently employ various methods, such as analysing network parameters and metadata, adding this layer of textual analysis and adopting a multilayered security approach could reduce the opportunities for attackers to bypass their existing defences and potentially increase their overall robustness against phishing threats.

It is important to recognise the trade-offs from both user and provider perspectives. Excessive false positives hinder user experience, potentially causing important legitimate messages to be missed and eroding user trust. Conversely, false negatives pose severe security risks, allowing phishing emails to reach inboxes undetected. Striking the right balance is challenging: providers must maintain stringent security without excessively filtering out legitimate communications. Understanding these trade-offs is key to designing more refined and user-centric spam filters.

This highlights the need for balanced, informed detection strategies incorporating stylometric insights.

5.3. Machine learning results

This section presents a detailed analysis of the results from four machine learning algorithms applied to 60 stylometric features extracted from AI-generated phishing and legitimate emails.

Table 6 summarises the accuracy, precision, recall, F1-score, and AUC score for each of the four machine learning algorithms.

Table 7 summarises the confusion matrix values for each algorithm. *True Negatives* (Legitimate Emails Correctly Identified), *False Positives* (Legitimate Emails Incorrectly Classified as Phishing), *False Negatives* (Phishing Emails Incorrectly Classified as Legitimate), and *True Positives* (Phishing Emails Correctly Identified) are reported.

From the classification report in Table 6, the XGBoost model achieved an accuracy of 96%, correctly classifying the vast majority of emails in the test set. This high accuracy demonstrates the model's effectiveness in distinguishing phishing emails from legitimate ones. The Logistic Regression model achieved 92% accuracy, showing it to be similarly effective. The Random Forest model also achieved 92%, performing on par with Logistic Regression. The SVM model had an accuracy of 85%, indicating it performed reasonably well but less effectively than the other models.

For precision, the XGBoost model achieved a perfect score of 1.00 for legitimate emails (Label 0), meaning all emails classified as legitimate were indeed legitimate. For phishing emails (Label 1), its precision was 0.93, indicating that 93% of emails classified as phishing were correctly identified as such. Logistic Regression showed a precision of 1.00 for legitimate emails and 0.87 for phishing, while the Random Forest model mirrored this performance. The SVM model had a perfect precision score for legitimate emails but a lower precision of 0.76 for phishing emails, meaning 24% of emails classified as phishing were false positives.

Regarding recall, XGBoost correctly identified 92% of legitimate emails, with 8% misclassified as phishing. For phishing emails, it achieved a perfect recall of 1.00, meaning no phishing emails were misclassified as legitimate. Logistic Regression had a recall of 0.85 for legitimate emails and a perfect 1.00 for phishing. The Random Forest model exhibited the same recall performance as Logistic Regression. However, The SVM model had a lower recall of 0.69 for legitimate

Table 4
% of AI-generated phishing emails flagged by email providers and source.

Phishing source	Attempts	Delivered	Email provider	Flagged emails	% Flagged as spam
Gmail Account	63	59	Gmail	8	13.6%
			Outlook	2	3.4%
			Yahoo	53	89.8%
Yahoo Account	63	62	Gmail	0	0%
			Outlook	1	1.61%
			Yahoo	17	27.4%
Outlook Account	63	62	Gmail	1	1.61%
			Outlook	1	1.61%
			Yahoo	50	80.6%

Table 5
% of AI-generated legitimate emails flagged by email providers and source.

Legitimate source	Attempts	Delivered	Email provider	Flagged emails	% Flagged as spam
Gmail Account	63	63	Gmail	1	1.6%
			Outlook	0	0%
			Yahoo	42	66.7%
Yahoo Account	63	63	Gmail	1	1.6%
			Outlook	0	0%
			Yahoo	11	17.5%
Outlook Account	63	63	Gmail	12	19%
			Outlook	0	0%
			Yahoo	37	58.7%

Table 6
Performance for each machine learning algorithm.

Algorithm	Accuracy	Precision	Recall	F1-Score	AUC-Score
Logistic Regression	0.92	0.93	0.92	0.92	0.98
SVM	0.85	0.88	0.85	0.84	0.96
Random Forest	0.92	0.93	0.92	0.92	0.99
XGBoost	0.96	0.96	0.96	0.96	0.99

Table 7
Confusion matrix for each algorithm.

Algorithm	TN	FP	FN	TP
Logistic Regression	11	2	0	13
SVM	9	4	0	13
Random Forest	11	2	0	13
XGBoost	12	1	0	13

emails, indicating that 31% were misclassified as phishing, though it achieved a perfect recall for phishing emails.

F1-scores further illustrate the performance of each model. XGBoost achieved 0.96 for both legitimate and phishing emails, indicating balanced and high performance across both categories. Logistic Regression and Random Forest had F1-scores of 0.92 for legitimate emails and 0.93 for phishing. The SVM model scored 0.82 for legitimate emails and 0.87 for phishing, reflecting its lower precision and recall for legitimate emails due to a higher rate of false positives.

The confusion matrix provides additional insights into the models' performance. XGBoost correctly identified 12 legitimate emails, with 1 incorrectly flagged as phishing, and identified all 13 phishing emails. Logistic Regression also misclassified 2 legitimate emails as phishing but correctly identified all phishing emails. The SVM model misclassified 4 legitimate emails but correctly identified all phishing emails. Similarly, Random Forest misclassified 2 legitimate emails while correctly classifying all phishing emails.

Conclusively, the XGBoost model, with its comprehensive use of stylometric and content-based features, demonstrated the best performance with a 96% accuracy rate. Its perfect recall for phishing emails and high precision make it a robust algorithm for detecting and preventing phishing attacks. Logistic Regression and Random Forest also performed reliably, maintaining high precision and recall across both classes. While the SVM model effectively detects phishing emails,

its higher false positive rate makes it less favourable for applications requiring a balance between sensitivity and specificity.

5.3.1. Feature importance

As shown in Fig. 7, the five most significant features contributing to the accuracy of the XGBoost model are: *imperative verbs count*, *clause density*, *first person pronoun count*, *pronoun density*, and *urgency markers count*.

imperative verbs count holds a relative importance of 16%. This feature counts the occurrences of imperative verbs (e.g., “click”, “verify”), which are frequently used in phishing emails to prompt immediate action from the recipient. *clause density*, with a relative importance of 14%, measures sentence complexity by counting the number of clauses per sentence. Higher clause density can suggest more sophisticated phishing attempts. *first person pronoun count* has a relative importance of 12% and tracks the use of first-person pronouns (e.g., “I”, “we”), often employed in phishing emails to create a sense of familiarity or trust. *pronoun density*, at 10% relative importance, captures the frequency of pronouns, which phishing emails may use more frequently to convey urgency or build rapport. Lastly, *urgency markers count* has a relative importance of 9%. This feature identifies words or phrases conveying urgency (e.g., “immediately”, “urgent”), commonly used to pressure recipients into quick action.

5.3.2. Ablation study: Impact of features on model performance

This ablation study aimed to evaluate the impact of different stylometric features on the performance of the XGBoost classifier in detecting AI-generated phishing emails. By systematically removing the top 10, 20, and 30 features, the study assessed how excluding these features affected the model's precision, recall, and F1-score. The goal was to understand the contribution of specific features to the classifier's ability to accurately distinguish between AI-generated phishing and legitimate emails.

The results from the ablation study in Table 8 demonstrate that removing the top stylometric features significantly impacts the performance of the XGBoost classifier in detecting AI-generated phishing emails. When all 60 stylometric features are utilised, the model performs exceptionally well, achieving a high precision, recall, and F1-score of 0.96 across the board. However, when the top 10 features are excluded, there is a noticeable drop in performance. The precision, recall, and F1-score all decrease to 0.85. This decline indicates that

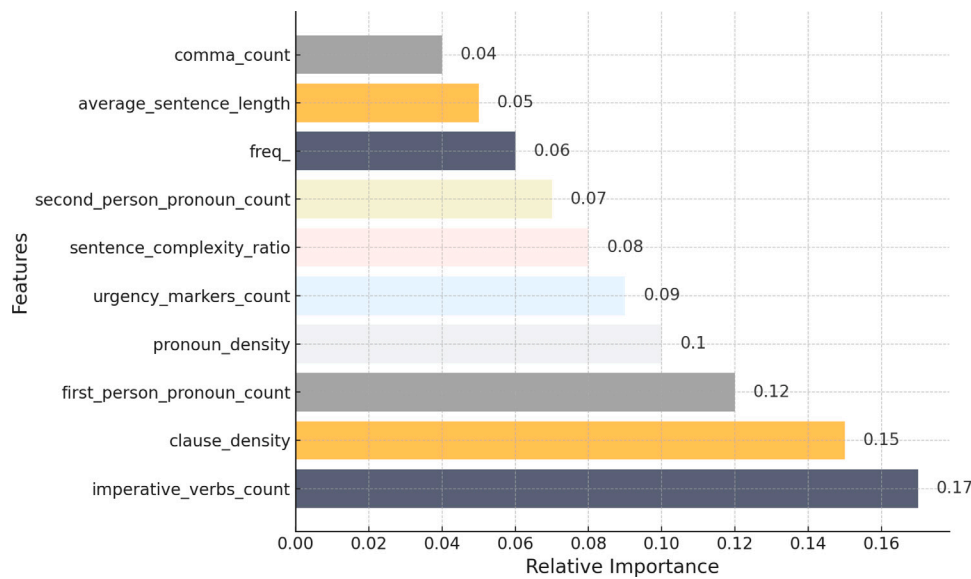


Fig. 7. Top 10 feature importances.

Table 8

Performance of XGBoost with removed features.

Features removed	Precision	Recall	F1-Score
Main (All Features)	0.96	0.96	0.96
Without Top 10 Features	0.85	0.85	0.85
Without Top 20 Features	0.82	0.81	0.81
Without Top 30 Features	0.85	0.85	0.85

the top 10 features are crucial for the model's accuracy, although their removal does not wholly impair its performance, suggesting that the remaining features still contribute meaningfully.

As the number of removed features increases to 20, the model's performance declines further, particularly in the recall, which falls to 0.81. This suggests that the top 20 features play an even more critical role in correctly identifying phishing emails, and their removal significantly affects the model's ability to differentiate between AI-generated and legitimate emails.

Interestingly, removing the top 30 features produces a slight recovery in performance, with the metrics rising again to 0.85. This implies that some of the remaining features still contain valuable discriminative information, and the top 30 features alone may not account for all the important patterns necessary for accurate classification.

5.4. The advantages and limitations of the proposed approach

Our proposed approach offers several advantages to the phishing detection field. First, it focuses on stylistic features, moving beyond traditional reliance on external indicators like URLs and domain reputation. This emphasis on intrinsic textual properties makes the method particularly effective against zero-day phishing attacks, which may not exhibit known external malicious markers.

Additionally, using interpretable machine learning models, such as Logistic Regression and XGBoost, provides transparency in decision-making processes. This interpretability builds trust and facilitates adoption among cybersecurity practitioners, as they can understand and validate the features driving the classification decisions. Unlike "black-box" deep learning models, these models offer clear insights into how phishing emails are detected.

A significant strength of this study is the use of a modern dataset comprising AI-generated phishing emails crafted by GPT-4o. This addresses a critical limitation in existing research, which often relies

on outdated datasets that fail to reflect the sophistication of current AI-driven phishing tactics. By generating phishing and legitimate emails using the same Large Language Model (LLM), this research ensures stylistic and linguistic consistency, enhancing the robustness and validity of the results.

Despite these advancements, the study has limitations. One limitation of this study is the dataset size. With only 63 phishing emails and 63 legitimate emails, the dataset may be too small to ensure broad generalisation. However, the primary focus was determining whether email providers would flag emails as spam based solely on the AI-generated textual content. Future work could expand on this by conducting a large-scale investigation of AI-generated phishing email attacks.

Another acknowledged limitation is that the emails were generated using only GPT-4o. However, research has shown that other LLM models will have a low deviation in the context of phishing (Heiding et al., 2024), Chataut et al. (2024). Additionally, we hope to investigate other LLMs (such as Anthropic's LLM, PaLM/Gemini, and LLaMA) in future studies.

Furthermore, the study was conducted on Gmail, Outlook, and Yahoo, which use sophisticated email filtering systems. These systems may not represent all webmail services, meaning the results might not generalise to other email clients or systems with different filtering mechanisms. We acknowledge this limitation and propose addressing it through further experiments in future studies.

6. Conclusion

As AI-generated phishing emails grow more sophisticated, a key question arises: How well can current email systems detect these threats? This study analysed 63 AI-generated phishing emails crafted using GPT-4o and evaluated the performance of major email services, Gmail, Outlook, and Yahoo, in filtering these messages. The results show that Gmail and Outlook have higher bypass rates than Yahoo, revealing weaknesses in existing email filtering systems.

To address these challenges, this study applied 60 stylistic features across four machine learning models Logistic Regression, SVM, Random Forest, and XGBoost. XGBoost proved the most effective, achieving 96% accuracy and a near-perfect AUC score of 99%. This strong performance is attributed to the model's effective use of stylistic features, such as imperative verb usage, clause density, and first-person pronouns, which are critical in identifying phishing intent.

Legitimate Email: Promotional Offers and Deals	
PROMPT:	Generate a professional email from a reputable organisation informing the recipient, [Recipient's Name], about a promotional offer or deal that they might find valuable. Encourage them to learn more by [call to action]. The email should be clear, friendly, and trustworthy.
<p><i>Subject: Exclusive Promotional Offer Just for You, Jonathan</i></p> <p>Dear Jonathan,</p> <p>I trust this email finds you well. As a valued patron of our organisation, we are excited to present you with an exclusive promotional offer that we believe would provide significant value to you. Given our understanding of your requirements and preferences, our team has thoroughly curated this offer to complement your needs suitably.</p> <p>To start with, we are providing a 20% discount on all our premium services or products for the next six months. We believe this would give you the opportunity not only to enjoy the benefits of our advanced features but also help you save considerably.</p> <p>Furthermore, we are including complimentary access to our new features set to be launched in the next quarter. This means before those services are available to the general public, you will get an exclusive chance to try them out and provide us with your valuable feedback.</p> <p>The intent behind these offers aligns with our commitment to providing you with better value-for-money solutions. However, we understand that every client's needs are unique. Hence, we encourage you to write back to us if you have any questions or need specific details about these offers. We assure you of our prompt support and guidance to help you make an informed decision.</p> <p>To avail of the promotional offer, simply reply to this email expressing your interest, or directly reach out to our customer service team at 1-800-123-4567. A member of our team would be more than happy to address your queries and guide you through the next steps.</p> <p>Please don't miss out on this opportunity as the offer ends on September 30th, 2023.</p> <p>We genuinely value your association with our organization and wish to serve you with the best possible offerings.</p> <p>Looking forward to your affirmative response.</p> <p>Best Regards,</p>	

Fig. A.1. Example of an AI-generated legitimate email: Promotional offers and deals.

The ablation study demonstrates that the top 20 stylometric features contribute significantly to the model's high performance. However, the model remains resilient to some extent when even half of the features are removed, indicating that the stylometric features collectively provide rich information for detecting phishing emails. Future work would focus on further fine-tuning feature selection to balance model complexity and performance. Furthermore, we aim to investigate other large language models like Gemini and Claude and explore the impact of AI-generated phishing emails on different email services such as iCloud mail.

CRediT authorship contribution statement

Chidimma Opara: Conceptualization, Data curation, Methodology, Software, Visualization, Writing – original draft preparation & editing. **Paolo Modesti:** Methodology, Investigation, Writing – review & editing, Validation. **Lewis Golightly:** Methodology, Conceptualization, Writing – review & editing, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Stylometric analysis of sample phishing and legitimate emails

This appendix presents a stylometric analysis comparing samples of AI-generated phishing emails with legitimate emails.

The phishing email exhibits a lower Flesch Reading Ease score (45.15 vs. 60.75) and higher complexity indices, including an SMOG index of 14.3 compared to 12.6 and a higher Gunning Fog score. These metrics indicate that phishing emails use more complex language to create a sense of authority or induce confusion. In contrast, legitimate emails are simpler, with language designed to be clear, accessible, and easy to understand.

In terms of word category features, the phishing email employs imperative verbs (count: 1), modal verbs (count: 2), an urgency marker (count: 1), and even an aggressiveness marker (count: 1). By comparison, the legitimate email contains no imperative verbs, urgency markers, or aggressiveness markers. This stark contrast highlights how phishing emails rely on language intended to prompt immediate action and create a sense of urgency or pressure. In contrast, legitimate emails maintain a neutral and professional tone (see Figs. A.1 and A.2).

Also, the phishing email exhibits greater punctuation variety (6 vs. 3), more exclamation marks (2 vs. 0), and a higher dash count (12 vs 5). Such patterns may indicate a more erratic or emotionally

Phishing Email: Promotional Offers and Deals
<p>PROMPT: Generate a phishing email that appears to be from a reputable organisation, informing the recipient, [Recipient's Name], about an exclusive promotional offer or deal and urging them to take immediate action by [call to action]. The email should be convincing and persuasive. The company name should sound real.</p>
<p>Subject: <i>Your Exclusive Limited Bonus Deal from GlobalTech Solutions is Here!</i></p> <p>Dear Peter,</p> <p>We hope this message finds you well. I am reaching out from GlobalTech Solutions, the trusted name in software innovation.</p> <p>In the spirit of deeply valuing unique members like yourself, we've arranged an exclusive, once-in-a-lifetime promotional bundle that you won't want to miss! This handpicked special includes premium access to cutting-edge software tools that would otherwise cost you thousands.</p> <p>However, due to the exclusive nature of this offer, we need to verify the rightful recipients of this benefit to prevent fraudulent access on our platform. Therefore, we can only unlock this top-tier bonus collection once your account undergoes a mandatory verification.</p> <p>Please immediately call us at 240-517-6106 for a speedy verification process and to secure your exclusive bonus. Please note, failure to complete verification within 24 hours may cause forfeiture of this opportunity. Should you need help at any stage, our support team is ready to assist you 24/7 - because your satisfaction is our priority.</p> <p>Remember, GlobalTech Solutions is committed to your success and enabling you to leverage state-of-the-art technology. Let's seize this opportunity together,</p> <p>Please don't miss out on this opportunity as the offer ends on September 30th, 2023.</p> <p>The GlobalTech Solutions Team GlobalTech Solutions Management 2865, Cyber Park Boulevard TechTown, TS-1340</p> <p>Note: To ensure your security, NEVER share your verification code, password or sensitive information over the phone or via email. Contact us directly for support. GlobalTech Solutions. Think Innovation. Think Future.</p>

Fig. A.2. Example of an AI-generated phishing email: Promotional offers and deals.

charged writing style characteristic of phishing attempts, as they often use punctuation to emphasise specific points or create urgency. Both emails are relatively lexically rich for the lexical indicators, but the phishing email has a higher lexical diversity (0.7258 vs 0.6205) and a slightly higher proportion of complex words (88 vs 84). This close similarity in lexical attributes highlights the challenge of stylistometric analysis and the importance of using all identified features and external attributes for phishing classification.

Appendix B. Time complexity analysis

Given the parameters of the study:

- Average length of an email (n) = 300,
- Number of features (m) = 60,
- Number of training samples (k) = 103,
- Number of test emails (t) = 26,

The total time complexity for the proposed pipeline is calculated as follows:

$$O(k \cdot n + k \cdot m \cdot \log(k) + t \cdot m \cdot \log(k))$$

B.1. Step-by-step calculation

Step 1: Data preprocessing. The time complexity for preprocessing k training samples is: $O(k \cdot n) = 103 \cdot 300 = 30,900$

Step 2: Feature extraction. The time complexity for extracting m features from k training samples is: $O(k \cdot m \cdot n) = 103 \cdot 60 \cdot 300 = 1,854,000$

Step 3: Classifier training. The time complexity for training the XGBoost classifier is: $O(k \cdot m \cdot \log(k)) = 103 \cdot 60 \cdot \log_2(103)$ Approximating $\log_2(103) \approx 6.7$, we get: $103 \cdot 60 \cdot 6.7 \approx 41,526$

Step 4: Classification. The time complexity for classifying t test samples is: $O(t \cdot m \cdot \log(k)) = 26 \cdot 60 \cdot \log_2(103)$ Approximating $\log_2(103) \approx 6.7$, we get: $26 \cdot 60 \cdot 6.7 \approx 10,452$

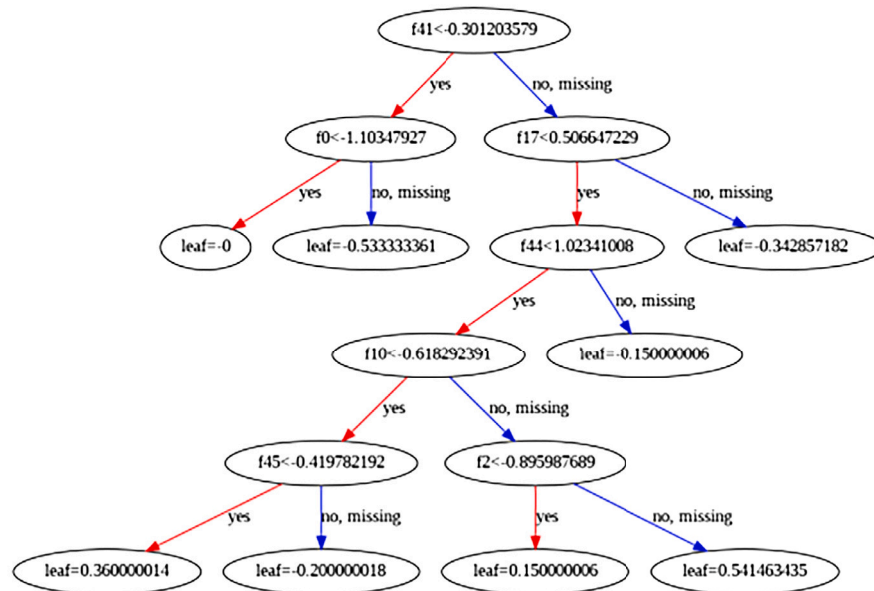
B.2. Total time complexity

Summing all the components: $30,900 + 1,854,000 + 41,526 + 10,452 = 1,936,878$

The total estimated time complexity for the algorithm with the given parameters is approximately: 1.94×10^6 operations. This result indicates that the computational cost is heavily dominated by the feature extraction stage, which is expected given the number of features and the dataset size.

Appendix C. XGBoost decision threshold and features

The XGBoost model first splits on *imperative verbs count* (f41), which has a very high gain (40.23) and is therefore the most influential feature. For observations with a lower imperative verb count, the model further refines its decision using *word count* (f0). For the remaining observations, the split on *clause density* (f17) indicates the importance of sentence complexity. Subsequent splits on *first person pronoun count* (f44), *comma count* (f10), *second person pronoun count* (f45), and *average word length* (f2) further refine the decision boundaries.



XGBoost Decision Threshold and Features

Appendix D. Abbreviations

AUC Area Under the Receiver Operating Characteristic Curve

BERT Bidirectional Encoder Representations from Transformers

GPT-4o Specific LLM variant employed in this study

LLaMA Large Language Model Meta AI

LLM Large Language Model

LR Logistic Regression

PaLM Pathways Language Model

ROI Return on Investment

SVM Support Vector Machine

XGBoost eXtreme Gradient Boosting

Data availability

We have included the link to the dataset within the paper in Subsection 1.1.

References

- Abdelhamid, N., Ayesha, A., & Thabtah, F. (2014). Phishing detection based associative classification data mining. *Expert Systems with Applications*, 41(13), 5948–5959.
- Alhogail, A., & Alsabih, A. (2021). Applying machine learning and natural language processing to detect phishing email. *Computers & Security*, 110, Article 102414. <http://dx.doi.org/10.1016/j.cose.2021.102414>.
- An, R., Yang, Y., Yang, F., & Wang, S. (2023). Use prompt to differentiate text generated by ChatGPT and humans. *Machine Learning with Applications*, 14, Article 100497.
- APWG (2024). *Phishing activity trends report, 1st quarter 2024: Technical report*, Anti-Phishing Working Group.
- Arifa Islam, C. (2023). Phishing email curated datasets. <https://zenodo.org/records/8339691>. (Accessed 13 August 2024).

- Barraclough, P. A., Hossain, M. A., Tahir, M., Sexton, G., & Aslam, N. (2013). Intelligent phishing detection and protection scheme for online transactions. *Expert Systems with Applications*, 40(11), 4697–4706.
- Bethany, M., Galiopoulos, A., Bethany, E., Karkevandi, M. B., Vishwamitra, N., & Najafirad, P. (2024). Large language model lateral spear phishing: A comparative study in large-scale organizational settings. <http://dx.doi.org/10.48550/arxiv.2401.09727>, arXiv preprint arXiv:2401.09727.
- Bogert, J. (1985). In defense of the fog index. *The Bulletin of the Association for Business Communication*, 48(2), 9–12.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45, 5–32.
- Chataut, R., Gyawali, P. K., & Usman, Y. (2024). Can ai keep you safe? a study of large language models for phishing detection. In *2024 IEEE 14th annual computing and communication workshop and conference* (pp. 0548–0554). IEEE, <http://dx.doi.org/10.1109/ccwc60891.2024.10427626>.
- Chew, C.-J., Lin, Y.-C., Chen, Y.-C., Fan, Y.-Y., & Lee, J.-S. (2024). Preserving manipulated and synthetic deepfake detection through face texture naturalness. *Journal of Information Security and Applications*, 83, Article 103798. <http://dx.doi.org/10.1016/j.jisa.2024.103798>.
- Cidon, A., Gavish, L., Bleier, I., Korshun, N., Schweighauser, M., & Tsitkin, A. (2019). High precision detection of business email compromise. In *28th USENIX security symposium* (pp. 1291–1307).
- Cofense (2023). Most common phishing email themes of 2023. <https://cofense.com/blog/most-common-phishing-email-themes-of-2023/>. (Accessed 11 August 2024).
- Cortes, C. (1995). Support-vector networks. *Machine Learning*, <http://dx.doi.org/10.1007/bf00994018>.
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Bidirectional encoder representations from transformers. arXiv preprint arXiv:1810.04805.
- Dewan, P., Kashyap, A., & Kumaraguru, P. (2014). Analyzing social and stylistic features to identify spear phishing emails. In *2014 apwg symposium on electronic crime research* (pp. 1–13). IEEE, <http://dx.doi.org/10.1109/ecrime.2014.6963160>.
- Drake, C. E., Oliver, J. J., & Koontz, E. J. (2004). Anatomy of a phishing email. In *CEAS*.
- Duman, S., Kalkan-Cakmakci, K., Egele, M., Robertson, W., & Kirda, E. (2016). Emailprofiler: Spearphishing filtering with header and stylistic features of emails. In *2016 IEEE 40th annual computer software and applications conference*, vol. 1 (pp. 408–416). IEEE, <http://dx.doi.org/10.1109/compsac.2016.105>.
- Dutta, M., & Bansal, K. (2016). A review paper on various search engines (Google, Yahoo, AltaVista, Ask and Bing). *International Journal on Recent and Innovation Trends in Computing and Communication*, 4(8), 190–195.
- Form, L. M., Chiew, K. L., Tiong, W. K., et al. (2015). Phishing email detection technique by using hybrid features. In *2015 9th international conference on IT in Asia* (pp. 1–5). IEEE.
- Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, 1189–1232. <http://dx.doi.org/10.1214/aos/1013203451>.
- Gallo, L., Gentile, D., Ruggiero, S., Botta, A., & Ventre, G. (2024). The human factor in phishing: Collecting and analyzing user behavior when reading emails. *Computers & Security*, 139, Article 103671. <http://dx.doi.org/10.1016/j.cose.2023.103671>.

- Gallo, L., Maiello, A., Botta, A., & Ventre, G. (2021). 2 Years in the anti-phishing group of a large company. *Computers & Security*, 105, Article 102259. <http://dx.doi.org/10.1016/j.cose.2021.102259>.
- Giray, L., Sevnanayan, K., & Ranjbaran Madiseh, F. (2024). Beyond policing: AI writing detection tools, trust, academic integrity, and their implications for college writing. *Internet Reference Services Quarterly*, 1–34.
- Heiding, F., Schneier, B., Vishwanath, A., Bernstein, J., & Park, P. S. (2024). Devising and detecting phishing emails using large language models. *IEEE Access*, <http://dx.doi.org/10.1109/access.2024.3375882>.
- Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied logistic regression*. John Wiley & Sons, <http://dx.doi.org/10.1002/9781118548387>.
- Jagatic, T. N., Johnson, N. A., Jakobsson, M., & Menczer, F. (2007). Social phishing. *Communications of the ACM*, 50(10), 94–100. <http://dx.doi.org/10.1145/1290958.1290968>.
- Jain, A. K., & Gupta, B. B. (2018). Towards detection of phishing websites on client-side using machine learning based approach. *Telecommunication Systems*, 68, 687–700.
- Jakobsson, M., & Myers, S. (2006). *Phishing and countermeasures: understanding the increasing problem of electronic identity theft*. John Wiley & Sons.
- Karim, A., Azam, S., Shanmugam, B., Kannoopatti, K., & Alazab, M. (2019). A comprehensive survey for intelligent spam email detection. *IEEE Access*, 7, 168261–168295.
- Kirova, V. D., Ku, C. S., Laracy, J. R., & Marlowe, T. J. (2024). Software engineering education must adapt and evolve for an llm environment. In *Proceedings of the 55th ACM technical symposium on computer science education v. 1* (pp. 666–672). <http://dx.doi.org/10.1145/3626252.3630927>.
- Liau, T. L., Bassin, C. B., Martin, C. J., & Coleman, E. B. (1976). Modification of the coleman readability formulas. *Journal of Reading Behavior*, 8(4), 381–386.
- Maneriker, P., Stokes, J. W., Lazo, E. G., Carutasu, D., Tajaddodianfar, F., & Gururajan, A. (2021). Urltran: Improving phishing url detection using transformers. In *MILCOM 2021-2021 IEEE military communications conference* (pp. 197–204). IEEE, <http://dx.doi.org/10.1109/milcom52596.2021.9653028>.
- Moradpoor, N., Clavie, B., & Buchanan, B. (2017). Employing machine learning techniques for detection and classification of phishing emails. In *2017 computing conference* (pp. 149–156). IEEE.
- Opara, C. (2024). Styloai: Distinguishing AI-generated content with stylometric analysis. In A. M. Olney, I.-A. Chounta, Z. Liu, O. C. Santos, & I. I. Bittencourt (Eds.), *Artificial intelligence in education. posters and late breaking results, workshops and tutorials, industry and innovation tracks, practitioners, doctoral consortium and blue sky* (pp. 105–114). Cham: Springer Nature Switzerland, ISBN: 978-3-031-64312-5.
- Parsons, K., McCormac, A., Pattinson, M., Butavicius, M., & Jerram, C. (2013). Phishing for the truth: A scenario-based experiment of users' behavioural response to emails. In *Security and privacy protection in information processing systems: 28th IFIP TC 11 international conference, SEC 2013, Auckland, New Zealand, July 8-10, 2013. proceedings 28* (pp. 366–378). Springer, http://dx.doi.org/10.1007/978-3-642-39218-4_27.
- Patel, H., Rehman, U., & Iqbal, F. (2024). Evaluating the efficacy of large language models in identifying phishing attempts. In *2024 16th international conference on human system interaction* (pp. 1–7). IEEE, <http://dx.doi.org/10.1109/hsi61632.2024.10613528>.
- Raman, R., Calyam, P., & Achuthan, K. (2024). ChatGPT or bard: Who is a better Certified Ethical Hacker? *Computers & Security*, 140, Article 103804. <http://dx.doi.org/10.1016/j.cose.2024.103804>.
- Roy, S. S., Thota, P., Naragam, K. V., & Nilizadeh, S. (2023). From Chatbots to PhishBots?—Preventing Phishing scams created using ChatGPT, Google Bard and Claude. <http://dx.doi.org/10.48550/arxiv.2310.19181>, arXiv preprint arXiv:2310.19181.
- Toolan, F., & Carthy, J. (2010). Feature selection for spam and phishing detection. In *2010 eCrime researchers summit* (pp. 1–12). IEEE, <http://dx.doi.org/10.1109/ecrime.2010.5706696>.
- Verizon (2024). 2024 data breach investigations report: Results and analysis. <https://www.verizon.com/business/en-gb/resources/reports/dbir/2024/results-and-analysis-intro/>. (Accessed 11 August 2024).
- Vishwanath, A., Herath, T., Chen, R., Wang, J., & Rao, H. R. (2011). Why do people get phished? Testing individual differences in phishing vulnerability within an integrated, information processing model. *Decision Support Systems*, 51(3), 576–586. <http://dx.doi.org/10.1016/j.dss.2011.03.002>.
- Wang, Y., Zhu, W., Xu, H., Qin, Z., Ren, K., & Ma, W. (2023). A large-scale pretrained deep model for phishing url detection. In *ICASSP 2023-2023 IEEE international conference on acoustics, speech and signal processing* (pp. 1–5). IEEE, <http://dx.doi.org/10.1109/icassp49357.2023.10095719>.