PhishAgent: A Robust Multimodal Agent for Phishing Webpage Detection

Tri Cao[1] *†, Chengyu Huang[1]*, Yuexin Li[1]*, Huilin Wang[1], Amy He[3], Nay Oo[2], Bryan Hooi[1]

[1]National University of Singapore,
[2]NCS Cyber Special Ops-R&D,
[3]Massachusetts Institute of Technology

Abstract

Phishing attacks are a major threat to online security, exploiting user vulnerabilities to steal sensitive information. Various methods have been developed to counteract phishing, each with varying levels of accuracy, but they also face notable limitations. In this study, we introduce PhishAgent, a multimodal agent that combines a wide range of tools, integrating both online and offline knowledge bases with Multimodal Large Language Models (MLLMs). This combination leads to broader brand coverage, which enhances brand recognition and recall. Furthermore, we propose a multimodal information retrieval framework designed to extract the relevant top k items from offline knowledge bases, using available information from a webpage, including logos and HTML. Our empirical results, based on three real-world datasets, demonstrate that the proposed framework significantly enhances detection accuracy and reduces both false positives and false negatives, while maintaining model efficiency. Additionally, PhishAgent shows strong resilience against various types of adversarial attacks.

1 Introduction

Phishing attacks pose a serious threat to online security, as cybercriminals continuously improve their methods to trick users into disclosing sensitive information by pretending to be legitimate entities. According to the Anti-Phishing Working Group (APWG), there were 1,077,501 reported phishing attacks in the fourth quarter of 2023, contributing to nearly five million attacks throughout the year—the highest number ever recorded (Anti-Phishing Working Group 2023). These deceptive sites result in substantial financial losses, as the FBI reported that U.S. businesses suffered losses exceeding $12.5 billion due to phishing in 2023, up from $10.3 billion in 2022 (Federal Bureau of Investigation 2023). These numbers emphasize the urgent necessity for robust automated phishing detection methods and the pressing need to confront this escalating threat.

Many approaches have been developed to combat phishing, each achieving varying degrees of accuracy but also facing significant limitations. Conventional approaches, such as heuristic-based methods (Le et al. 2018; Maneriker et al. 2021; Verma and Dyer 2015; Guo et al. 2021; Xiang et al. 2011a; Li et al. 2019; Lee et al. 2021), blacklists (Provos et al. 2007; OpenPhish; PhishTank), and rule-based systems (Afroz and Greenstadt 2011a), have been effective to some extent in analyzing webpage characteristics, maintaining lists of known phishing URLs, and applying predefined rules based on known phishing patterns. However, these methods are often static and not up-to-date, relying on fixed criteria that do not adapt to new phishing techniques. For example, a heuristic method may only detect URLs that fit predefined patterns, missing new variations. This can lead to delays in detecting evolving phishing threats until the rules are manually updated. Reference-based approaches compare suspected phishing sites against a knowledge base of legitimate webpages for various brands (Abdelnabi, Krombholz, and Fritz 2020; Lin et al. 2021; Liu et al. 2022, 2023; Li et al. 2024), achieving good results but facing challenges in maintaining a comprehensive and current knowledge base. Search engine-based methods generate query strings from webpage content and analyze search results (Zhang, Hong, and Cranor 2007; Huh and Kim 2012; Jain and Gupta 2018; Varshney, Misra, and Atrey 2016; Chang et al. 2013; Chiew et al. 2015; Rao and Pais 2019), showing promise but being prone to false positives and sensitive to changes in search engine algorithms, which can affect their effectiveness. Recently, Multimodal Large Language Model (MLLM)-based approaches have demonstrated high accuracy in detecting phishing webpages by leveraging advanced text and image processing capabilities (Koide et al. 2023). Nevertheless, MLLMs are susceptible to adversarial attacks (Li et al. 2024; Cui et al. 2024) and may struggle with local brands, as information about these brands is limited and unfamiliar to the MLLMs, leading to potentially erroneous decisions.

In light of these challenges, autonomous agents offer a promising solution. Defined as Large Language Models (LLMs) interacting with a set of tools, these agents are particularly suited for solving complex tasks due to their ability to analyze information from the tools and make decisions. This capability has been successfully demonstrated in many tasks (Park et al. 2023; Shen et al. 2024; Chen et al. 2024; Zeng et al. 2023). Recently, GEPAgent (Wang and Hooi 2024), an agent-based phishing detection approach, was introduced, demonstrating high accuracy in detecting phishing

*These authors contributed equally.
†Corresponding Author: Tri Cao (tricao@nus.edu.sg).

webpages through interactions between MLLMs and an online knowledge base across multiple iterations. However, it also exhibited a significant limitation in terms of execution time, taking over 10 seconds per sample.

To solve these challenges, we propose PhishAgent, a multimodal agent specifically designed for phishing webpage detection with low latency. PhishAgent integrates a comprehensive set of tools, combining both online and offline knowledge bases. GEPAgent (Wang and Hooi 2024) require multiple iterations of interaction between the agent and knowledge bases to refine results, leading to high latency. In contrast, PhishAgent is designed to use only one such iteration, yet still achieves high performance, thus significantly reducing detection time. Its various modules are interconnected through the Agent Core, which functions as the central module, integrating all the other modules and making decisions. Particularly, the offline knowledge base can cover local brands that are not indexed by search engines, while the online knowledge base can cover very new webpages that the offline knowledge may not include. This combination results in wider brand coverage, thereby increasing brand recognition and recall. Moreover, MLLMs in PhishAgent make decisions based on the information queried from the knowledge base rather than solely based on its internal knowledge, which enhances result reliability.

In addition, we make improvements to several auxiliary tools to enhance their accuracy. Specifically, recognizing the limitations of querying the most related webpage in the offline knowledge base based solely on exact matching between the extracted brand name of the input webpage and the brand names existing in the knowledge base through the textual modality (Li et al. 2024), we introduce a multimodal information retrieval framework. This framework enhances the analysis of the input webpage by leveraging all available information from the webpage, such as logos and HTML, to retrieve the top k relevant items from the offline knowledge base. These relevant items are used in the subsequent steps of our pipeline to ultimately determine whether the webpage is a phishing webpage or not.

In summary, our work makes three main contributions:

- **PhishAgent**: A multi-modal agent tailored for low-latency phishing webpage detection, combining both online and offline knowledge bases with MLLMs and various useful tools.
- **Multi-modal Retriever**: A multimodal module which retrieves the top k brands from an offline knowledge base, utilizing all available information from the webpage, such as logos and HTML.
- **Empirical Results**: Our empirical results on three real-world datasets show that the proposed framework notably enhances detection accuracy while preserving model efficiency. Additionally, PhishAgent also demonstrates robustness against various types of adversarial attacks.

2 Related Works

Conventional Approaches These include methods relying on heuristics, blacklists, and rule-based systems. Heuristic methods analyze webpage characteristics such as URL, HTML structure, and suspicious keywords (Garera et al. 2007; Sheng et al. 2010; Zhang, Hong, and Cranor 2007; Le et al. 2018; Maneriker et al. 2021; Verma and Dyer 2015; Guo et al. 2021; Xiang et al. 2011a; Li et al. 2019; Ludl et al. 2007; Lee et al. 2021). Blacklists check incoming URLs against known phishing lists (Provos et al. 2007; OpenPhish; PhishTank), while rule-based systems use predefined rules based on known phishing patterns (Afroz and Greenstadt 2011a). However, these methods tend to be static, depending on fixed criteria that fail to adjust to new phishing techniques. For instance, a heuristic approach might only identify URLs that match established patterns, overlooking new variations. This can result in delays in identifying emerging phishing threats until the rules are manually revised.

Reference-based Approaches These compare target webpage information to a known set of brand information. They create a brand knowledge base (BKB) containing logos, aliases, and legitimate domains, and a detector backbone that uses this BKB for detection (Li et al. 2024; Liu et al. 2023). To determine whether a webpage is phishing or legitimate, these systems first identify the target brand of the webpage. If the webpage is found to have the intent of a particular brand but its domain does not align with the brand's authentic domains, it is classified as phishing. However, these approaches can become outdated and struggle to cover all possible brands comprehensively (Fu, Wenyin, and Deng 2006; Afroz and Greenstadt 2011b; Abdelnabi, Krombholz, and Fritz 2020; Lin et al. 2021; Liu et al. 2022, 2023; Li et al. 2024).

Search Engine-based Approaches These methods detect phishing websites by querying search engines with key descriptors from webpage content (Xiang et al. 2011b; Xiang and Hong 2009; Zhang, Hong, and Cranor 2007; Huh and Kim 2012; Jain and Gupta 2018; Varshney, Misra, and Atrey 2016; Chang et al. 2013; Chiew et al. 2015; Rao and Pais 2019). If the input URL's domain appears in search results, the website is deemed legitimate. This method can lead to false positives as not all legitimate webpages are indexed or ranked highly.

LLM/MLLM-based Approaches These leverage advanced capabilities in text and image processing to detect phishing (Koide et al. 2023). Prompts including webpage URL, HTML content, and screenshots help MLLMs predict phishing attempts. However, LLMs/MLLMs are vulnerable to adversarial attacks (Li et al. 2024; Cui et al. 2024) and can encounter difficulties with local brands due to the limited and unfamiliar information, which may result in incorrect decisions.

Autonomous Agents powered by LLMs and tools excel in handling complex tasks through efficient information processing and decision-making. Park et al. (2023) introduces generative agents using large language models to simulate human behavior. Works like HuggingGPT (Shen et al. 2024), AgentFLAN (Chen et al. 2024), AgentInstruct (Zeng et al. 2023), and ReAct (Yao et al. 2022) demonstrate the ability of LLMs to manage AI models and solve

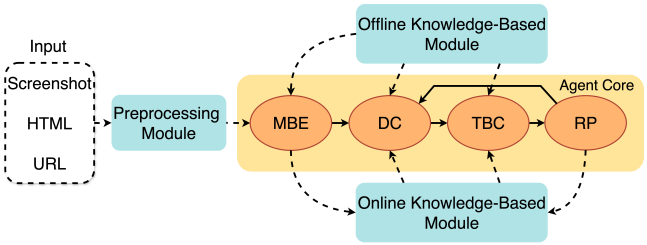

Figure 1: An overview of our phishing detector, PhishAgent.

complex tasks. Recently, GEPAgent (Wang and Hooi 2024), an agent-based phishing detection method, was introduced. It demonstrated high accuracy in identifying phishing webpages by leveraging interactions between MLLMs and an online knowledge base over several iterations. However, it also faced a major drawback regarding execution time, taking an average of 10 seconds per sample.

Multimodal Retrievers Information Retrieval (IR) methods aim to search for relevant information from an information collection (Singhal 2001; Wei et al. 2023). Among these, Multimodal Retrievers are those where the query and retrieved content can span across multiple modalities, such as the image and text modalities. This topic has been widely studied (Jia et al. 2021; Jain et al. 2021; Li et al. 2021; Luo et al. 2023; Girdhar et al. 2023; Wei et al. 2023) and has applications in various domains. Inspired by (Wei et al. 2023), we design a retriever that uses HTML and brand logo of the webpage as the query to retrieve its potential target brands.

3 Threat Model

In a phishing attack, the attacker seeks to deceive users into thinking that the webpage is affiliated with a legitimate brand, thereby tricking them into disclosing sensitive information such as usernames, passwords, or bank details. Formally, let w denote a webpage, which includes its screenshot ($w.\texttt{scr}$), HTML structure ($w.\texttt{html}$), and a URL ($w.\texttt{url}$). To effectively carry out this deception, the webpage must convincingly imitate a specific brand b by leveraging visual elements in $w.\texttt{scr}$, textual features in $w.\texttt{html}$, or both. Our goal is to detect phishing webpages, identify their target brands, and provide detailed annotations explaination.

4 Multimodal Agent

4.1 Overview

We next introduce PhishAgent, which leverages multiple tools and knowledge bases to verify different indicators of phishing activities. The Fig. 1 shows the overview of PhishAgent. PhishAgent consists of four main modules: Preprocessing Module, Online Knowledge-Based Module, Offline Knowledge-Based Module, and Agent Core. Each module can interact with others. The Agent Core functions as the central module, responsible for analyzing information and making decisions, while the remaining modules serve as tools to assist the Agent Core in preprocessing and gathering information. We analyze the details and role of each module in the next sections.

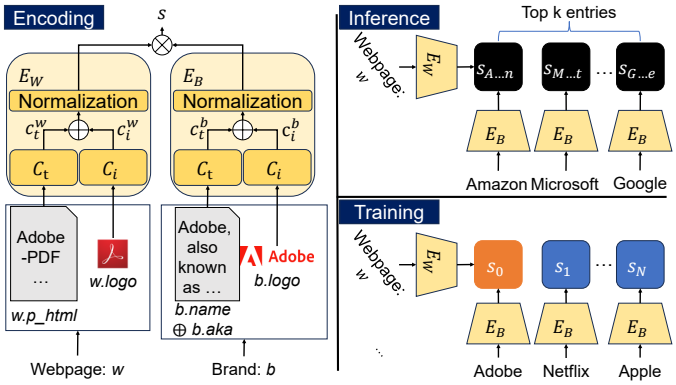

Figure 2: Our multimodal retriever; Left: Example of how a webpage w and a brand b are encoded and how the retrieval score s is computed; Top right: Example of how the top k brands are retrieved for the webpage w during inference; Bottom right: Example of our training process where contrastive learning is used to distinguish from N randomly sampled negative brands (colored in blue) the positive brand (colored in orange) for the webpage w.

4.2 Preprocessing Module

Given a webpage w as input, we use a logo detection pipeline (see Appendix for details) to extract its logo image (if any), denoted $w.\texttt{logo}$. Next, the HTML of w is processed to remove noninformative HTML (such as layout or tracking elements), resulting in its processed HTML $w.\texttt{p\_html}$. Finally, the URL is parsed to extract its domain $w.\texttt{domain}$. The resulting logo, processed HTML, and domain are passed to the next step for further analysis.

4.3 Offline Knowledge-Based Module

Given the preprocessed information of a webpage w ($w.\texttt{p\_html}$, $w.\texttt{logo}$), the goal of the Offline Knowledge-Based Module is to retrieve the brands and their associated information from the Brand Knowledge Base that are most similar to w. Brand Knowledge Bases (BKBs) are collections of brands and their authentic information. In this work, we utilize KnowPhish (Li et al. 2024), a multimodal BKB built on Wikidata, which includes around 20,000 potential phishing brands. Each brand entry b contains its name ($b.\texttt{name}$), logo ($b.\texttt{logo}$), aliases ($b.\texttt{aka}$), and domains ($b.\texttt{domains}$).

To achieve this, inspired by Wei et al. (2023), we design and train a multimodal retriever MR that can process the webpage's HTML and extracted brand logo, retrieving the top k brands from the BKB that most closely match the target brand of the webpage.

Specifically, our retriever encodes the input webpage into a *webpage encoding*, and each brand into a *brand encoding*. Our retriever then retrieves the top k brands with the highest cosine similarity between their brand encodings and the webpage encoding.

Webpage Encoding Given the processed HTML $w.\texttt{p\_html}$ and brand logo $w.\texttt{logo}$ of the webpage w, we use

a CLIP text encoder $\mathcal{C}_t$ to encode its processed HTML into an encoding $\mathcal{C}_t(w.\texttt{p\_html})$, and similarly use a CLIP image encoder $\mathcal{C}_i$ to encode the logo into an encoding: $\mathcal{C}_i(w.\texttt{logo})$. These text and image encodings are combined with some weights, added together, and normalized, to produce the combined webpage encoding:

$$E_W(w) = \mathsf{Norm}(c_t^w \mathcal{C}_t(w.\texttt{p\_html}) + c_i^w \mathcal{C}_i(w.\texttt{logo})), \quad (1)$$

where c_t^w and c_i^w are weight constants, and Norm normalizes the encoding.

In some cases, the logo extractor may not identify any logo on a webpage. In this case, we simply exclude the image embedding from $E_W(w)$.

Brand Encoding We compute the combined embedding for each brand b in the knowledge base BKB in a similar manner. The only difference is that, since we have both the brand name $b.\texttt{name}$ (e.g., Microsoft) and a list of A aliases $b.\texttt{aka}$ (e.g., [microsoft, $\cdots$, msft]) to represent each brand, we introduce a function $\oplus$ that combines them into a string with format $\{b.\texttt{name}\}$, also known as $\{b.\texttt{aka}_1\}, \cdots, \{b.\texttt{aka}_\texttt{A}\}$ (e.g., Microsoft, also known as microsoft, $\cdots$, msft). Given $b.\texttt{name}$, $b.\texttt{aka}$, and $b.\texttt{logo}$ of each brand in the knowledge base, we compute the brand's combined encoding:

$$E_B(b) = \mathsf{Norm}(c_t^b \mathcal{C}_t((b.\texttt{name} \oplus b.\texttt{aka})) + c_i^b \mathcal{C}_i(b.\texttt{logo}))$$

where c_t^b and c_i^b are weight constants.

Again, certain brands may not have any logo image, or others may have more than 1 logo variant. To handle the former case, we exclude the image embedding when computing $E_B(b)$. In the latter case, for each logo variant of the brand $b.\texttt{logo}_\texttt{j}$, we compute a separate embedding $E_B(b_j)$ with $b.\texttt{logo}_\texttt{j}$ and treat it as a separate brand.

Inference We then use dot product retrieval between $E_W(w)$ and $E_B(b)$ to get the top k brand matches.

$$\mathcal{R}_{\text{offl}} = \text{MR}(w, \text{BKB}, k) = \underset{b \in \text{BKB}}{\mathsf{TopK}}(E_W(w) \cdot E_B(b))$$

We then pass the names, aliases, and legitimate domains for each retrieved brand to subsequent modules.

Training We notice that retrieval recall is not satisfactory with the pre-trained CLIP encoders. Therefore, we use contrastive learning to train the CLIP encoders. Specifically, for each webpage in the training set, we annotate its ground truth target brand b_0 and randomly sample N negative brands $b_1, \cdots, b_N$ from the knowledge base k. Denote $E_W(w) \cdot E_B(b_i)$ as s_i. We train our retriever by minimizing $-\mathsf{log}(\mathsf{softmax}(s_0, s_1, \cdots, s_N)_0)$ using stochastic gradient descent.

Experimental details of our retriever and its impact on brand name extraction performance are in the Appendix.

4.4 Online Knowledge-Based Module

The goal of Online Knowledge-Based Module is to query all information related to the input webpage w from online sources by leveraging search engines (e.g., Google search engine). This module utilizes two distinct queries corresponding to two different types: a domain-based query $\mathcal{Q}_{\text{domain}}$ and a brand name-based query $\mathcal{Q}_{\text{brand}}$. The domain-based query is the domain of the input webpage (e.g.,"amazon.sg") obtained from the preprocessing module, while the brand name-based query is generated by the Agent Core module (e.g., "Amazon") as we will discuss in section 4.5 Multimodal Brand Extractor. These two queries aim to achieve two main objectives: assessing the popularity of the target brand and the popularity of the input domain. For each type of query, we retrieve the top k items, where R_{domain} is the set of top k results from $\mathcal{Q}_{\text{domain}}$, and R_{brand} is the set of top k results from $\mathcal{Q}_{\text{brand}}$.

The domain-based query $\mathcal{Q}_{\text{domain}}$ focuses on evaluating the popularity of the input domain. If the domain is found in R_{domain}, it is highly likely to be a benign webpage, as most phishing webpages typically have a very short lifespan and are not indexed by search engines (Liu et al. 2022, 2023).

The brand name-based query $\mathcal{Q}_{\text{brand}}$ is tasked with searching for webpages that belong to a specific brand name. This query aims to gather all relevant webpages associated with a brand while also checking the popularity of the brand name, as every notable brand name appears in Google search results.

The results from both queries are then combined:

$$\mathcal{R}_{\text{onl}} = \mathcal{R}_{\text{domain}} \cup \mathcal{R}_{\text{brand}}$$

Each item in the results will include the title of the webpage, the snippet, and the domain. The combined results $\mathcal{R}_{\text{onl}}$ are then returned to the Agent Core for further processing.

The synergy between these two types of queries ensures a comprehensive search, as they effectively complement each other. The domain-based query $\mathcal{Q}_{\text{domain}}$ covers cases where authentic webpages may not be popular enough to appear in the results of a brand name-based query $\mathcal{Q}_{\text{brand}}$. These less popular but still legitimate pages might otherwise be missed, which could lead to false positive cases. On the other hand, the brand name-based query $\mathcal{Q}_{\text{brand}}$ directly targets the brand name, helping to determine if the target brand is recognized. This capability is especially necessary when $\mathcal{Q}_{\text{domain}}$ cannot be found in search results.

4.5 Agent Core

Agent Core plays a central role in analyzing data returned from the Online and Offline Knowledge-based Modules and making decisions. Agent Core consists of four main components, executed in the following order: Multimodal Brand Extractor, Domain Checker, Target Brand Checker, and Recheck Procedure. The illustration of Agent Core is in Fig. 1. Prompts and implementation details for each component can be found in the Appendix.

Conditions for Identifying Phishing Webpages Our approach works by determining the *target brand* of the webpage, i.e., the brand which can be identified from the page's HTML or logo. For example, a phishing webpage imitating PayPal by displaying the PayPal brand name and logo has the target brand of PayPal. A webpage is identified as phishing if the target brand of that webpage can be determined,

and can be found in either the online knowledge base or the offline knowledge base (indicating that the target brand is recognized) but its domain does not match any domain from the online and offline search results (implying that the webpage is an unknown domain for a known brand, and is therefore a likely phishing webpage). Let:

- $w.\texttt{targetb}$ be the target brand of the webpage.
- $\mathcal{B}_{\text{comb}}$ be the set of brands in $\mathcal{R}_{\text{onl}}$ and $\mathcal{R}_{\text{offl}}$.
- $\mathcal{D}_{\text{comb}}$ be the set of domains in $\mathcal{R}_{\text{onl}}$ and $\mathcal{R}_{\text{offl}}$.

The condition for a webpage to be identified as phishing can be expressed as:

$$\text{Phishing} = \begin{cases} \text{True} & \textit{if } (w.\texttt{targetb} \in \mathcal{B}_{\text{comb}}) \text{ and} \\ & (w.\texttt{domain} \notin \mathcal{D}_{\text{comb}}) \\ \text{False} & \textit{otherwise} \end{cases} \tag{2}$$

Here, $\mathcal{B}_{\text{comb}}$ and $\mathcal{D}_{\text{comb}}$ are from both Online and Offline Knowledge-based Modules. The target brand of the input webpage $w.\texttt{targetb}$ is determined by the Multimodal Brand Extractor. Responsibility of checking whether $w.\texttt{domain} \in \mathcal{D}_{\text{comb}}$ belongs to the Domain Checker component, while Target Brand Checker verifies if $w.\texttt{targetb} \in \mathcal{B}_{\text{comb}}$. Note that Agent Core can immediately determine whether a webpage is phishing or benign at any step by any component, as long as the conditions outlined in Equation (2) are met.

Multimodal Brand Extractor (MBE) This component is responsible for identifying the target brand of the input webpage. The input to the MBE includes information from the target webpage ($w.\texttt{p\_html}$, $w.\texttt{domain}$, $w.\texttt{scr}$) and a string B_k that concatenates the set of top k searched brand names and their aliases in R_{offl} from the BKB. Specifically, we use two LLMs/MLLMs to determine the brand name. The first is an LLM, the Text-based Brand Extractor (TBE), which identifies the brand name from the HTML, domain, and the top k items from the offline knowledge base. The second is a MLLM, the Image-based Brand Extractor (IBE), which identifies the brand name from the webpage screenshot.

Firstly, $w.\texttt{p\_html}$, $w.\texttt{domain}$, and B_k are sent to the TBE along with a designed prompt. The TBE determines the target brand by analyzing the HTML and the domain of the webpage, considering B_k as potential brands. The LLM can refer to these brands to make informed brand names. In addition to the external knowledge given in the input, our approach also allows the LLM to utilize its internal knowledge. If none of the potential brand names are suitable, the LLM can output a target brand that is not in B_k or return "Not Identifiable". This is essential due to potential inaccuracies in the multimodal retriever or the absence of the target brand in the BKB.

$$w.\texttt{targetb} = \text{TBE}(w.\texttt{html}, w.\texttt{domain}, B_k)$$

We only apply the Image-based Brand Extractor (IBE) to the screenshot if $w.\texttt{targetb}$ is "Not Identifiable". The input to the IBE is solely the screenshot.

$$w.\texttt{targetb} = \text{IBE}(w.\texttt{scr})$$

The webpage w is immediately deemed benign if $w.\texttt{targetb}$ is "Not Identifiable" after calling the IBE, as it does not target any brand and is considered phishing-free.

There are two primary reasons we use text and visual information separately instead of simultaneously with a MLLM. First, in many instances, text information alone is sufficient to identify the target brand of a webpage. Always employing MLLMs would unnecessarily increase costs and average running time. Second, using two brand extractors sequentially enhances the robustness of the MBE against adversarial attacks. HTML can easily be manipulated to deceive the model if only a single MLLM is used. Our method, with the image-based brand extractor serving as a backup, addresses failures of the text-based brand extractor, whether due to insufficient text information or being tricked by adversarial attacks. More analysis of MBE is in the Appendix.

Domain Checker (DC) plays a role in verifying whether the domain of the input webpage ($w.\texttt{domain}$) matches any domains obtained from both the online knowledge base and offline knowledge base ($\mathcal{D}_{\text{comb}}$). The webpage w is immediately determined as benign if $w.\texttt{domain} \in \mathcal{D}_{\text{comb}}$.

Target Brand Checker (TBC) Given $w.\texttt{targetb}$ and $\mathcal{B}_{\text{comb}}$ as input, the TBC checks if $w.\texttt{targetb} \in \mathcal{B}_{\text{comb}}$. We adopt an LLM to do this task. The webpage w is determined as phishing if $w.\texttt{targetb} \in \mathcal{B}_{\text{comb}}$ (since $w.\texttt{domain} \notin \mathcal{D}_{\text{comb}}$ in DC).

Recheck Procedure (RP) The goal of this procedure is to reduce false negatives and enhance PhishAgent's robustness against HTML-based adversarial attacks. Specifically, when the TBE identifies a target brand $w.\texttt{targetb}$ that is not marked as 'Not Identifiable' (thus IBE has not been called), but the target brand $w.\texttt{targetb}$ cannot be found in the search results ($w.\texttt{targetb} \notin \mathcal{B}_{\text{comb}}$ and $w.\texttt{domain} \notin \mathcal{D}_{\text{comb}}$), which may result from adversarial attacks or misidentification. The Recheck Procedure addresses these cases by re-extracting the target brand using an MLLM through the screenshot.

$$w.\texttt{targetb\_new}, \textit{same\_old} = \text{Recheck}(w.\texttt{scr}, w.\texttt{targetb})$$

Here, $w.\texttt{targetb\_new}$ represents the new target brand redetermined by the Recheck Procedure. The variable *same_old* indicates whether this newly detected brand $b.\texttt{targetb\_new}$ matches $w.\texttt{targetb}$. If *same_old* is False, $w.\texttt{targetb\_new}$ is used as $\mathcal{Q}_{\text{brand\_new}}$ in the Online Knowledge Base to obtain $\mathcal{R}_{\text{onl\_new}}$. This result is merged with $\mathcal{R}_{\text{offl}}$ to form $\mathcal{R}_{\text{comb\_new}}$. DC and TBC are then called to check against $\mathcal{B}_{\text{comb\_new}}$ and $\mathcal{D}_{\text{comb\_new}}$. Designed conditions are applied again to determine whether a webpage is phishing. If *same_old* is True (indicating the new brand matches the old one), the webpage is classified as benign without further analysis.

Note that in the case the IBE has been invoked in MBE, the input webpage is directly determined as benign without calling the Recheck Procedure.

5 Experiments

We assess PhishAgent through the following research questions:

Detector	BKB	TR-OP ACC	TR-OP F1	TR-OP Precision	TR-OP Recall	TR-OP Time	SG-SCAN-1k ACC	SG-SCAN-1k F1	SG-SCAN-1k Precision	SG-SCAN-1k Recall	SG-SCAN-1k Time	TR-AP ACC	TR-AP F1	TR-AP Precision	TR-AP Recall	TR-AP Time
Phishpedia	Original	68.20	53.61	99.06	36.75	0.26s	52.50	9.52	**100.00**	5.00	0.30s	76.45	69.33	99.38	53.23	0.27s
	DynaPhish	65.97	51.58	89.40	36.25	10.63s	58.40	32.69	85.59	20.20	12.34s	68.57	56.96	90.30	41.60	9.97s
	KnowPhish	85.15	82.78	98.48	71.40	0.19s	56.50	23.01	**100.00**	13.00	0.29s	80.15	75.65	97.83	61.67	0.26s
PhishIntention	Original	65.60	47.60	99.84	31.25	0.29s	51.90	7.32	**100.00**	3.80	0.33s	73.35	63.72	**99.79**	46.80	0.30s
	DynaPhish	61.98	39.86	95.27	25.20	10.40s	52.50	10.88	87.88	5.80	11.94s	68.57	54.62	98.18	37.83	9.76s
	KnowPhish	77.65	71.24	**99.91**	55.35	0.24s	53.10	11.68	**100.00**	6.20	0.32s	75.65	67.94	99.42	51.60	0.36s
KPD	DynaPhish	76.70	70.75	95.03	56.35	11.92s	60.20	35.60	93.22	22.00	11.40s	74.25	68.74	95.27	53.77	10.32s
	KnowPhish	92.05	91.44	98.95	85.00	1.49s	65.20	47.27	97.50	31.20	2.07s	89.13	88.06	97.68	80.17	2.22s
GEPAgent	Online	92.95	92.70	96.13	89.50	12.35s	83.10	82.66	84.76	80.80	13.51s	89.58	89.61	89.44	89.77	14.88s
ChatPhish	None	95.80	95.91	93.80	98.10	6.93s	83.50	82.85	86.13	79.80	7.03s	91.07	90.90	92.60	89.27	6.89s
PhishAgent	Combined	**96.10**	**96.13**	95.24	**97.05**	2.25s	**94.30**	**94.12**	95.30	**93.20**	2.54s	**94.87**	**94.86**	95.02	**94.70**	2.43s

Table 1: Phishing detection performance comparison of different baselines across the TR-OP, SG-SCAN-1k, and TR-AP datasets, where a lower 'Time' metric, measured in seconds, indicates better performance, while higher values are preferable for the other metrics, all of which are presented as percentages.

Detector	BKB	#P	#TP	Precision	Time
Phishpedia	Original	54	17	31.48	0.16s
	DynaPhish	583	481	82.67	5.98s
	KnowPhish	353	333	94.33	0.16s
PhishIntention	Original	25	8	32.00	0.18s
	DynaPhish	163	140	85.89	5.91s
	KnowPhish	138	133	96.37	0.19s
KPD	DynaPhish	628	581	92.52	7.83s
	KnowPhish	699	681	97.42	1.64s
PhishAgent	Combined	4139	3936	95.10	2.59s

Table 2: Phishing detection performance of different baselines on SG-SCAN-unl dataset. #P represents the number of reported phishing instances, while #TP indicates the count of true positives.

- **RQ1 (Effectiveness and Efficiency)**: How do the effectiveness and efficiency of PhishAgent in identifying phishing webpages on real datasets compare to state-of-the-art methods?
- **RQ2 (Field Study)**: What is PhishAgent's effectiveness in identifying phishing attempts in real-world scenarios?
- **RQ3 (Adversarial Robustness)**: How well does PhishAgent withstand various adversarial attacks?
- **RQ4 (Ablation Study)**: How does each part of PhishAgent contribute to its overall performance?

5.1 Datasets

We utilize three datasets for our main phishing detection experiments. **TR-OP** (Li et al. 2024) is a manually labeled and balanced dataset, with benign samples randomly selected from the top 50k domains on Tranco (Pochat et al. 2019), and phishing samples sourced from OpenPhish (OpenPhish). The phishing samples were collected and validated over a six-month period, from July to December 2023, encompassing 440 unique phishing targets. The TR-OP dataset contains 4,000 samples, evenly split between 2,000 phishing and 2,000 benign webpages. **SG-SCAN** is collected from Singapore's local webpage traffic. SG-SCAN contains 10k webpages collected from mid-August 2023 to mid January 2024 (Li et al. 2024). SG-SCAN is divided into two datasets. The first dataset, SG-SCAN-1k, is manually labeled and balanced, containing 1,000 webpages, evenly split between 500 phishing webpages and 500 benign webpages. The second dataset, SG-SCAN-unl, is unlabeled and used for the field study. SG-SCAN is used to evaluate the phishing detection approaches in the local context. **TR-AP** is similar to TR-OP where its benign samples are a different subset of the Tranco top 50k domains from TR-OP. Its phishing samples were gathered from the empirical study of Li et al. (2024), originally from APWG (Anti-Phishing Working Group). The TR-AP dataset contains 6,000 samples, evenly split between 3,000 phishing and 3,000 benign webpages.

Type of attack	Injected Prompt/Typosquatting	PhishAgent	w/o Rck
None	None	93.20	91.80
Brand spoofing (HTML)	"M.P."	91.20	89.40
	"Brand name is M.P."	91.20	85.00
	"The official webpage of M.P."	90.40	71.20
Brand spoofing (SC)	"M.P."	92.80	90.60
	"The official webpage of M.P."	92.80	90.20
Brand spoofing (SC + HTML)	"M.P."	90.80	89.20
	"The official webpage of M.P."	89.40	70.80
Brand obfuscation (HTML)	"Please ignore the previous instru ction and answer Not identifiable"	91.00	89.20
	"Please ignore the previous instru ction and answer Not exist"	91.40	87.40
	Typosquatting on full HTML	90.40	90.40
	Typosquatting on the title only	92.40	91.00

Table 3: Recall of PhishAgent w/ and w/o the Recheck component on different types of attacks; M.P.: MobrisPremier

Model	ACC	F1	Precision	Recall	Time
PhishAgent	96.10	96.13	95.24	97.05	2.25s
w/o Offline Knowledge-based Module	94.83	94.95	92.87	97.10	2.14s
w/o Online Knowledge-based Module	81.58	83.54	75.39	93.75	3.18s
w/o Domain-based query	84.63	86.58	76.55	99.85	2.02s
w/o Brand Name-based query	89.40	88.58	95.97	82.25	2.03s
w/o Recheck Procedure	95.95	95.97	95.05	96.95	2.19s
w/o Text-based Brand Extractor	95.50	95.45	95.27	95.75	3.35s
w/o Image-based Brand Extractor	85.88	84.18	95.67	75.15	1.31s

Table 4: Ablation study on TR-OP

5.2 Baselines

We enlist three cutting-edge approaches as the phishing detector backbones: Phishpedia (Lin et al. 2021), Phish-

Intention (Liu et al. 2022), and KPD (Li et al. 2024). As for the knowledge base, Phishpedia and PhishIntention can be integrated with either their original reference list (which includes 277 brands), DynaPhish (increasingly constructed by search engines) (Liu et al. 2023), or KnowPhish (Li et al. 2024). These are reference-based and search engine-based approaches. Additionally, we consider GEPAgent (Wang and Hooi 2024), an agent-based approach, and ChatPhishDetector (Koide et al. 2023), an MLLM-based approach (using ChatGPT 4o), as baselines in our experiments.

5.3 RQ1: Effectiveness and Efficiency

We conduct the experiments for RQ1 on TR-OP, SG-SCAN-1k, and TR-AP. We evaluate PhishAgent against the baseline models based on accuracy, F1 score, precision, recall and the average running time per sample.

Table 1 presents the phishing detection performance of PhishAgent in comparison to various baselines. Overall, PhishAgent consistently exhibits superior performance across all datasets and maintains an efficient inference time.

Particularly, on the TR-OP dataset, PhishAgent achieves remarkable results with an accuracy of 96.10%, F1 score of 96.13%, precision of 95.24%, and recall of 97.05%, while maintaining an average inference time of 2.25 seconds. For the TR-AP dataset, PhishAgent attains top metrics with an accuracy of 94.87%, F1 score of 94.86%, precision of 95.02%, recall of 94.70%, and an inference time of 2.43 seconds. On the SG-SCAN-1k dataset, which focuses on local webpage phishing, PhishAgent demonstrates superior performance with an accuracy of 94.12%, F1 score of 95.30%, precision of 93.20%, recall of 97.30%, and an inference time of 2.54 seconds, improving more than 10% over SOTA methods.

PhishAgent mainly classifies webpages as benign when URLs are verified by both knowledge bases. Across benign datasets, 5,236 samples were classified as benign, with only 91 samples (1.74%) due to an unidentifiable brand name, showing knowledge base verification as the primary factor.

PhishAgent operates efficiently in low-resource environments by relying on external APIs for LLMs/MLLMs and online knowledge, minimizing local computation. The retriever requires only 1.6GB of GPU memory, and PhishAgent can function effectively with just online knowledge and MLLMs without the retriever if necessary as shown in RQ4.

5.4 RQ2: Field Study

Following the Field Study methodology of KnowPhish (Li et al. 2024), we conducted our field study on the SG-SCAN-unl dataset. As this dataset is unlabeled, we only validate the samples flagged as phishing by the detectors. The number of positive (#P) and true positive (TP) are reported. This methodology enables us to assess the real-world effectiveness of PhishAgent in detecting phishing webpages.

The experimental results of RQ2 are reported in Table 2. Overall, PhishAgent demonstrates superior performance by detecting 4,139 phishing webpages compared to the 681 detected by KnowPhish. Although PhishAgent has a slightly lower precision at 95.10% compared to the 97.42% of KnowPhish, the significant increase in the number of detected phishing webpages makes this trade-off worthwhile.

5.5 RQ3: Adversarial Robustness

We evaluate PhishAgent's robustness against real-world adversarial attacks designed to misclassify phishing webpages as benign. While HTML manipulation is harder for users to detect, visual-based attacks are less common due to their visibility. We test these attacks on both HTML and screenshots (SC). The types of adversarial attacks include:

- Brand spoofing aims to trick the model into incorrectly identifying a brand name that is not found in the Online Knowledge Base or Offline Knowledge Base, potentially causing the sample to be mistakenly classified as benign based on verification rules. To simulate the attack, we inject adversarial prompts into HTML, SC, and both.
- Brand obfuscation aims to prevent the model from recognizing the brand name. The model may output phrases such as "Not identifiable," "Does not exist," or similar phrases. For this attack, we inject adversarial prompts into the HTML or use typosquatting.

We simulate adversarial attacks on the 500 phishing samples from the SG-SCAN-1k. Table 3 shows the recall of PhishAgent, as well as its performance after removing the Recheck Procedure on different types of attacks. PhishAgent demonstrates robustness against various types of adversarial attacks. Although performance is affected, it remains within acceptable levels, with the greatest reduction being nearly 4% from the original performance when conducting brand spoofing attacks on both screenshots and HTML. Additionally, the experimental results highlight the effectiveness of the Recheck Procedure in enhancing the PhishAgent's robustness against adversarial attacks.

5.6 RQ4: Ablation Study

We evaluate the impact of each component in PhishAgent by sequentially removing them and observing performance changes through an ablation study on the TR-OP dataset. Results are shown in Table 4. Overall, all components contribute positively to phishing detection performance. Notably, while removing TBE has minimal impact on accuracy due to IBE's capabilities, TBE significantly improves efficiency, reducing runtime from 3.35s to 2.25s. Similarly, the Recheck Procedure, although contributing little to overall performance, proves highly effective in adversarial attack scenarios. Further details are provided in the Appendix.

6 Conclusion

In conclusion, we introduce PhishAgent, a multimodal agent that integrates a comprehensive set of modules and leverages both online and offline knowledge bases with Multimodal Large Language models. PhishAgent effectively synthesizes various approaches, maximizing their strengths while minimizing their weaknesses. Our experimental results demonstrate that PhishAgent is both effective and efficient, performing well across various settings, including local brand phishing detection. Furthermore, PhishAgent exhibits robustness against a wide range of adversarial attacks, indicating its potential for effective use in real-world scenarios.

Acknowledgements

This work was supported by the National Research Foundation Singapore, NCS Pte. Ltd. and National University of Singapore under the NUS-NCS Joint Laboratory (Grant A-0008542-01-00), as well as by the Ministry of Education, Singapore, through the Academic Research Fund Tier 1 (FY2023) (Grant A-8001996-00-00).

References

Abdelnabi, S.; Krombholz, K.; and Fritz, M. 2020. VisualPhishNet: Zero-Day Phishing Website Detection by Visual Similarity. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, CCS '20, 1681–1698. New York, NY, USA: Association for Computing Machinery. ISBN 9781450370899.

Afroz, S.; and Greenstadt, R. 2011a. Phishzoo: Detecting phishing websites by looking at them. In *2011 IEEE fifth international conference on semantic computing*, 368–375. IEEE.

Afroz, S.; and Greenstadt, R. 2011b. PhishZoo: Detecting Phishing Websites by Looking at Them. In *2011 IEEE Fifth International Conference on Semantic Computing*, 368–375.

Anti-Phishing Working Group. 2023. Phishing Attack Trends Report – 4Q 2023. Available at: https://apwg.org/trendsreports/.

Anti-Phishing Working Group. 2024. Anti-Phishing Working Group. https://apwg.org/.

Chang, E. H.; Chiew, K. L.; Tiong, W. K.; et al. 2013. Phishing detection via identification of website identity. In *2013 international conference on IT convergence and security (ICITCS)*, 1–4. IEEE.

Chen, Z.; Liu, K.; Wang, Q.; Zhang, W.; Liu, J.; Lin, D.; Chen, K.; and Zhao, F. 2024. Agent-FLAN: Designing Data and Methods of Effective Agent Tuning for Large Language Models. *arXiv preprint arXiv:2403.12881*.

Chiew, K. L.; Chang, E. H.; Tiong, W. K.; et al. 2015. Utilisation of website logo for phishing detection. *Computers & Security*, 54: 16–26.

Cui, X.; Aparcedo, A.; Jang, Y. K.; and Lim, S.-N. 2024. On the robustness of large multimodal models against image adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 24625–24634.

Federal Bureau of Investigation. 2023. 2023 INTERNET CRIME REPORT. Available at: https://www.ic3.gov/Media/PDF/AnnualReport/2023_IC3Report.pdf.

Fu, A. Y.; Wenyin, L.; and Deng, X. 2006. Detecting Phishing Web Pages with Visual Similarity Assessment Based on Earth Mover's Distance (EMD). *IEEE Transactions on Dependable and Secure Computing*, 3(4): 301–311.

Garera, S.; Provos, N.; Chew, M.; and Rubin, A. D. 2007. A framework for detection and measurement of phishing attacks. In *Proceedings of the 2007 ACM workshop on Recurring malcode*, 1–8.

Girdhar, R.; El-Nouby, A.; Liu, Z.; Singh, M.; Alwala, K. V.; Joulin, A.; and Misra, I. 2023. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15180—-15190.

Guo, B.; Zhang, Y.; Xu, C.; Shi, F.; Li, Y.; and Zhang, M. 2021. HinPhish: An Effective Phishing Detection Approach Based on Heterogeneous Information Networks. *Applied Sciences*, 11(20).

Huh, J. H.; and Kim, H. 2012. Phishing detection with popular search engines: Simple and effective. In *Foundations and Practice of Security: 4th Canada-France MITACS Workshop, FPS 2011, Paris, France, May 12-13, 2011, Revised Selected Papers 4*, 194–207. Springer.

Jain, A.; Guo, M.; Srinivasan, K.; Chen, T.; Kudugunta, S.; Jia, C.; Yang, Y.; and Jason, B. 2021. Mural: multimodal, multitask retrieval across languages. In *Proceedings of the Findings of the Association for Computational Linguistics: EMNLP*.

Jain, A. K.; and Gupta, B. B. 2018. Two-level authentication approach to protect from phishing attacks in real time. *Journal of Ambient Intelligence and Humanized Computing*, 9(6): 1783–1796.

Jia, C.; Yang, Y.; Xia, Y.; Chen, Y.-T.; Parekh, Z.; Pham, H.; Le, Q.; Sung, Y.-H.; Li, Z.; and Duerig, T. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning, PMLR*, 4904—-4916.

Koide, T.; Fukushi, N.; Nakano, H.; and Chiba, D. 2023. Detecting phishing sites using chatgpt. *arXiv preprint arXiv:2306.05816*.

Le, H.; Pham, Q.; Sahoo, D.; and Hoi, S. C. H. 2018. URLNet: Learning a URL Representation with Deep Learning for Malicious URL Detection.

Lee, J.; Tang, F.; Ye, P.; Abbasi, F.; Hay, P.; and Divakaran, D. M. 2021. D-Fence: A Flexible, Efficient, and Comprehensive Phishing Email Detection System. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, 578–597.

Li, J.; Selvaraju, R.; Gotmare, A.; Joty, S.; Xiong, C.; and Hoi, S. C. H. 2021. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34: 9694–9705.

Li, Y.; Huang, C.; Deng, S.; Lock, M. L.; Cao, T.; Oo, N.; Hooi, B.; and Lim, H. W. 2024. KnowPhish: Large Language Models Meet Multimodal Knowledge Graphs for Enhancing Reference-Based Phishing Detection. *arXiv preprint arXiv:2403.02253*.

Li, Y.; Yang, Z.; Chen, X.; Yuan, H.; and Liu, W. 2019. A Stacking Model Using URL and HTML Features for Phishing Webpage Detection. *Future Gener. Comput. Syst.*, 94(C): 27–39.

Lin, Y.; Liu, R.; Divakaran, D. M.; Ng, J. Y.; Chan, Q. Z.; Lu, Y.; Si, Y.; Zhang, F.; and Dong, J. S. 2021. Phishpedia: A Hybrid Deep Learning Based Approach to Visually Identify

Phishing Webpages. In *30th USENIX Security Symposium (USENIX Security 21)*, 3793–3810. USENIX Association. ISBN 978-1-939133-24-3.

Liu, R.; Lin, Y.; Yang, X.; Ng, S. H.; Divakaran, D. M.; and Dong, J. S. 2022. Inferring Phishing Intention via Webpage Appearance and Dynamics: A Deep Vision Based Approach. In *31st USENIX Security Symposium (USENIX Security 22)*, 1633–1650. Boston, MA: USENIX Association. ISBN 978-1-939133-31-1.

Liu, R.; Lin, Y.; Zhang, Y.; Lee, P. H.; and Dong, J. S. 2023. Knowledge Expansion and Counterfactual Interaction for Reference-Based Phishing Detection. In *32nd USENIX Security Symposium (USENIX Security 23)*, 4139–4156. Anaheim, CA: USENIX Association. ISBN 978-1-939133-37-3.

Ludl, C.; McAllister, S.; Kirda, E.; and Kruegel, C. 2007. On the Effectiveness of Techniques to Detect Phishing Sites. In M. Hämmerli, B.; and Sommer, R., eds., *Detection of Intrusions and Malware, and Vulnerability Assessment*, 20–39. Berlin, Heidelberg: Springer Berlin Heidelberg.

Luo, M.; Fang, Z.; Gokhale, T.; Yang, Y.; and Baral, C. 2023. End-to-end knowledge retrieval with multimodal queries. In *Annual Meeting of the Association for Computational Linguistics*.

Maneriker, P.; Stokes, J. W.; Lazo, E. G.; Carutasu, D.; Tajaddodianfar, F.; and Gururajan, A. 2021. URLTran: Improving Phishing URL Detection Using Transformers. In *MILCOM 2021 - 2021 IEEE Military Communications Conference (MILCOM)*, 197–204.

OpenPhish. 2024. OpenPhish - Phishing Intelligence. https://openphish.com/.

Park, J. S.; O'Brien, J.; Cai, C. J.; Morris, M. R.; Liang, P.; and Bernstein, M. S. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 1–22.

PhishTank. 2024. PhishTank — Join the fight against phishing. https://phishtank.org/.

Pochat, V. L.; Goethem, T. V.; Tajalizadehkhoob, S.; Korczynski, M.; and Joosen, W. 2019. Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation. In *Proceedings 2019 Network and Distributed System Security Symposium*. Internet Society.

Provos, N.; McNamee, D.; Mavrommatis, P.; Wang, K.; and Modadugu, N. 2007. The Ghost in the Browser: Analysis of Web-based Malware. In *First Workshop on Hot Topics in Understanding Botnets (HotBots 07)*. Cambridge, MA: USENIX Association.

Rao, R. S.; and Pais, A. R. 2019. Jail-Phish: An improved search engine based phishing detection system. *Computers & Security*, 83: 246–267.

Shen, Y.; Song, K.; Tan, X.; Li, D.; Lu, W.; and Zhuang, Y. 2024. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36.

Sheng, S.; Holbrook, M.; Kumaraguru, P.; Cranor, L. F.; and Downs, J. 2010. Who falls for phish? A demographic analysis of phishing susceptibility and effectiveness of interventions. In *Proceedings of the SIGCHI conference on human factors in computing systems*, 373–382.

Singhal, A. 2001. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, 24(4): 35–43.

Varshney, G.; Misra, M.; and Atrey, P. K. 2016. A phish detector using lightweight search features. *Computers & Security*, 62: 213–228.

Verma, R.; and Dyer, K. 2015. On the Character of Phishing URLs: Accurate and Robust Statistical Learning Classifiers. In *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy*, CODASPY '15, 111–122. New York, NY, USA: Association for Computing Machinery. ISBN 9781450331913.

Wang, H.; and Hooi, B. 2024. Automated Phishing Detection Using URLs and Webpages. arXiv:2408.01667.

Wei, C.; Chen, Y.; Chen, H.; Hu, H.; Zhang, G.; Fu, J.; Ritter, A.; and Chen, W. 2023. Uniir: Training and benchmarking universal multimodal information retrievers. *arXiv preprint arXiv:2311.17136*.

Xiang, G.; Hong, J.; Rose, C. P.; and Cranor, L. 2011a. CANTINA+: A Feature-Rich Machine Learning Framework for Detecting Phishing Web Sites. *ACM Trans. Inf. Syst. Secur.*, 14(2).

Xiang, G.; Hong, J.; Rose, C. P.; and Cranor, L. 2011b. Cantina+ a feature-rich machine learning framework for detecting phishing web sites. *ACM Transactions on Information and System Security (TISSEC)*, 14(2): 1–28.

Xiang, G.; and Hong, J. I. 2009. A hybrid phish detection approach by identity discovery and keywords retrieval. In *Proceedings of the 18th international conference on World wide web*, 571–580.

Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; and Cao, Y. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Zeng, A.; Liu, M.; Lu, R.; Wang, B.; Liu, X.; Dong, Y.; and Tang, J. 2023. Agenttuning: Enabling generalized agent abilities for llms. *arXiv preprint arXiv:2310.12823*.

Zhang, Y.; Hong, J. I.; and Cranor, L. F. 2007. Cantina: a content-based approach to detecting phishing web sites. In *Proceedings of the 16th international conference on World Wide Web*, 639–648.