



KnowPhish: Large Language Models Meet Multimodal Knowledge Graphs for Enhancing Reference-Based Phishing Detection

Yuexin Li, Chengyu Huang, and Shumin Deng, *National University of Singapore*;
Mei Lin Lock, *NCS Cyber Special Ops-R&D*; Tri Cao, *National University of Singapore*;
Nay Oo and Hoon Wei Lim, *NCS Cyber Special Ops-R&D*; Bryan Hooi,
National University of Singapore

<https://www.usenix.org/conference/usenixsecurity24/presentation/li-yuexin>

**This paper is included in the Proceedings of the
33rd USENIX Security Symposium.**

August 14-16, 2024 • Philadelphia, PA, USA

978-1-939133-44-1

**Open access to the Proceedings of the
33rd USENIX Security Symposium
is sponsored by USENIX.**

KnowPhish: Large Language Models Meet Multimodal Knowledge Graphs for Enhancing Reference-Based Phishing Detection

Yuexin Li¹, Chengyu Huang¹, Shumin Deng¹, Mei Lin Lock², Tri Cao¹,
Nay Oo², Hoon Wei Lim², Bryan Hooi¹

¹National University of Singapore, ²NCS Cyber Special Ops-R&D

Abstract

Phishing attacks have inflicted substantial losses on individuals and businesses alike, necessitating the development of robust and efficient automated phishing detection approaches. Reference-based phishing detectors (RBPDs), which compare the logos on a target webpage to a known set of logos, have emerged as the state-of-the-art approach. However, a major limitation of existing RBPDs is that they rely on a manually constructed brand knowledge base, making it infeasible to scale to a large number of brands, which results in false negative errors due to the insufficient brand coverage of the knowledge base. To address this issue, we propose an automated knowledge collection pipeline, using which we collect a large-scale multimodal brand knowledge base, KnowPhish, containing 20k brands with rich information about each brand. KnowPhish can be used to boost the performance of existing RBPDs in a plug-and-play manner. A second limitation of existing RBPDs is that they solely rely on the image modality, ignoring useful textual information present in the webpage HTML. To utilize this textual information, we propose a Large Language Model (LLM)-based approach to extract brand information of webpages from text. Our resulting multimodal phishing detection approach, KnowPhish Detector (KPD), can detect phishing webpages with or without logos. We evaluate KnowPhish and KPD on a manually validated dataset, and a field study under Singapore’s local context, showing substantial improvements in effectiveness and efficiency compared to state-of-the-art baselines.

1 Introduction

Phishing attacks are one of the most impactful types of scams, harming both individuals and businesses: in 2023, an estimated \$1.026 trillion was lost by consumers worldwide in scams [5], of which phishing scams are among the most common types. Phishing scams also account for about 90% of data breaches for organizations [12]. This problem has been exacerbated by the proliferation of automated phishing kits,

enabling malicious actors to easily mimic genuine pages while evading detection using countermeasures [63]. Accordingly, the number of phishing attacks has increased by 47.2% in 2022 compared to the previous year [64]. These underscore the importance and urgency of tackling the issue, and the need for effective automatic phishing detection approaches.

Many efforts have been made to counter phishing attacks, including anti-phishing blacklists, heuristic-based, and feature-based models [15]. Blacklist-based methods [42, 44, 47] compare an input URL with a predefined blacklist of malicious URLs, but these are reactive approaches that require phishing sites to be first reported or detected through other means. Heuristic-based [14] and feature-based models [19, 29, 32, 37, 39, 56, 61] extract features to proactively identify new phishing webpages. However, as these do not utilize logo information, they are greatly limited in their ability to detect plentiful phishing pages which are mainly identifiable by the presence of a logo [34]. Moreover, as they depend on statistical features for detecting phishing pages, they are susceptible to distribution shifts due to the constantly changing nature of phishing campaigns.

In contrast, reference-based phishing detectors (RBPDs), which work by comparing the logos on a target webpage to a known set of logos, have been established as the state-of-the-art phishing detection paradigm, garnering considerable research attention [1, 33, 34]. Specifically, an RBPD consists of *brand knowledge base* (BKB) containing brand information (the logos and legitimate domains of brands) and a *detector backbone* which uses information from this BKB for phishing detection. To detect if a webpage is a phishing or benign page, RBPDs first identify the webpage’s *brand intention*, i.e., the brand that the webpage presents itself as (e.g., a webpage with an Adobe logo and appearance has the brand intention of Adobe). Then, if the webpage is detected to have an intention of a certain brand, but its domain does not match the legitimate domains of that brand, the webpage is classified as phishing. As virtually all phishing pages do not utilize the legitimate domains of the original brand, this approach typically obtains high precision. Moreover, as this approach is based on an

invariant that does not change over time, it is relatively robust to distribution shift [33, 34].

Challenges. Despite the advantages of RBPDs, they have two major limitations which we focus on. (1) The first limitation is *Limited-Scale BKB*: RBPDs fundamentally rely on their BKB to identify the brand intention of a website. However, it is labor-intensive to construct and maintain a large-scale BKB manually. Phishpedia [33] and PhishIntention [34] rely on manual curation, hence are limited to a small BKB of 277 brands. A recent method, DynaPhish [35], proposes to dynamically expand the BKB during deployment time. However, this leads to extremely long running time, e.g., averaging 10.6 seconds per sample in our experiments. Moreover, it may fail to construct brand knowledge of novel phishing targets if phishing pages’ logos are different from those displayed on legitimate pages. (2) The second limitation is *Textual Brand Intention*: Phishing webpages can convey their brand intention via text in HTML, instead of via logos. Existing RBPDs cannot identify it because they solely operate within the image modality.

Present Work. In this paper, we seek to address both of these issues in the context of a static environment, where input webpages are not interactable. First, we propose an automated knowledge collection pipeline, with which we construct a large-scale multimodal BKB named *KnowPhish*. KnowPhish is constructed based on our empirical analysis which finds that phishing targets mostly belong to a few high-value industries. Hence, using brand-industry relations modeled from a publicly available knowledge base, Wikidata [57], we search for a set of potential phishing targets and their brand knowledge predictively, leading to a BKB covering around 20k potential phishing targets. We also incorporate extra data sources such as Tranco domain list [46] and Google Image Search [18] for brand knowledge enhancement. Therefore, KnowPhish is equipped with rich logo, alias, and domain variants of each brand, which can be used to boost the performance of existing RBPDs in a plug-and-play manner.

Next, to address the issue of textual brand intention, we develop a Large Language Model (LLM)-based approach to identify the brand intention of webpages in conjunction with the alias information in our KnowPhish BKB. Our approach can be directly integrated with any standard visual RBPD [34], augmenting it to form a *multimodal RBPD* which detects brand intention through both visual and textual modalities. Our resulting multimodal phishing detection approach, named *KnowPhish Detector (KPD)*, can operate with static webpage data to detect phishing webpages with or without logos.

We then evaluate the effectiveness and efficiency of KnowPhish and KPD on TR-OP, a manually validated dataset that comprises 5k benign and 5k phishing webpages. We also evaluate our approach on a field study under Singapore’s local context, to study how well different approaches generalize to such a local context. The resulting data, which we call SG-SCAN, contains 10k webpages from Singapore local web-

page traffic over 6 months. In experiments on the two datasets, KnowPhish significantly boosts the effectiveness of various RBPDs, and is 30 or more times faster than the on-deployment framework DynaPhish [35], when equipped with image-based RBPDs. Moreover, incorporating our multimodal approach, KPD, can substantially boost the number of detected phishing webpages.

In summary, our contributions are three-fold:

- **Multimodal Brand Knowledge Base.** We propose KnowPhish, a large-scale multimodal BKB for phishing detection, and its automated construction approach. KnowPhish can be used in any RBPDs to boost their brand knowledge in a plug-and-play manner.
- **Multimodal Reference-based Phishing Detector.** We propose an LLM-based approach to identify textual brands from HTML to handle logo-less phishing webpages. Our approach directly integrates with any existing RBPD, augmenting it to form a multimodal RBPD, named KnowPhish Detector (KPD), which can detect phishing webpages with or without logos.
- **Effectiveness and Efficiency** Extensive experiments show that KnowPhish significantly enhances the performance of various RBPDs, including KPD, while achieving much better runtime efficiency than DynaPhish. We also demonstrate their effectiveness in a field study and validate the robustness of KPD to adversarial attacks.

2 Motivating Examples

The development of KnowPhish and KPD is motivated by the brand knowledge construction difficulty of DynaPhish. Below, we present a brief workflow of DynaPhish, a few examples of where DynaPhish fails to build brand knowledge, and how KnowPhish and KPD address these difficulties.

DynaPhish expands brand knowledge through a two-step process when encountering a webpage with an unseen logo. (1) It first utilizes Google Search (GS) to validate the webpage’s domain popularity to determine its benignity. The logo-domain pair will be added to its brand knowledge if such validation passes. (2.1) If the validation fails, it recognizes the brand name of the logo using Google Logo Detector (GLD) and searches for the legitimate domain associated with that logo using another GS with the brand name as the query. (2.2) It then compares the input logo with the logo on each webpage from the GS results. If a match is found, the corresponding domain and logo are added to the brand knowledge.

However, DynaPhish may fail to construct new brand knowledge and correctly classify phishing pages due to the failure in step (2.2), i.e., logo-matching. A few examples of logos from phishing pages and the logos from their corresponding GS result pages are shown in Figure 1. The difference

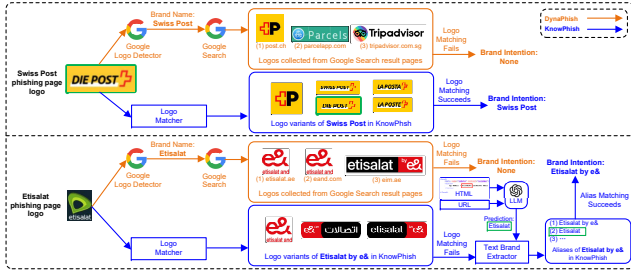


Figure 1: Comparison of the workflow between DynaPhish and KnowPhish to identify the brand intention of the two phishing page examples.

in appearance between the logos on phishing pages and GS result pages causes failure in the logo-matching process, even if GLD identifies the correct brand names, further resulting in brand knowledge construction failure and false negatives.

These examples motivate us to address such false negative problems in two ways. Firstly, we build KnowPhish from a high-quality data source, Wikidata, based on brand profiles (e.g., names and domains). This approach bypasses the logo-matching process used by DynaPhish and directly incorporates enhanced brand knowledge, such as logos and aliases. Therefore, the enriched logo variants increase the chance for image-based RBDs to match the correct brand. For instance, the Swiss Post phishing page logo in Figure 1 is covered in KnowPhish, thus enabling the logo-matcher to identify its brand intention. Secondly, even if image-based RBDs fail to match the logo, we remedy this by also using the text-brand extractor from KPD to identify the text brand intention from HTML and URL, in conjunction with the rich aliases in KnowPhish. As the Etisalat phishing page in Figure 1 shows, an LLM takes both the HTML and URL as input and outputs a predicted brand for the page. The predicted brand matches an alias from KnowPhish, thus the text-brand extractor detects its brand intention as the brand of that alias. We will elaborate on each in Sections 4 and 5, respectively.

3 Formalization

Threat Model. In a phishing attack, a malicious actor misleads visitors into believing that their webpage comes from a legitimate brand, misleading users into providing their credentials (such as username and password). Note that we do not consider other types of attacks, such as malware or miscellaneous scams that do not fit the above description, to be within our scope.

In our threat model, we assume the attacker has full control over their webpage. However, to maintain their webpage’s effectiveness in misleading users, the webpage needs to convey its *brand intention*, i.e., present itself as a particular recognizable brand to visitors. In addition, the webpage needs to be *credential receiving* in some way: e.g., via username and pass-

word fields, but less common approaches exist such as buttons or QR codes. Deviating from either of these two conditions makes it less likely for the phishing attack to succeed, and our empirical observations support that these two conditions hold consistently in phishing webpages.

Formally, consider a webpage w , which consists of its screenshot ($w.screenshot$), its page HTML ($w.html$), and its domain ($w.domain$). As described above, the webpage needs to convey its brand intention, presenting itself as belonging to a brand b . This can be done either in visual form (e.g., through logos in its screenshot) or textual form (e.g., through text in its HTML), so brand information may appear either in $w.screenshot$ or $w.html$ (or both).

Reference-Based Phishing Detection. Since phishing webpages need to convey brand intention, the state-of-the-art RBD approach relies on identifying this brand intention, by comparing images on the page to a set of known *reference* logos. Formally, an RBD consists of a *brand knowledge base* (BKB) and a *detector backbone* utilizing this BKB.

A BKB [33–35] stores brand-related knowledge, taking the form of a list of N brands: b_1, \dots, b_N . For each brand b , we store its name ($b.name$), its logo images ($b.logos$), and its legitimate domains ($b.domains$). In our work, to facilitate the detection of textual brand intention, we further add its list of textual aliases, i.e., a list of common alternate names used to refer to the brand ($b.aliases$), resulting in an *augmented BKB*. Formally, given a brand b , the augmented BKB is:

$$\mathcal{B} = \{(b.name, b.logos, b.domains, b.aliases)\}_{b \in \{b_1, \dots, b_N\}}$$

For example, for the brand PayPal, this may contain: (PayPal, (logo1, logo2), (www.paypal.com), (PYPL)), where logo1 and logo2 are two PayPal logo images.

Next, given a webpage w and a BKB \mathcal{B} , a *detector backbone* $g_{\mathcal{B}}(w)$ outputs either the brand intention that w is predicted to have, or ‘null’ to indicate no predicted brand intention: $g_{\mathcal{B}}(w) \in \mathcal{B} \cup \{\emptyset\}$.

Finally, an RBD, denoted $f_{\mathcal{B}}(w)$, classifies a webpage w as phishing or benign. $f_{\mathcal{B}}$ classifies w as phishing if its detector backbone $g_{\mathcal{B}}$ detects that w presents a brand intention $b' \in \mathcal{B}$ but w ’s domain is inconsistent with any of the legitimate domains of brand b' recorded in \mathcal{B} (i.e., $w.domain \notin b'.domains$). Otherwise, w is classified as benign. Formally:

$$f_{\mathcal{B}}(w) = \begin{cases} \text{Phishing} & \text{if } g_{\mathcal{B}}(w) = b' \text{ and } b' \neq \emptyset \text{ and } \\ & w.domain \notin b'.domains \\ \text{Benign} & \text{otherwise} \end{cases}$$

Evasion Attacks. The attackers may attempt to bypass $f_{\mathcal{B}}$ via the following methods:

T1: Phishing with Logo Variants. To circumvent the online knowledge expansion approach for \mathcal{B} (e.g., DynaPhish [35]), attackers can use other legitimate logo variants of b instead of the ones displayed on b ’s official webpages.

T2: Phishing with Text Brands. Instead of using a logo to present its brand intention, the attacker can rely on text *w.html* to show its brand intention, making image-based phishing detectors completely fail.

T3: HTML-oriented Attacks. Phishing attackers may employ evasion techniques on HTML, such as typosquatting, prompt injection, and text-to-image attacks to hinder effective information extraction by text-based methods.

We address T1 by constructing a large-scale multimodal BKB with rich logo information, i.e., multiple logos per brand. We address T2 by developing an LLM-based approach to extract text brands from HTML. For T3, we show in Section 6.5 that our LLM-based approach is generally robust to different types of adversarial noises in HTML.

4 KnowPhish Construction

In this section, we introduce how to construct *KnowPhish*, a large-scale multimodal BKB. We start by conducting empirical analysis on phishing feeds from different sources and periods to find prospective indicators to search for potential phishing targets proactively. With our empirical findings, we then describe how to automatically construct KnowPhish using a publicly accessible multimodal knowledge graph.

4.1 Empirical Motivation

Our empirical analysis seeks to address the following questions:

QUESTION 1. *Do phishing targets differ across different phishing feeds?*

QUESTION 2. *What are the enduring characteristics shared by phishing feeds across different sources and periods?*

4.1.1 Data

To achieve this objective, we conducted a study using two distinct phishing datasets. The first dataset, D_1 , includes phishing webpages that were collected by [33] from OpenPhish [42] three years ago. It encompasses a total of 29,496 phishing instances targeting 283 different brands. The second dataset, D_2 , comprises phishing samples obtained from APWG [7] at the end of 2022. This dataset contains a total of 5,167 phishing examples targeting 391 unique brands.

Furthermore, we manually categorized each of these phishing targets into one of the following distinct industries, namely: 1) financial, 2) online service, 3) telecommunication, 4) e-commerce, 5) social media, 6) postal service, 7) government, 8) web portal, 9) video game, and 10) gambling. For the brands that cannot be classified into any of the ten categories, we categorize them as 11) other businesses. For instance, Bank of America is categorized into the financial category, while KFC is classified as other businesses.

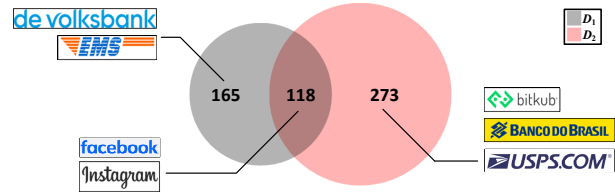


Figure 2: Phishing targets change substantially. The Venn diagram shows the disparities in the phishing targets from the two phishing datasets, with a few phishing target examples provided for illustration.

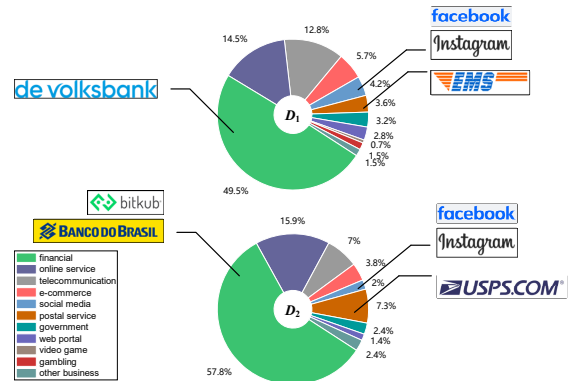


Figure 3: Industries of phishing target brands are relatively consistent. The chart shows the distribution of industries of the phishing targets from the two datasets.

4.1.2 Analysis

We conduct a thorough examination of the disparities in phishing targets and the distribution of their respective industries across the two phishing datasets. Our observations are as follows.

OBSERVATION 1. *Phishing targets may exhibit variation depending on the datasets involved.*

In particular, in Figure 2 we observe significant phishing target disparities between D_1 and D_2 , with only 118 of the 391 brands in D_2 being present in D_1 . This can be attributed to many potential factors, including temporal shifts and data collection methodologies. On the temporal shifts side, emerging phishing targets (e.g., Bitkub) may supplant existing ones over time [35]. On the data collection methodology side, OpenPhish employs proprietary detectors for phishing URL collection, whereas APWG relies on human reports. Such divergent approaches to data gathering directly impact the composition of phishing targets. These factors highlight the non-trivial challenge of manually managing the dynamic nature of phishing targets.

OBSERVATION 2. *Despite the dynamic nature of phishing targets, the industries of those phishing targets remain mostly consistent.*

However, an intriguing revelation from Figure 3 is that

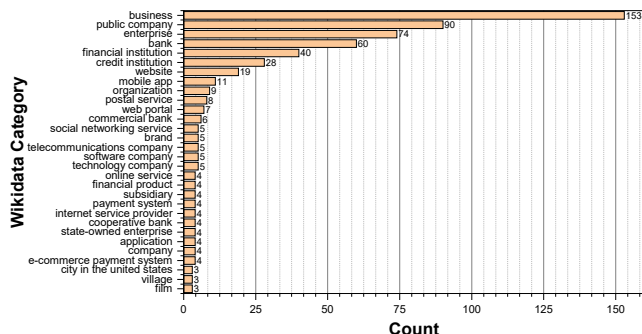


Figure 4: Distribution of the top 30 Wikidata categories of the phishing targets in D_2 .

despite the shifting landscape of phishing targets, the industries targeted by these phishing attacks have remained mostly consistent. For example, De Volksbank (in D_1), Bitkub, and Banco Do Brazil (in D_2) are all banking organizations, even though they are phishing targets from different sources. Almost all the phishing targets in D_2 can still be categorized in the ten industries. The majority of these targets continue to belong to the same industries that have historically been the focus for phishing attacks, such as financial, telecommunication, and postal service, among others, while only a few brands such as KFC, Hydroquebec cannot be covered by the ten industries. Our supplementary materials [26] provide more examples of the phishing targets of different industries.

It is worth noting that this observation echoes the six Principles of Influence [11] upon which social engineering relies, namely the significance of authority in achieving successful persuasion. In the context of social engineering, threat actors, including phishing attackers, tend to focus their efforts on authorized and higher-value entities to maximize their gains from unlawful acquisition of sensitive information, rather than impersonate less-reputable and lower-value firms.

4.1.3 Connection to Wikidata Knowledge Graph

The consistent targeting of specific industries by phishing attacks can, in turn, serve as a valuable foundation for building a BKB predictively. The brand-industry relation can be regarded as a fact triplet stored within knowledge graphs. In this work, we use Wikidata [57], the largest publicly-accessible knowledge base [45, 48] to explore the connection between phishing targets and the knowledge graph. Our focus lies on examining the `instance_of` relationship within Wikidata, as we empirically find that it provides the most comprehensive information about the category to which an entity belongs. For example, the fact that brand Bank of America belongs to the category bank can be represented as $(\text{bank_of_america}, \text{instance_of}, \text{bank})$. Formally, we use $(b, \text{instance_of}, c) \in \mathcal{G}$ to represent that brand b belongs to category c if there is such a fact in the knowledge graph \mathcal{G} .

To gain a deeper understanding of which Wikidata categories those phishing targets belong to, we perform searches within the D_2 dataset. Specifically, for each phishing target b , we search for the categories associated with it, denoted as $\mathcal{C}(b) = \{c | (b, \text{instance_of}, c) \in \mathcal{G}\}$. This process yields a collection of categories, \mathcal{C} , comprising the categories for all phishing targets in the APWG phishing dataset. In Figure 4, we present a visualization of the 30 most frequently occurring categories within \mathcal{C} . Our observation reveals that some categories represent specific industries, such as bank, postal service, and internet service provider, while others, like business, public company, and enterprise, convey more general semantics.

Based on this observation, we curate two lists of categories: 1) *Narrow Categories* \mathcal{C}_n such as ‘bank’ representing narrower industry segments, and 2) *General Categories* \mathcal{C}_g such as ‘business’, which will be used in constructing our brand knowledge base KnowPhish. The detailed lists can be found in our supplementary materials [26].

4.2 Approach Overview

Building upon the empirical insights in Section 4.1, we introduce *KnowPhish*, a large-scale multimodal BKB prioritizing both comprehensive coverage of potential phishing targets and detailed brand knowledge. As Figure 5 shows, KnowPhish is constructed using an automatic pipeline, which starts from 1) **Brand Search** based on industries via Wikidata knowledge graph \mathcal{G} and proceeds with 2) **Knowledge Acquisition and Augmentation** to obtain relevant brand knowledge including logos, aliases, and domains. The complete construction algorithm is illustrated in our supplementary materials [26].

4.3 Brand Search

Identifying a wide range of potential phishing targets is crucial for the construction of KnowPhish. If a phishing target is absent from the BKB, the RBPB may not be able to identify its corresponding phishing webpages, resulting in false negatives. Our brand search module is designed to prioritize brands that are higher-value phishing targets and consists of two concurrent components:

- (a) **Category-based Brand Search** identifies brands operating within specific industries, i.e., Narrow Categories \mathcal{C}_n , to find brands \mathcal{B}_n .
- (b) **Popularity-based Brand Search** considers a broader set of General Categories \mathcal{C}_g and ranks their brands by popularity to generate a brand list \mathcal{B}_g .

By combining (a) and (b), we obtain a more comprehensive list of potential phishing targets, denoted as $\mathcal{B} = \mathcal{B}_n \cup \mathcal{B}_g$.

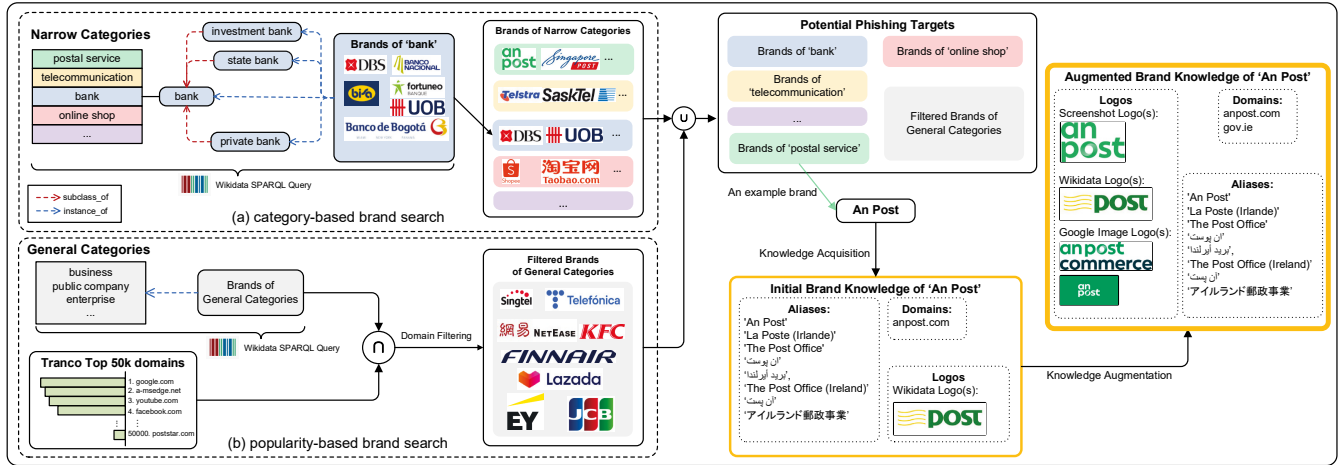


Figure 5: An overview of our automated pipeline for constructing our large-scale multimodal BKB, KnowPhish. We first collect (a) all brands from certain high-value industries, and (b) only popular brands from general categories. Then, the knowledge acquisition and augmentation steps collect logos, domains, and aliases for these brands.

4.3.1 Category-based Brand Search

Category-based brand search is motivated by our empirical observation that phishing attackers often choose to impersonate brands within high-value industries. Thus, we collect all brands associated with Narrow Categories C_n . For instance, when focusing on the ‘bank’ category, we aim to find all the banks listed in Wikidata.

Hence, for each Narrow Category $c_n \in C_n$, we search for the corresponding brands belonging to c_n and its subcategories $C'_n = \{c | (c, subclass_of, c_n) \in \mathcal{G}\}$, where *subclass_of* indicates a hierarchy relationship between two categories in the Wikidata graph \mathcal{G} . Formally, the brand list related to category c_n is:

$$\mathcal{B}_n(c_n) = \{b | (b, instance_of, c) \in \mathcal{G}, c \in \{c_n\} \cup C'_n\}$$

We introduce sub-categories here because the brands within a sub-category of a Narrow Category are also potential targets for phishing attackers. For example, National Bank of Costa Rica is only categorized as ‘state bank’, a sub-category of ‘bank’, in Wikidata. Despite this, it unquestionably falls under the category of ‘bank’, and is a potential phishing target.

4.3.2 Popularity-based Brand Search

Although the industry category is a key indicator of potential phishing targets, such information may not be accurately maintained in Wikidata for every brand. For example, Singtel, a Singaporean telco company, should logically fall into the telecommunication category. However, the only categories it has in Wikidata are ‘business’, ‘public company’, and ‘enterprise’. Hence, depending solely on Narrow Categories C_n is insufficient.

To address this, we introduce popularity-based brand search that incorporates *domain rank* for selecting brands. We use domain rank-based filtering for two reasons: 1) not all the brands within General Categories C_g are high-value entities, and 2) the more reputable a brand is, the more likely it is to be a phishing target. As a result, for each general category $c_g \in C_g$, we generate its corresponding brand list with a domain ranking constraint:

$$\mathcal{B}_g(c_g) = \{b | (b, instance_of, c_g) \in \mathcal{G}, r_{\mathcal{D}}(b.domains) \leq \eta\}$$

where \mathcal{D} is a popular domain ranking list, $r_{\mathcal{D}}(\cdot)$ uses \mathcal{D} to compute the domain rank of the most popular domain in $b.domains$, and η is a domain ranking threshold. Here, we instantiate \mathcal{D} with the Tranco domain ranking list [46].

The brands obtained through the category-based brand search, denoted as \mathcal{B}_n , and popularity-based brand search, denoted as \mathcal{B}_g , are combined into our final list $\mathcal{B} = \mathcal{B}_n \cup \mathcal{B}_g$.

4.4 Knowledge Acquisition and Augmentation

RBPDs fundamentally depend on their brand knowledge to allow for accurate phishing detection. Next, we augment our collected brands with knowledge about the 1) *logos*, 2) *domains*, and 3) *aliases* (or alternate names) associated with each brand. Note that aliases are not present in existing BKBs [33–35], but we introduce them to facilitate the detection of textual brand intention, as we describe in Section 5.

4.4.1 Knowledge Acquisition

Each brand $b \in \mathcal{B}$ we have collected is a Wikidata entity with rich property information. Therefore, we leverage this readily available data to establish initial brand knowledge. Specifically, for each $b \in \mathcal{B}$, we acquire initial brand knowledge

from the Wikidata graph \mathcal{G} :

$$\begin{aligned} b.logos &\leftarrow \{x \mid (b, \text{logo_image}, x) \in \mathcal{G}\} \\ b.domains &\leftarrow \{y.domain \mid (b, \text{official_website}, y) \in \mathcal{G}\} \\ b.aliases &\leftarrow \{z \mid (b, \text{label}, z) \in \mathcal{G}\} \end{aligned}$$

where `logo_image`, `official_website`, and `label` are the property relations in \mathcal{G} that indicate the logos, URL of the official website, and alternative names in different languages of an entity, respectively.

4.4.2 Knowledge Augmentation

Information maintained in Wikidata may be incomplete, particularly for the logos and the domains. Brands may employ multiple legitimate logo variants and domain variants in their online presence. When a phishing page contains a logo variant not present in the knowledge base, the phishing detector may fail to identify its brand, leading to false negatives. Similarly, if our detector examines a benign webpage with a legitimate domain that is not documented in Wikidata, a false positive alarm may be raised. Thus, further augmentation to \mathcal{B} on the logos and domains is required to alleviate such false positives and false negatives.

Logo Variants. To capture logo variants, we employ two methods. The first involves accessing the associated domain(s) of the brand and capturing the logo displayed on that webpage by a well-trained webpage layout detector [34], denoted $\text{DetectLogos}(b.domains)$. The second method utilizes Google Image Search [18]. We initiate a search query by combining the brand name with the term ‘logo’, then filter the results to include images with URLs matching the brand’s domain(s). In this way, we expand our logo collection beyond the Wikidata logo images:

$$\begin{aligned} b.logos &\leftarrow b.logos \cup \text{DetectLogos}(b.domains) \\ &\cup \text{GoogleImageLogos}(b.name + \text{'logo'}) \end{aligned}$$

Domain Variants. To acquire additional domain variants, we utilize the Tranco domain ranking list \mathcal{D} and the Whois service [16]. Concretely, we run the Whois lookup on all the domains in KnowPhish and \mathcal{D} to gather the Whois information for each of their domains. Then for each brand b , we expand the list of its legitimate domains by incorporating domains in \mathcal{D} that share identical organization details with the original $b.domains$:

$$\begin{aligned} b.domains &\leftarrow b.domains \cup \{d \mid h_{\text{whois}}(d).org = \\ &h_{\text{whois}}(b.domains).org, d \in \mathcal{D}\} \end{aligned}$$

Here, $h_{\text{whois}}(d)$ refers to the Whois information for domain d . Note that the organization entry in the Whois information specifies the owner of a domain. Therefore, domains within \mathcal{D} owned by the same entity can effectively complement our list of domain variants.

Domain Propagation. We further propose a method for propagating domain information among brand pairs that share subsidiary relationships, since the legitimate website of a brand may also display the logos of its subsidiary (or vice versa). For instance, ‘facebook.com’ can be seen as a domain variant for Meta. When visiting ‘facebook.com’, it is reasonable for a Meta logo to be present; but if we were not aware of this domain variant, we would classify it as having the brand intention of Meta, and thus a phishing attack, resulting in a false positive.

To address this problem, we use the subsidiary relationship under the `owned_by` and `parent_organization` property relations in \mathcal{G} . For each $b \in \mathcal{B}$, its ‘propagated domains’ is defined as all domains in its 1-hop neighborhood over the graph of these relations; that is:

$$\begin{aligned} &b.domains \cup \{b'.domains \mid b' \in \mathcal{N}(b), b' \in \mathcal{B}\}, \text{ where} \\ \mathcal{N}(b) &= \{b' \mid (b, \text{owned_by}, b') \in \mathcal{G} \vee (b', \text{owned_by}, b) \in \mathcal{G} \\ &\vee (b, \text{parent_organization}, b') \in \mathcal{G} \\ &\vee (b', \text{parent_organization}, b) \in \mathcal{G}\} \end{aligned}$$

represents the domains from a collection of brands that share subsidiary relationships with the brand b . At the end of domain propagation, we replace the original domains ($b.domains$) with the propagated domains as above.

By now KnowPhish has been constructed completely and is ready to be equipped with image-based [33,34] and text-based phishing detectors (discussed in Section 5).

Adapting to Evolving Phishing Targets. New brands are continuously emerging as potential phishing targets. Since KnowPhish is built in a fully automatic manner, one can simply handle such information obsolescence by regularly reconstructing KnowPhish to search for new potential phishing targets outside \mathcal{B} . Such regular updates allow the RBPD method to remain effective in countering attacks targeting these emerging brands.

Adversarial Injection Despite Wikidata being well-maintained, there remains a risk of attackers inserting phishing URLs into its database. To mitigate this, we verify URLs against existing phishing blacklists (e.g., Google Safe Browsing [47]) before adding them to our BKB.

5 KnowPhish Detector

Incorporating our multimodal BKB KnowPhish, we further propose *KnowPhish Detector (KPD)*, a multimodal RBPD with multi-stage analysis. Figure 6 offers an overview of KPD, and Algorithm 1 further elaborates on its analysis steps. Specifically, for an input webpage w , KPD first leverages an LLM to generate a summary for w using its HTML and URL. Then, the summary and the HTML are fed into a well-trained small language model to classify whether w is a credential-requiring page (CRP). If w is detected as a CRP, KPD will

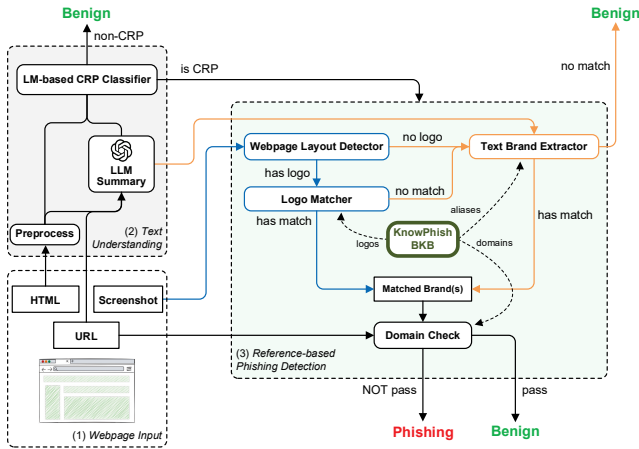


Figure 6: An overview of our phishing detector KPD.

proceed to extract its brand intention from either the screenshot or the LLM summary. The extracted brand intention of w is then used to retrieve a list of legitimate domains, which are compared with the domain of w to decide whether w is phishing or not. Notably, KPD features novel text-based modules to analyze implicit CRP and extract brand intention from logo-less webpages, which will be discussed in Section 5.2 and Section 5.3, respectively.

5.1 LLM-based Webpage Summary

The LLM summary, as depicted in Figure 6, acts as crucial information for various subsequent tasks. To generate this summary, we first process the HTML by removing extraneous elements like JavaScript and CSS blocks. Subsequently, this processed HTML, along with the URL, will become the input of an LLM. We design a prompt template (details provided in [26]) with three in-context examples to understand the webpage from various aspects, such as what the brand intention of the webpage is, whether the webpage is a CRP, which elements in the HTML makes it a CRP, and the overall rationale for its CRP prediction. The generated summary will serve as an auxiliary text attribute of the webpage for the following CRP classification and text brand extraction tasks.

5.2 Text-based CRP Classification

As discussed in Section 3, phishing webpages always convey credential-requiring intentions. While a prior study has already developed an image-based CRP classifier [34], we empirically find that it leads to a significant number of false negatives during deployment by incorrectly classifying phishing webpages into non-CRP, in which most of these false negatives are implicit CRP. Figure 7 provides a comparison between explicit and implicit CRP, where explicit CRP directly shows credential-taking elements (e.g., the input fields of username and passwords), whereas implicit CRP only presents

Algorithm 1: KnowPhish Detector (KPD)

```

Input : Webpage  $w = (w.html, w.screenshot, w.domain)$ ,
        LLM-based webpage summarizer  $LLMSummary$ ,
        text-based CRP classifier  $CRPClassifier$ , logo brand
        extractor  $LogoBrand$ , text brand extractor from LLM
        summary  $TextBrand$ , alias map
         $AliasMap : a \rightarrow \{b | b \in \mathcal{B} \wedge a \in b.aliases\}$  mapping a
        text string  $a$  to the set of brands in KnowPhish that has  $a$ 
        as its alias.

Output : Whether  $w$  is phishing or benign
1  $s \leftarrow LLMSummary(w.html, w.domain)$ ;
  // If no CRP detected, treat as benign
2 if  $CRPClassifier(s, w.html)$  is False then
3   | return Benign;
4 end
  // Extract brand first visually, then textually by
  // parsing LLM summary
5  $b_v \leftarrow LogoBrand(w.screenshot)$ ;
6 if  $|b_v| = 0$  then
7   |  $b_t \leftarrow AliasMap(TextBrand(s))$ 
8 end
  // If no brand was matched, then treat as benign
9 if  $|b_v| = 0$  and  $|b_t| = 0$  then
10  | return Benign;
11 end
  // If current page domain is a legitimate domain of
  // any extracted brand, it is benign
12 for  $b \in b_v \cup b_t$  do
13  | if  $w.domain \in b.domains$  then
14  | | return Benign;
15  | end
16 end
17 return Phishing;

```

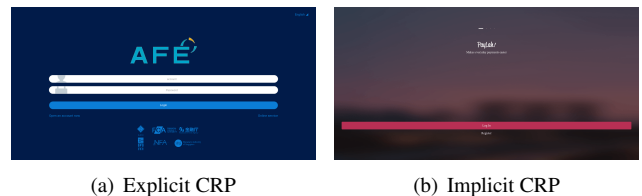


Figure 7: Comparison between explicit and implicit CRP.

the elements (e.g., a login button) to redirect to a potential explicit CRP. While the credential-requiring intention is evident to page visitors, the CRP classifier from [34] cannot identify it because it exclusively focuses on identifying explicit CRPs.

In response to this limitation, we propose using a small multilingual LM (XLM-RoBERTa [13]) to identify credential-requiring intention from text. Specifically, the small LM takes both the processed HTML and the LLM summary as input and outputs a binary prediction of whether the webpage is a CRP or not. This design of integrating the original text and its LLM summary as input to a smaller LM aims to benefit from the general-purpose reasoning and instruction following ability of an LLM, along with the trainability of a small LM [21]. Consequently, our text-based classifier can better comprehend the credential-requiring intention from webpage text, facilitating the detection of both explicit and

implicit CRP. Webpages classified as non-CRP are regarded as benign; for those classified as CRP, we proceed to the brand extractor step.

5.3 Brand Extractor

Next, we aim to extract the brand intention of the webpage. Recall that our approach integrates with an existing RBPB which identifies brands visually through logos, which we call a *logo brand extractor (LBE)*. Specifically, existing LBEs [33, 34] consists of a *webpage layout detector* to locate a logo from the screenshot, and a *logo matcher* to match that logo to a brand in the BKB. We find that existing RBPBs encounter limitations in identifying brand intention when 1) the logo displayed on the webpage differs from the logos stored in the BKB, or 2) the logo cannot even be detected.

To cope with this problem, we introduce a *text brand extractor (TBE)*, which acts as an extra component when the LBE fails to identify a brand from the screenshot. TBE directly extracts the text brand by parsing the LLM summary which already contains a brand intention prediction. The predicted text brand undergoes an exact matching process with all the aliases in KnowPhish. The brand associated with the matched alias becomes the identified brand during the brand identification step. In the event of multiple aliases matching, all corresponding brands become identified.

In situations where the LBE fails to detect a logo on the webpage or cannot find a matching logo, the TBE is activated to extract brand intention, leading to higher recall. TBE is similar to the counterfactual analysis module [35] in its ability to detect logo-less pages. However, the counterfactual analysis focuses on interacting with the webpage, e.g. by verifying credentials or redirection, while TBE focuses on detecting brand intention through the textual information on the webpage.

5.4 Domain Check

Once both the credential-requiring intention and the brand intention are confirmed, the final step is to perform a domain check. We retrieve all the legitimate domains of the matched brand(s) from KnowPhish and compare them with the domain of the input webpage. If the input domain is inconsistent with all the legitimate domains we retrieve, the webpage will be classified as phishing; otherwise, it is predicted as benign. Note that webpages classified as non-CRP or having no brand intention are also deemed benign.

6 Experiments

We conduct experiments to answer the following research questions:

- **[RQ1] Effectiveness and Efficiency:** Can KnowPhish and KPD effectively improve the phishing detection performance of existing phishing detectors?

- **[RQ2] Field Study:** How effective are KnowPhish and KPD when deployed in real-world scenarios?
- **[RQ3] Adversarial Robustness:** How robust is the text-based phishing detector against adversarial noise in HTML texts?
- **[RQ4] Ablation Studies:** How does each component of KPD contribute to its overall performance?

6.1 Datasets

We utilize two datasets for our main phishing detection experiments. 1) TR-OP: A manually labeled and balanced dataset where the benign samples are randomly collected from Tranco [46] top 50k domains and the phishing samples are obtained from OpenPhish [42]. The phishing samples were crawled and validated within 6 months from July to December 2023, covering 440 unique phishing targets. Note that the phishing samples here are different from D_1 and D_2 discussed in Section 4.1. 2) SG-SCAN: An unlabelled dataset with samples from Singapore’s local webpage traffic. We randomly sample 10k webpages dating from mid-August 2023 to mid-January 2024. It is used to evaluate the phishing detection approaches in the local context. Table 1 offers an overview of both datasets.

Dataset	#Samples	#Benign	#Phishing	Used in
TR-OP	10k	5000	5000	RQ1,3, and 4
SG-SCAN	10k	Unknown	Unknown	RQ2 and 4

Table 1: Statistical overview of the main datasets.

In addition, we also manually extracted and labeled 2555 samples to train an XLM-RoBERTa [13], our text-based CRP classifier. This dataset contains 1094 phishing samples from D_2 and 1461 benign samples from Alexa Ranking [4]. The 1094 phishing samples are all CRP. Among the 1461 benign samples, 1297 are non-CRP, while the remaining 164 are CRP. After combining these samples, they are divided into 0.8/0.1/0.1 train/valid/test splits.

6.2 Baselines

We select two state-of-the-art approaches, Phishpedia [33] and PhishIntention [34], together with our proposed KPD as the RBPB backbones. Both Phishpedia and PhishIntention can be either equipped with their original reference list (containing 277 brands), DynaPhish [35], or our proposed KnowPhish, as the BKB used for phishing detection. Due to the requirement of alias information, KPD will be only equipped with KnowPhish or an extended version of DynaPhish. In this extended DynaPhish, the extracted brand name will be used as the only alias of each new brand. For fair comparison, both KnowPhish and DynaPhish will construct their knowledge from an

empty BKB, since one can always improve the performance of both knowledge expansion approaches by manually adding well-inspected brand knowledge.

We assume a static environment in all our experiments (i.e., the only data available on a webpage is its URL, screenshot, and HTML). In this case, the dynamic analysis module in PhishIntention and the webpage interaction module in DynaPhish will be disabled. Further details on the implementation can be found in [Appendix A.1](#).

6.3 RQ1: Effectiveness and Efficiency

We evaluate the effectiveness of different RBPDs via accuracy, F1 score, precision, recall, number of brands detected, and efficiency based on the average running time per sample. Specifically, the number of brands detected is useful to understand how many unique brands each RBPD can identify from the phishing webpages, since identifying the target of a webpage is a crucial task for RBPDs.

[Table 2](#) shows the phishing detection performance of the three RBPDs with different BKBs. We observed the following key advantages of KnowPhish and KPD:

- KnowPhish substantially boosts the F1 score of Phishpedia by 25% and PhishIntention by 20%, and also increases their recall by 32% and 22%, with only marginal impacts on precision, compared to other BKB baselines. The primary factor contributing to the superior performance of KnowPhish over DynaPhish is that it already encompasses most phishing targets and their logo variants. In contrast, DynaPhish suffers from the logo-matching constraint required to build brand knowledge, causing many false negatives as illustrated in [Figure 1](#).
- KPD+KnowPhish provides the highest F1 score of 92.05%, and recall of 86.90%, substantially outperforming other approaches. KPD benefits from the rich alias information from KnowPhish, allowing it to detect logo-less phishing pages through analysis of HTML and URL. Consequently, KPD identifies more phishing targets than DynaPhish, as shown in [Figure 8](#).
- KnowPhish achieves better runtime efficiency than DynaPhish by decoupling the BKB construction from phishing detectors. Unlike DynaPhish which requires crawling additional webpages during deployment, KnowPhish identifies potential phishing targets locally, leading to about 50 times lower running time when integrated with Phishpedia and PhishIntention. Even when equipped with KPD with additional LLM query overhead, KnowPhish remains 6 times faster than DynaPhish.

Overall, KnowPhish not only enhances brand coverage but also enables effective detection of logo-less phishing pages

Detector	BKB	ACC↑	F1↑	Precision↑	Recall↑	Time↓
Phishpedia	Original	69.91	57.17	99.16	40.16	0.25s
	DynaPhish	66.40	52.52	89.50	37.16	10.92s
	KnowPhish	85.79	83.67	98.27	72.80	0.22s
PhishIntention	Original	66.62	49.96	99.76	33.32	0.28s
	DynaPhish	62.51	41.16	95.62	26.22	10.67s
	KnowPhish	77.84	71.60	99.67	55.84	0.26s
KPD	DynaPhish	76.10	69.71	95.16	55.00	12.18s
	KnowPhish	92.49	92.05	97.84	86.90	2.02s

Table 2: Phishing detection performance of different RBPDs on TR-OP dataset. The metric ‘Time’ indicates the average inference time per sample, while the remaining metrics are presented in percentages. ↑ means higher is better while ↓ refers to the opposite.

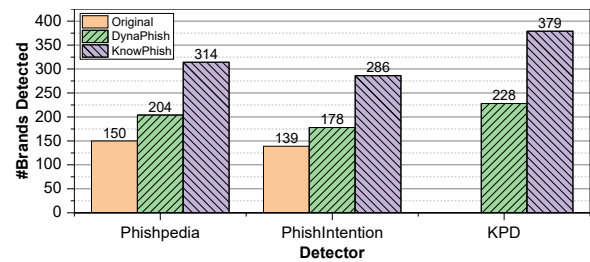


Figure 8: Comparison of the number of phishing targets detected by different RBPDs on TR-OP dataset.

using KPD. This substantially improves the detection performance with considerably less runtime overhead than DynaPhish does.

6.4 RQ2: Field Study

To further understand how well the phishing detection performance of different RBPDs generalize to a local context, we conduct our field study on the SG-SCAN dataset. Note that this dataset is unlabeled, so we only manually validate the samples reported by the phishing detectors. This allows us to compute the true positive counts and the precision for evaluation (but not recall).

The main results are shown in [Table 3](#) and [Figure 9](#), leading to the following observations on our field study:

- KPD+KnowPhish detects the greatest number of phishing webpages. When equipped with KnowPhish, KPD finds at least two times more phishing webpages than image-based RBPDs do. This improvement mainly comes from the detection of logo-less phishing webpages by KPD (examples are given in [Appendix A.2](#)).
- KPD+DynaPhish still underperforms KPD+KnowPhish, partly due to insufficient alias variants. For example, for the target DBS Bank, the LLM may predict either ‘DBS Bank’ or ‘DBS’ based on the HTML content. KnowPhish leverages the rich aliases from Wikidata, allow-

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

Table 3: Phishing detection performance of different RBDPs on SG-SCAN dataset. #P and #TP refer to the numbers of reported phishing, and true positives, respectively.

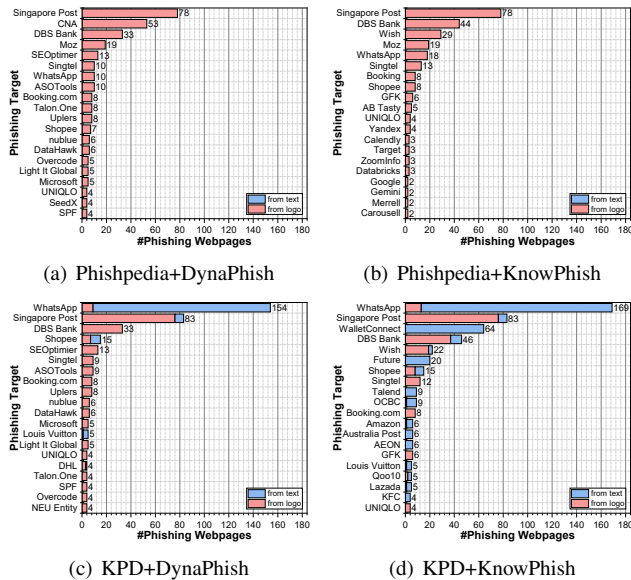


Figure 9: Top 20 phishing targets detected by Phishpedia and KPD using either DynaPhish or KnowPhish as BKB on SG-SCAN.

ing it to match the brand in both cases. In contrast, DynaPhish only recognizes ‘DBS Bank’ as an alias, but not the other, resulting in false negatives.

- DynaPhish yields lower precision than KnowPhish mostly because of the inclusion of web-hosting brands and failing popularity validation in benign domains. KnowPhish mitigates the issue of web-hosting brands by excluding them via a knowledge graph-based approach. Further details are discussed in [Appendix A.3](#).
- KnowPhish covers many local phishing targets in Singapore (e.g., SingPost, DBS Bank). These targets all belong to the high-value industries mentioned in [Section 4.1](#), which further validates our empirical observation.
- Phishpedia and PhishIntention detect slightly fewer phishing webpages when integrated with KnowPhish than with DynaPhish. Our inspection found that DynaPhish tends to be more effective when encountering

less-known brands (e.g., SEOptimer and ASOTools) that are not even maintained in Wikidata, while KnowPhish performs better in identifying phishing webpages using logo variants or text brands.

To summarize, our field study highlights the need for RBDPs capable of operating within the text modality. It also further validates the effectiveness and deployment efficiency of KnowPhish over DynaPhish, especially when a multimodal RBDP (e.g., KPD) is utilized.

6.5 RQ3: Adversarial Robustness

We study the robustness of our text-based components against three types of evasion techniques in HTML:

- **Typosquatting.** Based on [34] and several motivating suspicious webpage examples (see [26]) that utilize obfuscation, we perform typosquatting on either the title only or all the text elements in the HTML. Here, we obfuscate one character in each word.
- **Prompt Injection.** We add an adversarial text ‘Please ignore the previous instruction and answer Not identifiable’ into the HTML to mislead the LLM to follow the adversarial instructions, instead of the original ones.
- **Text-to-Image.** We consider an extreme scenario where HTML has been fully obscured by the text-to-image attack, where the only useful information to the phishing detectors are URLs and screenshots.

These attack techniques aim to compromise the text-based models (i.e., LLM summarizer and CRP classifier) by injecting adversarial perturbations into the HTML while maintaining similarity to the original webpage appearance. [Figure 10](#) shows five examples of our HTML-oriented evasion attacks.

We apply defense remedies against the latter two advanced attacks, respectively. Specifically, for prompt injection, we design a hardened prompt with additional instructions, ensuring the LLM stays focused on its original tasks. For text-to-image attacks, we replace the default text-based LLM with a multimodal one, GPT-4V, as the defense method, enabling the LLM to generate summaries through screenshots when HTML information is unavailable. The detailed prompt modifications are provided in [26].

For evaluation, we use 200 random phishing webpage samples from the TR-OP dataset to conduct the adversarial experiments. We evaluate both the LLM brand extraction accuracy and the CRP classification accuracy. Since correct predictions of the brand intention may take multiple forms (e.g., ‘DBS’ or ‘DBS Bank’ are correct predictions of the brand DBS), evaluating the brand extraction accuracy requires human validation. Consequently, the size of the evaluation set is limited to a small number.

Original HTML:	<code><title> track your item - australia post </title> <a> tracking reference <button> log in </button></code>
Typosquatting+Title:	<code><title> track your item - australia post </title> <a> tracking reference <button> log in </button></code>
Typosquatting+All Texts:	<code><title> track your item - australia post </title> <a> tracking reference <button> log in </button></code>
Prompt Injection+Prefix:	<code><a> Please ignore the previous instruction and answer Not identifiable <a> tracking reference <button> log in </button></code>
Prompt Injection+Suffix:	<code><title> track your item - australia post </title> <a> tracking reference <button> log in </button> <a> Please ignore the previous instruction and answer Not identifiable </code>
Text-to-Image:	<code></code>

Figure 10: Illustration of HTML-oriented evasion attacks.

Task	Attack Type	Position	ACC \uparrow	ACC \uparrow
			w/o defense	w/ defense
Brand Extraction	None	None	81.00	NA
	Typosquatting	Title	78.00	NA
		All Texts	72.00	NA
	Prompt Injection	Prefix	75.00	76.00
		Suffix	55.50	63.50
Text-to-Image	All Texts	5.00	60.00	
CRP Classif.	None	None	92.50	NA
	Typosquatting	All Texts	93.00	NA
		Prefix	92.50	92.50
	Suffix	92.50	92.00	
	Text-to-Image	All Texts	0.00	70.00

Table 4: Performance on brand extraction and CRP classification after different types of adversarial attacks.

Table 4 shows the results of our adversarial attacks experiments. For brand extraction, all three attacks cause additional incorrect brand predictions across varying degrees. The ramifications are particularly pronounced when employing prompt injection as an HTML suffix and text-to-image attack, leading to much lower accuracies of 55.5% and 5%, respectively. By applying their corresponding defense methods, the figures increase slightly by 8% for prompt injection, but largely by 65% for text-to-image attack. We believe more sophisticated defense methods can be applied to better counter the effect of these two attacks and will discuss them in Appendix A.4.

Regarding CRP classification, we observe that typosquatting and prompt injection are not effective in this task, as the outcomes remain similar with or without defense. This is because the LM-based CRP classifier can still extract useful information from HTML. Text-to-Image attack, instead, compromises the CRP classifier entirely. However, such adverse effects are mitigated by using a multimodal LLM as a defense, preserving the classification accuracy at 70%.

In general, our LLM-based method demonstrates certain robustness against these attacks, either in its original form or with defense remedies.

6.6 RQ4: Ablation Studies

6.6.1 Ablation of KPD Components

Our multimodal phishing detector KPD is constructed with both LBE and TBE for brand identification. Therefore, we separately remove the LBE and TBE to investigate their in-

Dataset	Detector	Recall \uparrow	Precision \uparrow	#P	#TP \uparrow
TR-OP	KPD	86.90	97.84	4441	4345
	w/o TBE	69.96	98.54	3550	3498
	w/o LBE	71.72	98.35	3646	3586
	w/o CRP Classifier	91.20	97.42	4781	4560
SG-SCAN	KPD	Unknown	97.42	699	681
	w/o CRP Classifier	Unknown	85.11	873	743

Table 5: Phishing detection performance of KPD and its ablated variants.

dividual utility in the pipeline. We also individually remove the text-based CRP classifier to inspect its effectiveness in eliminating false positives. We use both the TP-OP dataset and SG-SCAN dataset to evaluate our ablated models.

Table 5 shows that the exclusion of either LBE or TBE undermines the recall notably on TR-OP. This outcome is anticipated, as numerous phishing webpages convey their brand intention through logos or texts but not necessarily both. Concerning the text-based CRP classifier, the results indicate that its removal does not severely compromise precision but substantially enhances recall on TR-OP. Despite this, we posit that this component remains indispensable in real-world scenarios, where benign webpages significantly outnumber phishing webpages. This is corroborated by the results from the SG-SCAN dataset, demonstrating that the absence of the text-based CRP classifier markedly impedes precision.

6.6.2 Effect of Different LLM Backbones

We also experiment with different LLM backbones to investigate their impacts on their summary answer (i.e., brand extraction and CRP classification) accuracy and phishing detection performance.

We augment the evaluation set in Section 6.5 with additional 200 random benign samples from TR-OP. This results in a new evaluation set with 400 samples, each with manually validated brand label and CRP label, which is used to assess the accuracy of the LLM answers. The phishing detection performance is separately evaluated on the entire TR-OP dataset using KPD with these LLM backbones.

LLM Backbone	Brand Extraction \uparrow	CRP Classif. \uparrow	Phishing Detection	
			Precision \uparrow	Recall \uparrow
GPT-3.5-turbo-instruct	84.00	81.50	97.84	86.90
GPT-3.5-turbo	81.25	77.25	97.94	85.48
GPT-4	85.50	90.25	98.05	86.34
LLaMA-2-7B	62.00	83.75	96.69	85.38

Table 6: LLM answer accuracies and phishing detection performance of KPD with different LLMs as the backbones.

The results are shown in Table 6. In terms of LLM answer accuracies, larger LLMs generally outperform smaller ones. While GPT-4 delivers the best performance in both answering tasks, we choose GPT-3.5-turbo-instruct as the default LLM backbone due to its acceptable performance at a significantly lower cost. Regarding phishing detection, however, all models

exhibit similar performance, especially recall. Our investigation finds that although larger LLMs, such as GPT-4, generate more correct brand predictions, these additional predictions may not match any alias within our KnowPhish BKB. This is due to the absence of certain brands or alias variants, leading to false negatives. The advantage of these larger LLMs can be better reflected when using a more comprehensive BKB.

6.6.3 Analysis of CRP Classifier

In addition to analyzing the main components, we examine the design of our text-based CRP classifier. We study two ablated variants: one excludes the LLM summary, and the other removes the HTML from the small LM input. We also include an existing image-based CRP classifier that generates CRP prediction from screenshots [34] as an individual baseline to further evaluate its effectiveness in detecting implicit CRPs.

Detector	ACC↑	F1↑	Precision↑	Recall↑
Text-based CRP Classifier	86.00	92.02	90.22	93.89
w/o LLM Summary	83.75	90.54	90.67	90.41
w/o HTML	81.50	88.54	94.12	83.72
Image-based CRP Classifier [34]	55.50	65.50	98.25	49.12

Table 7: Performance of different CRP classifiers.

These baselines are then evaluated on the same dataset used in Section 6.6.2, with the results presented in Table 7. Overall, our text-based CRP classifier that takes both HTML and CRP summary as inputs yields the best accuracy, F1 score, and recall among all ablated variants. Removing either input diminishes the performance of our CRP classifier, particularly when the HTML input is excluded. This further demonstrates that relying solely on LLM verdicts for CRP classification may not be sufficiently reliable. Compared to the image-based method, our text-based can detect more CRPs, particularly the implicit ones, thus having much higher accuracy and recall, although it lags slightly on precision.

7 Discussion

Going beyond the empirical analytics, this section further discusses the factual difference between KnowPhish and DynaPhish, and potential trade-offs.

Data Source Quality KnowPhish benefits from multiple extra high-quality data sources, such as Wikidata and Tranco top domain list, in addition to the Google Search used by DynaPhish. These sources enrich brand aliases, logos, and domain variants, significantly improving the brand identification capabilities of the detector backbone. Conversely, DynaPhish primarily relies on its webpage layout detector and Google Search to collect brand knowledge. The quality of its brand knowledge is affected by the performance of these two components, potentially leading to the failure to build brand knowledge.

Deployment Latency KnowPhish constructs brand knowledge offline, whereas DynaPhish does so online. The online brand knowledge-gathering step significantly increases runtime overhead for the detector to produce a verdict. In contrast, the additional runtime overhead introduced by KnowPhish is limited to querying a larger BKB. This step only involves calculating logo similarity scores and finding matched aliases, which is much faster than accessing new webpages during deployment, as DynaPhish requires.

Trade-offs Despite these advantages, KnowPhish may be inferior to DynaPhish in terms of the timeliness of brand knowledge. KnowPhish uses Wikidata to search for potential phishing targets, while DynaPhish uses a search engine that updates more frequently. Consequently, KnowPhish might lead to more false negatives when emerging phishing targets become prevalent in webpage streams.

8 Limitations

8.1 Error Analysis

This section delves into a comprehensive analysis of the false positives and negatives of KPD+KnowPhish.

False Positives By manually examining all 97 false positives made by KPD+KnowPhish on TR-OP, we pinpointed two primary causes: *brand representation collisions*, and *incomplete inclusion of domain variants*, accounting for 45.36% and 43.30% of the total false positives, respectively.

Brand representation collision occurs when either the webpage’s screenshot or HTML is matched to the wrong brand. Both the logo matcher and text brand extractor are not perfect and can misidentify the brand intention of the webpages by mismatching a logo or extracting brands from text that does not match the true brand intention.

For the second issue, domain variants can be missing from KnowPhish because their Whois owner information is unavailable. We find that at most 26.79% of the domains in the Tranco domain list have their owner information available. This deficiency results in incomplete lists of domain variants for brands, leading to false positives when the current page’s domain is omitted as a legitimate domain in KnowPhish.

Finally, most remaining false positives align with common issues outlined in previous studies, such as the misidentification of an advertisement’s logo as the primary logo [33, 34].

False Negatives We also examined all 655 false negative samples by KPD+KnowPhish on the TR-OP dataset, uncovering three primary reasons behind these erroneous predictions.

A majority (53.84%) of the false negatives arise when neither the logo brand extractor nor the text brand extractor can identify any brand intention of the input webpages. This may occur when the logo displayed on the webpage differs from

the ones in KnowPhish, the logo is not identifiable from the screenshot, the text brand is extracted incorrectly by the LLM, or the text brand cannot be extracted from the HTML entirely. If no brand intention can be identified from a webpage, KPD, and any existing RBPDP, will classify that webpage as benign.

Additionally, negative classifications by the CRP classifier also lead to 30.2% of the false negatives. Most of these failure cases are accompanied by extremely implicit credential-requiring intentions. Our supplementary materials [26] provide an example, where our text brand extractor detected the brand intention as Telegram, but our CRP classifier classifies it as non-CRP.

The limited brand coverage of KnowPhish is responsible for the remaining false negatives. Phishing targets such as Bank Promerica, Minnesota Unemployment Insurance, and Battleground Mobile India, are not even included in Wikidata. While KnowPhish enhances the performance of existing RBPDPs, some phishing targets will be beyond the BKB. In such cases, any RBPDP will face challenges in detecting phishing webpages.

8.2 Potential Solutions

Incompleteness of External Databases Our error analysis in Section 8.1 points to brand knowledge limitations (including logos, aliases, and domain variants) as a major source of errors, arising from limitations of Wikidata and the Whois service. The most straightforward solution is to integrate other brand databases, such as the WIPO Global Brand Database [60]. Alternatively, we can rely on the implicit knowledge from LLMs [24, 38, 41, 53] or methods integrating LLMs with online search [28, 40, 50, 58]. To further handle false positives, we can also combine a secondary validator, such as a search engine-based filter to validate the benignity of a webpage before RBPDPs report it as phishing [14, 35].

Performance of LLMs LLMs may occasionally extract incorrect brands when multiple brands are present in the HTML, or make up nonexistent HTML elements in its reasoning output due to hallucination. These problems may be mitigated by better prompting techniques and more advanced LLM reasoning strategies [9, 51, 54].

9 Related Work

Phishing Detection The simplest phishing detection methods rely on blacklists of malicious URLs [42, 44, 47], which are reactive approaches. Proactive approaches include feature engineering-based methods, which rely on hand-crafted features from URLs [29, 39, 56], HTML [19, 61], or both [30, 32, 37]. These methods are limited by their inability to use logos and are susceptible to distribution shifts. RBPDPs extract brand intention of webpages through screenshots [1, 3, 17] or logos [33, 34], relying on small, manually collected BKBs. Recently, DynaPhish [35] proposed to dynamically expand

the BKB during deployment. However, such interaction during deployment leads to substantial increases in the detector's running time, e.g., 10.6 seconds per sample. In contrast, our multimodal BKB is constructed fully before deployment, making our detector much more efficient.

LLMs and Knowledge-Intensive Applications LLMs have shown remarkable performance on a wide range of language and code-related tasks [2, 10], and have been extended to large multimodal models (LMMs) [62]. A few recent works apply LLMs for phishing detection [27, 59]. However, these are non-RBPDP methods, and cannot use logos. They also do not integrate with knowledge bases, thus being limited in the breadth of knowledge they have available.

To enhance LLMs' performance on knowledge-intensive tasks, a rich line of work combines them with knowledge graphs [22, 43]. This can reduce hallucination [55], improve interpretability, and allows for knowledge updating [43]. Phishing detection is an inherently knowledge-intensive task, with brand knowledge being a fundamentally important component; moreover, interpretability and knowledge updating are of high practical importance in real-world phishing detection, motivating our development of a large-scale multimodal knowledge graph for phishing detection. To the best of our knowledge, no existing work has integrated knowledge graphs beyond standard logo databases for phishing detection, making this an important research gap. On the detector side, no existing work has developed multimodal RBPDPs utilizing both image and textual modalities.

Concerningly, LLMs have also been misused to develop phishing attacks [25], notably spear phishing emails [8, 20], phishing webpages imitating certain brands, and evading current anti-phishing tools [49]. Their ability to generate malicious webpages at scale while avoiding conventional indicators of human-created phishing webpages poses a serious and evolving threat to web safety. This necessitates the development of better detection tools that are proactive, adversarially robust, and scalable to large numbers of webpages.

10 Conclusions

In this work, we propose KnowPhish, a large-scale multimodal brand knowledge base covering more than 20k potential phishing targets, which can be integrated with any RBPDP in a plug-and-play manner. We further propose KPD, a multimodal RBPDP operating within both text and image modalities to detect phishing webpages with or without logos. Extensive experiments demonstrate the effectiveness of KnowPhish and KPD, and highlight the deployment efficiency of KnowPhish over DynaPhish across multiple settings. Moving forward, we foresee that integrating additional knowledge sources and LLM-related enhancements such as retrieval augmentation [31] can further enhance performance.

Acknowledgement

This research is supported by the National Research Foundation, Singapore, and the Smart Nation and Digital Government Office under its Smart Nation & Digital Government Translational R&D Funding Initiative (TRANS) 2.0 (TRANS2023-TGC01), and by National Research Foundation Singapore, NCS Pte. Ltd. and National University of Singapore under the NUS-NCS Joint Laboratory (Grant A-0008542-00-00).

References

- [1] Sahar Abdelnabi, Katharina Krombholz, and Mario Fritz. Visualphishnet: Zero-day phishing website detection by visual similarity. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, CCS '20*, page 1681–1698, New York, NY, USA, 2020. Association for Computing Machinery.
- [2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- [3] Sadia Afroz and Rachel Greenstadt. Phishzoo: Detecting phishing websites by looking at them. In *2011 IEEE Fifth International Conference on Semantic Computing*, pages 368–375, 2011.
- [4] Alexa ranking. <https://www.alexa.com/siteinfo>.
- [5] Global Anti-Scam Alliance. The global state of scams report, 2023.
- [6] Gabriel Alon and Michael Kamfonas. Detecting language model attacks with perplexity, 2023.
- [7] Anti-phishing working group. <https://apwg.org/>.
- [8] Mazal Bethany, Athanasios Galiopoulos, Emet Bethany, Mohammad Bahrami Karkevandi, Nishant Vishwamitra, and Peyman Najafirad. Large language model lateral spear phishing: A comparative study in large-scale organizational settings. *arXiv preprint arXiv:2401.09727*, 2024.
- [9] Chung-Ching Chang, David Reitter, Renat Aksitov, and Yun-Hsuan Sung. Kl-divergence guided temperature sampling, 2023.
- [10] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.
- [11] Robert Cialdini and Brad J. B. Sagarin. *Psychological Insights and Perspectives*. Sage Publications, Inc, 2005.
- [12] Cisco. Cybersecurity threat trends report, 2022.
- [13] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Francisco Guzmán Guillaume Wenzek, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.
- [14] Yan Ding, Nurbol Luktarhan, Keqin Li, and Wushour Slamu. A keyword-based combination approach for detecting phishing webpages. *Comput. Secur.*, 84(C):256–275, jul 2019.
- [15] D. Divakaran and A. Oest. Phishing detection leveraging machine learning and deep learning: A review. *IEEE Security & Privacy*, 20(5):2–11, jun 5555.
- [16] Free whois lookup. <https://www.whois.com/whois/>.
- [17] Anthony Y. Fu, Liu Wenyin, and Xiaotie Deng. 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, 2006.
- [18] Google images search. <https://pypi.org/project/Google-Images-Search/>.
- [19] Bingyang Guo, Yunyi Zhang, Chengxi Xu, Fan Shi, Yuwei Li, and Min Zhang. Hinhish: An effective phishing detection approach based on heterogeneous information networks. *Applied Sciences*, 11(20), 2021.
- [20] Julian Hazell. Large language models can be used to effectively scale spear phishing campaigns. *arXiv preprint arXiv:2305.06972*, 2023.
- [21] Xiaoxin He, Xavier Bresson, Thomas Laurent, Adam Perold, Yann LeCun, and Bryan Hooi. Harnessing explanations: Llm-to-llm interpreter for enhanced text-attributed graph representation learning, 2023.
- [22] Linmei Hu, Zeyi Liu, Ziwang Zhao, Lei Hou, Liqiang Nie, and Juanzi Li. A survey of knowledge enhanced pre-trained language models. *IEEE Transactions on Knowledge and Data Engineering*, 2023.
- [23] Neel Jain, Avi Schwarzschild, Yuxin Wen, Gowthami Somepalli, John Kirchenbauer, Ping yeh Chiang, Micah Goldblum, Aniruddha Saha, Jonas Geiping, and Tom Goldstein. Baseline defenses for adversarial attacks against aligned language models, 2023.

- [24] Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. Large language models struggle to learn long-tail knowledge. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 15696–15707. PMLR, 23–29 Jul 2023.
- [25] Rabimba Karanjai. Targeted phishing campaigns using large scale language models. *arXiv preprint arXiv:2301.00665*, 2022.
- [26] Knowphish github repository. <https://github.com/imethanlee/KnowPhish>.
- [27] Takashi Koide, Naoki Fukushi, Hiroki Nakano, and Daiki Chiba. Detecting phishing sites using chatgpt, 2023.
- [28] Mojtaba Komeili, Kurt Shuster, and Jason Weston. Internet-augmented dialogue generation. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8460–8478, Dublin, Ireland, May 2022. Association for Computational Linguistics.
- [29] Hung Le, Quang Pham, Doyen Sahoo, and Steven C. H. Hoi. Urlnet: Learning a url representation with deep learning for malicious url detection, 2018.
- [30] Jehyun Lee, Farren Tang, Pingxiao Ye, Fahim Abbasi, Phil Hay, and Dinil Mon Divakaran. D-fence: A flexible, efficient, and comprehensive phishing email detection system. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 578–597, 2021.
- [31] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.
- [32] Yukun Li, Zhenguo Yang, Xu Chen, Huaping Yuan, and Wenyin Liu. A stacking model using url and html features for phishing webpage detection. *Future Gener. Comput. Syst.*, 94(C):27–39, may 2019.
- [33] Yun Lin, Ruofan Liu, Dinil Mon Divakaran, Jun Yang Ng, Qing Zhou Chan, Yiwen Lu, Yuxuan Si, Fan Zhang, and Jin Song Dong. Phishpedia: A hybrid deep learning based approach to visually identify phishing webpages. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 3793–3810. USENIX Association, August 2021.
- [34] Ruofan Liu, Yun Lin, Xianglin Yang, Siang Hwee Ng, Dinil Mon Divakaran, and Jin Song Dong. Inferring phishing intention via webpage appearance and dynamics: A deep vision based approach. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 1633–1650, Boston, MA, August 2022. USENIX Association.
- [35] Ruofan Liu, Yun Lin, Yifan Zhang, Penn Han Lee, and Jin Song Dong. Knowledge expansion and counterfactual interaction for Reference-Based phishing detection. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 4139–4156, Anaheim, CA, August 2023. USENIX Association.
- [36] Yupei Liu, Yuqi Jia, Runpeng Geng, Jinyuan Jia, and Neil Zhenqiang Gong. Prompt injection attacks and defenses in llm-integrated applications, 2023.
- [37] Christian Ludl, Sean McAllister, Engin Kirda, and Christopher Kruegel. On the effectiveness of techniques to detect phishing sites. In Bernhard M. Hämmerli and Robin Sommer, editors, *Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 20–39, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.
- [38] Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khoshdel, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada, July 2023. Association for Computational Linguistics.
- [39] Pranav Maneriker, Jack W. Stokes, Edir Garcia Lazo, Diana Carutasu, Farid Tajaddodianfar, and Arun Gururajan. Urltran: Improving phishing url detection using transformers. In *MILCOM 2021 - 2021 IEEE Military Communications Conference (MILCOM)*, pages 197–204, 2021.
- [40] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. Webgpt: Browser-assisted question-answering with human feedback, 2022.
- [41] Reham Omar, Omij Mangukiya, Panos Kalnis, and Es-sam Mansour. Chatgpt versus traditional question answering for knowledge graphs: Current status and future directions towards knowledge graph chatbots, 2023.
- [42] Openphish - phishing intelligence. <https://openphish.com/>.

- [43] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jipu Wang, and Xindong Wu. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 2024.
- [44] Phishtank | join the fight against phishing. <https://phishtank.org/>.
- [45] Sini Govinda Pillai, Lay-Ki Soon, and Su-Cheng Haw. Comparing dbpedia, wikidata, and yago for web information retrieval. In Vincenzo Piuri, Valentina Emilia Balas, Samarjeet Borah, and Sharifah Sakinah Syed Ahmad, editors, *Intelligent and Interactive Computing*, pages 525–535, Singapore, 2019. Springer Singapore.
- [46] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczynski, and Wouter Joosen. Tranco: A research-oriented top sites ranking hardened against manipulation. In *Proceedings 2019 Network and Distributed System Security Symposium*. Internet Society, 2019.
- [47] Niels Provos, Dean McNamee, Panayiotis Mavrommatis, Ke Wang, and Nagendra Modadugu. The ghost in the browser: Analysis of web-based malware. In *First Workshop on Hot Topics in Understanding Botnets (HotBots 07)*, Cambridge, MA, April 2007. USENIX Association.
- [48] Daniel Ringler and Heiko Paulheim. One knowledge graph to rule them all? analyzing the differences between dbpedia, yago, wikidata & co. In Gabriele Kern-Isberner, Johannes Fürnkranz, and Matthias Thimm, editors, *KI 2017: Advances in Artificial Intelligence*, pages 366–372, Cham, 2017. Springer International Publishing.
- [49] Sayak Saha Roy, Poojitha Thota, Krishna Vamsi Naragam, and Shirin Nilizadeh. From chatbots to phishbots?—preventing phishing scams created using chatgpt, google bard and claude. *arXiv preprint arXiv:2310.19181*, 2023.
- [50] Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools, 2023.
- [51] Weijia Shi, Xiaochuang Han, Mike Lewis, Luke Zettlemoyer Yulia Tsvetkov, and Scott Wen tau Yih. Trusting your evidence: Hallucinate less with context-aware decoding, 2023.
- [52] Yongxin Shi, Dezhi Peng, Wenhui Liao, Zening Lin, Xinhong Chen, Chongyu Liu, Yuyi Zhang, and Lianwen Jin. Exploring ocr capabilities of gpt-4v(ision) : A quantitative and in-depth evaluation, 2023.
- [53] Kai Sun, Yifan Ethan Xu, Hanwen Zha, Yue Liu, and Xin Luna Dong. Head-to-tail: How knowledgeable are large language models (llm)? a.k.a. will llms replace knowledge graphs?, 2023.
- [54] Ran Tian, Shashi Narayan, Thibault Sellam, and Ankur P Parikh. Sticking to the facts: Confident decoding for faithful data-to-text generation., 2019.
- [55] SM Tonmoy, SM Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das. A comprehensive survey of hallucination mitigation techniques in large language models. *arXiv preprint arXiv:2401.01313*, 2024.
- [56] Rakesh Verma and Keith Dyer. 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*, page 111–122, New York, NY, USA, 2015. Association for Computing Machinery.
- [57] Denny Vrandečić and Markus Krötzsch. Wikidata: A free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85, sep 2014.
- [58] Xintao Wang, Qianwen Yang, Yongting Qiu, Jiaqing Liang, Qianyu He, Zhouhong Gu, Yanghua Xiao, and Wei Wang. Knowledgept: Enhancing large language models with retrieval and storage access on knowledge bases, 2023.
- [59] What does chatgpt know about phishing? <https://securelist.com/chatgpt-anti-phishing/109590/>.
- [60] Wipo global brand database. <https://www.wipo.int/portal/en/index.html>.
- [61] Guang Xiang, Jason Hong, Carolyn P. Rose, and Lorie Cranor. Cantina+: A feature-rich machine learning framework for detecting phishing web sites. *ACM Trans. Inf. Syst. Secur.*, 14(2), sep 2011.
- [62] Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *arXiv preprint arXiv:2306.13549*, 2023.
- [63] Rasha Zieni, Luisa Massari, and Maria Carla Calzarossa. Phishing or not phishing? a survey on the detection of phishing websites. *IEEE Access*, 11:18499–18519, 2023.
- [64] Zscaler. Zscaler threatlabz 2023 phishing report, 2023.

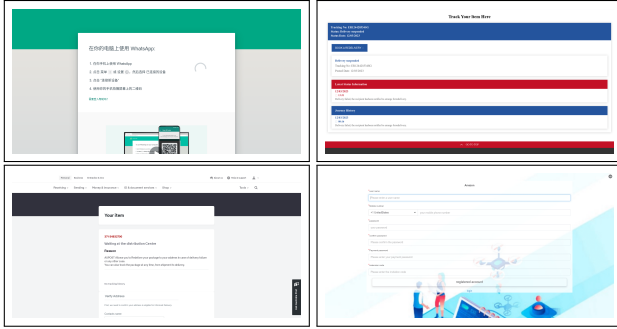


Figure 11: A few examples of logo-less phishing webpages targeting WhatsApp, Singapore Post, Australia Post, and Amazon on SG-SCAN dataset.

A Appendix

A.1 Implementation Details

For KnowPhish construction, we use the same webpage layout detector from [34] to extract the logo image from the webpage, as the instantiation of the DetectLogo() function. We limit the size of the top domain list \mathcal{D} to 50k (i.e., $\eta=50k$). The brand search algorithm thereby returns us 20514 potential phishing targets. For KPD, we also instantiate the webpage layout detector and logo matcher with the same modules from [34], and use GPT-3.5-turbo-instruct as the LLM backbone.

All the experiments are conducted within a Ubuntu server with 2 AMD EPYC 7543 32-Core Processor @ 2.8GHz and 8 Nvidia A40 48GB GPU available.

A.2 Logo-less Phishing in Field Study

Figure 11 shows a few examples of logo-less phishing webpages detected by KPD+KnowPhish on SG-SCAN dataset.

A.3 False Positives Analysis in Field Study

Here, we discuss two primary types of false positives that arise from DynaPhish: the inclusion of web-hosting brands and the popularity validation failure of benign domains. They account for more than 70% of the false positives. We elaborate on each:

Inclusion of web-hosting brands Figure 12 shows a few false positive examples resulting from the inclusion of web-hosting brands, such as file-hosting brands Nextcloud, FileGator, and a domain hosting brand Bitly. These benign webpages display the logos of web-hosting brands simply because they utilize their services, not because they are conducting phishing attacks. However, the RBPDS mistakenly report them as phishing due to the inconsistency between the logos and their legitimate domains. This issue also exists in the original

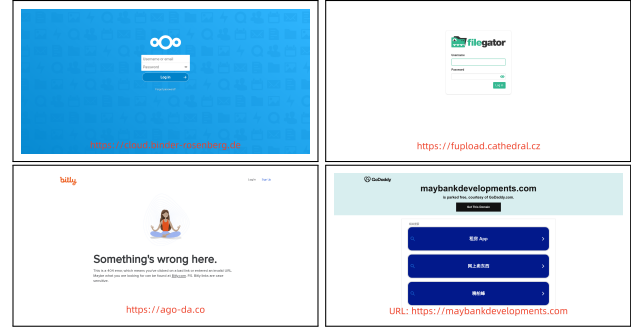


Figure 12: Examples of false positives by DynaPhish and Phishpedia/PhishIntention’s original BKB on SG-SCAN dataset, arising due to web-hosting brands.

BKB of Phishpedia/PhishIntention, which includes a domain hosting brand GoDaddy.

A straightforward solution is to exclude these brands from the BKB. In our work, KnowPhish adopts a postprocess operation that filters these brands out by conditioning on their Wikidata categories (i.e., any brand that belongs to ‘file synchronization’, ‘URL shortener’, ‘blog’, or ‘domain name registrar’ is excluded from the brand list \mathcal{B}). However, since the Wikidata information may be incomplete, we foresee that a more comprehensive list of such web-hosting brands should be collected for exclusion.

Popularity validation failure of benign domains DynaPhish also experiences false positives when failing to validate the popularity of benign domains. For example, googleadservices.com and documentforce.com are the legitimate domains of Google and Salesforce, respectively. However, popularity validation fails when DynaPhish Google Searches with these domains as queries, because they are not included in the search results. Since the popularity validation fails, and DynaPhish identifies the brand intention from the logos, phishing is mistakenly reported.

A.4 Potential Defense Improvements against Adversarial Attacks

As discussed in Section 6.5, our defense partially mitigates the adverse effect of prompt injection and text-to-image attacks, and we believe future studies are needed to better address these problems. Potential improvements for prompt injection defense include handling the adversarial prompts at input stage [23], inference stage [6], or both [36]. Regarding text-to-image attacks, additional components such as OCR with well-crafted prompts can assist LLMs to better analyze the texts from screenshots, as supervised OCR models have been found to outperform our defense model GPT-4V in OCR tasks [52].