Phishing in the Noise: Analysis of CT-based Phishing Detection Performance on Free Hosting Platforms

Maksymilian Nowak
Warsaw University of Technology
Warsaw, Poland
maksymilian.nowak2.stud@pw.edu.pl

Wojciech Mazurczyk*
Warsaw University of Technology
Warsaw, Poland
wojciech.mazurczyk@pw.edu.pl

Ewa Syta
Trinity College
Hartford, Connecticut, USA
ewa.syta@trincoll.edu

Abstract

Free Hosting Platforms (FHPs) let users publish websites with minimal cost and configuration, but the same provider-managed infrastructure can also be used to host phishing websites. We study how this setting affects Certificate Transparency (CT)-based phishing detection by analyzing X.509, CT, and URL features across FHP phishing, FHP benign, non-FHP phishing, and popular benign websites. Reflecting the shared and wildcard certificate practices common in this setting, we analyze the certificate-level and domain-level data separately. Our measurement study shows that many apparent phishing indicators instead reflect hosting-provider characteristics: in our domain-level correlation analysis, hosting-platform status is more strongly associated with the extracted X.509, CT, and URL features than phishing status is, with the strongest FHP-associated feature (subdomain levels, $\eta = 0.54$) exceeding the strongest phishing-associated feature (certificate validity period, $\eta = 0.31$). We further evaluate two representative CT-based phishing detection frameworks on FHP-only data and discover that provider-managed infrastructure creates challenges for applying them directly. These findings show the need for future work on CT-based phishing detection methods that account for this deployment setting, especially as AI tools lower the effort required to create convincing phishing websites at scale.

1 Introduction

Phishing, broadly understood as the impersonation of a trusted service to deceive users into disclosing credentials, payment information, or other sensitive data [9], remains one of the most common cyber attacks worldwide, with no signs of slowing down [35]. To be effective, a phishing website must resemble the service it impersonates and be accessible in a way that appears legitimate to users and browsers. In the past, appearing "legitimate" simply meant resembling a trusted service in terms of a URL, website design, and surrounding context. However, in practice, the widespread deployment of HTTPS [18] has changed what "legitimate" means. Because TLS certificates can be inexpensive or even free, including through Certificate Authorities (CAs) such as Let's Encrypt [2, 18], phishing websites can present valid TLS certificates and avoid the warnings that modern browsers would otherwise show for sites served only over HTTP [37]. As a result, the use of HTTPS and, consequently, TLS certificates is no longer a signal that a website is benign; rather, it is part of the baseline for legitimate and phishing websites.

This observation is particularly relevant in the context of Certificate Transparency (CT) [24]. Since 2018, Google and Apple have required newly issued certificates to appear in CT logs for websites to be treated as secure in Chrome and Safari, respectively [5, 8, 36]; Mozilla began enforcing a similar policy for Firefox desktop platforms in 2025 [22]. CT logs serve as a global public database of certificates issued by CAs worldwide. This visibility has made CT a valuable source for Internet measurements and security analyses [29]: certificate issuance patterns, domain names, and certificate features and metadata can reveal useful information about web infrastructure and individual services. Previous work has also used X.509 certificate, CT and domain features to identify phishing websites [3, 10, 14]. At the same time, visibility does not necessarily imply interpretability. For example, the same certificate can carry different meanings depending on who controls the infrastructure, how domains are assigned, and whether a certificate represents a single website or a larger hosting platform. This means that CT-derived features are not intrinsically meaningful: their interpretation depends on the deployment setting that produced them. A short validity period, a shared certificate, or a particular subdomain structure may indicate attacker behavior in one setting, but provider automation in another.

This distinction matters for Free Hosting Platforms (FHPs). Driven by global digitalization, maintaining a web presence has become essential for businesses and individuals alike. To lower the barrier to entry, FHPs, encompassing both Free Website Builders (FWB) for non-technical users and traditional Free Web Hosting (FWH) for developers, have gained massive popularity [6]. Such platforms allow users to easily build and deploy websites, handling storage, domain names, routing, and often certificate management without financial costs. With such services, a phishing campaign can be created in a very short time and for free, since these services are often bundled with a free subdomain name; for example, when creating a Wix account, if a user chooses `example` as the username, the user receives a free `example.wixsite.com` subdomain [41].

In this paper, we distinguish between two deployment settings. In the *conventional setting*, attackers control their own domain names and much of the externally visible infrastructure around the phishing website. In the *FHP setting*, the infrastructure is provider-controlled: the platform assigns subdomains, manages routing, and may issue individual certificates for hosted sites or cover many hosted sites with shared or wildcard certificates. This practice can weaken or distort CT-derived detection signals that are informative in the conventional setting. For example, a certificate included in CT may identify the hosting provider and its automation practices more clearly than it identifies the individual hosted website where phishing content appears. Similarly, shared or wildcard certificates can obscure the relationship between the global certificate-level view and the local subdomain-level site that a user actually visits.

*Also with IDEAS Research Institute.

Although phishing detection has been extensively studied, most studies assume that attackers control their own domain names and infrastructure [3, 14] or focus on general URL features [4, 10, 23]. Some of these frameworks use the CT ecosystem, but do not isolate the FHP setting. Prior work on phishing in free hosting services has shown that these platforms are actively abused [27, 32], but this line of work has focused primarily on URL-level or platform-level classification rather than on how CT-derived features behave in the FHP setting. As a result, it remains unclear whether phishing detection frameworks trained or designed for the conventional setting transfer to FHP-hosted phishing websites.

The key question is therefore not whether phishing websites can be identified from CT logs in general, but whether the existing CT-based phishing detection frameworks still perform when the infrastructure used to host phishing websites is provider-controlled. In this paper, we treat FHP-hosted phishing websites not merely as a special case of URL classification, but as a distinct deployment setting that can change how CT-derived signals should be interpreted. We therefore address the following research question:

RQ: *How do existing CT-based phishing detection frameworks perform on FHP-hosted websites?*

To answer this question, we compare FHP-hosted phishing websites with FHP-hosted benign websites, phishing websites outside FHPs, and popular benign websites. We analyze X.509, CT, and URL features, while separating certificate-level and domain-level analyzes to reflect the shared and wildcard certificate practices common in the FHP setting. Our domain-level dataset contains 84,899 entries in four groups: 41,568 FHP phishing, 4,155 FHP benign, 31,000 other phishing, and 8,176 popular benign websites. We will make our dataset publicly available upon acceptance. For X.509 and CT feature analysis, we deduplicate certificates by Subject Key Identifier, resulting in 24,725 certificate-level entries. We then evaluate two existing CT-based phishing detection frameworks on FHP-only data: a pattern-based clustering approach by Sakurai et al. [33] and a learning-based approach by Fasllija et al. [14].

Our analysis shows that the FHP setting changes the way certificate-derived features should be interpreted. Several X.509 and CT features reflect hosting-provider characteristics, such as automated issuance and shared certificate management, rather than phishing intent alone. URL characteristics, in turn, often distinguish FHP-hosted websites from non-FHP websites more strongly than they distinguish phishing from benign FHP websites. In particular, hosting-platform status is more strongly associated with the extracted X.509, CT, and URL features than phishing status is: the most discriminative FHP feature, subdomain levels, reaches $\eta = 0.54$, compared to $\eta = 0.31$ for the most discriminative phishing feature, certificate validity period. As a result, existing CT-based detection frameworks can produce misleading results in the FHP setting: pattern-based clustering tends to learn provider-specific characteristics, while learning-based models relying on conventional certificate and URL features struggle to separate benign and phishing subdomains.

In summary, this paper makes the following contributions:

- We identify FHP-hosted phishing as a distinct deployment setting for CT-based phishing detection and show that CT-derived and URL-level features must be interpreted in relation to provider-managed infrastructure. Our measurement study of X.509, CT, and URL features shows that many apparent phishing indicators instead reflect hosting-provider behavior.
- We introduce a study design tailored to the FHP setting and construct a dataset that spans FHP phishing, FHP benign, non-FHP phishing, and popular benign websites. In our analysis, we use separate certificate-level (24,725 samples) and domain-level (84,899 samples) datasets to account for shared and wildcard certificates. We will make these datasets publicly available upon acceptance.
- We evaluate two existing CT-based phishing detection frameworks in the FHP setting and demonstrate that the provider-managed infrastructure poses challenges to their direct application. Our results motivate future work on phishing detection frameworks that account for the specific properties of FHP-hosted websites.

This paper is organized as follows. Section 2 discusses related work. Section 3 presents our study design, describes our datasets and features used for the analysis, and discusses limitations. Section 4 analyzes certificate-level and domain-level feature analysis across our comparison datasets. Section 5 evaluates two existing CT-based detection frameworks on FHP data. Finally, Section 6 concludes and discusses future work.

2 Related Work

Prior work on phishing hosted on Free Hosting Platforms (FHPs) has shown that attackers already use these platforms at scale. Saha Roy et al. [32] study phishing attacks that use Free Website Builder (FWB) services and identify more than 30,000 phishing URLs from 17 unique platforms. Their work demonstrates the relevance of FWBs as a phishing infrastructure, but focuses on malicious URLs distributed through advertisements on social media platforms such as Facebook and Twitter. Lohar et al. [27] similarly study phishing websites hosted on traditional Free Web Hosting (FWH) services and propose a machine learning model for distinguishing FWH and non-FWH domain names. However, their evaluation relies on a manually updated dataset and excludes certificate information. These studies establish FHP-hosted phishing as an important problem, but they do not evaluate whether CT-based phishing detection remains effective in the FHP setting.

CT-based phishing detection has been studied from pattern-based and learning-based perspectives. Sakurai et al. [33] detect phishing campaigns by clustering groups of similar domain names, extracting regular-expression patterns, and matching those patterns against certificate *Common Name* fields to identify new domains from existing campaigns. Fasllija et al. [14] take a learning-based approach in Phish-Hook, applying machine learning directly to certificate features extracted from the CT logs and reporting a success rate of over 90%. AlSabah et al. [3] extend content-agnostic phishing detection by combining CT logs with passive DNS data, achieving high performance (98% precision, 90% recall, 0.3% false positive rate), but their dataset explicitly excluded the FHP subdomains, leaving the impact of these subdomains on performance unknown. Drichel et al. [10] present a complete CT-based phishing detection pipeline that includes dataset creation, model training,

classification, and intelligence modules, showing how different classifiers can be embedded into a broader detection framework. These systems show that information from CT logs can support phishing detection, but they leave open the FHP-specific research question considered in our paper.

Most CT-based approaches are designed for and evaluated in the conventional setting, where domain names, certificate fields, and issuance behavior are more likely to reflect attacker-controlled infrastructure. In contrast, FHP-hosted phishing is deployed within provider-controlled infrastructure, where platform choices can independently shape X.509, CT, and URL features, regardless of phishing activity. AlSabah et al. explicitly exclude FHP subdomains from their dataset, and the remaining CT-based systems do not treat FHP-hosted websites as a distinct deployment setting. Our work, therefore, complements prior studies by evaluating existing CT-based phishing detection frameworks on FHP-hosted websites.

3 Study Design

We design our study to evaluate CT-based phishing detection in the FHP setting. Our study asks whether X.509, CT, and URL features, which are useful for detecting phishing in the conventional setting, remain useful when interpreted in the FHP setting. Our hypothesis is that certificate issuance and management practices for websites hosted on FHPs differ and that this difference creates a gap in existing CT-based phishing detection frameworks.

Our methodology has three parts. First, we construct a dataset that compares FHP-hosted phishing websites with benign FHP websites, phishing websites outside FHPs, and popular benign websites. Second, we identify, extract, and analyze X.509, CT, and URL features to assess which features behave as phishing signals and which instead reflect hosting-provider characteristics. Third, we evaluate two representative CT-based phishing detection frameworks on FHP-hosted websites: a pattern-based approach by Sakurai et al. [33] and a learning-based approach based on Phish-Hook [14].

3.1 Datasets

We compare four groups, defined by whether a website is hosted on an FHP and whether it is classified as phishing or benign. *FHP Phishing* and *FHP Benign* contain phishing and benign websites hosted on FHPs, respectively. *Other Phishing* contains phishing websites not hosted on FHPs. *Tranco* contains popular benign, non-FHP websites drawn from the Tranco Top 50000 list [25, 38]. This comparison approach provides two main axes: FHP phishing can be compared with FHP benign websites to analyze phishing status within the same deployment setting, and with other phishing websites to analyze differences associated with FHP hosting. The Tranco group provides a popular benign baseline outside the FHP setting. For feature extraction and analysis, we use the 2024 dataset by Hranický et al. [21], which draws phishing websites from trusted sources such as PhishTank and OpenPhish, and benign websites from sources such as Cisco Umbrella and CESNET.

Because benign FHP websites are underrepresented in that dataset, we augment them using Tranco lists from late 2023 through the first half of 2024 with pay-level-domain filtering disabled [39, 40]. We then extract all entries whose domains match our list of FHP apex domains in Appendix B, and check the added benign FHP subdomains with Google Safe Browsing API [19] not to misclassify a phishing website as benign. Since the Tranco lists do not contain X.509 certificates, we used `crt.sh`, a web-based tool for querying archival certificates logged in the CT ecosystem [26], for each subdomain. In total, the complete dataset has 84,899 entries: 41,568 *FHP Phishing*, 4,155 *FHP Benign*, 31,000 *Other Phishing* and 8,176 *Tranco*. We refer to this dataset as the *domain-level dataset*.

3.2 Certificate-level and Domain-level Approach for Analysis

The FHP setting commonly involves shared and wildcard certificates. This means that a single certificate may represent many hosted websites. We therefore analyze the data at two levels: certificate-level and domain-level. This separation is important because certificate-level and domain-level observations answer different questions in provider-managed settings.

For X.509 and CT feature analysis, we use a certificate-level dataset obtained by deduplicating certificates by Subject Key Identifier (SKI). This reflects that certificate fields and CT metadata describe the certificate, not each individual hosted subdomain. Without deduplication, a single wildcard certificate covering thousands of subdomains would inflate per-group statistics in proportion to the subdomain count. After deduplication, our *certificate-level dataset* consists of 6,306 FHP Phishing, 457 FHP Benign, 11,068 Other Phishing and 6,894 Tranco certificates - 24,725 entries total.

For URL feature analysis, we use the full domain-level dataset because URL features focus on the individual domain names and are independent across subdomains, even if those subdomains share the same certificate. In this case, deduplication would discard real lexical diversity. We compute URL features after removing the public suffix so that the features describe the hosted domain or subdomain rather than the provider's apex domain. To achieve this, we utilized the Public Suffix List [17].

3.3 Evaluation Inputs for CT-based Frameworks

Each of the frameworks we evaluate requires a different set of data, which we draw from the source data accordingly. The pattern-based framework (subsection 5.1) performs unsupervised clustering, so it does not rely on phishing labels. We run it on all records in the Hranický et al. source dataset [21], including those dropped for feature analysis due to missing X.509 data or not belonging to any of the groups, that match one of the FHP apex domains listed in Appendix B. We also apply the same threshold of a minimum of 10 characters in the domain name as the framework authors, which brings the final input size to 69,126 samples. The learning-based framework (subsection 5.2) is supervised and requires labeled data; we evaluate it on *FHP Benign* and *FHP Phishing* groups of the domain-level dataset.

3.4 Features

We extract 14 features grouped into three categories, shown in Table 1. *X.509 features* capture structural properties of the certificate itself, such as the issuing CA, validity period, certificate type, extensions, and Subject Alternative Name (SAN) entries. *CT features* capture how certificates appear in Certificate Transparency logs, including the number of Signed Certificate Timestamps (SCTs) and

the time between SCT inclusion and the certificate's `notBefore` date. *URL features* capture lexical properties of domain names; we adapt from prior phishing-detection work that used lexical phishing indicators [3, 14], and apply them to the domain name in Common Name certificate field. It is worth noting the Hranický et al. dataset does not directly expose the Common Name field; instead, they provide a `domain_name` column carrying the same information. This lets a single shared or wildcard certificate appear in multiple records while preserving the original domain name that has been observed in the wild. Feature selection was guided by prior use in related work [3, 10, 11, 14], availability across our datasets, and hypothesized relevance to the FHP setting.

3.5 Experiments

We conduct our study in two stages. First, we compare the 14 extracted features across the 4 dataset groups to determine which features distinguish FHP-hosted websites from non-FHP websites and which distinguish phishing from benign websites within the FHP setting. Second, we evaluate 2 existing frameworks on FHP-hosted websites. Sakurai et al. [33] uses a pattern-based clustering approach on certificate Common Names to derive regular-expression templates. Phish-Hook by Fasllija et al. [14], is a learning-based approach that uses certificate and domain features observable through CT logs. Neither framework was designed specifically for FHP-hosted websites, making them useful test cases for whether existing CT-based methods transfer to this deployment setting. We compare our results to those reported in the original publications, both of which were evaluated on general, non-FHP websites.

3.6 Limitations

Our study is limited to evidence available through CT logs, certificate metadata, and domain names. This scope makes the evaluation directly relevant to CT-based detectors, but it also means we do not analyze page content, redirection behavior, hosting-account metadata, or takedown dynamics.

We deduplicate certificates using the Subject Key Identifier (SKI) because certificate serial numbers and fingerprints are not available in the source dataset. Since SKI identifies the public key rather than the certificate itself, it is possible that distinct certificates that share the same key would be incorrectly classified as duplicates; this is unlikely but possible. After deduplication, *FHP Benign* is considerably smaller than the other certificate-level groups (457 certificates versus 6,306 *FHP Phishing*, 11,068 *Other Phishing* and 6,894 *Tranco*). This reflects that many benign FHP subdomains utilize a small number of providers that reuse certificates across many hosted sites, so the group represents a distinct certificate configuration for such websites rather than just presenting a sample of them. We interpret the results with this group size difference in mind.

Finally, the source code for both evaluated frameworks is unavailable. Thus, we reproduce them from their published descriptions, adapting undisclosed implementation details as closely as possible to the original design. These changes are documented alongside each framework's evaluation results in Section 5.

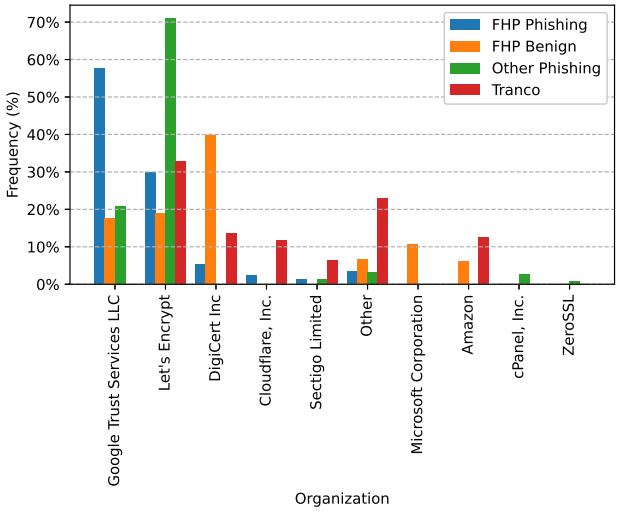


Figure 1: Normalized distribution of top-5 CAs per group.

4 Feature Analysis [ES: reviewing]

We compare X.509, CT, and URL features across the four dataset groups to understand which features distinguish phishing from benign websites and which instead reflect the FHP setting. We analyze X.509 and CT features at the certificate level, and URL features at the domain level, as previously described.

4.1 X.509 Features

Issuing Certificate Authority. The choice of a CA varies across groups. Let's Encrypt is the most popular issuer for *Other Phishing*, with 71% group share, as shown in Figure 1. However, contrary to popular belief [12, 20], using this specific CA does not necessarily mean a website is malicious; it is also the most popular choice for *Tranco* websites (33%), albeit within a more diverse set of CAs. Top 5 issuers account for only about 77% of all certificates in *Tranco*, compared to over 93% for the remaining groups. For FHPs, the results diverge: *FHP Phishing* is led by Google Trust Services (57.8%), with Let's Encrypt at only 29.9%, while 39.8% of *FHP Benign* certificates are issued by DigiCert. Let's Encrypt is again second, with 19% share. The prevalence of Google Trust Services in *FHP Phishing* is attributable to Blogger, a Google-owned FHP that issues certificates through Google's own CA, suggesting that Blogger accounts for a large share of FHP-hosted phishing websites and is likely a popular choice for FHP-hosted phishing campaigns. The country of origin does not provide a useful discriminative signal. The vast majority of the CAs are US-based, with only the *Tranco* group having about 15% of certificates from other countries, which we attribute to the long tail of certificates not from the Top 5 issuers for this group.

Validity period. This is the most discriminating X.509 feature that distinguishes phishing from benign websites, regardless of the hosting environment. Both phishing groups cluster around 90 days (see Figure 2), which is the maximum validity period for Let's Encrypt certificates [1]. They both have the median and 75th percentile at the value of 89 days, meaning at least three quarters

Table 1: List of analyzed features.

Feature Group	Features
X.509 Features	Issuing CA, CA Country of Origin, Validity Period, Certificate Extensions, Certificate Type, Number of SAN Entries
CT Features	Number of SCTs, Inclusion Time Difference
URL Features	Entropy, URL Length, Subdomain Levels, Number of Hyphens, TLD, Hidden TLD in Domain

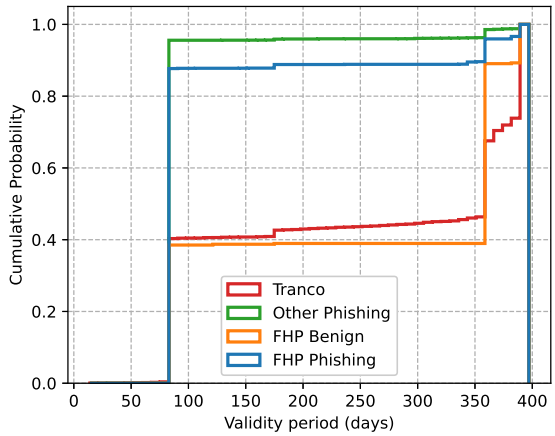


Figure 2: Empirical CDF of certificate validity periods.

of *FHP Phishing* and *Other Phishing* certificates are the same length or shorter than this threshold. On the other hand, *FHP Benign* and *Tranco* groups both have a median of 364 days. This is consistent with our correlation analysis (subsection 4.4), in which the validity period is the most discriminating feature for the phishing target label ($\eta = 0.31$). This pattern reflects Let's Encrypt's automated issuance policy rather than a property specific to phishing intent; the 90-day ceiling applies equally to any domain that chooses this CA, regardless of purpose.

We also compared certificates with 2 and 3 SCTs separately. Among 2-SCT certificates, all groups have a median of 89 days. This is expected, as CT browser policies tie the required number of SCTs to the certificate's validity period [8, 36]. However, for 3-SCT certificates, the previously discovered pattern still holds: phishing groups remain at 89 days, and benign groups shift to 364 and 365 days. Because the validity gap between phishing and benign groups persists even after conditioning on SCT count, we interpret this as a deliberate choice to include more SCTs than necessary, rather than a browser policy artifact.

Certificate type. Domain Validated (DV) certificates dominate both phishing groups: 95% of *FHP Phishing* and 97% of *Other Phishing* certificates are DV-only. Benign groups show substantially more Organizational Validation (OV): 37.9% for *FHP Benign* and 49.5% for *Tranco*. This is consistent with two factors: OV issuance requires CAs to verify the requesting organization's legal existence [16], which phishing actors are unlikely to undergo, and Let's Encrypt, the most popular CA for phishing, issues only DV certificates [13].

Other X.509 features. Certificate extension counts and SAN entries show negligible differences across groups for the certificate-level dataset; all groups have a median of 9 to 10 extensions and 2 SAN entries. We do not draw conclusions from these features alone. Overall, X.509 features provide some separation between phishing and benign websites, especially through validity period and certificate type, but several of these differences are shaped by CA and provider practices rather than phishing status alone.

4.2 CT Features

SCT Count. The number of Signed Certificate Timestamps broadly mirrors the validity period pattern. This is because the CT browser policy requires at least two SCTs for certificates valid for less than 180 days and three for longer ones [8, 36]. Since the majority of phishing certificates are 90 days long or shorter, it comes as no surprise that the vast majority of them have only 2 SCTs embedded in them; specifically, 87.3% of *FHP Phishing* and 89.7% of *Other Phishing*. Benign groups predominantly have 3 SCTs, with the exact number being 72.4% for *FHP Benign* and 86.4% for *Tranco*.

SCT and `notBefore` time difference. While all groups tend to be logged before their `notBefore` date, the actual lead times follow a broad gradient: *Tranco* certificates are provisioned furthest in advance (median 2,558 h, ≈107 days), consistent with avoiding any lapse in certificate validity, while both phishing groups show shorter lead times (*Other Phishing* median 113 h, *FHP Phishing* 339 h), and *FHP Benign* sits between the two (919 h). However, the standard deviations substantially exceed the medians across all groups, indicating highly overlapping distributions. At a 24-hour threshold, *FHP Phishing* and *FHP Benign* are nearly identical (75.7% vs. 76.8%), but the gap widens at longer horizons: at 30 days, 54.7% of *FHP Benign* certificates were provisioned at least that far in advance, compared to only 28.8% of *FHP Phishing*. Therefore, CT inclusion timing provides weak discrimination between FHP phishing and benign traffic at longer provisioning horizons, but is not reliable enough to serve as a standalone signal. The gradient itself likely reflects differences in provisioning automation. Provider-managed FHP infrastructure issues certificates on demand at subdomain creation, while established operators and bulk phishing campaigns do so in advance. Together, these CT features mainly reflect certificate automation and operational timing, which can support analysis but do not provide a direct phishing signal in the FHP setting.

4.3 URL Features

URL features are computed on the domain-level dataset because they capture independent per-domain properties.

Entropy and URL length. Both FHP groups exhibit higher Shannon entropy than *Tranco* and *Other Phishing*, with median values of 4 and 3, respectively. This suggests that user-chosen FHP

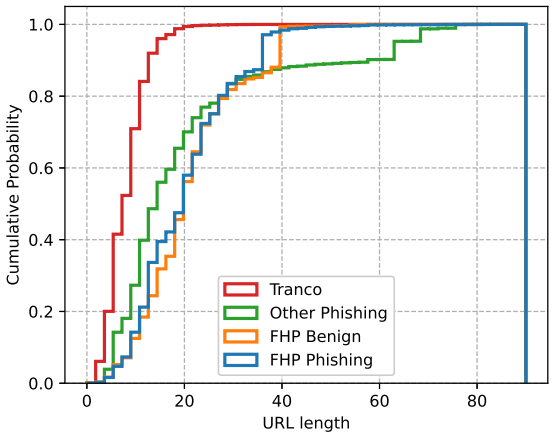


Figure 3: Empirical CDF of URL lengths excluding the public suffix.

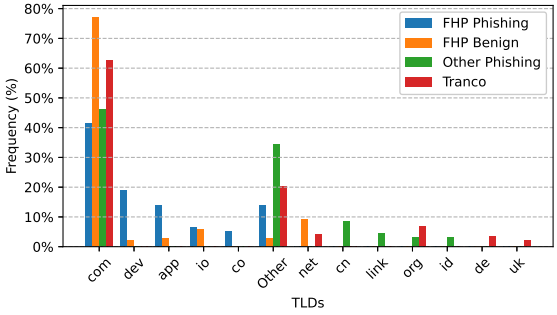


Figure 4: Normalized distribution of top-5 TLDs per group.

subdomains are not constrained by a specific naming convention. *Tranco* domains are significantly shorter, with an average of 8.78 characters, compared to the remaining groups, which have means between 20 and 22 characters. This also reflects the tendency of established brands and businesses to favor short, memorable names. When looking at the cumulative distribution of URL lengths in Figure 3, we also notice the FHP groups are almost indistinguishable from each other. Moreover, at around 40 characters, their cumulative probability abruptly jumps to almost 1.00, indicating the potential character limit for FHP subdomain names.

Subdomain levels. Subdomain levels separate FHP from non-FHP traffic most cleanly: *Tranco* has a median of 0 and *Other Phishing* a median of 1, while *FHP Phishing* and *FHP Benign* have medians of 1 and 2, respectively — a structural consequence of FHP platforms requiring at least one subdomain level to function. Both FHP groups also use more hyphens than non-FHP groups (means of 0.87-0.91 vs. 0.44 for *Other Phishing* and 0.05 for *Tranco*).

Top-level domain. In Figure 4, `.com` is clearly the most popular top-level domain across all groups. For *FHP Phishing*, it sits at 41.4% share. *Other Phishing* has 46.3% of `.com` domains, while both benign groups sit much higher, with 77.2 and 62.7% for *FHP Benign* and *Tranco*, respectively. Both FHP groups use a distinct set of TLDs that is largely absent from other groups. This set includes `.dev` (19.1% of *FHP Phishing*, 2.9% of *FHP Benign*), `.app` (13.8% of *FHP Phishing*, 2.8% of *FHP Benign*), `.io` (6.6% of *FHP Phishing*, 5.8% of *FHP Benign*) and `.co` (5.2% for *FHP Phishing*). Moreover, in the case of *FHP Phishing* domains, the combined TLDs listed above account for 44.7% of all domains, which is more than the most popular `.com`. Both non-FHP groups show a long tail of largely regional TLDs, such as `.cn`, `.id`, `.de` and `.uk`. Top 5 TLDs account for 65.6% of *Other Phishing*, and for 79.6% of *Tranco*.

Hidden TLD. Contrary to the assumption in Fasllija et al. [14], the hidden-TLD feature behaves unexpectedly: phishing groups show low rates (6.2% for *FHP Phishing*, 5.5% for *Other Phishing*), while *FHP Benign* shows a rate of 41.4%. We attribute this to CDN and infrastructure hostnames in the benign group, whose domain strings contain cloud-region identifiers that coincide with country-code TLDs (e.g., `us-east-1.amazonaws.com`). This illustrates how a feature designed as a phishing indicator in the conventional setting can instead capture benign provider-managed infrastructure in the FHP setting. To better understand this result, we apply a simple pattern-based check for infrastructure-like hostnames, including node identifiers, IP address fragments, cloud region strings, and service keywords. This identifies 592 of 4,155 (14.2%) *FHP Benign* domains as infrastructure-related, of which 98% belong to Microsoft's services, primarily SharePoint (89% of all infrastructure-related domains) or Azure (9%).

Non-infrastructure FHP domains can be divided into three groups: Microsoft 365 enterprise portals, which account for almost 50% of all such domains; blog platforms such as Blogger and WordPress.com; and application deployments, mainly via Azure Static Web Apps and GitHub Pages. This suggests that genuinely user-created benign FHP subdomains are rare, with the group dominated by enterprise platform tenancy rather than voluntary FHP adoption. Overall, URL features most clearly distinguish FHP-hosted websites from non-FHP websites, while offering weaker separation between phishing and benign websites within the FHP setting.

4.4 Correlation Between Features

We compute a mixed-type correlation matrix using Pearson's r for numerical-numerical pairs, the correlation ratio η for numerical-nominal pairs, and Theil's U for nominal-nominal pairs [15, 31]. We use `dython` library [42] for this task. The matrix is computed on the full domain-level dataset. For X.509 and CT features, each certificate's values are shared by all subdomains it covers, so the corresponding correlations should be read as domain-level associations with certificate-level properties rather than associations across distinct certificates (see Section 3.4). We present the calculated matrix in Figure 5 (Appendix C).

`is_fhp` consistently exhibits stronger associations than `phishing`, which is expected, as prior work has already established that phishing classification from certificate features alone is not feasible [11], while FHP hosting leaves structural traces in the domain name itself. The most discriminative feature for FHP classification is `subdomain_levels` ($\eta = 0.54$), followed by `extension_count` ($\eta = 0.44$), `tld` ($U = 0.37$), `hyphens_in_domain` ($\eta = 0.27$) and *entropy* ($\eta = 0.25$). These results are consistent with our findings in Sections

4.1 and 4.3. In comparison, the most discriminative `phishing` feature is `valid_len` with $\eta = 0.31$, closely followed by `num_of_scts` ($\eta = 0.29$) and `entropy` ($\eta = 0.27$).

We also observe several strong inter-feature correlations. `organization` shows near-perfect correlation with both `valid_len` ($\eta = 0.98$) and `extension_count` ($\eta = 0.99$), most likely due to free CA policies that restrict certificate lifetime to 90 days [1], making the issuing organization a near-deterministic predictor of validity period. Certificate type features (dv and ov) exhibit the uncertainty coefficient of $U = 0.96$, which is expected, since they are mutually exclusive. Another strongly correlated pair is `entropy` and `url_length` ($r = 0.80$), since longer domain names naturally yield higher Shannon entropy. Furthermore, `num_of_scts` and `valid_len` show a notable correlation ($r = 0.54$), reflecting the browser CT policy requirement that longer-lived certificates carry more SCTs [5, 8, 36]. Another notable pair is `SANnumber` and `num_of_scts`, with $r = -0.25$. We attribute this anomaly to wildcard certificates that cover multiple regional domains of the same service (such as Blogger's regional TLDs [7]) and are subject to a shorter validity period.

5 Evaluation of Existing CT-Based Detection Frameworks

In this section, we evaluate two representative CT-based phishing detection frameworks:

- a pattern-based approach by Sakurai et al. [33] that clusters certificate Common Names into regular-expression templates for matching newly logged certificates; and
- a learning-based approach (Phish-Hook) by Fasllija et al. [14] that extracts certificate and domain features from CT data and trains classifiers on them.

We selected these frameworks because they represent both pattern-based and learning-based approaches to phishing detection, and importantly, neither framework was designed specifically for the FHP setting. We evaluate each framework using the input that matches its requirements: unlabeled FHP records extracted from the Hranický et al. dataset (see Section 3.3) for pattern-based approach, and labeled FHP groups from our domain-level dataset for learning-based approach.

5.1 Pattern-based Approach

Sakurai et al.'s framework [33] relies on clustering analysis of certificates from existing phishing websites. After grouping the certificates by their Common Names, the framework generates phishing campaign templates as regular expressions. These templates are then used to filter new certificates appearing in the CT logs.

Sakurai et al. use DBSCAN as the clustering algorithm and apply the Ratcliff-Obershelp distance as the heuristic function. Furthermore, clustering is performed in groups of certificates that share the same number of dots in the Common Name. In other words, there are actually multiple DBSCAN algorithms running at the same time for groups of certificates with one dot, two dots, etc.

Since this is not a traditional machine learning framework, standard evaluation metrics such as accuracy, precision, recall, etc., do not apply. Our findings for this framework are therefore qualitative. As a result, we focus on the clustering results for the dataset consisting only of FHP subdomains. We also calculate the noise percentage, which indicates the proportion of certificates in the dataset that could not be grouped into any cluster. Because Sakurai et al. rely only on the Common Name field of each certificate, we use the terms *certificate* and *subdomain* interchangeably throughout. Given that the framework's source code is not publicly available, we reproduced the experiments described in the original paper on unlabeled FHP data (see Section 3.3).

Table 2: Statistics of clusters and noise percentage for original and tweaked versions of [33] on the FHP-only dataset.

Statistic	Original	Tweaked
count	1939	4529
mean	25.12	5.52
std	870.69	56.94
min	2	2
25%	2	2
50%	2	2
75%	3	3
max	38292	3376
Noise %	29.53	50.06

Results are presented in Table 2 ('Original' column). Almost 30% of all certificates in the dataset were noise: 20,413 samples were classified into cluster no. -1, which according to `scikit-learn`'s documentation should be interpreted as noisy neighbors. In total, 1939 clusters were created with an average size of 25 certificates. We note that this number is artificially inflated by a single large cluster of more than 38,000 certificates. Moreover, the median value is only 2 certificates, which is the minimum cluster size allowed in both the Sakurai et al. paper and our reproduction.

We investigated the generated clusters and concluded that: *(i)* most of the certificates are grouped together by the FHP they use; *(ii)* the largest cluster (38,292 certificates) is very generic and contains Common Names that are not related to each other (e.g., fshraha.pages, zxdpage.pages, port02-adf.pages, dis01.pages, moitamssklogin.gitbook, etc.), yet they were not classified as noise.

Then, following the original Sakurai et al. approach, we performed an entropy reduction to eliminate overly generic templates. We discovered that the following pattern persists: the certificates are grouped by their FHP, not, for example, by their lexical features or the keywords they use (see Table 3). When we used the same 55-bit threshold as Sakurai et al., half of the clusters were eliminated, including the largest one. This suggests that the original parameter setup does not generalize to the FHP setting: removing half of the groups indicates that many templates were not sufficiently specific, while the remaining ones reflect hosting provider characteristics rather than those of phishing domain names. Furthermore, it is not a result of this specific threshold value, as changing it would not alter how the clusters were grouped in the first place.

We also noticed an anomaly regarding the Blogger subdomains. When a user visits a specific blog, they may be redirected to a local URL rather than `blogspot.com` [7]. This behavior is reflected in our dataset, where multiple Blogger URLs share the same subdomain string, which, when stripped of the TLD, is identical. Sakurai et al. also observed anomalous behavior on FHP websites. In their paper, they state that if a template contains the FHP subdomain,

they manually strip it. This behavior results in an artificial cluster with high URL-string entropy, such as cluster no. 4 in Table 3.

Taking the above into account, we conclude that the original clustering implementation by Sakurai et al. does not work well in the FHP setting. Therefore, we modified the algorithm to better suit the specific properties of the FHP subdomains, which we refer to as the *tweaked version*. The most notable change is the method used to group certificates before clustering. Instead of the number of dots, we use the number of hyphens in the subdomain part of the URL. Additionally, since a few FHP providers allow users to create sub-subdomains, the heuristic function is the weighted sum of Ratcliff-Obershelp similarity, as in the original implementation, and the difference in the number of dots between the two Common Names at a given time. We use a weight of 0.2 for the *dot difference* because it primarily serves as a penalty rather than a significant component of the clustering process. We also strip the FHP's provider's apex domain from the URL to avoid overly generic groups, as seen in the original implementation. Because the apex domain is stripped before filtering, the minimum 10-character requirement removes more samples than in the original version, reducing the tweaked input size to 50,087 samples. After running experiments for the tweaked version, we noticed that 50% of the certificates in the dataset were classified as noise (see Table 2, 'Tweaked' column). The number of clusters increased to 4,529, with the largest cluster containing 3,376 certificates, almost ten times fewer than the largest cluster in the original implementation. However, most of them still have only two certificates per cluster.

After removing overly generic patterns through entropy reduction, we noticed that most clusters actually consist of duplicate copies of the same string (Table 3). There are two causes for this behavior. First, it is the same case as in the original approach. Blogger websites have multiple URLs that share the same subdomain and multiple regional TLDs. When we remove the apex domains from the FHP URLs, the remaining substrings are identical. Second, now that FHPs' domains have been stripped out, URLs with the same subdomain name but hosted on multiple FHPs are clustered together because the only remaining part is identical, and, to the algorithm, they are similar.

In summary, this pattern-based approach does not yield satisfactory results in the FHP setting. The templates it generates are often either too generic or too specific to consistently identify new phishing FHP subdomains. Together with the fact that many FHP subdomains use shared or wildcard certificates, which reduce the number of unique domain names found in CT logs, this suggests that the pattern-based approach requires substantial redesign before it can be applied to FHP-only data.

5.2 Learning-based Approach

For a learning-based approach, we test the Phish-Hook framework by Fasllija et al. [14] on the FHP-only dataset. The authors collect incoming certificates from CT logs and extract eight binary features: small Levenshtein distance of sub-words in the domain to a list of popular suspicious keywords, large number of subdomain levels in the domain, usage of a free Certificate Authority, suspicious Top Level Domain, presence of a hidden TLD inside the domain name, presence of any of the suspicious keywords listed in the paper, high Shannon entropy of the URL and an unusually high number of hyphens in the domain name. Afterwards, the authors train multiple classifiers on the extracted features and evaluate their results. Because the original source code was not available, we reproduced the experiments described in the original paper.

Since the original features are binary, but the certificate properties used to compute them are not, Fasllija et al. established threshold values for each feature. However, the thresholds themselves are not disclosed in the paper, so we decided to change some features from binary to continuous. Moreover, some of the features used are not appropriate for an FHP-only dataset. Specifically, the `issued_from_free_CA` feature is not a valid phishing indicator for FHP subdomains, since only a minority of such domains use free Certificate Authorities. Considering the above, we slightly modified the features to address missing information, while keeping the model as close to the original as possible. In particular, we applied the following modifications:

- `small_levenshtein_distance` contains the smallest Levenshtein distance value for all the words listed in the original paper. If no similarity between the strings is found (the Levenshtein distance is $+\infty$), we assign the feature's value to the largest finite distance in the dataset +1.
- `deeply_nested_subdomains` is classified as true if the number of dots in the URL (excluding the apex domain) is > 1.
- For `issued_from_free_CA`, the actual list of the Certificate Authorities classified as *free* has not been disclosed, so we used the following CAs [28]: Let's Encrypt, SSL for Free, Basic SSL, ZeroSSL, BuyPass Go SSL.
- `suspicious_keywords` checks for suspicious keywords only in the subdomain part of the FHP URL.
- `high_shannon_entropy` contains the actual Shannon entropy value of the subdomain part of the FHP's URL, since the threshold value has not been disclosed, similarly to `small_levenshtein_distance`.
- `hyphens_in_subdomain` has been converted to a continuous feature, since the threshold is not known.

In total, of the eight original features, we changed three from binary to continuous (`small_levenshtein_distance`, `high_shannon_entropy`, and `hyphens_in_subdomain`), read directly from the dataset, and normalized afterwards, while `deeply_nested_subdomains` keeps a binary form with the structural cutoff stated above. The remaining features are kept as in the original design, except that `issued_from_free_CA` uses our own list of free CAs, since the original list was not disclosed. Finally, Fasllija et al. used a multi-class target label set (*legitimate*, *potential*, *likely*, *suspicious*, *highly-suspicious*), but their evaluation dataset has only binary labels. As a result, they decided to add a class threshold, which defeats the purpose of the multi-class label.

To ensure that our changes do not introduce bias into the experiment, we first wanted to train the modified model on the original training dataset, evaluate it on the UCI dataset [30], and then evaluate it on the FHP-only dataset. However, like the feature thresholds, the original training dataset has not been made available. Moreover, it relied on live CT data, so replicating it would be impossible without querying archival CT logs for data from a period similar to the publication date. Thus, we decided on the following approach: *(i)* Perform *train-test split* on the UCI dataset; *(ii)* Train and evaluate

No.	Common Names	Regex Pattern
Original version of Sakurai et al.		
1	mashealthcenters-pdf.pages, mashealthcenters-clu.pages, mashealth-file.pages, mashealthcenters-m.pages, mashealthcenters.pages	mashealthcenters[a-z]{0,5}\.pages
2	generatorfreeaccounts.blogspot, generatorfreeaccounts.blogspot, generatorfreeaccounts.blogspot, generatorfreeaccounts.blogspot, generatorfreeaccounts.blogspot, generatorfreeaccounts.blogspot, generatorfreeaccounts.blogspot, generatorfreeaccounts.blogspot, generatorfreeaccounts.blogspot, generatorfreeaccounts.blogspot	generatorfreeaccounts\.blogspot
3	policies-community1873-bf8.pages, policies-community-scrure.pages, policies-community1873.pages, policies-community1873-bf9.pages, policies-community192ar6.pages	policies\-community[a-z0-9]{4,8}\.pages
4	blki-9ws2bui09-tessenwout-gmailcom.vercel, durr-5nlnryzca-tessenwout-gmailcom.vercel, ci-q9aoco4gj-tessenwout-gmailcom.vercel, ret-o5bllzmoy-tessenwout-gmailcom.vercel, swet-m2pz3jmqh-tessenwout-gmailcom.vercel, nack-eyckju61l-tessenwout-gmailcom.vercel, car-grik8z9ay-tessenwout-gmailcom.vercel	[a-z0-9]{12,14}\-tessenwout\-gmailcom\.vercel
5	debank-deban.github, debank-debanj.github	debank\-deban[a-z]{0,1}\.github
Tweaked version of Sakurai et al.		
6	generatorfreeaccounts, generatorfreeaccounts, generatorfreeaccounts, generatorfreeaccounts, generatorfreeaccounts, generatorfreeaccounts, generatorfreeaccounts, generatorfreeaccounts, generatorfreeaccounts, generatorfreeaccounts	generatorfreeaccounts
7	instagramprofileurl, instagramprofileurl, instagramprofileurl, instagramprofileurl, instagramprofileurl, instagramprofileurl, instagramprofileurl, instagramprofileurl, instagramprofileurl, instagramprofileurl	instagramprofileurl
8	stywsysdyszyswyhrh, stywsysdyszyswyhrh	stywsysdyszyswyhrh
9	srtyerywyery, srtyerywyery, stryerywyery, srtyerywyery, srtyerywyery	srtyerywyery
10	internationaldomainrepresentationmainserstexts2, internationaldomainrepresentationmainserstexts3	internationalrepresentationmainserstexts[0-9]{1}

Table 3: Sample clusters after entropy reduction for original and tweaked versions of Sakurai et al. [33].

the model on respective subsets; *(iii)* Finally, run the model predictions on the FHP-only dataset. This approach allows us to use the same evaluation data as Fasllija et al. and assess the feasibility of this method for detecting FHP-based phishing.

Since the UCI dataset does not contain certificate data, we retrieved matching data from the live TLS connection, if the website was still live, or crt.sh database [26]. If the certificate data were not available from either source, we dropped the domain from the dataset. In the end, the UCI dataset originally contained 227,014 domains; for 20,159 (8.88%) of them, no matching certificate was found, bringing the final number of samples used for training and evaluation to 206,855.

After extracting the feature matrix, we prepared the dataset for the experiment's training phase. First, we split the dataset into training and testing sets using `scikit-learn`'s `train_test_split` function [34]. We set the test size to 20% and additionally used the `stratify` parameter, since the UCI dataset is imbalanced, with about twice as many benign domains as phishing domains. Afterward, we trained and evaluated the same models as in the original publication. The authors used decision trees (DTs), support vector machines (SVMs), *k*-nearest neighbors (KNN), and multilayer perceptrons (MLP) from the `scikit-learn` Python library.

On the UCI dataset, our reproduced model achieved performance between 4 and 15 percentage points lower than reported by Fasllija et al. in [14]. This variance represents an intentional methodological trade-off: by converting select features from binary to continuous, we avoided introducing arbitrary, tuned thresholds that would artificially inflate our baseline performance on the UCI dataset. While this continuous representation enriches the feature space with more granular data, it introduces a conscious trade-off in replication fidelity relative to the original binary model. Nonetheless, this design choice ensures a highly transparent and reproducible baseline that does not rely on undisclosed, dataset-specific threshold parameters. We attribute the performance decrease to the use of a different training dataset and the change of some features from binary to continuous, which we adopted instead of reconstructing Phish-Hook's undisclosed thresholds. Because we did not tune any thresholds ourselves, threshold choice is not a source of the observed gap. We therefore consider our version a close approximation to the published version, while recognizing it is not an exact replication.

The results on the FHP-only dataset vary between the models used. The results presented in Table 4 show that all DTs and MLPs have lower accuracy compared to the original results reported in [14]. Interestingly, the overall performance of SVM is much lower on the FHP-only dataset (62%) than for other model families.

However, the results in Table 4 show the overall performance for the combined classes of *Phishing* and *Benign*. When we examine the metrics for each class separately (see Table 5), all the evaluated models achieve much worse results on *Benign* subdomains. Beyond simple overfitting, this behavior indicates a domain shift. Features that serve as strong indicators of malice in conventional, self-hosted environments, such as deeply nested subdomains and elevated Shannon entropy, are structural necessities of provider-managed FHP environments. Consequently, models trained on conventional datasets fail to generalize, classifying almost all FHP subdomains as phishing because the benign infrastructure structurally masquerades as self-hosted threats.

The range of benign precision varies from 0.02 for KNN-10 to 0.40 for MLP-3.5. Additionally, almost all models show a recall of 0.00 for benign FHP domains, with the only exceptions being the SVM models, which have a recall of 0.70, and KNN-10, which has a recall of 0.01. Consequently, the F1 scores for the *Benign* group remain low for all models tested. Half of the models have a score of 0.00 due to low recall, and the maximum score for the remaining models is 0.25. The metrics for the Phishing group remain consistently high, with precision ranging from 91% to 95% and recall up to 100%. However, this performance must be interpreted in the context of the 10:1 class imbalance in our FHP test dataset (41,568 phishing vs. 4,155 benign subdomains). In a highly skewed distribution, a classifier that defaults to predicting the majority class will yield deceptively high overall accuracy and phishing recall, while hiding complete failure on the minority benign class. This stark divergence

Model	Accuracy			Precision			Recall			F1 score		
	Phish-Hook	UCI	FHP	Phish-Hook	UCI	FHP	Phish-Hook	UCI	FHP	Phish-Hook	UCI	FHP
DT-2	91.06	76.46	90.88	91.12	79.07	84.02	91.06	76.46	90.88	91.07	73.33	86.58
DT-5	91.42	81.45	90.88	91.46	83.49	84.17	91.42	81.45	90.88	91.39	79.78	86.58
DT-10	89.39	81.83	90.87	89.44	83.81	83.74	89.39	81.83	90.87	89.40	80.25	86.58
SVM-0_03	91.62	79.80	62.41	91.68	82.21	88.14	91.62	79.80	62.41	91.58	77.67	70.38
SVM-0_3	91.29	79.80	62.41	91.37	82.21	88.14	91.29	79.80	62.41	91.25	77.67	70.38
SVM-1	91.39	79.80	62.41	91.45	82.21	88.14	91.39	79.80	62.41	91.35	77.67	70.38
KNN-1	86.41	75.05	90.84	86.40	75.18	83.58	86.41	75.05	90.84	86.37	75.12	86.57
KNN-3	86.38	79.24	90.86	86.40	78.83	83.50	86.38	79.24	90.86	86.32	78.72	86.57
KNN-10	87.23	82.94	88.66	87.23	84.03	82.72	87.23	82.94	88.66	87.19	81.83	85.54
MLP-3_5	90.08	81.93	90.91	90.16	83.85	86.29	90.08	81.93	90.91	90.03	80.39	86.59
MLP-5_10	89.55	81.99	90.88	89.94	84.10	83.84	89.55	81.99	90.88	89.44	80.41	86.58
MLP-1_100	89.06	82.02	90.88	89.63	84.09	84.02	89.06	82.02	90.88	88.92	80.45	86.58

Table 4: Prediction results for "Phish-Hook": original results reported by Fasllija et al; "UCI": modified models trained and evaluated on the UCI dataset; "FHP": modified models trained on the UCI dataset and evaluated on the FHP-only dataset.

highlights the danger of relying on aggregate metrics in provider-controlled environments without evaluating per-class metrics. SVM is the only model family that did not fail on the *Benign* group, but at the cost of the *Phishing* group, with metric values of 0.70 and 0.62. This can be attributed to the use of a linear kernel in the models, which may be poorly suited to the FHP feature distribution. F1 scores range from 0.75 for SVMs to 0.95 for DTs and MLPs.

Model	Precision		Recall		F1 score	
	Benign	Phishing	Benign	Phishing	Benign	Phishing
DT-2	0.15	0.91	0.00	1.00	0.00	0.95
DT-5	0.17	0.91	0.00	1.00	0.00	0.95
DT-10	0.12	0.91	0.00	1.00	0.00	0.95
SVM-0_03	0.15	0.95	0.70	0.62	0.25	0.75
SVM-0_3	0.15	0.95	0.70	0.62	0.25	0.75
SVM-1	0.15	0.95	0.70	0.62	0.25	0.75
KNN-1	0.10	0.91	0.00	1.00	0.00	0.95
KNN-3	0.09	0.91	0.00	1.00	0.00	0.95
KNN-10	0.02	0.91	0.01	0.97	0.01	0.94
MLP-3_5	0.40	0.91	0.00	1.00	0.00	0.95
MLP-5_10	0.13	0.91	0.00	1.00	0.00	0.95
MLP-1_100	0.15	0.91	0.00	1.00	0.00	0.95

Table 5: Phish-Hook results on FHP-only dataset for *Benign* and *Phishing* classes.

Because SVM alone retains non-trivial recall on both classes under the same class distribution, this divergence across model families indicates that the failure is not purely an artifact of class imbalance, but reflects real differences in how each algorithm separates the two classes given the FHP feature distribution.

This behavior suggests that several Phish-Hook features lose discriminative value in the FHP setting, especially when combined with imbalanced data and provider-controlled URL and certificate patterns. If deployed directly for FHP-hosted websites, the Phish-Hook model would likely flag many legitimate subdomains as phishing, making it unsuitable without substantial redesign.

6 Conclusions and Future Work

In this paper, we study how Free Hosting Platforms affect CT-based phishing detection. Our results show that this setting changes how X.509, CT, and URL features should be interpreted. Many features that appear useful for phishing detection instead capture properties of provider-managed infrastructure, showing that CT-derived information must be interpreted in relation to the respective deployment setting. Our evaluation of two existing CT-based detection frameworks further demonstrates the challenges of applying these methods directly to FHP-hosted websites: pattern-based clustering tends to identify provider-specific characteristics, while the learning-based approach struggles to distinguish benign and phishing FHP subdomains. These results do not imply that CT data are irrelevant for FHP-hosted phishing, but emphasize that further effort is required to address this issue.

Future work should develop CT-based phishing detection methods that account for the deployment and management of FHP-hosted websites. In particular, such methods should account for automated certificate issuance, shared or wildcard certificates, and platform-specific URL structure while also incorporating additional evidence such as page content, redirection behavior, provider metadata, and temporal changes in deployed websites. This direction is especially important, as modern AI tools reduce the effort required to generate convincing phishing emails and landing pages at scale. When combined with free hosting infrastructure, these tools can enable attackers to create a large number of more credible phishing websites with less manual effort, making deployment-aware detection increasingly important.

References

[1] Josh Aas. 2015. Why ninety-day lifetimes for certificates? https://letsencrypt.org/2015/11/09/why-90-days.html

[2] Josh Aas, Richard Barnes, Benton Case, Zakir Durumeric, Peter Eckersley, Alan Flores-López, J. Alex Halderman, Jacob Hoffman-Andrews, James Kasten, Eric Rescorla, Seth Schoen, and Brad Warren. 2019. Let's Encrypt: An Automated Certificate Authority to Encrypt the Entire Web. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS '19)*. Association for Computing Machinery, New York, NY, USA, 2473–2487. doi:10.1145/3319535.3363192

[3] Mashael AlSabah, Mohamed Nabeel, Yazan Boshmaf, and Euijin Choo. 2022. Content-Agnostic Detection of Phishing Domains using Certificate Transparency and Passive DNS. In *Proceedings of the 25th International Symposium on Research in Attacks, Intrusions and Defenses (RAID '22)*. Association for Computing Machinery, New York, NY, USA, 446–459. doi:10.1145/3545948.3545958

[4] Kholoud Althobaiti, Ghaidaa Rummani, and Kami Vaniea. 2019. A Review of Human- and Computer-Facing URL Phishing Features. In *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. 182–191. doi:10.1109/EuroSPW.2019.00027

[5] Nikita Blagov. 2020. State of the Certificate Transparency Ecosystem. (2020). doi:10.2313/NET-2020-11-1_09

[6] BuiltWith. 2026. Simple Website Builder Web Usage Distribution on the Entire Internet. https://trends.builtwith.com/cms/simple-website-builder/traffic/Entire-Internet

[7] Alex Chitu. 2012. Country-Specific Blogger URLs. https://googlesystem.blogspot.com/2012/01/country-specific-blogger-urls.html

[8] Google Chrome. 2026. Chrome Certificate Transparency Policy. https://googlechrome.github.io/CertificateTransparency/ct_policy.html

[9] Rachna Dhamija, J. D. Tygar, and Marti Hearst. 2006. Why phishing works. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06)*. Association for Computing Machinery, New York, NY, USA, 581–590. doi:10.1145/1124772.1124861

[10] Arthur Drichel, Vincent Drury, Justus von Brandt, and Ulrike Meyer. 2021. Finding Phish in a Haystack: A Pipeline for Phishing Classification on Certificate Transparency Logs. In *Proceedings of the 16th International Conference on Availability, Reliability and Security (ARES '21)*. Association for Computing Machinery, New York, NY, USA, 1–12. doi:10.1145/3465481.3470111

[11] Vincent Drury and Ulrike Meyer. 2019. Certified Phishing: Taking a Look at Public Key Certificates of Phishing Websites. 211–223. https://www.usenix.org/conference/soups2019/presentation/drury

[12] Robert Duncan. 2017. Let's Encrypt and Comodo issue thousands of certificates for phishing | Netcraft. https://www.netcraft.com/blog/lets-encrypt-and-comodo-issue-thousands-of-certificates-for-phishing

[13] Let's Encrypt. 2026. FAQ. https://letsencrypt.org/docs/faq/#will-let-s-encrypt-issue-organization-validation-ov-or-extended-validation-ev-certificates

[14] Edona Fasllija, Hasan Ferit Enişer, and Bernd Prünster. 2019. Phish-Hook: Detecting Phishing Certificates Using Certificate Transparency Logs. In *Security and Privacy in Communication Networks*, Songqing Chen, Kim-Kwang Raymond Choo, Xinwen Fu, Wenjing Lou, and Aziz Mohaisen (Eds.). Springer International Publishing, Cham, 320–334. doi:10.1007/978-3-030-37231-6_18

[15] Ronald A. Fisher. 1954. *Statistical Methods for Research Workers* (twelfth ed.). Oliver and Boyd.

[16] CA/Browser Forum. 2013. Information for the Public. https://cabforum.org/about/information/consumers/

[17] Mozilla Foundation. 2026. Public Suffix List. https://publicsuffix.org/

[18] Google. 2026. HTTPS encryption on the web – Google Transparency Report. https://transparencyreport.google.com/https/overview?hl=en

[19] Google. 2026. Overview | Safe Browsing APIs (v4). https://developers.google.com/safe-browsing/v4

[20] Scott Helme. 2017. Let's Encrypt are enabling the bad guys, and why they should. https://scotthelme.co.uk/lets-encrypt-are-enabling-the-bad-guys-and-why-they-should/

[21] Radek Hranický, Ondřej Ondryáš, Adam Horák, Petr Pouč, Kamil Jeřábek, Tomáš Ebert, and Jan Polišenský. 2024. A Dataset of Information (DNS, IP, WHOIS/RDAP, TLS, GeoIP) for a Large Corpus of Benign, Phishing, and Malware Domain Names 2024. doi:10.5281/ZENODO.13330073

[22] Dana Keeler. 2025. Certificate Transparency is now enforced in Firefox on desktop platforms starting with version 135. https://groups.google.com/a/mozilla.org/g/dev-security-policy/c/OagRKpVirsA/m/Q4c89XG-EAAJ

[23] Taeri Kim, Noseong Park, Jiwon Hong, and Sang-Wook Kim. 2022. Phishing URL Detection: A Network-based Approach Robust to Evasion. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. ACM, Los Angeles CA USA, 1769–1782. doi:10.1145/3548606.3560615

[24] Ben Laurie, Adam Langley, and Emilia Kasper. 2013. Certificate Transparency. RFC 6962. https://www.rfc-editor.org/rfc/rfc6962

[25] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński, and Wouter Joosen. 2019. Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation. In *Proceedings of the 26th Annual Network and Distributed System Security Symposium (NDSS 2019)*. doi:10.14722/ndss.2019.23386

[26] Sectigo Limited. 2026. crt.sh | Certificate Search. https://crt.sh/

[27] Rohit Lohar, Meenakshi Tripathi, and Hossein Shirazi. 2025. Free-Phish: detecting phishing websites hosted on free web hosting domains. *International Journal of Computers and Applications* 47, 2 (Feb. 2025), 162–168. doi:10.1080/1206212X.2024.2448835

[28] Winnie Ondara. 2024. 11 Best Free SSL Certificate Providers in 2024. https://www.tecmint.com/best-ssl-certificate-authorities/

[29] Stijn Pletinckx, Thanh-Dat Nguyen, Tobias Fiebig, Christopher Kruegel, and Giovanni Vigna. 2023. Certifiably vulnerable: Using certificate transparency logs for target reconnaissance. In *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*. IEEE, 817–831.

[30] Arvind Prasad and Shalini Chandra. 2024. PhiUSIIL: A diverse security profile empowered phishing URL detection framework based on similarity index and incremental learning. *Computers & Security* 136 (Jan. 2024), 103545. doi:10.1016/j.cose.2023.103545

[31] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. 1992. *Numerical Recipes in C: The Art of Scientific Computing* (second ed.). Cambridge University Press, Cambridge.

[32] Sayak Saha Roy, Unique Karanjit, and Shirin Nilizadeh. 2023. Phishing in the Free Waters: A Study of Phishing Attacks Created using Free Website Building Services. In *Proceedings of the 2023 ACM on Internet Measurement Conference*. ACM, Montreal QC Canada, 268–281. doi:10.1145/3618257.3624812

[33] Yuji Sakurai, Takuya Watanabe, Tetsuya Okuda, Mitsuaki Akiyama, and Tatsuya Mori. 2020. Discovering HTTPSified Phishing Websites Using the TLS Certificates Footprints. In *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. 522–531. doi:10.1109/EuroSPW51379.2020.00077

[34] scikit learn. 2026. train_test_split. https://scikit-learn/stable/modules/generated/sklearn.model_selection.train_test_split.html

[35] Statista. 2024. Number of global phishing attacks 2024. https://www.statista.com/statistics/266155/number-of-phishing-attacks-worldwide/

[36] Apple Support. 2026. Apple's Certificate Transparency policy. https://support.apple.com/en-us/103214

[37] Chris Thompson, Mustafa Emre Acer, Serena Chen, Joe DeBlasio, Emily Stark, David Adrian, and Chrome Security Team. 2025. HTTPS by default. https://security.googleblog.com/2025/10/https-by-default.html

[38] Tranco. 2026. Information on the Tranco list with ID G23YK. https://tranco-list.eu/list/G23YK/50000

[39] Tranco. 2026. Information on the Tranco list with ID Q7NK4. https://tranco-list.eu/list/Q7NK4

[40] Tranco. 2026. Information on the Tranco list with ID WL9Y9. https://tranco-list.eu/list/WL9Y9

[41] Wix. 2026. About Your Site URL Prefix | Help Center | Wix.com. https://support.wix.com/en/article/about-your-site-url-prefix

[42] Shaked Zychlinski. 2025. shakedzy/dython: Version 0.7.12. doi:10.5281/zenodo.17956912

A Ethical Considerations

This paper does not involve any ethical issues.

B List of FHP Apex Domains

Weebly: weebly.com, weeblysite.com
000webhost[1]: 000webhostapp.com, 000webhost.com
Blogger: blogspot.*[2]
Wix: wixsite.com, wixstudio.io
GitHub Pages: github.io
Firebase: web.app, firebaseapp.com
SquareUp: square.site
WordPress.com: wordpress.com, wpcomstaging.com
Microsoft SharePoint: onmicrosoft.com, sharepoint.com
Yola: yolasite.com
GoDaddy: godaddysites.com
MailChimp: mailchimpsites.com, mailchi.mp
Glitch[3]: glitch.me, glitch.com, gomix.com, gomix.me
Webnode: webnode.com
Heroku: herokuapp.com, herokudns.com
Website.com: website1.me, website2.me, website3.me
Netlify: netlify.app
HPage: hpage.com
Infinityfree: infinityfreeapp.com, gt.tc, kesug.com, lovestoblog.com, rf.gd, free.nf, 42web.io, wuaze.com, ct.ws, infinityfree.me, page.gd, gamer.gd, fwh.is, xo.je, great-site.net
ByetHost: 10001mb.com, 22web.org, 2kool4u.net, 66ghz.com, a0001.net, fast-page.org, html-5.me, humorme.info, iblogger.org, is-best.net, is-great.net, is-great.org, joomlahost.org, likesyou.org, loveslife.biz, my-board.org, my-style.in, mydiscussion.net, nichesite.org, social-networking.me, synergize.co, talk4fun.net, totalh.net, web1337.net
HyperPHP: hyperphp.com
Awardspace: atwebpages.com, getenjoyment.net, medianewsonline.com, myartsonline.com, mygamesonline.org, mypressonline.com, mywebcommunity.org, onlinewebshop.net, scienceontheweb.net, sportsontheweb.net
Freehostia: freehostia.com
Free Web Hosting Area: orgfree.com, 6te.net, ueuo.com, eu5.org, noads.biz, coolpage.biz, freeoda.com, freevar.com, freetzi.com, xp3.biz
GoogieHost: googiehost.com, c1.is
Dreamhost: dreamhosters.com
hosting.com[4]: a2hosted.com
Vercel: vercel.app
Cloudflare Pages: pages.dev, workers.dev
Cloudflare: r2.dev[5]
Wasmer: wasmer.app
GitLab Pages: gitlab.io
Codeberg Pages: codeberg.page
Statichost: statichost.eu, statichost.page
EasyWP: ewp.live
Buffer: start.page
Replit: repl.co, replit.app
EU.org: eu.org
Miarroba: webcindario.com
uKit: ukit.me
Pantheon: pantheonsite.io
Surge: surge.sh
Webflow: webflow.io
Bluehost: bluehost.com, my.bluehost.com, mybluehost.me
Microsoft Azure[6]: azurewebsites.net
Oracle Cloud[7]: homeunix.org
web.com: wcomhost.com
W3Schools Spaces: w3spaces.com
Studio: studio.site
GitBook: gitbook.io
BitBucket Cloud: bitbucket.io
Hostinger[8]: builder-preview.com

C Mixed-type Correlation Matrix

Figure 5 reports the full matrix discussed in Section 4.4.

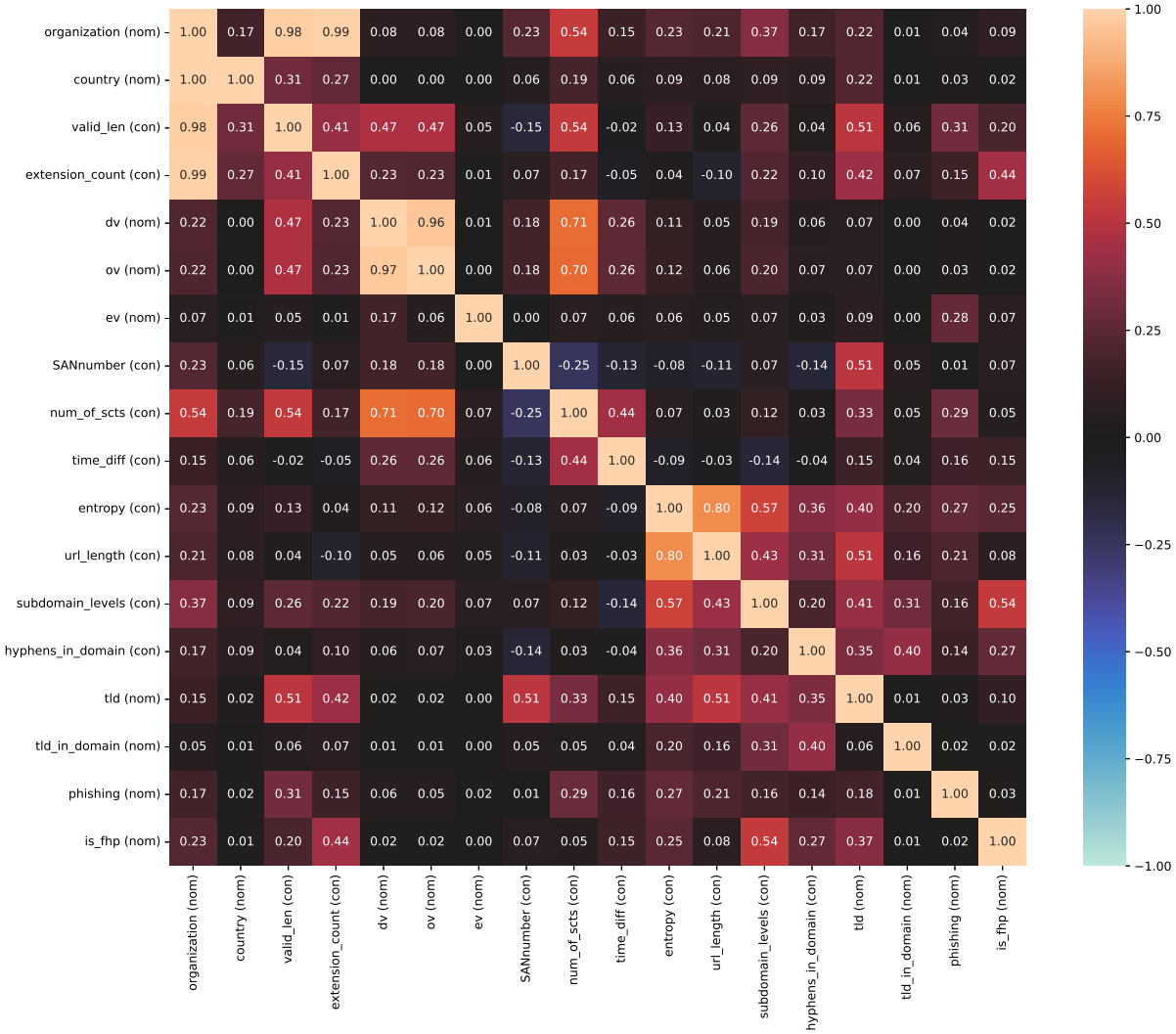


Figure 5: Mixed-type correlation matrix (Pearson's r, correlation ratio η, Theil's U) of all features and two target labels. `is_fhp` shows stronger associations than `phishing`, with `subdomain_levels` being the strongest ($\eta = 0.54$). Theil's U is asymmetric; read row-to-column.