Expert Systems With Applications 207 (2022) 118010

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Phishing websites detection using a novel multipurpose dataset and web technologies features

Manuel Sánchez-Paniagua [*], Eduardo Fidalgo, Enrique Alegre, Rocío Alaiz-Rodríguez

Department of Electrical, Systems and Automatics Engineering, University of León, Spain
Researcher at INCIBE (Spanish National Institute of Cybersecurity), León, Spain

A R T I C L E   I N F O

Keywords:
Phishing detection
Phishing dataset
Web technologies
Machine learning
Login

A B S T R A C T

Phishing attacks are one of the most challenging social engineering cyberattacks due to the large amount of entities involved in online transactions and services. In these attacks, criminals deceive users to hijack their credentials or sensitive data through a login form which replicates the original website and submits the data to a malicious server. Many anti-phishing techniques have been developed in recent years, using different resource such as the URL and HTML code from legitimate index websites and phishing ones. These techniques have some limitations when predicting legitimate login websites, since, usually, no login forms are present in the legitimate class used for training the proposed model. Hence, in this work we present a methodology for phishing website detection in real scenarios, which uses URL, HTML, and web technology features. Since there is not any updated and multipurpose dataset for this task, we crafted the Phishing Index Login Websites Dataset (PILWD), an offline phishing dataset composed of 134,000 verified samples, that offers to researchers a wide variety of data to test and compare their approaches. Since approximately three-quarters of collected phishing samples request the introduction of credentials, we decided to crawl legitimate login websites to match the phishing standpoint. The developed approach is independent of third party services and the method relies on a new set of features used for the very first time in this problem, some of them extracted from the web technologies used by the on each specific website. Experimental results show that phishing websites can be detected with 97.95% accuracy using a LightGBM classifier and the complete set of the 54 features selected, when it was evaluated on PILWD dataset.

1. Introduction

Phishing is one of the main social engineering attacks, where attackers create a fake website to deceive users and obtain their passwords or other sensitive data regarding a specific brand or web service (Mohammad et al., 2015b). Phishing may occur through different attack vectors, including emails, instant messaging, Short Message Service (SMS) and many others. Moreover, one of the most important vectors are the web pages (Chiew, Yong et al., 2018; Gupta et al., 2018), where attackers emulate the website of a well-known company to obtain user information, usually via login or a sign-up form. Since many phishing attack vectors contain a Uniform Resource Locator (URL) pointing to a website, we can identify websites as the final endpoint of the attacks. Therefore, in this work, we focus on detecting those deceitful websites.

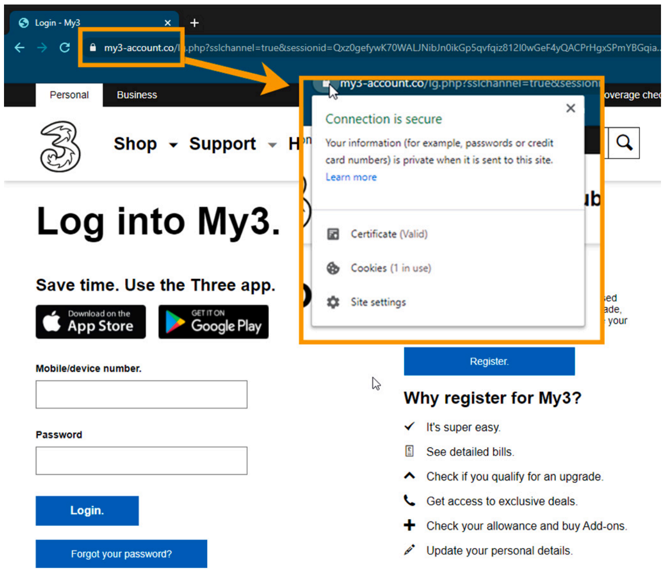
The Anti-Phishing Working Group (APWG) detected up to 611,877 unique phishing websites during the last quarter of 2020. Financial institutions were the main targets of those attacks (24.9%) followed by social media (23.6%), SAAS (Software As A Service) and webmail services (19.6%) and payment platforms (8.5%) (Anti-Phishing Working Group, 2021). Phishing campaigns have a noticeable impact since the private data revealed derives into economical losses that affect either way to corporations, with more than 411 million US$ (Bose & Leung, 2014), and to users, with millions of US$ (Jain & Gupta, 2017; Shaikh et al., 2016).

Also, the percentage of phishing websites using Secure Sockets Layer protocol (SSL) has increased up to 83% (Anti-Phishing Working Group, 2021), which means that they display the green padlock on the browser navigation bar. This is a misleading security indicator for users that

* Corresponding author at: Department of Electrical, Systems and Automatics Engineering, University of León, Spain.
E-mail addresses: msancp@unileon.es (M. Sánchez-Paniagua), eduardo.fidalgo@unileon.es (E. Fidalgo), enrique.alegre@unileon.es (E. Alegre), rocio.alaiz@unileon.es (R. Alaiz-Rodríguez).
URL: https://gvis.unileon.es/ (E. Alegre).

[1] https://safebrowsing.google.com/.
[2] https://www.phishtank.com/.
[3] https://bit.ly/2OJDYBS.

https://doi.org/10.1016/j.eswa.2022.118010
Received 25 October 2021; Received in revised form 15 June 2022; Accepted 27 June 2022
Available online 30 June 2022


Login - My3
my3-account.co/lg.php?sslchannel=true&sessionid=Qxz0gefywK70WALJNibJn0ikGp5qvfqiz812l0wGeF4yQACPrHgxSPmYBGqia...
Personal
Business
Shop
Support
Log into My3.
Save time. Use the Three app.
Download on the App Store
GET IT ON Google Play
Mobile/device number.
Password
Login.
Forgot your password?
Register.
Why register for My3?
It's super easy.
See detailed bills.
Check if you qualify for an upgrade.
Get access to exclusive deals.
Check your allowance and buy Add-ons.
Update your personal details.
Connection is secure
Learn more
Certificate (Valid)
Cookies (1 in use)
Site settings

Your information (for example, passwords or credit card numbers) is private when it is sent to this site.

Fig. 1. A new phishing website not yet registered on blocklists, displays the green padlock and shows no previous warning to the user.

commonly identify this icon with a secure website. This statement also compromises the effectiveness of previous phishing works which used HTTPS as a key feature in their approaches (Adebowale et al., 2019; Li et al., 2019; Rao et al., 2020).

Current defences for users against phishing websites are implemented on browsers and operating systems like Windows. The most common are Google SafeBrowsing[1] for Google Chrome, PhishTank[2] for Firefox and SmartScreen[3] for Windows. These systems rely on blocklists where phishing URLs stack up when they are reported. Therefore, if a user visits a recent phishing website that has not been reported yet, it will not be detected. Then, the user will enter into the fake website with no previous warning, exposing credentials to the attackers if they are introduced. Fig. 1 displays an example with a recent phishing encounter not listed on those blocklists.

To address this issue, researchers have developed different approaches to improve phishing detection, including list-based techniques (Hong et al., 2020; Prakash et al., 2010) and also artificial intelligence-based approaches, where they used different inputs such as URLs (Abutair & Belghith, 2017; Marchal et al., 2016; Moghimi & Varjani, 2016; Rao et al., 2020; Sahingoz et al., 2019), the source code or HTML (Adebowale et al., 2019; Li et al., 2019; Rao & Pais, 2019b) and images (Bozkir & Aydos, 2020; Chiew et al., 2015; Dalgic et al., 2018; Dunlop et al., 2010).

Some previous works (Adebowale et al., 2019; Sadique et al., 2020; Xiang et al., 2011), have also used third-party services like Google or WHOIS to include features such as the website ranking on Google Browsing Index or the age of the domain registry. However, these implementations have –at least– four drawbacks. The first one is their reliance on services that could be offline or slow on their response time, setting a boundary on the real-time application. Second, some features have flaws for a specific group of websites, like the case of the domain age feature, which may incorrectly classify legitimate websites with a short life. A third drawback is that these techniques may fail to detect phishing attacks hosted on compromised legitimate domains (Rao & Pais, 2017). Finally, WHOIS features have some additional limitations due to the rising of the WHOIS Privacy service, which hides this information and denies feature extraction for works relying on that data.

One of the most relevant concerns about the state-of-the-art approaches for phishing detection is the need for a standard, updated and representative dataset for researchers to compare their works. The comparison of phishing detection systems entails implementing other methods and the usage of the same dataset, otherwise, comparisons are relative and might obtain misleading conclusions. Also, most of the studies presented tailored datasets to meet their approach, i.e., a work using URL features usually presents a URL dataset (Sahingoz et al., 2019). The first of these premises leads to a time-consuming task due to the complexity of current works. The second one implies using the same approach (URL, HTML, images or specific data) since a work using HTML code cannot use a dataset built with only URLs. Therefore, fair comparisons are limited to use the same type of data.

In our previous work (Sanchez-Paniagua et al., 2022), we identified the problem of using outdated datasets for new model developments. Constantly evolving phishing techniques degrade the performance of models, which can no longer maintain the generalization performance estimated at the learning stage. In order to ensure that new models perform correctly on current environments, researchers should use updated datasets. Phishing campaigns last an average of 21 h (Oest et al., 2020), while URLs are stored for long periods of time. For this reason, phishing websites should be collected as they appear to obtain the data before they are seized or close. As far as we are aware, only one known work is focused on building an offline phishing dataset (Chiew, Chang et al., 2018). However, we found some issues regarding the legitimate URLs and the screenshots, which is discussed later in Section 2.

To address the drawbacks of published datasets and the lack of offline datasets with raw data, in this work we introduce and make publicly available[4] PILWD-134K (Phishing Index Login Websites Dataset). This dataset comprises 134,000 verified samples collected from August 2019 to September 2020. PILWD-134K covers six different kinds of raw resources, including URLs, HTML code, screenshots, a copy of the website files, web technologies analysis and extra metadata regarding the phishing reports[5] This new dataset can be used both as a large sample of real-life websites and as a standard corpus for evaluating the results using different phishing detection techniques over the same websites, something that was not available until the creation of the introduced dataset. Jointly with PILWD-134K, we provide baseline results using traditional Machine Learning descriptors and a brief analysis of the dataset itself.

Additionally to other approaches that commonly focus exclusively on index pages, we propose a new approach for the phishing detection task, trying to simulate a real-world situation when a user finds a phishing website. A review of collected phishing websites shows that most of them (77%) requested credentials through a login form. However, most datasets used in the literature do not include the same amount of login forms in their legitimate class, which causes bias in the classification due to the significant differences between both types of web pages. El Aassal et al. (2020) included a significant number of login websites to avoid this bias in the legitimate class, However, the proportion of login forms was still distant between both classes. We address this issue by crawling and creating two legitimate subsets, one is built up with homepages, and the other contains login websites. Then, we identify the pages with a login form to merge them and build a final legitimate class with the same number of forms as the phishing one.

Finally, we proposed a phishing website detection method using and comparing legacy features used in previous works (Li et al., 2019;

[1] https://safebrowsing.google.com/.
[2] https://www.phishtank.com/.
[3] https://bit.ly/2OJDYBS.

[4] Dataset will be available on our website https://gvis.unileon.es/dataset/pilwd-134k/ once the paper is accepted.
[5] Phishtank's reports contain information related to the date, the brand affected or the IP address.

Sahingoz et al., 2019) and the novel features proposed in this paper, such as the technology analysis and new HTML features. These features were designed to improve detection performance, reliability and resilience against upcoming phishing attacks and bypass methods, such as clones and phishing kits (Bijmans et al., 2021). We use a LightGBM (Light Gradient Boosting Machine) model to build an independent third-party system capable of detecting phishing websites among legitimate login web pages.

The main contributions of this work are summarized in the following items:

- Novel hand-crafted features: Attackers take advantage from published phishing countermeasures to modify their websites and bypass detection systems. We propose 27 novel features with relevant information to achieve high performance for today's phishing detection tasks. We include a novel type of feature, web technology analysis.
- Real-case phishing detection scenario. Almost half of the collected phishing websites have login forms for users to input their credentials. For this reason, we focus on crawling legitimate login forms. This way, we reproduce a challenging real-case scenario that is representative to achieve the actual objective: Detecting whether a website with a login form is legitimate or not.
- A publicly available dataset, with offline legitimate and phishing samples. We present the Phishing Index Login Websites Dataset (PILWD-134K), a balanced dataset with a high ratio of login forms in both classes, that includes 134,000 verified samples and can be used in a wide variety of phishing detection approaches. Each sample contains raw data such as the URL, HTML, screenshots, technology analysis and an offline version of the website. Since the samples are taken into a verification process we also include the remaining dismissed samples for researchers to implement their methodology.

This paper is structured as follows. Section 2 presents a review of the literature and related work. Section 3 introduces the dataset, its structure, properties and the quality filters. Section 4 explains the methodology, the features proposed and the metrics used. Section 5 introduces the experiments and the obtained results. Section 6 contains the conclusion along with the limitations and future work.

2. Related work

In previous years, researchers have developed phishing detection systems using different techniques and data sources. Next, we review these approaches together with the datasets presented.

2.1. Phishing website detection

Zhang et al. (2007) implemented CANTINA, a phishing detection system that uses TF-IDF (Term Frequency - Inverse Document Frequency) technique to extract five signature words from a website and use them into Google Search engine. If the analysed website domain was within the first 30 results, the website was classified as legitimate. In the following years, Xiang et al. (2011) presented CANTINA+, an enhanced version which added two filters and 15 features related to the URL, the HTML and web (PageRank, copyright and WHOIS). The proposed system achieved 92% accuracy on an 8000 sample dataset using a Bayesian Network.

Influenced by CANTINA works, He et al. (2011) selected 12 features extracted from the URL, the requests carried out by the website and the results of a search engine. Together with an SVM (Support Vector Machine), these features led to a 97% accuracy using a small dataset with 525 samples.

Gowtham and Krishnamurthi (2014) implemented a system with three detection stages where the first two were intended to reduce calculation time, and the last one performed the actual classification. The first stage consisted of a pre-approved site identifier, which checked the website with an allowlist maintained by the user. The second stage was a Login Form finder where if no login was detected, the website is directly classified as legitimate. Finally, they extracted 15 features and achieved a 99.62% accuracy on SVM with a 2464 samples dataset.

Marchal et al. (2014a) proposed a real-time URL phishing rating system called PhishScore which focuses on intra-URL words relations. For this task, they used 12 features; the first six used the Jaccard index to calculate the similarity between different sets of words of the URL. The last six were used to calculate the popularity of the URL using Google, Yahoo and Alexa Website ranking. On a 96.018 URL samples dataset, they achieved a 95.22% accuracy using an RF (Random Forest) classifier.

Moghimi and Varjani (2016) implemented two sets of features, the first one contained nine legacy features from previous phishing detection works, and they were related to the URL and detecting a set of keywords within the domain name, the path and the query of the URL. The second set of eight proposed features included typosquatting features between the URL and the source of the elements loaded in the website (CSS, JavaScript, images and links) and the protocol used to load those resources. Combining both sets of features, they achieved an accuracy of 98.65% on a 1707 e-banking phishing dataset.

Rao and Pais (2017) presented *FeedPhish*, an application to detect phishing based on the response to fake credentials submission. To simulate the user input in the login form, the authors used Selenium WebDriver and created three modules. First, the *LoginCheck* module verifies if there is a login form in the website, then *FeedFakeCredentials* introduces a random account and credential and *HeuristicsCheck* module analyses the content of the response. Finally, *TargetDomainCheck* compute the identity of the domain based on the anchor links. Results showed that the proposed heuristics obtained 96.38% accuracy on a 2342 samples dataset. In subsequent studies, Rao and Pais (Rao & Pais, 2019a) proposed 16 legacy and novel features, which were divided into three sets: URL obfuscation, third-party-based and hyperlink-based. Combined with an RF classifier, they obtained 99.31% accuracy on a 3526 samples dataset.

Tan et al. (2018) compared five groups of features depending on their target: URL obfuscation on the HTML, domain obfuscation in the URL, HTML content, symbol exploit and web page URL properties. Using a C4.5 classifier, they found that the URL obfuscation on the HTML obtained the best performance with an accuracy of 95.97%, followed by the symbol exploitation with 82.03%. Those tests were performed on a 10,000 samples dataset.

Sahingoz et al. (2019) proposed two sets of features for phishing detection through the URL: First, 39 NLP (Natural Language Processing) features and second, a set of 102 word features. Using WEKA's RF and only the NLP set of features, they obtained 97.98% accuracy on a 73,575 phishing URL dataset collected on one of their previous work (Buber et al., 2018).

Li et al. (2019) proposed a stacking model using 20 features extracted from the URL and the HTML code along with the combination of GBDT, XGBoost and lightGBM models. In addition, they proposed novel features, like brand detection, consistency in the title and URL domain, and a string embedding extracted from the HTML using Word2Vec. The first implementation with 20 features reached 97.11% accuracy on their 50,000 samples dataset. Furthermore, they explored visual features using website screenshots with a small CNN (Convolutional Neural Network), improving their model accuracy to 98.60%.

Adebowale et al. (2019) proposed an ANFIS (Adaptative Neuro-Fuzzy Inference System) using three sets of features: (i) properties from website images, (ii) frame features, which aim to identify website behaviour in terms of redirections, pop-up windows and right-click disabled among others; (iii) and a text subset related mainly with the URL and third-party services like Google Page Rank and WHOIS. They obtained a 98.30% accuracy on Mohammad et al. (2015a) dataset for the frame and text sets along with a self-collected image dataset.

Ding et al. (2019) proposed an SHLR (Search & Heuristic Rule & Logistic Regression) method to filter and detect phishing websites. First, they search the website title into Baidu's top 10 results. If a domain is not found, URL heuristics are evaluated to determine if it is phishing or not. If not, they extract a total of 37 features to be evaluated with a logistic regression algorithm. They obtained 98.9% accuracy on their 20,384 samples dataset.

Rao et al. (2020) proposed CatchPhish, a phishing URL detection method that combines TF-IDF and 35 handcrafted features, 16 of which are focused on the hostname, and the rest use the path and the base URL. First, they created a list of keywords that had a high frequency on the phishing subset. Then, they used it to count the number of encounters within the hostname and in the URL. Using an RF classifier, they obtained 94.26% accuracy on their dataset, 98.25% on Sahingoz et al. URL dataset (Sahingoz et al., 2019) and a 97.49% on a URL dataset by Marchal et al. (2014b).

Aljofey et al. (2020) presented an RCNN model to classify phishing URLs. They used the URL as input for a tokenizer and then used a one-hot encoding to represent the URL as a matrix at a character level. They used a 310,642 URL dataset to feed the RCNN model and obtained a 95.02% accuracy using the aforementioned character embedding level features.

Yang et al. (2021) presented an Extreme Learning Machine (ELM) model along with three different types of features: (i) Surface features uses the information from the URL. Specifically, they used 12 URL handcrafted features and 4 Domain Name System (DNS) features related to the registration date and the DNS records for the target domain; (ii) 28 Topological features related to the structure of the website. Finally, 12 deep features (iii) were obtained from the text, image and overall similarity. Combining these features and the ELM classifier, they obtained 97.5% accuracy on a 60,000 samples dataset.

Gupta et al. (2021) proposed a real-time system focused on running on constrained devices. They designed nine lexical features from the URL, including length-related features and counting several elements such as digits, dots and symbols. They obtained 99.57% accuracy using RF classifier on the ISCXURL-2016 dataset which allocates 19,964 URLs.

Sadique et al. (2020) presented a framework for real-time phishing detection using four sets of URL features: (i) Lexical features based on the number of characters, dots and symbols found in different parts of the URL, (ii) host-based features related with the server and the IP where the website is hosted, (iii) WHOIS features related to the days after the registration date and days before the expiration date, and (iv) GeoIP-based features like the Autonomous System Number (ASN), the country or the city where the website is hosted. A total of 142 individual features were evaluated using 98,000 samples from Phishtank, where legitimate samples are also picked from false positives collected at PhishTank. They obtained a 90.51% accuracy on an RF classifier using the proposed descriptors.

2.2. Dataset manufacture

Phishing detection is a relevant area of research, however, the lack of large representative datasets has been an obstacle for uncovering the real problems and developing efficient solutions.

Recently, Chiew, Chang et al. (2018) worked on building an offline standard dataset for researchers to test their work. The dataset is composed of 30,000 samples, 15,000 for each class, and they contain the URL, HTML files, a Screenshot, an offline copy of the website retrieved with WGET and WHOIS information. APWG reports were used to match the different phishing categories with the represented ones on the dataset. They also used Alexa, DMOZ BOTZ and Phishtank as the source for the URLs and the domain names. Then, they built a web crawler to visit the websites and retrieve the information indicated. After a close lookup on this dataset, we found that many screenshots were repeated or blank on both classes. Also, the URLs collected for the legitimate class were only domain names and TLDs, with no subdomain, protocol or path. For this reason, we consider that researchers may find obstacles to use this dataset for testing their methodology.

3. Phishing index login websites dataset

As seen in the previous section, the small set of publicly accessible datasets forces researchers to build their own tailored dataset so it meets their specific requirements. To the best of our knowledge, there are no publicly available datasets that comprise multiple data from the same website: complete URL, HTML, screenshots and website resources. This makes comparison and benchmarking tasks unfeasible when used resources are different between approaches.

In this section, we explain the process of building the Phishing Index Login Websites Dataset (PILWD), with more than 134,000 verified samples collected from August 2019 to September 2020. It includes raw data from legitimate and phishing websites, allowing researchers to extract their features and develop their phishing detection systems. It covers a wide variety of raw data: URLs, source code, screenshots, technologies analysis and an offline replica of the website. In addition, other metadata is also included for each sample: time of recollection, filters, and PhishTank information in case of the phishing class. To collect the samples, a web crawler was developed using Python3 and Selenium WebDriver to visit the legitimate domains and the reported phishing URLs. By loading the website, our crawler has various advantages in terms of quality:

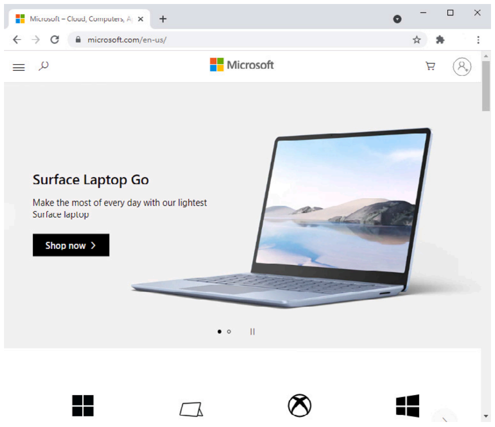
- The website was rendered to simulate the user access. Once the site was fully loaded, we extracted the content as it is, including browser screenshots. This way, we can include the same information displayed to users.
- Loading the website in the web driver avoided web obfuscation. Some phishers encrypt or encode the original HTML code and embed it into an empty website. In order to render the HTML, phishers include a JavaScript file used to decrypt or decode the payload. This obfuscation method affects recollection procedures that do not execute the website's JavaScript.
- URL resolution. Legitimate datasets are usually composed of a list of most visited domains. Nevertheless, once our crawler lands in the website, the complete and final URL is saved. In the case of phishing samples, phishers try to confuse users with re-directions. By visiting the website, we could collect the URL corresponding to the final phishing site, storing it along with the original reported one.

3.1. Legitimate class

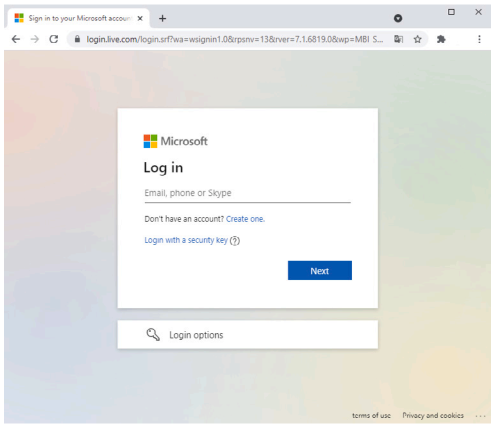
State-of-the-art papers generally use different sources for collecting legitimate websites. They choose the most visited domains from Alexa Topsites,[6] DMOZ,[7] and many others. We used Quantcast Top Sites[8] and The Majestic Million[9] as a source for the crawler. These services provide the domains with the most referring subnets. Therefore, we can assume that it is hard for a phishing website to enter these lists since there are not enough websites pointing to them. From the first source, we generated a list with 150,000 domains ordered by the number of visitors and then appended the top million domains from Majestic. After merging both lists, we removed the repeated domains.

Current phishing detection works (Aljofey et al., 2020; Li et al., 2019; Rao et al., 2020) use most visited domains without further crawling into the domain, i.e., they use legitimate homepages (Fig. 2(a)) and phishing samples (Fig. 2(c)) to feed their algorithms. Nevertheless, phishing attacks focus on stealing credentials, and most homepages do not have login forms. Hence, instead of collecting only the legitimate homepage, we decided to get closer to the real-case scenario (Sanchez-Paniagua et al., 2022), where the user doubts whether a login page

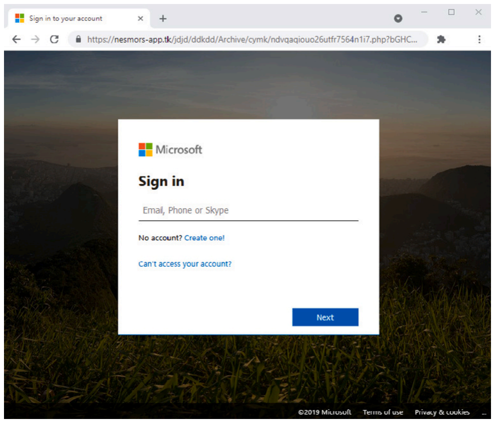
[6] https://www.alexa.com/topsites.
[7] https://dmoz-odp.org/.
[8] https://www.quantcast.com/top-sites/.
[9] https://majestic.com/reports/majestic-million.


Microsoft – Cloud, Computers, A
microsoft.com/en-us/
Microsoft
Surface Laptop Go
Make the most of every day with our lightest Surface laptop
Shop now

(a) Legitimate homepage


Sign in to your Microsoft accoun
login.live.com/login.srf?wa=wsignin1.0&rpsnv=13&rver=7.1.6819.0&wp=MBI S...
Microsoft
Log in
Email, phone or Skype
Don't have an account? Create one.
Login with a security key
Next
Login options
terms of use
Privacy and cookies

(b) Legitimate login page



(c) Phishing web page

Fig. 2. Pages collected for phishing detection. Legitimate homepage samples (a) differs from the Phishing ones (c) while Legitimate login pages (b) has the same structure in the URL and HTML look and feel.

(Fig. 2(b)) is legitimate or phishing (Fig. 2(c)). Fig. 2 depicts the similarity between the legitimate login page and the phishing one, specifically in the URL and the structure of the rendered HTML. In the case of the legitimate homepage, differences are remarkable in both items. Due to the similarity, we believe that this scenario is more challenging for artificial intelligence algorithms.

For creating the homepage subset, the crawler visited each domain from the list and retrieved the information on the landing page. For the login subset, the crawler looked for the sign-in form of the service. If it was not on the main page, it searched for any login link on the website until it reached the objective. Unfortunately, not all websites had login forms; therefore, they were not included in this subset.

In addition to this process, most websites had a large banner containing the cookies consent, which could occlude information of the captured screenshot. To enhance dataset quality, we improved the crawler allowing it to accept the cookies automatically and remove the banner. Since there are many types of cookies banners, we focused on the most common ones.

3.2. Phishing dataset

As the majority of other research works (Chiew, Chang et al., 2018; Ding et al., 2019; Li et al., 2019; Sahingoz et al., 2019), we used Phishtank as the source for all the phishing URLs. Chiew, Yong et al. (2018) used the *online* and *verified* indicators from Phishtank to select 15,000 phishing URLs to visit. However, the *online* indicator only ensures that the server is online, but the website content might be seized or removed, which could bias the collected data. Since phishing pages last an average of 21 h (Oest et al., 2020), we obtained all the reported websites as soon as they were reported, so the crawler hits the website on time before the attacker or the hosting service takes the web page down.

Phishtank verifies reported websites using user votes, increasing the chances of those samples being true phishing. As samples were collected, we retrieved hourly JSON reports. Moreover, we requested the API to verify the collected samples and retrieve some extra information from the JSON report, including the IP address, the affected brand, the server location and the announcing network. Unlike the legitimate dataset collection, we did not interact with the website to find the login form or remove the cookies banner. As a consequence, we collected only the reported land page. Samples on Phishtank marked as *spam*, or *disabled from voting* were omitted and not collected since the vast majority were forums for malware download or websites that were offline at the time of submission.

3.3. Structure

To make our dataset valuable for different phishing detection methods, we collected the following information:

url.txt: Contains the complete final URL of the visited website, with the Fully Qualified Domain Name (FQDN), the protocol and the path, if it exists. We collected the URL displayed on the browser once we reached the website since the reported URLs or the legitimate domain list are not entirely representative of the sample. Those reported URLs could have redirections or lack information like the path or the subdomain.

html_content.txt: Contains the source code of the website, including in the same file, the HTML, the JavaScript code and the CSS.

Screenshots: We take two screenshots of the website in high resolution (1848 pixels width and 911 pixels height), one from the top of the website and another from the bottom. These locations usually contain the brand logo, which can be helpful for phishing detection (Bozkir & Aydos, 2020; Chiew et al., 2015).

WGET: We retrieved the website with all the resources needed to recreate it offline. We used WGET[10] for this task with the following parameters: *–no-check-certificate* so it was collected even with invalid certificate, *–user-agent* was set to *"Mozilla/5.0 (Windows NT 10.0; WOW64)"* to simulate as better as possible the request from a browser, *-p* to retrieve the page requisites to load it offline (including images, sounds and other resources) and last, *-T 60 –tries = 3* to avoid overwhelming loading times.

tech.json: With a local instance of Wappalyzer,[11] we listed the technologies used by the website, and saved them with JSON format. These

[10] https://www.gnu.org/software/wget/.
[11] https://www.wappalyzer.com/.

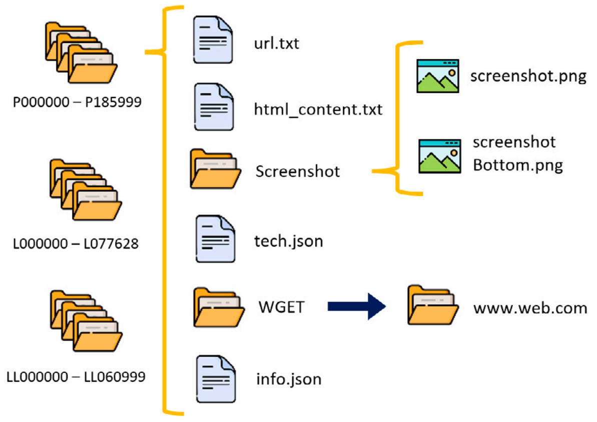


Fig. 3. Dataset structure. It contains the three subsets, labelled on the folder name with L, LL and P for Legitimate, Legitimate Login and Phishing respectively. The inner sample structure is uniform for every sample, no matter the class.

are detected due to the fingerprint generated by the technologies into the HTML code. The output file also includes the different categories of detected technologies such as E-commerce, databases or analytics.

info.json: It contains general information about the sample, including the collection date, the URL, the ID and if it was compliant with any of the filters applied. Moreover, we provide extra information for the phishing samples, including the original URL reported, an attribute indicating whether Phishtank verified the sample or not and when it was verified. Finally, the attribute "banned" described in detail later in this section.

To store the data above, we established a simple file structure represented in Fig. 3. We distributed the collected samples in different directories, labelled using the class as a prefix and the sample *id* within that class. For instance, the directory named as *P000000* stores the first sample corresponding to the phishing class, *P000001* stores the second sample and so on. Prefixes *L* and *LL* stands for Legitimate and Legitimate Login, respectively, on the labelling system. The distribution of the samples is described in the following sections.

Furthermore, we created a metadata file with relevant information for every sample. This metadata file contains the *id*, the URL, the filtering information for all subsets, and Phishtank verification status in the phishing subset. This way, researchers can fetch samples with a specific condition without iterating over the whole dataset. It is also worth noticing that the collected information is independent of external services, therefore, we did not include data from WHOIS or other services.

3.4. Filtering

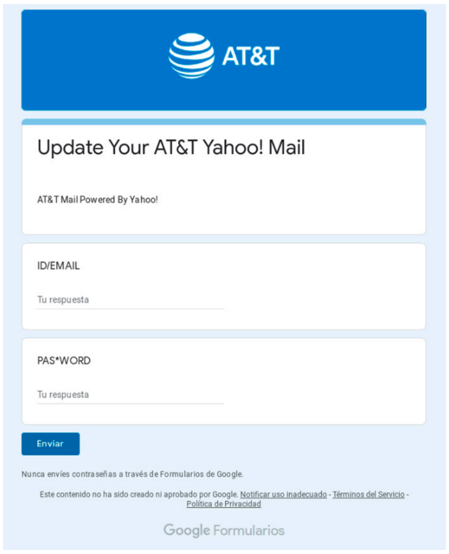
After a deep analysis of the dataset, we developed a three-filter system (F_1, F_2 and F_3) to increase the dataset quality. Information regarding any of the filters is provided in the metadata file, so authors can identify which samples are affected by each filter. Since no sample is deleted, researchers can develop their filtering methods or reach further conclusions. We proposed three filters:

3.4.1. Filter 1: Repeated legitimate websites in phishing reports

We visited all the phishing URLs as soon as they were reported for reaching the original online version of the website. We found many redirections to legitimate websites; therefore those collected samples were stored as phishing. This occurrence introduced the hypothesis of attackers using redirections to legitimate websites in order to confuse detection systems. We found a significant amount of legitimate domains within the phishing class; Table 1 shows the most repeated domains. We manually checked those Fully Qualified Domain Names (FQDNs)

Table 1
Most repeated domain names on phishing collection.

Domain name	Samples	Banned
Total samples	186,000	–
*.000webhostapp.com	5702	No
*.weebly.com	4044	No
docs.google.com	2849	No
*.godaddysites.com	1055	No
*.appspot.com	903	No
*.umbler.net	799	No
storage.googleapis.com	720	No
firebasestorage.googleapis.com	642	No
2m.ma	1680	Yes
www.google.com	1312	Yes
www.imdb.com	698	Yes
www.paypal.com	626	Yes
drive.google.com	386	Yes
www.amazon.co.jp	705	Yes
login.microsoftonline.com	356	Yes
www.icloud.com	325	Yes


AT&T

Update Your AT&T Yahoo! Mail

AT&T Mail Powered By Yahoo!

ID/EMAIL

Tu respuesta

PAS*WORD

Tu respuesta

Enviar

Nunca envíes contraseñas a través de Formularios de Google.

Este contenido no ha sido creado ni aprobado por Google. Notificar uso inadecuado - Términos del Servicio - Política de Privacidad

Google Formularios

Fig. 4. Google Form used for phishing, where attackers ask for users and passwords from different websites and services.

and inspected the samples to confirm their legitimacy. After this inspection, filter one was implemented, which banned the false positives found on the manually reviewed FQDN list. Also, it is noteworthy the significant number of phishing websites hosted on 000webhost, Weebly, googleapis, GoDaddySites, appspot and umbler. These services allow the attackers to host their website at no cost. Finally, we noticed that some phishers used Google Forms (docs.google.com) as a lazy method to retrieve credentials, even when Google advises not to send passwords through them as shown in Fig. 4.

3.4.2. Filter 2: Empty and error samples

Phishing websites are ephemeral (Oest et al., 2020), and even when we visited the website within the next five minutes of its report, some of them were already offline and could not be retrieved. Also, a few of the online websites were empty or displayed errors related to the following reasons: Empty Apache directories, Cloudflare access denied or 403/404 errors. Those samples have no content, and they could introduce bias in the data for HTML or image detection methods. Consequently, these samples were banned in compliance with filter two and include both cases: Blank samples (no content) and error websites.

Table 2
Dataset distribution, on the left the total samples collected, then F_1, F_2 and F_3 represents the number of samples affected by each filter. On the right, the final number of valid samples and the amount of login forms within them in %.

Subset	Original samples	F_1	F_2	F_3	Valid samples	Forms proportion	Login forms proportion
Legitimate homepage	77,629	86 (0.1%)	3648 (4.7%)	–	73,895	77.70%	10.70%
Legitimate Login	61,000	1072 (1.8%)	4104 (6.7%)	–	55,847	92.42%	74.26%
Phishing	186,000	24,644 (13.2%)	23,419 (12.6%)	104,762 (56.3%)	**66,964**	**72.44%**	**40.80%**

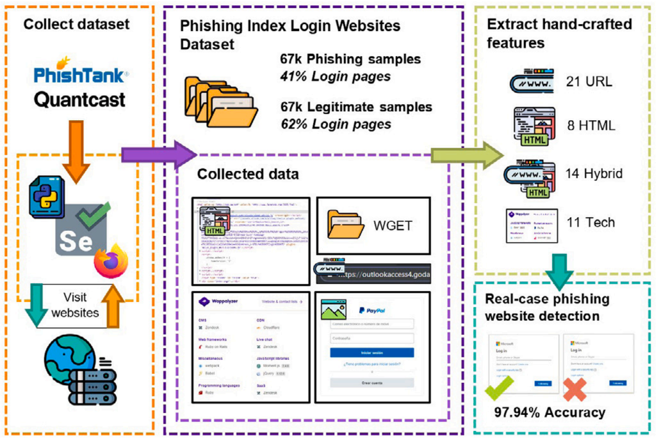


Fig. 5. Summary of the main stages and contributions of this work, from the sample gathering until the model generation.

3.4.3. Filter 3: Verified phishing samples

Not all reported phishing websites were real phishing. Phishtank implements an internal voting system used for verifying samples. This voting process can last up to a month, so samples within that time frame were checked regularly. For this task, we used two verification methods available at Phishtank:

- JSON report: Phishtank publishes an hourly report which contains extensive information from the last verified samples. If any sample in our dataset appeared in the report, we transferred the relevant information into the corresponding sample metadata, including the verified attribute.
- API request: The rest of the samples (not verified by the JSON report) were audited using Phishtank API. This method retrieves less information than JSON report, but includes the verification status of the sample.

After applying this last filter, we found out that a significant number of unverified websites were also real phishing. Since we have a large phishing set, we decided to work only with verified samples, representing 56.3% of the total collected phishing samples. Therefore, filter three was only applied to the phishing subset resulting in 104,762 unverified samples out of 186,000 that were discarded.

After the filtering process, we checked for forms on the HTML code and login forms with password fields (sign in forms). We found login forms on 40.80% of the phishing samples, on 74.26% of the legitimate login samples and on 10.71% of legitimate homepage samples. Table 2 summarize the available samples in the dataset.

4. Methodology

A representation of the complete work is provided in Fig. 5. First, the dataset is collected, organized and filtered as explained in Section 3. Then, the proposed features are extracted in the different sets explained in this section. Finally, experiments are presented, where we compare the performance of the different sets and features.

4.1. Features groups

Our approach relies upon combining four groups of features: URL, HTML, Hybrid and Technologies. The first three groups were built with some of the most effective state-of-the-art features (Chiew et al., 2019) and the new ones proposed in this study. Then, we introduced the novel technology-based features to the technologies group. We crafted 54 features, where half of them were legacy features from previous works and the other half were proposed to enhance detection performance. Each feature is labelled with a short tag composed of a letter representing the group ('U' for URL, 'H' for HTML, 'Y' for Hybrid and 'T' for technology) and a number to identify it within each group of features. The tag starts with an 'N' in the case of the novel ones.

4.1.1. URL features

The URL is a unique identifier of the website and locates it on the Internet. Attackers have different options to deploy their phishing sites: free hosting services, compromised websites and hired domains and servers (Jain & Gupta, 2022b; Moore & Clayton, 2007). When using free hosting services, phishers can customize the subdomains and the path. In a compromised server, attackers can set a specific path to locate the crafted phishing files. The last option implies hiring a server and a domain name for the attack. Since attackers cannot use the same domain name as the legitimate target (except for DNS spoofing attacks), they try to deceive users by using typosquatting techniques (Marchal et al., 2016), or a long deceptive subdomain that mimics the target URL structure (Oest et al., 2018).

HTTPS protocol has been used as a relevant feature in many phishing detection methods (Rao & Pais, 2019a; Rao et al., 2020). The increasing availability of free SSL certificates issued by Certificate Authorities (CA) like Let's Encrypt,[12] entailed a trend into requesting an SSL certificate for phishing sites. The number of phishing websites hosted under this protocol increased from 5% in the final quarter of 2016 to 83% at the beginning of 2021 (Anti-Phishing Working Group, 2021). Li et al. (2019) proved that this feature is no longer effective, and therefore, we do not include it in our work. The final set of URL features is described below:

Subdomain level (Chiew et al., 2019; Marchal et al., 2016; Moghimi & Varjani, 2016; Sahingoz et al., 2019): U1. Attackers use a long list of subdomains, leaving the first part of the URL as the target one to deceive users by hiding the original domain name. This technique has a higher impact on mobile devices since they can only display the lead part of the URL due to the narrow screen (Goel & Jain, 2018). We count the number of subdomains on the URL to include it in our feature vector.

Subdomain named '*com*' (Sahingoz et al., 2019): U2. Phishing URLs sometimes include '*com*' as a subdomain to mislead the end of the domain name and stop the user from reading the rest of the URL. We include a binary feature set to 1 in case there is a subdomain called '*com*'.

IP address (Jain & Gupta, 2018b; Li et al., 2019; Moghimi & Varjani, 2016; Rao et al., 2020): U3. In order to use a domain name on the website, attackers need to buy this service. If not, the website would only be accessible by introducing the IP address in the navigation

[12] https://letsencrypt.org/.

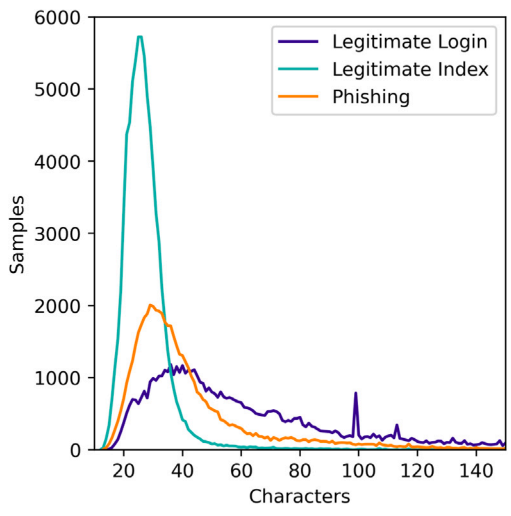


Fig. 6. URL length distribution in PILWD dataset. In this analysis we have only used valid samples presented on Table 2.

Table 3
Average digits found on the URLs from the proposed subset D1.

URL part	Legitimate	Phishing
Subdomain	0.0080	0.52800
Domain	0.0875	0.6073
Path	2.1311	11.1501

bar. If the domain name is an IP, this binary feature is set to 1 and 0 otherwise.

Common Top-Level Domain (TLD): NU4. The APWG (Anti-Phishing Working Group, 2021) reported a high number of websites hosted on common TLDs, but after an analysis of the collected dataset, we could count a high number of phishing samples hosted under uncommon TLDs. We consider the following Generic TLDs (gTLD) *'com', 'org', 'net', 'edu' and 'gob'* and all the Country Code TLDs (ccTLD) as ordinary. If the target URL TLD is not in the list, this binary feature is set to 1.

Length (Gupta et al., 2021; Rao et al., 2020; Sahingoz et al., 2019): McGrath and Gupta (2008) stated that phishing URLs were longer compared to legitimate ones. In Fig. 6 we depict the distribution of our filtered dataset URLs in terms of length. Most legitimate index URLs are distributed between 15 and 40 characters and legitimate login URLs are notably more distributed from 20 to 80 characters. Phishing URLs distribution is closer to the legitimate login URLs; hence, this feature is might not be valuable on the classification task between these two classes. This analysis supports our hypothesis that detecting phishing among legitimate login websites is more challenging than detecting them among legitimate index websites. We counted the characters within the subdomain (U5.1), the domain (U5.2) and the path (U5.3) to use them as three different features for our model.

Digits (Gupta et al., 2021; Rao et al., 2020; Sahingoz et al., 2019): Generally, phishing domain names have more digits than legitimate ones (Varshney et al., 2017). In Table 3 we provide the analysis of this premise over our dataset. We found that, in average, phishing URLs tend to incorporate digits, especially in the path. We counted the number of digits in the different parts of the URL, subdomain (U6.1), domain (U6.2) and path (U6.3).

Special characters (Chen et al., 2020; Fahmy & Ghoneim, 2011; Jain & Gupta, 2018b; Sahingoz et al., 2019): Phishing URLs usually include abnormal symbols (Verma & Dyer, 2015), some of them with a specific purpose like '@'. This symbol is used to append characters to the left of the URL which is ignored by the browser parser. Nine different symbols were picked to identify deceit techniques: '-' (U7.1),

Table 4
Different types of links and the average appearances on proposed dataset D1.

Link type	Avg. legitimate	Avg. phishing	Link example
External	86.83	25.01	⟨a href = "otherdomain.com"⟩⟨/a⟩
Internal	59.53	3.12	⟨a href = "thisdomain.com"⟩⟨/a⟩
Same page (#)	4.95	3.49	⟨a href = "#"⟩⟨/a⟩
Empty	0.57	0.53	⟨a href = ""⟩⟨/a⟩
Null	3.86	1.84	⟨a⟩⟨/a⟩

'.' (U7.2), '/' (U7.3), '@' (U7.4), '?' (U7.5), '=' (U7.6), '_' (U7.7), '&' (U7.8) and '~' (U7.9). These features are set to the times each symbol appeared on the entire URL.

Random words (Sahingoz et al., 2019): We included the detection of random domain names (U8.2) as a feature but also the detection of a randomly formed subdomain (NU8.1) since we observed a high number of phishing websites on free hosting services, which provide a randomly named subdomain to the phishers.

4.1.2. HTML features

Common free hosting services offer a low-resource server that includes limited number of CPU cores, low RAM and disk capacity. These variables limit the attacker capabilities to store and host complex websites. For this reason, phishing websites tend to be a single HTML page with no other content than the login form. Menus and links on those websites are usually empty to prevent users from leaving the target page. Moreover, techniques employed on phishing attacks aim to minimize the effort to build the attack. Since phishers only need the login form to be successful in the attack, they take a screenshot of the legitimate website and place the HTML form on top of that image (Rao & Pais, 2019b), omitting the implementation of the rest of the website. The complete list of HTML features is shown below:

Links (Gupta et al., 2018; He et al., 2011; Li et al., 2019; Mohammad et al., 2012; Whittaker et al., 2010): Phishing websites try to imitate a legitimate one, but almost all of them are hosted on a different domain name. In their way to deceive users, attackers use links from the original website resources to refine their websites while their sitemap is very limited so are the number of internal links. Significant legitimate websites store their resources on cloud services; therefore, the number of external links will increase. In the same way, these websites count on an extensive infrastructure and sitemap so the number of internal links is also expected to be high. Finally, empty links are used by phishing websites to simulate a wide domain content but preventing users from leaving the phishing site simulating a malfunction when they are clicked. These premises are confirmed on Table 4, where the proportion between types of links in our dataset noticeable. To differentiate between the types of links, we took the domain name from the URL and compared it with the ones in the links of the HTML code. We implemented five legacy features corresponding with the types of links described in Table 4. We counted the number of external links (H1.1), internal links (H1.2), same page links (H1.3), empty links (H1.4) and null links (H1.5).

Body length and tags: Li et al. (2019) used a method to compute the length of the websites by using the content of style, script, link, comments and form tags. An extensive sample inspection revealed that scripts and styles are similar in length between phishing and legitimate classes. For this reason, we changed the way to measure the length and implemented two features: First, body length (NH2.1) which counts the number of characters within the body and second, the body tags (NH2.2) where we count the number of HTML paired and single tags inside the body. We analysed the potential of these features on our dataset, and discovered that the average length of the legitimate websites on our dataset was $104,539.41$ characters and $42,243.36$ on the phishing websites, which is a significant difference. This difference is also noticeable in the average number of tags found in both classes, the legitimate websites had 823.23 HTML tags while phishing website are built with 258.41 HTML tags as average. This analysis confirm that phishing websites content is, in most cases, effortless in comparison with the legitimate one.

Base64 resources: NH3. We noticed some phishing samples with no resource files but they contained images on the HTML. In this case, the media content was attached with base64 codification. The reason behind this could be the limited number of files on the webserver. We define a binary feature that is set to 1 if the HTML have content with base64 encoding.

4.1.3. *Hybrid features*

Phishing websites are developed to look legitimate in appearance. To achieve this, attackers copy the HTML, CSS, JavaScript and images to incorporate them into their website. However, URLs are unique and domain names cannot be repeated, generating a naming inconsistency between the website and the URL in most cases.

Rao and Pais (2019a) used the title, copyright and website description to identify the brand instead of using TF-IDF technique as CANTINA+ (Xiang et al., 2011) did. Other works (Marchal et al., 2016; Rao & Pais, 2019a, 2019b; Xiang & Hong, 2009) aim to recognize the identity by using copyright/title in their approaches. After the brand is determined, Xiang and Hong (2009) look for the domain name corresponding to that brand on the Internet and compare it with the current phishing domain name. Since our goal is to develop a third-party independent system, we focused on the top three identifying elements of a website: The first one is the title, which usually contains the brand of the targeted company. The second one refers to copyright, where the brand appears again. Last, but not least, the URL, where the domain name is usually the brand itself. With these identifiers, we created three levels of consistency.

The first level corresponds to a *complete* consistency, where copyright and domain name are within the title. We noticed a great number of legitimate titles like “login” or “sign-in” since login websites are also included in the dataset. To overcome this issue, we also considered the copyright and domain name match to be fully consistent.

The second level of consistency is *misleading*, which is identified when the attacker use typosquatting techniques to deceive users (Spaulding et al., 2016). To detect it, we used the Levenshtein distance (Levenshtein, 1965) with a maximum distance of two between the brand declared in the copyright and the title or domain name. We refused the usage of any brand list since it biases the detection for non-famous companies.

The last level is *zero* consistency, where all website identifiers (title, domain and copyright) are mismatched or the copyright mark is missing in the source code.

Once we have described the levels of consistency, we created the following list of hybrid features used for classification:

Copyright in the HTML (Marchal et al., 2017, 2016; Xiang & Hong, 2009): Y1. Phishers objective is to spoof the brands to increase users’ confidence. We detect whether a website has the copyright disclaimer ©, or not.

Domain in HTML: NY2 Usually, legitimate websites have their domain name in the HTML, but phishing ones are unlikely to include their strange domain name. Users may suspect if they find names with no relationship with the current brand or company. Instead of using the raw HTML, we provided a novel approach and removed the comments in case attackers placed their domain name into the comments as a detection bypass.

Domain-Copyright: We created three specific features depending on the relation between the domain and the copyright: First, equal when they match completely (NY3.1). Second one is when the copyright is within the domain (NY3.2), a common practice in phishing websites. Finally, we detected typosquatting between both names (NY3.3).

Subdomain-Copyright: We used the same distribution as the domain but using the subdomain. This way, we created the same levels of similarity: Equal (NY4.1), within (NY4.2) and typosquatting (NY4.3). These three features aimed to detect cases where attackers use the brand name on the subdomain to deceive users, especially on mobile devices (Goel & Jain, 2018).

Path-Copyright: A deep analysis of the phishing data shows that some attackers use the brand name to identify their attacks on the same host. We implemented two more features, checking if the copyright brand match one part of the path (NY5.1) or if it is within any one of them (NY5.2).

Title-Domain-Copyright: NY6. Rao and Pais (2019a) used the title, website description and copyright as an identifier. We changed this feature and checked whether the copyright and domain name were the same and if they were also in the title.

Title-Domain (Li et al., 2019): Y7. As not all the websites have copyright on the HTML, we crafted a feature to compare these two identifiers: the title, which usually has the target brand, and the domain, which could be the brand in the legitimate case but not in the phishing one.

Domain in body (Li et al., 2019): Y8. As the previous domain in the HTML, this feature only uses the HTML body. This way, metadata is excluded, which may include information related to the domain.

Subdomain in the title: NY9. We noticed a high number of brands and keywords (secure, login, accounts, ...) that were in the title and also in the subdomain. We checked if any of the subdomains matched the title of the website.

4.1.4. *Technology-based features*

We introduced novel features based on web technology analysis to capture relevant information for phishing detection. Phishing websites could be similar in appearance to legitimate websites or direct clones from the original pages. However, they are created with effortless techniques and phishing kits (Bijmans et al., 2021) to save time and development costs (Yang et al., 2021). The usual method implies copying the HTML code, CSS and resources to pasting them into a raw HTML file (Jain & Gupta, 2018a). In the cloning process, the technologies used by the native websites is lost since attackers cannot clone server-side source code. Then, minimum changes are done to get the page running on a simple web server. Given this situation, followed by the low server resources mentioned above, we can assume most of the phishing websites run a simple web server software like Apache or Nginx to provide the necessary files and execute the attack. In other words, attackers do not use complex frameworks, web technologies or databases due to the time-consuming tasks involved.

With the Wappalyzer report for each website, we could identify the most significant technologies that represent both, the phishing category and the legitimate one. Additionally, the number of detected technologies is also registered since it could be an identifier of effort invested on the website. After a deep analysis of the dataset technologies, we came up with the following features:

Number of technologies detected: NT1. Phishing attacks are simple, the fewer technologies or libraries used, the more chances to be a phishing website.

Specific technologies: Some technologies are more frequently used on legitimate websites than on phishing ones. To select those technologies, we introduced the total of 134 detected technologies into *SelectKBest* library from scikit-learn and used the *f_classif* algorithm to create the technologies ranking. Results showed that the most valuable features were Google Analytics (NT2), Google Tag Manager (NT3), Facebook (NT4), PHP (NT5), Apache (NT6), Google Font API (NT7), jQuery Migrate (NT8), jQuery (NT9), Cloudflare (NT10) and Bootstrap (NT11).

Table 5
Total features implemented in this work. Features are represented by an id regarding its novelty and assigned group. "#" stands for the number of a given item, symbol or character.

Id	Name	Value	Description
U1	Subdomain level	Discrete	# of subdomains
U2	Subdomain "com"	Binary	Checks if there is a "com" subdomain
U3	IP address	Binary	Checks if domain name is an IP
NU4	Common TLD	Binary	Checks if TLD is in common TLD list
U5.1	Subdomain length	Discrete	# of characters in the subdomain.
U5.2	Domain length	Discrete	# of characters in the domain.
U5.3	Path length	Discrete	# of characters in the path.
U6.1	Subdomain digits	Discrete	# of digits in the subdomain.
U6.2	Domain digits	Discrete	# of digits in the domain.
U6.3	Path digits	Discrete	# of digits in the path.
U7.1	Character "-"	Discrete	# of "-" in the complete URL
U7.2	Character "."	Discrete	# of "." in the complete URL
U7.3	Character "/"	Discrete	# of "/" in the complete URL
U7.4	Character "@"	Discrete	# of "@" in the complete URL
U7.5	Character "?"	Discrete	# of "?" in the complete URL
U7.6	Character "="	Discrete	# of "=" in the complete URL
U7.7	Character "_"	Discrete	# of "_" in the complete URL
U7.8	Character "&"	Discrete	# of "&" in the complete URL
U7.9	Character "~"	Discrete	# of "~" in the complete URL
NU8.1	Random subdomain	Binary	Check if the any of the subdomains is a set of random characters
U8.2	Random domain	Binary	Checks if the domain is composed by a set of random characters
H1.1	External links	Discrete	# of links pointing to different domains
H1.2	Internal links	Discrete	# of links pointing to the actual domain
H1.3	Same page links "#"	Discrete	# of links referencing the actual page ("#")
H1.4	Empty links	Discrete	# of links with no content on the "href" attribute
H1.5	Null links	Discrete	# of links with no "href" attribute
NH2.1	Body length	Discrete	# of characters within the body tags of the HTML code
NH2.2	Body tags	Discrete	# of paired and single HTML tags inside the body
NH3	Base64 resources	Binary	Checks if any of the resources is encoded on base64
Y1	Copryright	Binary	Checks whether the HTML code contains a copyright disclaimer
NY2	Domain in HTML	Discrete	# of times that the domain appears in the HTML
NY3.1	Domain is equal to copyright	Binary	Checks if the domain is equal to the stated copyright brand
NY3.2	Copyright is within the domain	Binary	Checks if the copyright is as it is in the domain
NY3.3	Typosquatting domain-copyright	Binary	Checks if the Levenshtein distance between the domain and copyright is less or equal to two
NY4.1	Subdomain is equal to copyright	Binary	Checks if any of the subdomains is equal to the stated copyright brand
NY4.2	Copyright is within a subdomain	Binary	Checks if the copyright is as it is in any of the subdomains
NY4.3	Typosquatting subdomain-copyright	Binary	Checks if the Levenshtein distance between any subdomain and the copyright is less or equal to two
NY5.1	Copyright in path words	Binary	Checks if the copyright is equal to any of the words in the path
NY5.2	Copyright is within the path words	Binary	Checks if the copyright is in a word into the path
NY6	Title-domain-copyright	Binary	Checks if the three elements are equal
Y7	Title-domain	Binary	Checks if the domain is within the website title
Y8	Domain in body	Discrete	# of times that the domain appeared in the whole HTML code
NY9	Subdomain in title	Binary	Checks if any of the subdomains is in the website title
NT1	Number of technologies	Discrete	# of technologies detected in the web server
NT2	Google Analytics	Binary	Checks if Google Analytics is within the detected analysis
NT3	Google Tag Manager	Binary	Checks if Google Tag Manager is within the detected analysis
NT4	Facebook	Binary	Checks if Facebook is within the detected analysis
NT5	PHP	Binary	Checks if PHP is within the detected analysis
NT6	Apache	Binary	Checks if Apache is within the detected analysis
NT7	Google Font API	Binary	Checks if Google Font API is within the detected analysis
NT8	jQuery Migrate	Binary	Checks if jQuery Migrate is within the detected analysis
NT9	jQuery	Binary	Checks if jQuery is within the detected analysis
NT10	CloudFlare	Binary	Checks if CloudFlare is within the detected analysis
NT11	Bootstrap	Binary	Checks if Bootsrap is within the detected analysis

4.2. Feature extraction and vectorization

Once all features have been defined, we used a Python3 script to iterate over all valid samples (not banned). Each sample went over the four developed modules (one per group) and generated a JSON structure indicating the name of the descriptors as keys and their correspondent values.

Then, an n-dimensional feature vector is generated for each group, for instance, $U_v = [U_1, U_2, \dots, NU_{8.1}]$, $H_v = [H_{1.1}, H_{1.2}, \dots, H_3]$, $Y_v = [Y_1, NY_2, \dots, NY_9]$ and $T_v = [NT_1, NT_2, \dots, NT_{11}]$. Finally, the four vectors are concatenated to obtain the complete feature vector for a specific sample: $F_v = [U_v, H_v, Y_v, T_v]$. After all samples have been processed, they were arranged into an X matrix with MxN dimension, where M corresponds to the number of features and N is the number of total samples of the experiment. An overview of this process is shown in Fig. 7.

5. Experimentation and results

The proposed hand-crafted features were evaluated for the phishing detection task. In this section, we describe the environment, the different sets of samples used in the experiments and the obtained results.

5.1. Datasets

In order to arrange the samples used for experimentation, we defined four different data subsets depicted in Table 6.

The real-world scenario for phishing detection algorithms implies the classification of login web pages, where users have to determine if a login form is trustworthy or phishing. As seen in Table 2, only 10.70% of the collected legitimate homepages had a login form, while in the phishing class, these pages represent 40.80% of the samples.

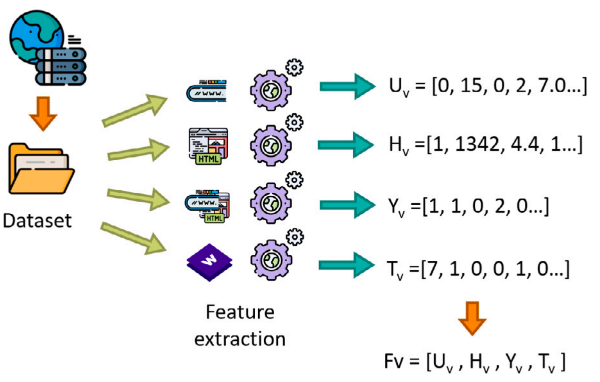

Dataset
Feature extraction
U_v = [0, 15, 0, 2, 7.0...]
H_v =[1, 1342, 4.4, 1...]
Y_v =[1, 1, 0, 2, 0...]
T_v =[7, 1, 0, 0, 1, 0...]
Fv = [U_v , H_v , Y_v , T_v]

Fig. 7. An overview of the extraction and vectorization process.

Table 6
Data subsets used in different experiments. D1 and D2 are used for model training and testing while D3 and D4 are used only for testing purposes.

Subset	Legitimate samples	Legitimate login forms (%)	Phishing samples	Phishing login forms (%)
D1	66,964	61.93	66,694	40.80
D2	66,964	10.71	66,694	40.80
D3	55,847	74.26	–	–
D4	62,729	10.71	–	–

The proposed subset, D1, was built to represent this scenario by using completely filtered and verified phishing samples (66,964) along with 55,847 legitimate login samples and 11,117 legitimate homepage samples, that combined, make a total of 66,964 instances for the legitimate class. In the phishing class, we found login forms asking for credentials but with no input of type password, therefore, they were not counted as login pages. Since we cannot review the complete set of phishing samples manually, we crafted the legitimate class so the proportion of login forms is above the phishing one. The result is a dataset where both classes, *Legitimate* and *Phishing*, have a similar amount of login websites, 61.93% the first one and 40.80% the second.

D2 depicts the methodology used in most state-of-the-art papers, where authors collect only the homepage of the most visited domains and the phishing websites. It was built using 66,964 filtered and verified phishing samples and 66,964 legitimate homepage samples, which have 10.70% of login forms in comparison with the 40.80% of the phishing one.

Subsets D3 and D4 were created to benchmark the behaviour of different proposals when facing a different set of legitimate websites. D3 contains only legitimate login samples, and D4 was built using legitimate homepage samples only. To arrange fair tests with D4, we have removed the first 11,117 samples used to build D1. Hence, D1 and D4 do not share any samples. D3 and D4 datasets have no phishing samples, and they were used only for benchmarking purposes.

5.2. Experimental setup

Experiments are executed on an Intel Core i3 9100F at 3.6 GHz and 16 GB of DDR4 RAM. We used scikit-learn[13] and Python 3 for the implementation of the different experiments.

We have selected nine classifiers, widely used in the literature (Das et al., 2020; Dou et al., 2017): Support Vector Machine (SVM), Logistic Regression (LR), Naïve Bayes (NB), Random Forest (RF), k-Nearest Neighbour (kNN) and Adaboost (ADA). We also included two powerful ensemble methods, LightGBM (LGBM) and XGBoost (XGB), that achieved a great performance in other works (Li et al., 2019).

We have selected the best parameters for each classifier empirically, using k-fold cross-validation (CV). The rest of the undefined parameters are set with the default values in scikit-learn. In the case of SVM we have used Radial Basis Function (RBF) kernel with parameters $C = 1$ and $gamma = 0.1$. For Logistic Regression we obtained the best results using Limited-memory Broyden–Fletcher–Goldfarb–Shanno (LBFGS) solver with default parameters. Naïve Bayes obtained its best results using Bernoulli implementation with the *binarize* parameter since most of our features are binary already. Random Forest was tuned using the number of estimators, where 300 was the optimal value. k-NN was tested using a different number of neighbours and the best results were obtained using $k = 1$. For AdaBoost, we found that 50 estimators obtained the best results for our problem. LightGBM was configured with 500 leaves and *binary* objective. Finally, XGBoost obtained its best performance using 200 estimators.

We scale the features vector using scikit-learn's *StandardScaler* over the complete set of training samples and features, then, we applied the obtained scaler to the test samples.

For the assessment of phishing detection models, we estimated the classifier performance using the k-fold cross-validation technique, with the following parameters: $k = 10$, $shuffle = True$ and $random\_state = 42$.

For measuring the impact of the login websites, first, we train and evaluate the classifier using $70 - 30\%$ training and test split with $random\_state = 42$. Then, benchmark datasets, D3 and D4, are introduced to the generated model to measure the accuracy.

5.3. Performance metrics

We assessed the performance of the phishing classification models employing accuracy, precision, recall and F-score. We denoted the phishing class as the positive class and the legitimate class as the negative one.

The classification accuracy, i.e. the number of correctly classified samples is taken as the main metric for evaluation purposes due to its common use in phishing detection works (Rao & Pais, 2019a; Sahingoz et al., 2019) and the fact that it is reliable on a balanced dataset like ours. It can be computed as shown in Eq. (1)

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

where True Positive (TP) indicates the number of phishing samples correctly classified whereas True Negative (TN) refers to the number of legitimate samples correctly identified as legitimate. False Positive (FP) represents the number of legitimate samples wrongly classified as phishing. False Negative (FN) denotes the number of phishing samples improperly classified as legitimate.

Precision is also a relevant metric in this field and it is defined as the fraction of correctly classified phishing samples over the number of items classified as phishing, as indicated in Eq. (2).

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

Recall refers to the fraction of correctly classified phishing samples over the total number of phishing instances as indicated in Eq. (3).

$$Recall = \frac{TP}{TP + FN}, \tag{3}$$

Finally, the F1 score of a class summarizes the two before-mentioned metrics as it refers to the harmonic mean of the precision and recall and it is computed as shown in Eq. (4).

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4}$$

[13] https://scikit-learn.org/stable/.

Table 7
Performance results for the classification algorithms.

Algorithm	Precision	Recall	F1-score	Accuracy
XGBoost	0.978	0.971	0.974	**97.42%**
LightGBM	0.983	0.976	0.980	**97.95%**
AdaBoost	0.940	0.935	0.937	93.76%
Random Forest	0.980	0.972	0.976	**97.63%**
kNN	0.931	0.948	0.940	93.91%
SVM	0.955	0.942	0.948	94.86%
Logistic Regression	0.890	0.901	0.899	89.79%
Naïve Bayes	0.828	0.911	0.868	86.12%

5.4. Assessment of phishing detection models

This experimentation involves the evaluation of eight machine learning algorithms using the proposed dataset D1, and the complete set of 54 features displayed on Table 5.

Table 7 shows the performance of each algorithm, where we can observe that ensemble methods outperformed the rest of the classifiers, with a significant performance gap between them.

The highest accuracy was 97.95%, achieved by the LightGBM algorithm, which is inline with results obtained by current state-of-the-art works (Li et al., 2019; Rao et al., 2020; Sahingoz et al., 2019). On the other hand, Random Forest, which usually brings a consistently high accuracy among other algorithms in phishing detection tasks (Basit et al., 2021), outperformed the remaining classifiers with an accuracy of 97.63%.

Note that the proposed model is the confluence of two new concepts on phishing detection: First, using a high number of legitimate login websites in our dataset and second, the implementation of new hand-crafted features plus the website technology analysis. Since the LightGBM algorithm obtained the best accuracy, we used it to carry out the rest of the experiments.

In terms of execution time, we split the total execution time in two: the feature extraction time and the time elapsed in the classification task. For this test, we used all samples in dataset D1. The proposed model generated using the LightGBM algorithm was able to classify one sample in 0.21 ms. on average. Feature extraction time since the final objective is to incorporate the complete algorithm in real-time environments. This extraction time is measured once all resources are loaded in memory, including the URL, the HTML and the JSON file with the technology analysis, therefore, we have discarded read time from the disk since the ideal implementation should process the data on the fly. Average extraction time was 56.43 ms. per sample. It is worth noting that the average extraction time for a legitimate sample was 81.76 ms. whereas a phishing sample achieved 31.11 ms. average. The main reason is the usage of regular expressions over extensive HTML code.

5.5. Impact of login websites on phishing detection

One of the main contributions of this paper is the high number of legitimate login websites in the proposed dataset that represents a real-world scenario. To demonstrate the capabilities of our approach, we assessed two different pipelines presented in Fig. 8. In the first one, we trained and tested a model with phishing and legitimate homepages (D2 used for the base model), then, the same model was tested again, but with legitimate login samples (D3 as the second dataset). In the second pipeline, we trained and tested a new model with the proposed dataset (D1) which contains phishing and legitimate login samples, then it was tested again with the legitimate homepages subset (D4).

Both pipelines were executed following these steps: First, we train and test a base model using the LightGBM algorithm with a 70 – 30% dataset split. Results from this step are shown in blue on Fig. 8. Once the model has been exported, we test it again with samples from the

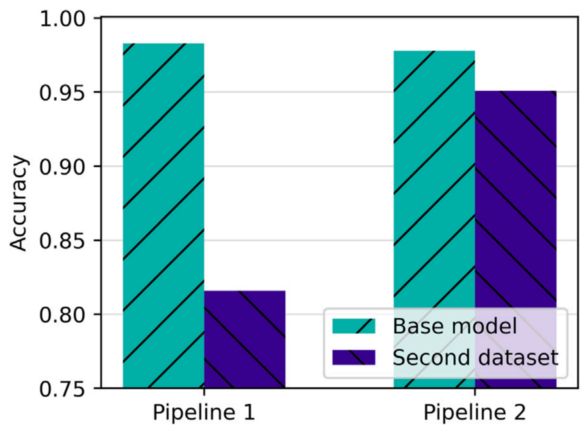


Fig. 8. The first pipeline depicts results for training a model with D2 and test it with D2 (on blue) and with D3 (on purple). The second pipeline shows results of using D1 for training and test (on blue), and testing later the same model with D4 (on purple).

second dataset, obtaining in this way, the performance results shown in purple on the same Fig. 8.

For the first pipeline, the model learnt using the D2 dataset, which is the common state-of-the-art approach dataset. Fig. 8 depicts the results obtained in this experiment, where the performance of the model trained with D2 was estimated as 98.27% in terms of accuracy. Next, we tested that model on D3 and results showed that the base model fails to classify legitimate login samples obtaining only 84.61% accuracy on this task. It misclassified 18.41% of legitimate login samples (10,273 out of 55,799). Hence, models trained with legitimate homepage samples are not trustworthy when used on real-world applications due to the high false-positive rate on their predictions for login websites.

We tested our approach in the second pipeline. We trained a phishing detection model using the D1 dataset with an estimated 97.94% accuracy. Then, we used this model for the homepage classification task (D4 dataset) and obtained 95.11% accuracy. This experiment demonstrates that our model focused on login websites, generalize better for the phishing detection problem and correctly detects phishing among any type of legitimate websites.

5.6. Feature relevance

In order to understand the impact of each feature group for detecting phishing, we used the proposed dataset (D1) and evaluated, one by one and with different combinations, the groups of features described in Section 4. Additionally, we added the legacy group representing the state-of-the-art features implemented in this work.

Results, in Fig. 9, show the increased performance of the model when taking into account more groups of features. The combination of all sets obtained 97.95% accuracy. The subsets with three groups had a negligible difference with the complete set, reducing its accuracy to 97.58% in case of removing the technology set, a 97.51% in case of removing the hybrid set and 97.40% by excluding the HTML set. Among the subsets with two groups, UH (URL and HTML groups) obtained the best performance with 96.80% accuracy. The standalone performance of the groups had significant differences. HTML group could reach 93.57% accuracy followed by hybrid (92.97%), URL group (91.47%) and finally the Tech group (83.63%). Overall results revealed that the combination of all subsets is necessary to obtain the best performance. Also, subsets where URL and HTML features were combined, tend to perform better than the rest. Technology-based features showed a bad standalone performance but significantly improved the detection, adding the mentioned endurance to bypass techniques.

Finally, the 28 legacy features obtained 97.05% accuracy, concluding that the novel 27 features enhance the results by 0.90% proving that they are helpful and provide robustness for the phishing detection task.

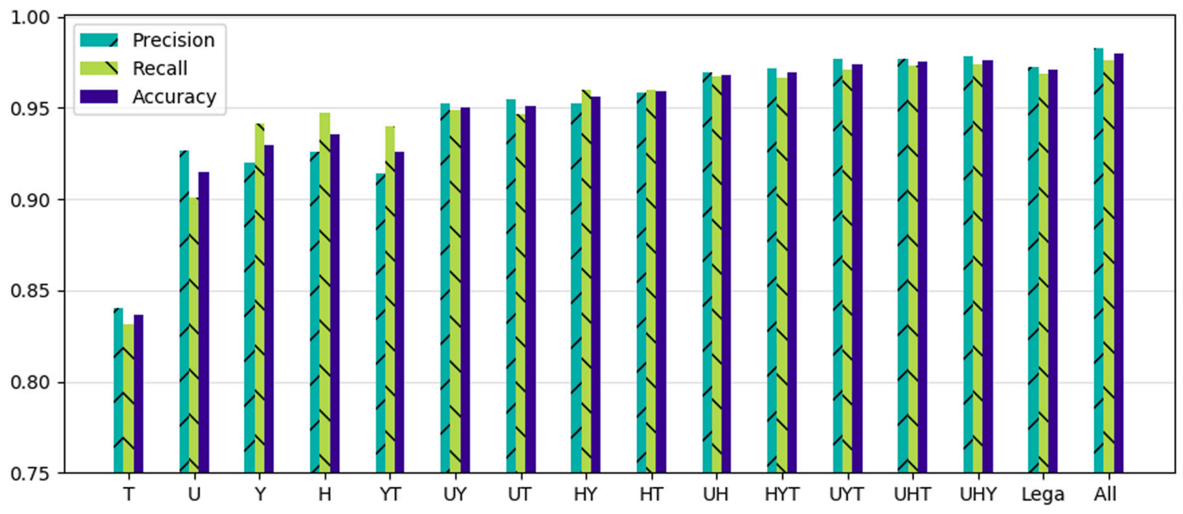


Fig. 9. Comparison of groups combination: 'All' represents the whole set of 54 features, 'Lega' denotes the legacy features and then sets are named using the letters of the groups involved: U stands for URL group, H for HTML, Y for Hybrid and *T* for Technology. For instance, UHY, represents the set composed by URL, HTML and Hybrid features.

5.7. Comparing feature importance

Li et al. (2019) used split feature importance coefficient[14] to generate a descriptor ranking. We used this method with the best ensemble model in our work, LightGBM. Fig. 10 shows the importance coefficient for each feature on the proposed model. The novel "body length" and "number of tags within the body" were the most valuable, which is directly related with the complexity of the websites and their content. Then, "external links", "path length", and "domain length" closed the top five and confirmed their use over the years in the phishing detection scene. The novel "number of technologies" was located within the six most valuable features, because that phishing pages are built effortlessly with no frameworks or additional user experience packages or libraries. Feature "domain in HTML" also had a high impact on the classification, this is probably caused by the fact that most phishing attacks are hosted on free-of-charge servers with fixed domain names or within compromised domains (Jain & Gupta, 2022b). Therefore, they cannot impersonate a company while placing another domain name in the HTML body since it arises suspicion. Novel features have proven to aggregate resilience to the model when facing phishers' modifications oriented to avoid detection or clone attacks. Overall, HTML features had a higher impact than the rest of the groups, proving that they are an essential group for phishing detection.

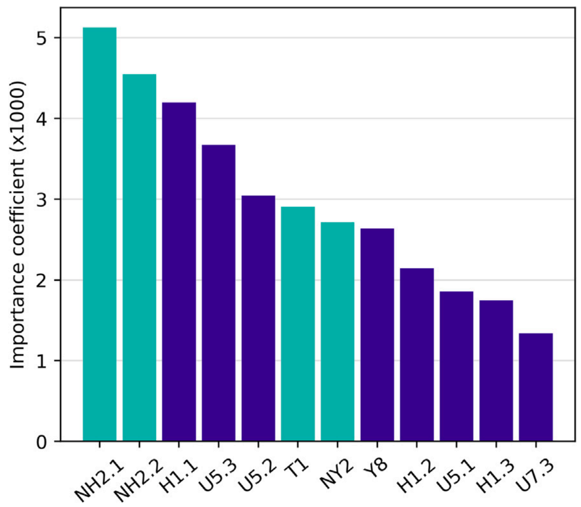


Fig. 10. Comparison of the 12 most important features measured with lightGBM's feature importance.

5.8. Comparative evaluation

In Table 8, we compare the performance of our method with recent works in the literature. Columns *URL*, *Indep* and *Login* represent works using only URLs, the independence from external services and the specific use of login pages in the methods. Yang et al. (2021) extended their dataset with artificial samples generated using the ADASYN algorithm. One of the main problems solved in this work is the absence of a public standard dataset useable on any methodology. Sahingoz et al. (2019) published an URL dataset which is a great baseline since most legitimate URLs include the complete path. This work is the only one that demonstrates its effectiveness in classifying legitimate login pages and phishing. Furthermore, we used and released the most complete and updated dataset in the literature, which allowed us to report the performance utilizing current phishing attacks. Finally, we believe that our dataset will allow other researchers to fairly compare their methods against the same dataset instead of reporting results using personal datasets.

6. Conclusions and future work

This paper addresses the problem of phishing detection with some contributions to this field: Firstly, we introduce PILWD, a high-quality offline dataset presenting a new phishing detection paradigm characterized by containing legitimate samples mainly from login pages. Second, a new group of features based on website technologies improved detection accuracy and enhanced resilience to the detection system. Third, we proposed additional novel features focused on the URL and HTML, combined with other state-of-the-art ones. Finally, the appropriate combination of the evaluated features for describing web pages combined with the selected LightGBM classifier establishes a new methodology that allows detecting phishing with a competitive 97.94% accuracy on very real-case scenarios.

The presented dataset, PILWD, is an extensive and updated phishing website dataset with 134,000 verified samples and up to 324,000 total samples for researchers to evaluate their proposals in this field. It is a more comprehensive benchmark for testing various approaches, providing researchers with an easy way to compare their work and solutions against the same dataset. The information about the collected samples includes URL, HTML, Screenshots, Technology Analysis, files

[14] https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.LGBMClassifier.html.

Table 8
Comparative evaluation of phishing detection techniques.

Work	Dataset year	Legitimate samples	Phishing samples	URL	Public dataset	Indep	Login	Accuracy	F1-score
Rao and Pais (2019a)	2016	1,407	2119	No	No	No	No	99.31	N/A
Moghimi and Varjani (2016)	2013	686	3066	No	No	Yes	No	98.70	99.0
Ding et al. (2019)	2017	5,883	2776	Yes	No	Yes	No	98.90	99.0
Adebowale et al. (2019)	2015	6,157	6843	No	No	No	No	98.30	98.3
Sahingoz et al. (2019)	2018	36,400	37,175	Yes	Yes	Yes	No	97.98	98.0
Yang et al. (2021)	2018	60,000	5000*	No	No	No	No	97.50	97.3
Li et al. (2019)	2009–2017	30,873	19,074	No	No	Yes	No	97.30	N/A
Rao et al. (2020)	2018	85,409	40,668	Yes	No	Yes	No	94.26	95.9
Sadique et al. (2020)	2019	60,000	38,000	No	No	No	No	90.51	N/A
This work	**2020**	**66,964**	**66,964**	No	**Yes**	**Yes**	**Yes**	**97.95**	**98.0**

and other metadata. The availability of this data will also prevent researchers from the time-consuming task of data collection.

Based on our observations, we introduced a new paradigm on phishing detection because we consider that the datasets to test any solution should contain a high number of legitimate login pages. Considering that the principal intention of phishing websites is to retrieve user data with login or sign-up forms, we determined that they have to be present in the legitimate class of any phishing dataset. Because of that, our presented dataset has 61.93% of login forms in legitimate class and 40.80 in the phishing one. This approach can be easily translated to real-world applications where users can be advised before introducing their credentials on a login website.

Additionally, technology analysis was proposed as a new group of features. Legitimate websites are carefully built, using a set of technologies and frameworks to offer the best functionality possible to users, while phishing websites are minimal, usually in a hurry and in a straightforward way. The features from the technology group improve detection and provide resilience against bypasses since attackers need to build more complex and resource-consuming websites.

Finally, we collected the most significant features on the state-of-the-art and proposed a set of novel features. Random subdomain detection, a new method for body length measure, the identity features, and the technology group have improved the performance compared to models where only legacy features were used.

An additional observation is that the proposed set of 54 features is independent of third-party services like Google Page Rank or WHOIS, preventing delays or malfunctions caused by those services. Using the complete set of features combined with the LightGBM algorithm, we have obtained a high 97.94% accuracy, outperforming some of the current state-of-the-art works.

Our approach has limitations. Due to the significant number of samples in the dataset, isolated false positives or negatives are possible. For that reason, we implemented the filters mentioned in this work, which can identify most of the compromised cases. Also, the verification process of the phishing samples is entirely dependent on PhishTank services and users; therefore, we cannot guarantee the correct verification of all the phishing samples. Also, it could be another limitation that we proposed several new hybrid features related to website identification using copyright, title and domain name. This approach may affect websites with no trademarks or companies behind them, even when we did not use any brand list.

For future works, one of the critical tasks for phishing detection is target recognition. We have used copyright and title to address this problem. However, brand and logo recognition methods can enhance typosquatting features to detect phishing or deceptive content that does not match legitimate domain names. Furthermore, since we retrieve the images of the websites, logo detection can be developed using PILWD samples. Finally, other researchers may find external services information interesting for their works. Therefore, our goal for future datasets is to retrieve favicon, SSL, WHOIS and Google Rank information to cover all approaches on the state-of-the-art, even if they are not suitable for real-time detection. Thus, an updated dataset could be helpful to keep up-to-date phishing detection research.

CRediT authorship contribution statement

Manuel Sánchez-Paniagua: Conceptualization, Methodology, Software, Formal analysis, Investigation, Data Curation, Validation, Writing - Original Draft. **Eduardo Fidalgo:** Conceptualization, Supervision, Project administration, Writing – review & editing. **Enrique Alegre:** Conceptualization, Funding acquisition, Supervision, Writing – review & editing. **Rocío Alaiz-Rodríguez:** Writing – review & editing.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgements

This research was funded by the framework agreement between the University of León and INCIBE (Spanish National Cybersecurity Institute) under Addendum 01.

Disclaimer

Reports on PhishTank are arranged by anonymous users and pages reported can contain malicious files or code within the source code, the HTML or the WGET files. An alternative dataset with no WGET files is provided to reduce the size and risk of the archives.

References

Abutair, H. Y., & Belghith, A. (2017). Using case-based reasoning for phishing detection. *Procedia Computer Science*, *109*, 281–288. http://dx.doi.org/10.1016/j.procs.2017.05.352, 8th International Conference on Ambient Systems, Networks and Technologies, ANT-2017 and the 7th International Conference on Sustainable Energy Information Technology, SEIT 2017, 16-19 May 2017, Madeira, Portugal.

Adebowale, M., Lwin, K., Sánchez, E., & Hossain, M. (2019). Intelligent web-phishing detection and protection scheme using integrated features of images, frames and text. *Expert Systems with Applications*, *115*, 300–313. http://dx.doi.org/10.1016/j.eswa.2018.07.067.

Aljofey, A., Jiang, Q., Qu, Q., Huang, M., & Niyigena, J.-P. (2020). An effective phishing detection model based on character level convolutional neural network from URL. *Electronics*, *9*(9), 1–24. http://dx.doi.org/10.3390/electronics9091514.

Anti-Phishing Working Group (2021). Phishing activity trends report 1Q. URL: https://docs.apwg.org/reports/apwg_trends_report_q1_2021.pdf. Accessed: 2022-06-06.

Basit, A., Zafar, M., Liu, X., Javed, A., Jalil, Z., & Kifayat, K. (2021). A comprehensive survey of AI-enabled phishing attacks detection techniques. *Telecommunication Systems*, *76*(1), 139–154. http://dx.doi.org/10.1007/s11235-020-00733-2.

Bijmans, H., Booij, T., Schwedersky, A., Nedgabat, A., & van Wegberg, R. (2021). Catching phishers by their bait: Investigating the dutch phishing landscape through phishing kit detection. In *30th USENIX security symposium (USENIX security 21)* (pp. 3757–3774). USENIX Association, URL: https://www.usenix.org/conference/usenixsecurity21/presentation/bijmans.

Bose, I., & Leung, A. C. M. (2014). Do phishing alerts impact global corporations? A firm value analysis. *Decision Support Systems*, *64*, 67–78. http://dx.doi.org/10.1016/j.dss.2014.04.006.

Bozkir, A. S., & Aydos, M. (2020). LogoSENSE: A companion HOG based logo detection scheme for phishing web page and E-mail brand recognition. *Computers & Security*, *95*, Article 101855. http://dx.doi.org/10.1016/j.cose.2020.101855.

Buber, E., Diri, B., & Sahingoz, O. K. (2018). Nlp based phishing attack detection from URLs. In A. Abraham, P. K. Muhuri, A. K. Muda, & N. Gandhi (Eds.), *Intelligent systems design and applications* (pp. 608–618). Cham: Springer International Publishing, http://dx.doi.org/10.1007/978-3-319-76348-4_59.

Chen, R., Gaia, J., & Rao, H. R. (2020). An examination of the effect of recent phishing encounters on phishing susceptibility. *Decision Support Systems*, *133*, Article 113287. http://dx.doi.org/10.1016/j.dss.2020.113287.

Chiew, K. L., Chang, E. H., Sze, S. N., & Tiong, W. K. (2015). Utilisation of website logo for phishing detection. *Computers & Security*, *54*, 16–26. http://dx.doi.org/10.1016/j.cose.2015.07.006, Secure Information Reuse and Integration & Availability, Reliability and Security 2014.

Chiew, K. L., Chang, E. H., Tan, C. L., Abdullah, J., Sheng, K., & Yong, C. (2018). Building standard offline anti-phishing dataset for benchmarking. *International Journal of Engineering & Technology*, *7*(4.31), 7–14. http://dx.doi.org/10.14419/ijet.v7i4.31.23333.

Chiew, K. L., Tan, C. L., Wong, K., Yong, K. S., & Tiong, W. K. (2019). A new hybrid ensemble feature selection framework for machine learning-based phishing detection system. *Information Sciences*, *484*, 153–166. http://dx.doi.org/10.1016/j.ins.2019.01.064.

Chiew, K. L., Yong, K. S. C., & Tan, C. L. (2018). A survey of phishing attacks: Their types, vectors and technical approaches. *Expert Systems with Applications*, *106*, 1–20. http://dx.doi.org/10.1016/j.eswa.2018.03.050.

Dalgic, F. C., Bozkir, A. S., & Aydos, M. (2018). Phish-IRIS: A new approach for vision based brand prediction of phishing web pages via compact visual descriptors. In *2018 2nd international symposium on multidisciplinary studies and innovative technologies (ISMSIT)* (pp. 1–8). http://dx.doi.org/10.1109/ISMSIT.2018.8567299.

Das, M., Saraswathi, S., Panda, R., Mishra, A. K., & Tripathy, A. K. (2020). Exquisite analysis of popular machine learning–based phishing detection techniques for cyber systems. *Journal of Applied Security Research*, 1–25. http://dx.doi.org/10.1080/19361610.2020.1816440.

Ding, Y., Luktarhan, N., Li, K., & Slamu, W. (2019). A keyword-based combination approach for detecting phishing webpages. *Computers & Security*, *84*, 256–275. http://dx.doi.org/10.1016/j.cose.2019.03.018.

Dou, Z., Khalil, I., Khreishah, A., Al-Fuqaha, A., & Guizani, M. (2017). Systematization of knowledge (SoK): A systematic review of software-based web phishing detection. *IEEE Communications Surveys & Tutorials*, *19*(4), 2797–2819. http://dx.doi.org/10.1109/COMST.2017.2752087.

Dunlop, M., Groat, S., & Shelly, D. (2010). GoldPhish: Using images for content-based phishing analysis. In *2010 fifth international conference on internet monitoring and protection* (pp. 123–128). http://dx.doi.org/10.1109/ICIMP.2010.24.

El Aassal, A., Baki, S., Das, A., & Verma, R. M. (2020). An in-depth benchmarking and evaluation of phishing detection research for security needs. *IEEE Access*, *8*, http://dx.doi.org/10.1109/ACCESS.2020.2969780.

Fahmy, H. M., & Ghoneim, S. A. (2011). PhishBlock: A hybrid anti-phishing tool. In *2011 international conference on communications, computing and control applications (CCCA)* (pp. 1–5). Hammamet, Tunisia: IEEE, http://dx.doi.org/10.1109/CCCA.2011.6031523.

Goel, D., & Jain, A. K. (2018). Mobile phishing attacks and defence mechanisms: State of art and open research challenges. *Computers & Security*, *73*, 519–544. http://dx.doi.org/10.1016/j.cose.2017.12.006.

Gowtham, R., & Krishnamurthi, I. (2014). A comprehensive and efficacious architecture for detecting phishing webpages. *Computers & Security*, *40*, 23–37. http://dx.doi.org/10.1016/j.cose.2013.10.004.

Gupta, B. B., Arachchilage, N. A. G., & Psannis, K. E. (2018). Defending against phishing attacks: taxonomy of methods, current issues and future directions. *Telecommunication Systems*, *67*, 247–267. http://dx.doi.org/10.1007/s11235-017-0334-z, Secure Information Reuse and Integration & Availability, Reliability and Security 2014.

Gupta, B. B., Yadav, K., Razzak, I., Psannis, K., Castiglione, A., & Chang, X. (2021). A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment. *Computer Communications*, *175*, 47–57. http://dx.doi.org/10.1016/j.comcom.2021.04.023.

He, M., Horng, S.-J., Fan, P., Khan, M. K., Run, R.-S., Lai, J.-L., Chen, R.-J., & Sutanto, A. (2011). An efficient phishing webpage detector. *Expert Systems with Applications*, *38*(10), 12018–12027. http://dx.doi.org/10.1016/j.eswa.2011.01.046.

Hong, J., Kim, T., Liu, J., Park, N., & Kim, S.-W. (2020). Phishing URL detection with lexical features and blacklisted domains. In *Adaptive autonomous secure cyber systems* (pp. 253–267). Cham: Springer International Publishing, http://dx.doi.org/10.1007/978-3-030-33432-1_12.

Jain, A. K., & Gupta, B. B. (2017). Phishing detection: Analysis of visual similarity based approaches. *Security and Communication Networks*, *2017*(i), 1–20. http://dx.doi.org/10.1155/2017/5421046.

Jain, A., & Gupta, B. (2018a). Towards detection of phishing websites on client-side using machine learning based approach. *Telecommunication Systems*, *68*(4), 687–700. http://dx.doi.org/10.1007/s11235-017-0414-0.

Jain, A. K., & Gupta, B. (2018b). PHISH-SAFE: URL features-based phishing detection system using machine learning. *Advances in Intelligent Systems and Computing*, *729*, 467–474. http://dx.doi.org/10.1007/978-981-10-8536-9_44.

Jain, A. K., & Gupta, B. (2022b). A survey of phishing attack techniques, defence mechanisms and open research challenges. *Enterprise Information Systems*, *16*(4), 527–565. http://dx.doi.org/10.1080/17517575.2021.1896786.

Levenshtein, V. I. (1965). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics. Doklady*, *10*, 707–710.

Li, Y., Yang, Z., Chen, X., Yuan, H., & Liu, W. (2019). A stacking model using URL and HTML features for phishing webpage detection. *Future Generation Computer Systems*, *94*, 27–39. http://dx.doi.org/10.1016/j.future.2018.11.004.

Marchal, S., Armano, G., Grondahl, T., Saari, K., Singh, N., & Asokan, N. (2017). Off-the-hook: An efficient and usable client-side phishing prevention application. *IEEE Transactions on Computers*, *66*(10), 1717–1733. http://dx.doi.org/10.1109/TC.2017.2703808.

Marchal, S., François, J., State, R., & Engel, T. (2014a). PhishScore: Hacking phishers' minds. In *10th international conference on network and service management (CNSM) and workshop* (pp. 46–54). http://dx.doi.org/10.1109/CNSM.2014.7014140.

Marchal, S., François, J., State, R., & Engel, T. (2014b). PhishStorm: Detecting phishing with streaming analytics. *IEEE Transactions on Network and Service Management*, *11*(4), 458–471. http://dx.doi.org/10.1109/TNSM.2014.2377295.

Marchal, S., Saari, K., Singh, N., & Asokan, N. (2016). Know your phish: Novel techniques for detecting phishing sites and their targets. In *2016 IEEE 36th international conference on distributed computing systems (ICDCS)* (pp. 323–333). http://dx.doi.org/10.1109/ICDCS.2016.10.

McGrath, D. K., & Gupta, M. (2008). Behind phishing: An examination of phisher modi operandi. In *First USENIX workshop on large-scale exploits and emergent threats, LEET '08, San Francisco, CA, USA, April 15, 2008, proceedings*. USENIX Association, URL: http://www.usenix.org/events/leet08/tech/full_papers/mcgrath/mcgrath.pdf.

Moghimi, M., & Varjani, A. Y. (2016). New rule-based phishing detection method. *Expert Systems with Applications*, *53*, 231–242. http://dx.doi.org/10.1016/j.eswa.2016.01.028.

Mohammad, R. M., Thabtah, F., & McCluskey, L. (2012). An assessment of features related to phishing websites using an automated technique. In *2012 international conference for internet technology and secured transactions* (pp. 492–497).

Mohammad, R. M., Thabtah, F., & McCluskey, L. (2015a). Phishing website dataset : UCI machine learning repository. URL: https://archive.ics.uci.edu/ml/machine-learning-databases/00327/Training%20Dataset.arff. accessed 28.10.2020.

Mohammad, R. M., Thabtah, F., & McCluskey, L. (2015b). Tutorial and critical analysis of phishing websites methods. *Computer Science Review*, *17*, 1–24. http://dx.doi.org/10.1016/j.cosrev.2015.04.001.

Moore, T., & Clayton, R. (2007). Examining the impact of website take-down on phishing. In *eCrime '07, Proceedings of the anti-phishing working groups 2nd annual ECrime researchers summit* (pp. 1–13). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/1299015.1299016.

Oest, A., Safei, Y., Doupe, A., Ahn, G. J., Wardman, B., & Warner, G. (2018). Inside a phisher's mind: Understanding the anti-phishing ecosystem through phishing kit analysis. In *2018 APWG symposium on electronic crime research (ECrime)* (pp. 1–12). http://dx.doi.org/10.1109/ECRIME.2018.8376206.

Oest, A., Zhang, P., Wardman, B., Nunes, E., Burgis, J., Zand, A., Thomas, K., Doupé, A., & Ahn, G.-J. (2020). Sunrise to sunset: Analyzing the end-to-end life cycle and effectiveness of phishing attacks at scale. In *USENIX security symposium*.

Prakash, P., Kumar, M., Kompella, R. R., & Gupta, M. (2010). PhishNet: Predictive blacklisting to detect phishing attacks. In *2010 proceedings IEEE INFOCOM* (pp. 1–5). IEEE, http://dx.doi.org/10.1109/INFCOM.2010.5462216.

Rao, R. S., & Pais, A. R. (2017). Detecting phishing websites using automation of human behavior. In *CPSS 2017 - proceedings of the 3rd ACM workshop on cyber-physical system security, co-located with ASIA CCS 2017* (pp. 33–42). http://dx.doi.org/10.1145/3055186.3055188.

Rao, R. S., & Pais, A. R. (2019a). Detection of phishing websites using an efficient feature-based machine learning framework. *Neural Computing and Applications*, *31*(8), 3851–3873. http://dx.doi.org/10.1007/s00521-017-3305-0.

Rao, R. S., & Pais, A. R. (2019b). Jail-Phish: An improved search engine based phishing detection system. *Computers & Security*, *83*, 246–267. http://dx.doi.org/10.1016/j.cose.2019.02.011.

Rao, R. S., Vaishnavi, T., & Pais, A. R. (2020). CatchPhish: detection of phishing websites by inspecting URLs. *Journal of Ambient Intelligence and Humanized Computing*, *11*(2), 813–825. http://dx.doi.org/10.1007/s12652-019-01311-4.

Sadique, F., Kaul, R., Badsha, S., & Sengupta, S. (2020). An automated framework for real-time phishing URL detection. In *2020 10th annual computing and communication workshop and conference, CCWC 2020* (pp. 335–341). http://dx.doi.org/10.1109/CCWC47524.2020.9031269.

Sahingoz, O. K., Buber, E., Demir, O., & Diri, B. (2019). Machine learning based phishing detection from URLs. *Expert Systems with Applications*, *117*, 345–357. http://dx.doi.org/10.1016/j.eswa.2018.09.029.

Sanchez-Paniagua, M., Fernandez, E. F., Alegre, E., Al-Nabki, W., & Gonzalez-Castro, V. (2022). Phishing URL detection: A real-case scenario through login URLs. *IEEE Access*, *10*, 42949–42960. http://dx.doi.org/10.1109/ACCESS.2022.3168681.

Shaikh, A. N., Shabut, A. M., & Hossain, M. (2016). A literature review on phishing crime, prevention review and investigation of gaps. In *2016 10th international conference on software, knowledge, information management & applications (SKIMA)* (pp. 9–15). IEEE, http://dx.doi.org/10.1109/SKIMA.2016.7916190.

Spaulding, J., Upadhyaya, S., & Mohaisen, A. (2016). The landscape of domain name typosquatting: Techniques and countermeasures. In *Proceedings - 2016 11th international conference on availability, reliability and security, ARES 2016* (pp. 284–289). http://dx.doi.org/10.1109/ARES.2016.84.

Tan, C. L., Chiew, K. L., Musa, N., & Ibrahim, D. H. A. (2018). Identifying the most effective feature category in machine learning-based phishing website detection. *International Journal of Engineering & Technology*, *7*(4.31), 1–6. http://dx.doi.org/10.14419/ijet.v7i4.31.23331.

Varshney, G., Misra, M., & Atrey, P. (2017). Improving the accuracy of search engine based anti-phishing solutions using lightweight features. In *2016 11th international conference for internet technology and secured transactions, ICITST 2016* (pp. 365–370). http://dx.doi.org/10.1109/ICITST.2016.7856731.

Verma, R., & Dyer, K. (2015). On the character of phishing URLs. In *Proceedings of the 5th ACM conference on data and application security and privacy - CODASPY '15* (pp. 111–122). New York, New York, USA: ACM Press, http://dx.doi.org/10.1145/2699026.2699115.

Whittaker, C., Ryner, B., & Nazif, M. (2010). Large-scale automatic classification of phishing pages. In *Proceedings of the network and distributed system security symposium, NDSS 2010, San Diego, California, USA, 28th february - 3rd march 2010*. The Internet Society.

Xiang, G., & Hong, J. I. (2009). A hybrid phish detection approach by identity discovery and keywords retrieval. In *WWW '09*, *Proceedings of the 18th international conference on world wide web* (pp. 571–580). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/1526709.1526786.

Xiang, G., Hong, J., Rose, C. P., & Cranor, L. (2011). CANTINA+: A feature-rich machine learning framework for detecting phishing web sites. *ACM Transactions on Information and System Security*, *14*(2), 1–28. http://dx.doi.org/10.1145/2019599.2019606.

Yang, L., Zhang, J., Wang, X., Li, Z., Li, Z., & He, Y. (2021). An improved ELM-based and data preprocessing integrated approach for phishing detection considering comprehensive features. *Expert Systems with Applications*, *165*, http://dx.doi.org/10.1016/j.eswa.2020.113863.

Zhang, Y., Hong, J., & Cranor, L. (2007). Cantina: A content-based approach to detecting phishing web sites. In *16th international world wide web conference, WWW2007* (pp. 639–648). http://dx.doi.org/10.1145/1242572.1242659.

Manuel Sánchez Paniagua received the B.Sc. degree in Computer Science and the M.Sc degree in Cybersecurity Research, both from the University of León in 2019 and 2021 respectively. He is currently a researcher in GVIS group at the University of León. His research interest includes anti-phishing solutions, fraud e-commerce website detection, web scrapping, cyber threat intelligence and machine learning.

Eduardo Fidalgo Fernández received the M. Sc. (2008) degree in Industrial Engineering and the Ph.D. degree in 2015, both from University of León. He is currently the Coordinator of the Group for Vision and Intelligent Systems (GVIS), whose objective is the research and development of solutions to problems related to cybersecurity for INCIBE (https://www.incibe.es/en), by using Artificial Intelligence. His current research interests are in the field of Natural Language Processing, Computer Vision, Machine and Deep Learning.

Enrique Alegre received the M.Sc. degree in electrical engineering from the University of Cantabria, in 1994, and the Ph.D. degree from the University of León, Spain, in 2000. Currently, he is the Head of the Research Group for Vision and Intelligent Systems (GVIS) and Full Professor with the Department of Electrical, Systems, and Automation, University of León. His research interests include computer vision and pattern recognition applications to medical and industrial problems and, more recently, machine learning and computer vision applications for crime control and prevention.

Rocío Alaiz-Rodríguez received the B.Sc. degree in Electrical Engineering from the University of Valladolid, Spain, in 1999 and the Ph.D. degree from Carlos III University of Madrid, Spain in 2005. She is currently an Associate Professor at the University of Leon, Spain. Her research interests include machine learning, statistical pattern recognition, neural networks and the dataset shift problem.