IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 23 January 2023, accepted 13 February 2023, date of publication 22 February 2023, date of current version 27 February 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3247135

SURVEY

Phishing or Not Phishing? A Survey on the Detection of Phishing Websites

RASHA ZIENI, LUISA MASSARI, AND MARIA CARLA CALZAROSSA, (Senior Member, IEEE)
Department of Electrical, Computer and Biomedical Engineering, Università di Pavia, 27100 Pavia, Italy

Corresponding author: Luisa Massari (luisa.massari@unipv.it)

This work was supported in part by the University of Pavia through the CRUI-CARE Agreement.

ABSTRACT Phishing is a security threat with serious effects on individuals as well as on the targeted brands. Although this threat has been around for quite a long time, it is still very active and successful. In fact, the tactics used by attackers have been evolving continuously in the years to make the attacks more convincing and effective. In this context, phishing detection is of primary importance. The literature offers many diverse solutions that cope with this issue and in particular with the detection of phishing websites. This paper provides a broad and comprehensive review of the state of the art in this field by discussing the main challenges and findings. More specifically, the discussion is centered around three important categories of detection approaches, namely, list-based, similarity-based and machine learning-based. For each category we describe the detection methods proposed in the literature together with the datasets considered for their assessment and we discuss some research gaps that need to be filled.

INDEX TERMS Phishing, security threat, phishing website, phishing detection, URL, blacklists, machine learning, page similarity, datasets, social engineering.

I. INTRODUCTION

Phishing is a dangerous security threat that exploits sophisticated psychological and social engineering techniques to trick individuals into clicking links of malicious websites and submit highly valuable sensitive information, such as personal or corporate information and account credentials.

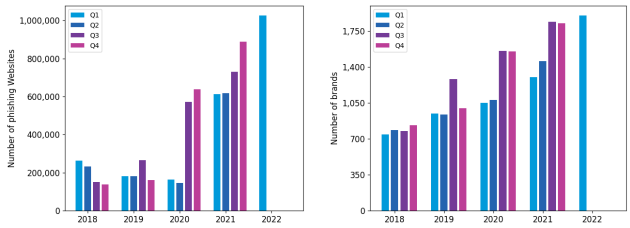
Phishing attacks are far from being technologically complex and their deployment requires little effort. Nevertheless, they are generally very effective. Attackers create well-crafted phishing websites with a look and feel of the legitimate sites they are trying to impersonate, thus making it very challenging for individuals to identify phishing sites. In addition, to avoid being detected, attackers have refined over the years their tactics and evasion techniques, as demonstrated in [1].

Phishing attacks have several direct and indirect impacts. They affect the individuals being phished, whose identity and accounts might be compromised, thus leading to money being stolen as well as to a potential crisis of trust towards online services. These attacks also affect the companies and organizations being impersonated, whose brands might be abused, thus leading to potential data breaches, financial losses and reputation damages.

A study by Enisa [2] reveals that phishing attacks are among the most common cyber incidents European small-medium enterprises are likely to be exposed to. In the Cybersecurity threat trends report [3] Cisco suggests that in 2020 phishing accounts for around 90% of data breaches. Moreover, 86% of organizations had at least one user try to connect to a phishing site. In fact, as discussed in [4], individuals tend to fall prey of phishing attacks especially because of the insufficient attention paid in assessing the legitimacy of a website and the lack of appropriate education.

According to the Phishing activity trends report [5] by Anti-Phishing Working Group (APWG), the total number of phishing websites observed in the first quarter of 2022 exceeds one million. As shown in Figure 1(a), this surge is particularly evident since the beginning of the Covid-19 pandemic. These effects are also evident in the unique brands being targeted by phishing campaigns whose number has increased significantly since the third quarter of 2020 (see Fig. 1(b)).

The associate editor coordinating the review of this manuscript and approving it for publication was Junho Hong.



(a) Phishing websites (b) Brands

FIGURE 1. Quarterly trends of the number of unique phishing websites detected (a) and of unique brands targeted by phishing campaigns (b) since 2018. [Source: *APWG*].

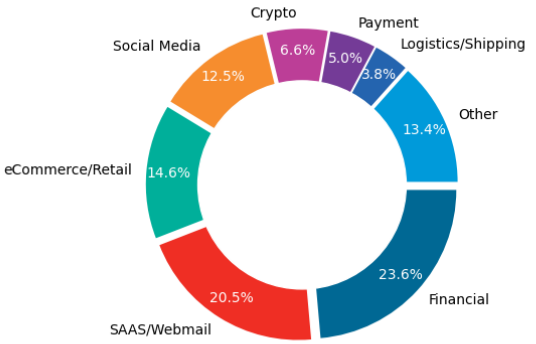


FIGURE 2. Most targeted industry sectors in the first quarter of 2022. [Source: *APWG*].

Another interesting result reported by APWG refers to the industry sectors most targeted by attackers. Financial services, which include banks, are particularly prone to phishing. As shown in Figure 2, in the first quarter of 2022, this sector was the most frequently victimized by phishing, with 23.6% of all attacks. Webmail and Software-As-A-Service providers have also been targeted by a large fraction of the attacks (i.e., 20.5%), whereas fewer attacks (i.e., 3.8%) were directed towards logistic/shipping sectors.

All these figures demonstrate that phishing is a very active security threat. Hence, to protect individuals against phishing attacks, powerful solutions capable of detecting and fighting these attacks in a timely manner become compelling. In this context, research is playing a key role for mitigating the impact of such a serious threat.

In the years, the detection of phishing websites has been widely investigated and a large body of the literature has addressed this challenging problem.

Our survey aims at providing a broad and comprehensive review of the state of the art in the area of phishing website detection by focusing on the most relevant solutions proposed in the literature. In particular, we subdivide these solutions in three main categories according to their target, namely:

- List-based;
- Similarity-based;
- Machine learning-based.

For each category we describe the suggested detection methods and the datasets considered for their assessment. In addition, we discuss the main strengths and weaknesses of these approaches and identify the most important research gaps that need to be filled.

The rest of this paper is organized as follows. After a brief overview of the anatomy of phishing attacks presented in Section II, Section III covers the methodological approach adopted in this survey. A comparison of our work with existing surveys is provided in Section IV, while a comprehensive review of the approaches proposed for detecting phishing websites is presented in the following sections. In particular, list-based approaches are covered in Section V, similarity-based approaches in Section VI, while feature representations and learning algorithms in Section VII. The main lessons learnt from the analysis of the state of the art are summarized in Section VIII. Finally, some concluding remarks and possible research directions are presented in Section IX.

II. ANATOMY OF PHISHING ATTACKS

As already pointed out, phishing attacks are technologically simple to implement. Figure 3 sketches a typical scenario of a phishing attack. In this scenario, the attacker has two main roles, namely, creating websites that look very similar to the sites being impersonated and spreading the corresponding links using various communication channels, such as email or social media. By clicking these links, individuals are directed to malicious websites where they might end up disclosing sensitive data. Eventually attackers will monetize this data either directly or indirectly. For example, they could leverage the hijacked accounts for performing illegal online transactions or simply sell the collected data on the marketplace.

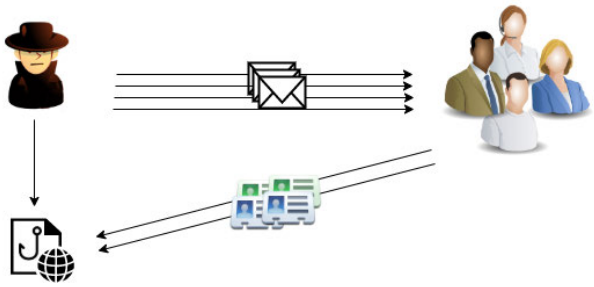


FIGURE 3. Main actors involved in a phishing attack.

To gain the trust of the individuals, attackers make the link and website appear legitimate using various tricks, such as typosquatting and combosquatting techniques. For example, they craft the patterns of the Uniform Resource Locator (URL) – shown in address bar of the browser – by inserting unnecessary punctuation marks (e.g., dash), misspelled words (e.g., paymet) or specific words (e.g., brand name being targeted) in incorrect positions. Sometimes, attackers replace English characters with identical looking characters from different alphabets. In fact, although malicious sites might be hosted on compromised servers, attackers might choose to register specific domains with appropriately crafted names. Moreover, attackers tend not to use phishing URLs multiple

times due to the low cost of generating new ones, thus making the detection of phishing websites even more challenging.

Let us recall that a URL is a human-readable string of characters – parsed by client programs in a standard way – uniquely identifying a resource on the web [6]. As shown in Figure 4, a URL consists of several components referring to the protocol used to access the resource identified by the path as well as the fully qualified domain name of the server hosting the resource – denoted as authority – that includes the Top-Level Domain name (TLD) and the second level domain name. The URL string also contains some optional parameters (i.e., query) used for specific purposes.

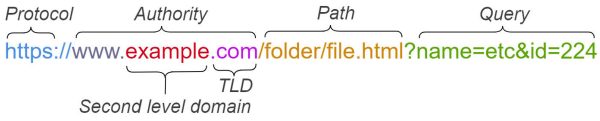


FIGURE 4. Example of a URL with its components.

To quickly design phishing pages whose layout and content mimic their legitimate counterparts, attackers often exploit phishing toolkits. These kits help them automate their phishing campaigns by providing many diverse functionalities, such as website cloning, page template creation and modification, templates in different languages, back-end operational support. As a consequence, phishing websites might appear as nearly flawless replicas of the targeted brands. Nevertheless, it is worth mentioning that attackers generally tend to focus on the visual appearance of their phishing pages, rather than on the actual content of the page source codes. Hence, it might happen that the content associated with HTML tags is not fully customized to the brand being targeted.

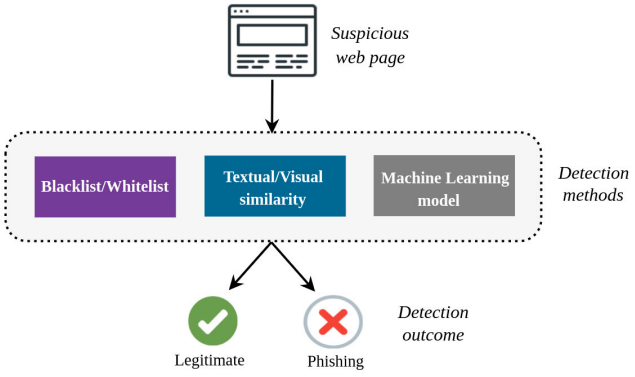


FIGURE 5. Phishing detection model.

These challenging scenarios call for the development of accurate phishing detection systems as well as for specific actions aimed at educating individuals and making them aware of this dangerous security threat. A general diagram of a phishing detection model is sketched in Figure 5. As can be seen, starting from a suspicious web page – described by its URL or source codes – different methods (i.e., list-based, similarity-based, machine learning-based) are applied to detect whether the page is legitimate or phishing. More precisely, the detection outcome can be provided by simply checking blacklists and whitelists, by comparing the content and visual appearance of the page with its legitimate counterpart or by a machine learning model. Note that these methods can also be used in combination to make the detection more effective.

In what follows, we discuss in detail the solutions proposed in the literature for detecting phishing websites.

III. METHODOLOGY

Our literature review is based on research and survey papers addressing the detection of phishing websites. We selected these papers from major bibliographic databases, such as ACM Digital Library, DBLP, Google Scholar, IEEE Xplore and Scopus. We started by searching the Scopus database – one of the largest abstract and citation database of peer-reviewed literature – through a very broad query consisting of one word only, namely, ''phishing''. From this query we obtained all papers – namely, about 4,400 – published until 2021 and including the word phishing in their title, abstract or keywords.

For refining the results of this query, we scrutinized these papers and performed a semi-automatic inspection that allowed us to identify the papers pertinent for our survey and discard the others. For example, we discarded papers not focused on phishing detection as well as papers addressing side topics, such as phishing email, spear phishing, phishing training. After this inspection, we retained a set of 600 papers, that is, 13.6% of the papers.

As a further refinement, we considered as a selection criterion the relevance of these papers. For this purpose we collected additional information from various bibliographic databases. In total, we retained 127 papers. This set includes pioneering works as well as papers selected according to characteristics, such as venues where they were published and interest of the research community – measured in terms of number of citations and number of views and downloads.

Figure 6 illustrates the distribution of these papers as a function of the publication year and of the detection approach being adopted, namely, list-based, similarity-based and machine learning-based. It is interesting to point out the prevalence of papers in the area of machine learning. In fact, these approaches are particularly suitable in identifying previously unseen phishing web pages, thus coping with zero-hour attacks.

IV. COMPARISON WITH EXISTING SURVEYS

Phishing is a popular topic that has been researched over the years under different perspectives. Several surveys have summarized the state of the art in this field. Some of them analyze general aspects, such as attack strategies, training and education approaches (see, e.g., [7], [8], [9]), whereas

TABLE 1. Summary of the surveys on phishing website detection. Papers are listed alphabetically according to the first author's lastname.

Paper	Objectives	Limitations
Basit et al. [14]	Review of machine learning, deep learning, hybrid learning and scenario-based approaches for phishing attack detection.	Very high-level analysis of the detection approaches. Review of papers published between 2011 and 2020. No discussion of pioneering works in the field of machine learning.
Das et al. [15]	Review of machine learning techniques for phishing detection of URLs, websites and emails from a security perspective.	Review of papers published until 2017. No comparison with alternative detection techniques.
Dou et al. [16]	Review of software-based web phishing detection schemes.	Review of papers published until 2016. Rather narrow, although detailed, analysis of the state of the art.
Jain and Gupta [17]	Review of web phishing detection approaches based on visual similarity.	Review of papers published until 2016. No discussion of extensively researched approaches, such as machine learning approaches.
Khonji et al. [18]	Review of various types of phishing mitigation techniques for email and web.	Review of papers published until 2011. Limited discussion on phishing website detection.
Sahoo et al. [11]	Review of feature representations and machine learning algorithms for malicious URL detection.	Review of papers published until 2019. No discussion of detection approaches based on the combined use of different types of content.
Varshney et al. [13]	Review of the strategies proposed for web phishing detection.	Review of papers published until 2015. Machine learning approaches covered to a rather limited extent.

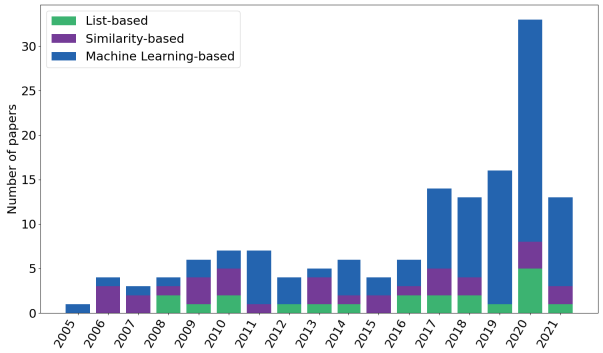


FIGURE 6. Distribution of the papers considered in this survey as a function of the publication year and of the detection approach adopted, i.e., list-based, similarity-based and machine learning-based.

some others specifically focus on detection and prevention approaches (see, e.g., [10], [11], [12], [13]).

In what follows we review the surveys mainly dedicated to the detection of phishing websites and we highlight their objectives and potential limitations (see Table 1 for a comparative summary).

One of the earliest surveys [18] presents an interesting review of the various types of phishing mitigation techniques applied to web as well as to email, namely, detection, offensive defense, correction and prevention. In the context of phishing detection, the focus is on software solutions as well on solutions based on user awareness. Software solutions are classified in four categories, i.e., blacklists, heuristics, visual similarity and data mining techniques. For each category, the approaches suggested in some papers are discussed and evaluated in detail. Nevertheless, the number of analyzed papers is very limited – especially for what concerns the detection of phishing websites based on machine learning techniques.

A detailed review of the strategies offered in the literature for the detection of phishing websites is presented in [13]. These strategies are subdivided in six categories according to the techniques they are based upon, namely, search-based, heuristics and machine learning, black and whitelists, DNS-based, visual similarity, and proactive phishing URL based techniques. The advantages and disadvantages of the various strategies are discussed in detail. Nevertheless, unlike our survey, this survey does not mention the selection criteria of the papers being discussed, thus their relevance is not clear. In addition, the survey mainly focuses the detection approaches based on heuristics, while machine learning approaches are covered to a rather limited extent.

Jain and Gupta [17] offer a comprehensive review of phishing detection approaches based on visual similarity. These approaches are classified according to the types of features (e.g., visual, pixel-based, hybrid) used to compare the suspicious web pages and their legitimate counterparts. The main advantages and limitations of the proposed solutions are discussed. Nevertheless, unlike our work, this survey addresses visual similarity approaches only and does not consider other prominent state of the art approaches.

In the framework of machine learning, Sahoo et al. [11] focus on page URLs and analyze the approaches adopted for detecting malicious URLs. A thorough discussion and categorization of feature representations and learning algorithms are presented. Some practical issues related to the design of detection systems (e.g., data volume, labeling process) are also discussed. Although the paper identifies features associated with different types of content (e.g., HTML features, visual features), the approaches that take advantage of multiple types of content have not been specifically discussed.

Another survey focused on machine learning approaches is offered by Das et al. [15]. This survey examines phishing detection for different attack vectors, namely, URLs, websites and emails, by taking a security perspective that considers the needs of this domain. The solutions proposed in the literature are classified according to the learning method

applied for the detection (e.g., supervised, unsupervised, rule-based). An interesting aspect suggested in this survey refers to the effects on the detection systems of the computation and storage requirements associated with feature extraction. Nevertheless, unlike our survey, this survey focuses on machine learning techniques only and does not provide any comparison with alternative detection methods.

AI-enabled phishing detection – based on machine learning, deep learning, hybrid learning and scenario-based techniques – is addressed in [14]. This survey only provides a very high-level analysis of the detection approaches proposed in the literature. In addition, despite the intense research in the field, very few papers are discussed and pioneering works based on machine learning have not been considered.

Dou et al. [16] present a systematic literature review of software-based web phishing detection schemes from different perspectives including the life cycle, taxonomy, evaluation datasets, detection features, detection techniques and evaluation metrics. In the context of detection techniques, the survey analyzes 41 paper selected according to three criteria, namely, pioneering character, attention and completeness. Twelve of these papers – chosen as representative examples of the various categories of detection methods (e.g., page content-based, URL-based, hybrid) – are explained thoroughly.

Our survey complements [16]. In fact, instead of focusing on few papers, we select a bigger set of relevant papers in the context of three main categories of detection methods, that is, list-based, similarity-based and machine learning-based. In fact, our primary objective is to allow readers to easily navigate the state of the art of this extensively investigated research field by providing them with a broad, comprehensive and up-to-date overview of the main solutions proposed in the literature for detecting phishing websites. We believe this is an important added value of our survey that, to the best of our knowledge, none of existing surveys offers. Another important added value is represented by the comparison of the three categories of detection methods which highlights their main strengths and weaknesses.

V. LIST-BASED DETECTION METHODS

The primary and simpler defense to protect individuals from dangerous websites consists in maintaining blacklists of known phishing URLs and whitelists of trusted URLs. Most browsers integrate by default regularly updated blacklists. For example, Google Chrome and Mozilla Firefox support the Google Safe Browsing service [19]. This service generates a warning whenever individuals try to navigate any known malicious website. An example of such a warning is shown in Figure 7.

Detection mechanisms based on blacklists are simple to implement, fast and accurate but they are not completely effective since they are inherently reactive, thus they fail to defend against the so-called zero-hour attacks. In fact, lists are updated at different speeds only after an attack takes place and sometimes even several hours later [20], [21].

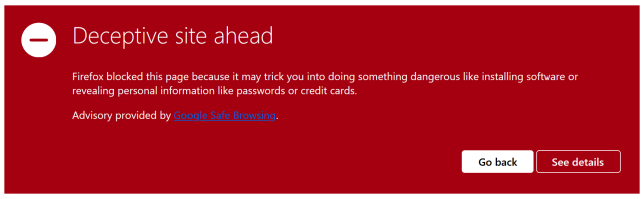


FIGURE 7. Warning provided by the Mozilla Firefox browser.

In addition, it has been demonstrated that cloaking techniques used by attackers to avoid detection often introduce delays in recognizing the attacks and in updating the blacklists [22], [23]. On the contrary, whitelists are very useful to identify trusted and suspicious websites. Any URL not included in a whitelist should be considered as suspicious. Nevertheless, it is practically unfeasible to maintain global lists of all possible websites.

A. LIST CREATION AND MAINTENANCE

To overcome the issues previously discussed, several papers suggest methods for creating and maintaining blacklists and whitelists (see, e.g., [20], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33]). The details of the main research efforts offered in this context are discussed in what follows, whereas an overview of these efforts is presented in Table 2.

1) BLACKLISTS

In the context of blacklists, the research efforts focus on improving the incompleteness of these lists by predicting the phishing URLs to be included. The approaches proposed to populate blacklists mainly differ in terms of the heuristics and techniques applied for this purpose.

For example, Prakash et al. [29] suggest an offline approach where new URLs are created from blacklisted ones by applying heuristics based on URL lexical similarity (e.g., top level domain equivalence, directory structure similarity). These heuristics take into account the behavior of attackers who tend to implement simple URL modifications. The URLs being generated are validated before including them into blacklists. This process – performed through DNS lookup and content matching – aims at discarding the URLs that are either non-existent or do not correspond to malicious web pages.

A probabilistic detection approach that exploits the content similarity of web pages built using phishing toolkits is presented in [33]. To identify near-duplicate phishing pages, the approach combines human-verified blacklists with the shingling algorithm. In addition, to further scrutinize potential phishing pages, search engines are queried with content extracted by means of information retrieval techniques.

Similarly, Rao and Pais [30] focus on variants of phishing web pages, namely, pages which nearly duplicate blacklisted ones. To detect phishing web pages and update blacklists, the fingerprints of suspicious and blacklisted pages are compared

TABLE 2. Overview of the papers addressing list-based approaches. Papers are listed in alphabetic order according to first author's lastname.

Paper	List type	Focus	Techniques	Third-party services
Cao et al. [25]	Whitelist	Login user interface	Machine learning	DNS
Jain and Gupta [27]	Whitelist	Hyperlinks	Custom	DNS
Lee et al. [28]	Blacklist	URL	Custom	None
Prakash et al. [29]	Blacklist	URL	Heuristics	DNS
Rao and Pais [30]	Blacklist	Page source code	Hashing	None
Sonowal and Kuppusamy [34]	Whitelist	URL	Information retrieval, edit distance	Search engine
Xiang et al. [33]	Blacklist	Page source code	Natural Language Processing, Information retrieval	Search engine

using the Hamming distance and a threshold. These fingerprints are generated using features extracted from the source codes of individual pages.

A different perspective is adopted in [28] to identify suspicious web pages and populate blacklists as early as possible. More precisely, new URLs are detected by tracking the redirections extracted from blacklisted URLs and by following the phishing forms iteratively.

2) WHITELISTS

In the context of whitelists, the research efforts focus on mechanisms for creating and maintaining individual whitelists. These mechanisms take into account specific aspects of the browsing behaviors of the users (e.g., login process, visited websites).

For example, to update the whitelist of a given user, Cao et al. [25] consider the login user interfaces of the websites visited by the user. More precisely, the information about these interfaces, e.g., URL, DNS-IP mapping, is automatically added to the whitelist after a certain number of successful logins of the user. A Naïve Bayesian classifier is applied to identify successful login processes.

The websites accessed by a given user are considered in [27] for auto-updating the corresponding whitelist. In detail, before updating the list, the legitimacy of the page is checked. This check is based on hyperlink features extracted from the page source code. In fact, hyperlinks are good for discriminating phishing and legitimate pages since phishing pages often include hyperlinks pointing to their legitimate counterparts.

The whitelist updates suggested in [34] are based on a multilayer model whose goal is to assess the legitimacy of a URL. The model consists of various filters, each dedicated to analyze a specific aspect of the URL (e.g., features, lexical signature). To verify a URL, the model also relies on the results of a search engine queried using the page signature. In fact, legitimate web pages are usually highly ranked by search engines.

B. DATASETS

As already discussed, blacklists and whitelists are populated using different approaches that take into account the behaviors of attackers as well as of the individuals. For the evaluation of these approaches, collections of phishing and legitimate websites taken from various sources are considered. For example, popular sources of malicious URLs are represented by PhishTank [35] – a community based phishing website reporting and verification system – and by the Safe Browsing lists provided by Google. Similarly, Alexa – a service providing top-ranked domains retired in May 2022 – and DMOZ – an open directory of the web discontinued in 2017 and now replaced by Curlie [36] – used to be the sources of benign URLs.

C. DISCUSSION

List-based approaches are valid for detecting phishing websites because of their high accuracy coupled with a low overhead. Nevertheless, these approaches suffer from limitations mainly related to their reactive nature that makes them unable to cope with zero-hour attacks.

In this context, the creation and maintenance of blacklists and whitelists play a key role. The analysis of the state of the art has highlighted some interesting findings that can be summarized by the following recommendations:

- List creation and updates should be based on lightweight mechanisms not to introduce delays in the detection process;
- Lists should be constantly updated to defend against newly discovered phishing attacks;
- Rules and heuristics devised for creating and updating the lists should reflect in a timely manner the evolution of the tactics adopted by attackers.

In conclusion, our investigation has shown the importance to properly populate black and whitelists. In fact, list-based approaches are often used in conjunction with other approaches to reduce false positive rate.

VI. PAGE SIMILARITY-BASED DETECTION METHODS

Page similarity has been extensively investigated in the literature to detect phishing web pages (see, e.g., [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], [47], [48], [49], [50], [51], [52], [53], [54], [55], [56], [57], [58]). In fact, as already pointed out, web pages are made in a way to look very similar or identical to their legitimate counterparts, thus their similarity is a good indicator of a phishing attack.

Similarity is commonly measured by comparing the content of suspicious pages with the content of legitimate pages identified as potential targets of the attack. As Figure 8 shows, the content to be matched can be subdivided into two main categories, namely:

- *Textual content* referring to text-based components describing the structure and text of web pages, such as HTML and CSS source codes, Document Object Models (DOM);
- *Visual content* referring to image-based components describing the appearance of web pages, such as page and logo region snapshots.

The techniques applied for quantifying the similarity, i.e., computing the similarity scores, vary and mainly depend on the type of content being considered. The final decision about suspicious pages is generally based on pre-defined thresholds associated with the similarity scores.

In what follows, we explore in detail the main approaches proposed in the literature to assess page similarity in the context of phishing detection (see Table 3 for an overview). We also discuss the main strengths and weaknesses of these approaches.

A. TEXTUAL CONTENT

In the framework of textual content, phishing detection is typically investigated by focusing on the *page text* and *layout* because of their direct relationships with page visual appearance. In particular, the text-based components and the content to be matched refer to:

- HTML source code whose content is represented by the markups used to structure and build a web page;
- Cascading Style Sheet (CSS) whose content is represented by the rules used to specify the appearance of a web page;
- Document Object Model (DOM) whose content is represented by a tree structure describing a web page.

Various techniques are applied to evaluate content similarity. For example, Zhang et al. [57] investigate the phishing detection problem by applying an information retrieval algorithm, i.e., Term Frequency-Inverse Document Frequency (*TF–IDF*) and some heuristics. We recall that the Term Frequency measures the importance of the term within a web page, whereas the Inverse Document Frequency measures the general importance of the term, that is, how common a term is across an entire collection of pages. Hence, the score for the term t_i in a given web page is computed as:

$$TF\text{–}IDF(t_i) = \frac{n_i}{N} \times \log \frac{M}{m_i}$$

where N and n_i denote the number of terms in the page and the frequency of the term t_i, while M and m_i refer to the number of pages in the collection and the number of pages that contain term t_i.

Starting from the textual content of a web page, the *TF–IDF* score is computed for each term that appears in the page and used to generate a lexical signature according to the Robust Hyperlink approach [59]. The signature – consisting of the terms with the highest scores – is fed to a search engine. In detail, the domains of the top results of these queries are matched with the domain of the suspicious page to assess whether it is phishing. In fact, phishing websites are seldom highly ranked by search engines since they are typically active for a very short time.

To distinguish between phishing and legitimate pages, Rosiello et al. [54] focus on the Document Object Model representations of the pages. Since phishing pages generally share the layout of their legitimate counterparts, DOM trees are particularly suitable for assessing layout similarity. In detail, the comparison of two DOM trees is based on HTML tags and isomorphic subtrees extracted by applying graph theory.

Page similarity has also been investigated by considering the influence of page elements on page layout and appearance. Starting from the CSS rules associated with a suspicious web page and with its target counterpart, Mao et al. [52] extract static features describing the visual layout of the pages and normalize these features into vectors. The similarity between these pages is evaluated by considering the complexity of the page layouts as well as their visual appearance.

To cope with the strategies adopted by attackers for avoiding detection, Liu et al. [50] analyze the content of web pages at different levels of detail, i.e., blocks, layout and style. More precisely, pages are segmented into blocks, each described by features associated with their textual and visual content (e.g., background and foreground colors, font size, family and style). The layout of the blocks is explained in terms of their spatial relations, whereas the overall page style is described by features referring to the page appearance (e.g., page title, dominant color of logo, border style and width). The comparison of the features extracted at each level provides a measure of similarity between legitimate and suspicious pages.

B. VISUAL CONTENT

In the framework of visual similarity, phishing detection is typically investigated as an *image matching problem* that takes advantage of traditional computer vision techniques [60], [61]. Two representations of the image-based components to be matched are generally considered, namely:

- *Keypoints*, i.e., points that define what is distinguishable in an image;
- *Signatures*, i.e., descriptors that quantify the image properties.

Techniques, such as Scale-Invariant Feature Transform (SIFT) [62], Speeded-Up Robust Features (SURF) [63], Contrast Context Histogram (CCH) [64], are applied to detect keypoints and construct descriptors of their visual characteristics. For example, in [39] the Harris-Laplacian corners are extracted as keypoints of the images of legitimate and suspicious web pages. Invariant information around the keypoints are captured and the corresponding descriptors

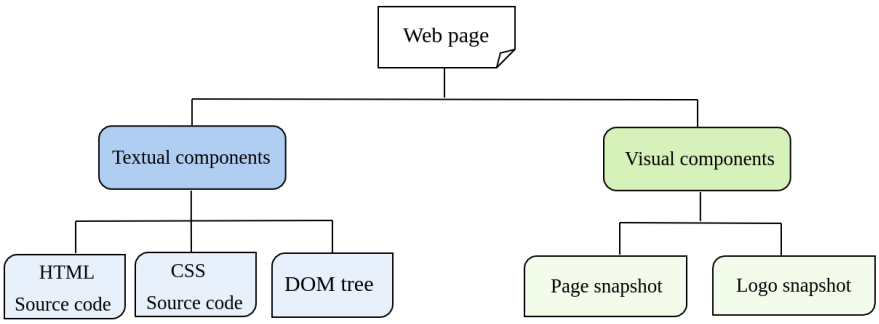


FIGURE 8. Types of content used for measuring page similarity.

TABLE 3. Overview of the papers addressing page similarity. Papers are listed in alphabetic order according to first author's lastname.

Paper	Content			Techniques	Threshold	Data sources	Third-party services
	Type	Representation	Focus				
Afroz and Greenstadt [37]	Textual, Visual	HTML content, URL, Keypoints	Page logo, Page content	TF − IDF, SIFT	Yes	PhishTank, Alexa	None
Chen et al. [39]	Visual	Keypoints	Entire page	Harris-Laplacian corners, CCH, k-means, Euclidean distance	Yes	Custom	None
Dunlop et al. [43]	Textual, Visual	Text	Top of the page	OCR	No	PhishTank, Custom	Google search
Fu et al. [44]	Visual	Signatures	Entire page	EMD	Yes	Custom	None
Huang et al. [46]	Textual, Visual	Keypoints, Signatures	Website	SIFT, Euclidean distance	Yes	Custom	None
Liu et al. [50]	Textual	HTML properties	Page blocks, Page layout, Page style	Custom	Yes	Custom	None
Mao et al. [52]	Textual	CSS rules	Page layout	Custom	Yes	PhishTank, Custom	None
Medvet et al. [53]	Textual, Visual	Signatures	Entire page	Levenshtein, 1-norm, Euclidean distances	Yes	PhishTank, Custom	None
Rosiello et al. [54]	Textual	DOM graphs	Page layout	Graph theory	Yes	PhishTank	None
Zhang et al. [57]	Textual	HTML text	Entire page	TF − IDF, Robust hyperlink	No	PhishTank, Alexa, 3Sharp	Google search
Zhou et al. [58]	Visual	Keypoints, Signatures	Logo, Visible region	SURF, Euclidean distance, EMD	Yes	PhishTank, Custom	None

are computed by applying a lightweight version of CCH. Keypoint matching takes into account both their similarity and spatial location. More precisely, similarity is measured in terms of Euclidean distance between descriptors, while k-means clustering algorithm is applied to find groups of keypoints close to each other.

A representation based on keypoints is also adopted in [37] to detect whether a suspicious web page contains a logo belonging to any legitimate page. In particular, the SIFT algorithm is used to extract scale-invariant features describing the images and to find matching keypoints that are similar to the keypoints of some pre-defined logos.

In the context of content representations based on signatures, Fu et al. [44] investigate the similarity between web pages by comparing the signatures obtained from their low resolution images. These signatures are derived from colors and coordinate features extracted from the pixel level of the images. The similarity between signatures is computed in terms of Earth Mover's Distance (*EMD*), a measure of distance between multi-dimensional distributions extensively applied for image matching [65]. In detail, given the signatures of a legitimate and a phishing web page, i.e., S_a and S_b, each described by m and n features, the *EMD* is computed as follows:

$$EMD(S_a, S_b) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} \times d_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}}$$

where f_{ij} and d_{ij} denote the flow and distance between feature a_i and feature b_j.

Several works propose a combined use of representations based on keypoints and signatures. To detect logo images, in [58] keypoints are extracted from suspicious and legitimate web pages – using the SURF algorithm – and matched according to the Euclidean distance. Moreover, to make the detection more accurate, the visible region of the pages is

described in terms of signatures. Similarly to [44], these signatures are created and compared using *EMD*.

Another combined use of signatures and keypoints is presented in [46]. In this work signatures refer to both textual and visual content of multiple pages of websites. In particular, common image blocks are considered as the distinctive characteristics of websites. To identify these blocks, keypoints are extracted for the images embedded into the web pages – using the SIFT algorithm – and pairwise matched according to the Euclidean distance.

Medvet et al. [53] visually compare signatures of suspicious and legitimate pages by considering both visible text sections and images. Signatures capture the characteristics of individual text pieces and images. Features, such as background and foreground colors, font size, position in the page, describe each individual text section, whereas features, such as color histograms, 2D Haar wavelet transformation, refer to each embedded image as well as to the image representing the overall page. Various types of distances, e.g., Levenshtein, 1-norm, Euclidean, are computed to compare homogeneous components of the signatures and to derive a similarity score.

Visual content has also been analyzed with the objective of extracting textual content. In [43] web page images are processed using optical character recognition (OCR) to convert these images into text. To check the identity and validity of the visited site, this text is submitted to the Google Search API and the page URL is compared with the top and second level domains of the top-ranked results of the search.

C. DATASETS

As already pointed out, the detection methods devised in the context of page similarity are often based on the comparison of suspicious web pages with the legitimate pages identified as potential targets of the phishing attacks. Hence, the datasets of the legitimate pages to be matched play a fundamental role.

Popular sources for creating these datasets are represented by services that provide top-ranked domains, such as Alexa, and by the results of customized queries to search engines, such as Google. In addition, the PhishTank archive has been extensively used as a source for creating datasets of phishing pages.

We outline that, to customize the data being collected according to their requirements, most researchers build their own datasets using one or multiple sources. In general, these datasets are rather small and not publicly available.

D. DISCUSSION

The analysis of the state of the art has demonstrated that approaches based on page similarity are valid for detecting phishing web pages. In fact, textual and visual content provide useful insights to assess the degree of similarity between legitimate and suspicious pages.

In general, detection mechanisms that focus on textual content are faster although easier to bypass. Moreover, they fail whenever text is replaced with images. On the contrary, visual similarity approaches are more robust since they are agnostic to the underlying textual content, even though their effectiveness strongly depends on the techniques adopted for describing the content. Moreover, these approaches are generally rather expensive in terms of computation and storage requirements.

The main weaknesses of similarity-based approaches can be summarized as follows:

- *Effectiveness*: the design of phishing web pages often exploits techniques aimed at evading detection, such as HTML code obfuscation, invisible content, image distortion, image rotation, replacement of HTML text with images or embedded objects, thus reducing the effectiveness of the detection mechanisms or even making them fail;
- *Subjectivity*: assessments based on thresholds might introduce a sort of ''subjectivity'', especially because these thresholds are seldom chosen as a function of the page being considered;
- *Speed*: the approaches based on external services, such as Google search engine, are slow by nature;
- *Storage*: the datasets used for matching suspicious pages should contain as many legitimate pages as possible, thus their storage requirements might become quite large.

In conclusion, a strength of page similarity approaches is the fact that they take precisely into account the behavior of attackers who tend to create websites with a look and feel of the legitimate ones. Nevertheless, because of the weaknesses previously outlined, these approaches have to be used with particular care and possibly combined with other approaches that allow real-time detection.

VII. MACHINE LEARNING-BASED DETECTION METHODS

Machine learning methods for detecting phishing web pages have been extensively researched. The detection is generally formulated as a binary classification problem. A large variety of machine learning algorithms have been applied to solve this problem and obtain models able to classify pages – according to the classification rules and the set of features chosen to describe the properties of the pages – as either legitimate or phishing.

In what follows, we review the features proposed in the literature as a function of the properties chosen to characterize web pages. We also focus on the diverse classification models and on the datasets used to evaluate the performance of the models. In addition, we discuss the main strengths and weaknesses of machine learning-based approaches.

A. FEATURE EXTRACTION

Feature extraction refers to the process of identifying the characteristics that distinguish phishing and legitimate web pages. This process is of paramount importance since the choice of the features influences the accuracy and speed of phishing detection.

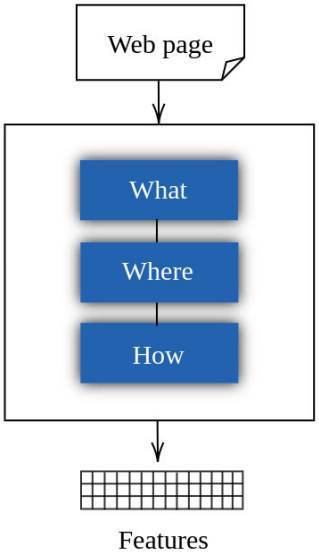


FIGURE 9. Main steps of the feature extraction process.

As Figure 9 shows, starting from a web page, the extraction process is formulated as a sequence of inter-dependent steps corresponding to the following questions:

- *What* are the properties useful for the detection?
- *Where* are the properties being identified derived from?
- *How* are features obtained from these properties?

In detail, *What* refers to the definition of web page properties relevant for the detection, *Where* to the selection of the data sources and *How* to the application of techniques and heuristics for encoding properties into features.

To effectively detect phishing, the properties describing web pages have to take into account the strategies and practices commonly adopted by attackers to create phishing web pages and the characteristics that differentiate these pages from their legitimate counterparts. Hence, a solid domain knowledge is required for this purpose.

Once properties have been identified, it is necessary to select the appropriate content for their description. Figure 10 shows the possible sources of this content. As can be seen, sources are classified in two main categories, namely:

- *Client side sources*: the content – referring to the page URL and source codes (e.g., URL string, page text and appearance) and to the traffic generated by the page downloads – is obtained from the client devices;
- *External sources*: the content – referring to the results of queries performed with specific web page components – is obtained from third-party services, such as search engines, DNS, WHOIS.

We outline that page URL and source codes are the main sources of content, whereas internal data provided by software agents and external data obtained from third-party services are generally used as a way to enrich the description of the web pages.

To characterize the identified properties and encode them into meaningful features, various types of techniques, such as lexical analysis, information retrieval, statistical techniques, Natural Language Processing and heuristics – customized to the diversity of phishing attacks – are applied. Table 4 summarizes the properties and the third-party services considered in some of the most relevant papers for extracting URL-based and HTML-based features. We can easily notice that a good number of papers considers both HTML-based and URL-based features and these features often refer to a variety of properties. The number of features used in the papers is also listed in the table. As can be seen, some papers use very few features, whereas some others use as many as thousands.

In what follows we provide details of the types of properties and of the approaches proposed to extract features.

1) URL-BASED FEATURES

Features that take into account the URL properties play a key role for detecting phishing web pages. In fact, as already discussed, attackers commonly construct the URLs of their pages in such a way that individuals believe these URLs belong to a trusted party.

Hints about the properties to be taken into account for extracting meaningful features are derived from the analysis of the anatomy of phishing URLs and domains. For example, McGrath and Gupta [89] show that properties, such as URL and domain lengths, presence of the brand name being targeted, character composition of domain names, domain registration and expiration dates, are particularly relevant for the detection. Similarly, Garera et al. [69] identify the properties associated with the obfuscation techniques exploited by attackers – who might replace hostnames with IP addresses or with other domains, add a large string of characters or use unknown or misspelled domains.

It is important to point out that the properties identified in these pioneering works have been used by many papers as the basis for extracting features.

In general, the properties for deriving URL-based features refer to the URL string, i.e., the URL textual content, and to the external data obtained by querying third-party services with specific URL components. Four main categories of properties can be associated with URLs, namely:

- *Lexical properties* describing the URL structure and composition;
- *Statistical properties* describing the URL patterns;
- *Network-based properties* describing the host and domain names specified in the URL;
- *Reputation properties* describing the ranking and popularity of the web page identified by the URL.

To obtain a detailed characterization of phishing web pages, features should refer to multiple categories of properties. For example, the ranking of a page is an important property, although by itself it is not sufficient to detect whether the page is phishing, thus this property needs to be complemented with specific properties referring to the structure of the URL.

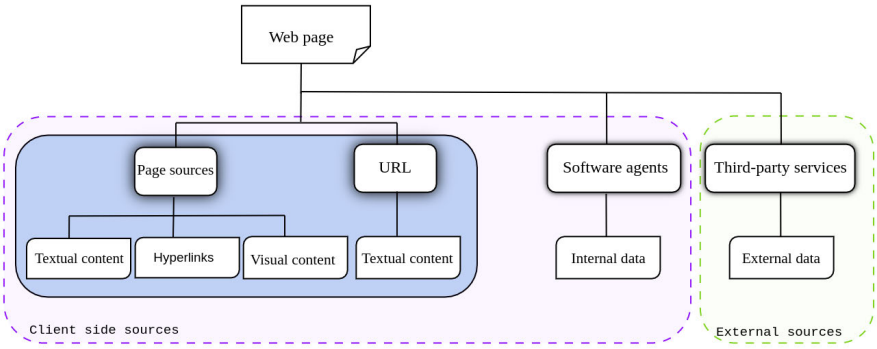


FIGURE 10. Client side and external sources used for describing the properties of web pages. The blue box groups the main sources.

TABLE 4. Overview of the features and third-party services used by some relevant papers in the context of machine learning-based detection approaches. Papers are listed in alphabetic order according to first author's lastname.

Paper	Features						Number of Features	Third-party services
	URL-based				HTML-based			
	Lexical properties	Statistical properties	Network properties	Reputation properties	Textual/visual properties	Traffic properties		
Blum et al. [66]	✓						> 1000	None
Corona et al. [67]					✓		> 1000	None
Dong et al. [68]						✓	42	None
Garera et al. [69]	✓			✓			18	Google PageRank
Gowtham and Krishnamurthi [70]	✓		✓	✓	✓		15	Search engine, WHOIS
Jain and Gupta [71]	✓				✓	✓	19	None
Kan and Thi [72]	✓						N/A	None
Le et al. [73]	✓		✓				N/A	WHOIS
Li et al. [74]	✓				✓		238	None
Ma et al. [75]	✓		✓				> 1000	DNS, WHOIS
Marchal et al. [76]		✓	✓	✓			12	Search engine, Alexa
Nagunwa et al. [77]	✓		✓	✓	✓		31	Search engine, WHOIS
Niakanlahiji et al. [78]					✓	✓	18	None
Pan and Ding [79]	✓		✓		✓		10	WHOIS
Rao and Pais [80]	✓		✓	✓	✓	✓	16	Search engine, WHOIS, Alexa
Rao et al. [81]	✓		✓		✓		17	WHOIS
Sahingoz et al. [82]	✓						> 1000	None
Shirazi et al. [83]	✓				✓		7	None
Tan et al. [84]					✓		17	None
Tian et al. [85]					✓		≈ 1000	Search engine
Verma and Das [86]	✓						> 1000	None
Verma and Dyer [87]	✓	✓					Dozens	None
Xiang et al. [88]	✓		✓	✓	✓		15	Search engine, WHOIS

Various types of approaches have been proposed in the literature to identify the properties associated with URLs and extract the corresponding features (see, e.g., [66], [69], [70], [72], [73], [75], [76], [77], [82], [86], [87], [90], [91], [92], [93], [94]). Details are provided in what follows.

a: LEXICAL PROPERTIES

Lexical properties refer to the structure and composition of URL strings and in particular of the tokens obtained by segmenting – according to specific delimiters, such as ".", "/", ":" – the entire string or its individual components (e.g., protocol, domain, pathname).

In an early work by Kan and Thi [72] features describe the properties of the tokens extracted through a baseline segmentation – using as delimiters whitespace and case change – augmented with entropy-based segmentation. These properties are related to URL orthographic patterns, length, originating URL components as well as to sequential and precedence dependencies between tokens.

Another perspective adopted for analyzing the composition of URL strings considers as tokens the words URLs consist of. Bag-of-words representations have been proposed for extracting features, namely, a binary feature is associated with each word. In particular, to distinguish words

appearing in different URL components, Ma et al. [75] suggest a bag-of-words representation for each of the identified components, that is, hostname, second level domain, top level domain, pathname and file extension. To model the order of the words inside URL domain and pathname components, Blum et al. [66] propose bag-of-words representations enhanced with bi-grams, i.e., sequences of two words, thus each URL is represented as a vector of binary features. Another enhancement is introduced in [93] where words are obtained by segmenting concatenated words with the objective of extracting brand names often abused by attackers.

URL strings have also been described in terms of properties specifically related to the tricks adopted by attackers. For example, in [73] numerical features are derived from obfuscation resistant properties referring to the individual URL components, e.g., dots in the entire URL, hyphens in the domain name, tokens in sub-directories, dots and other delimiters in the filename, variables in the query component. Similarly, in [86] binary features that capture the URL obfuscation, such as punctuation, misspellings, are extracted from character n-grams, that is, overlapping sequences of n consecutive characters. In [82] numeric and binary features are obtained by examining general and specific properties of the URLs and of the words they consist of. For example, properties refer to the word length, the presence of special or consecutive repeated characters in the URL as well as to the word usage obtained from word vectors.

b: STATISTICAL PROPERTIES

Statistical properties refer to the patterns existing in URL strings considered either at the character or at the word level. For example, in [87] the distributions of URL characters are analyzed from a statistical perspective to describe their usage and derive the corresponding features. In detail, the frequency distributions of English characters in legitimate and phishing URLs are compared with the distribution in standard English by applying tests, such as Kolmogorov-Smirnov and Kullback-Liebler divergence. The values of these similarity metrics are treated as features.

Marchal et al. [76] analyze the statistical properties of URLs at the word level to derive features able to identify phishing URLs relying on registered domains not related to the brand being targeted. In particular, to explain the relationships among the words composing the registered domain and the part of the URL that can be freely defined, the concept of intra-URL relatedness is introduced. The relatedness features are obtained by computing the Jaccard index pairwise, that is, considering these two sets of words as well as the sets of words related or associated with them and derived by querying search engines.

c: NETWORK-BASED PROPERTIES

Network-based properties refer to the characteristics of the hostname and domain name associated with a URL. Unlike lexical and statistical properties, these properties are not derived directly from the URL strings. Instead, some components of the URL string are used to obtain external data by querying third-party services, such as DNS, WHOIS.

Numerous properties are derived from this external data, such as registration and expiration dates of the corresponding domain, domain registrar and registrant, Autonomous System and geographic location. In most papers (see, e.g., [75], [79], [88], [95], [96]), the numeric and binary features extracted from these properties are mainly used to enhance the URL description.

d: REPUTATION PROPERTIES

Reputation properties refer to the ranking of the page identified by the URL, that is, its importance as seen from external services, such as search engines. Similarly to network-based properties, third-party services are involved in the assessment of these properties.

Some papers (see, e.g., [69], [70]) consider as feature the value of the ranking provided by the Google's PageRank. In other papers (see e.g., [76], [77]), the features refer to the results obtained by checking domain popularity lists (e.g., Alexa) or by querying search engines using various components of the URL, such as hostname, domain name.

As already mentioned, these features are seldom used in isolation, they are seen as a way to improve of the description of a web page.

2) HTML-BASED FEATURES

Features related to page source codes (e.g., HTML files) and to traffic generated by page downloads are particularly relevant for phishing detection because they capture the strategies exploited by attackers for creating web pages that mimic the layout and characteristics of their legitimate counterparts.

The properties for deriving these features can be grouped in three main categories, namely:

- *Textual properties* describing the content of the source codes and the relationships among their components;
- *Visual properties* describing the appearance of the page;
- *Traffic properties* describing the HTTP responses received from web servers by software agents.

As already mentioned, to make the detection more effective, many papers consider both HTML-based and URL-based features related to multiple types of properties.

Various types of approaches have been proposed in the literature to identify the properties associated with page source codes and extract the corresponding features (see, e.g., [67], [71], [74], [78], [79], [80], [81], [83], [84], [85], [88], [95], [96], [97], [98], [99], [100], [101], [102]). Details are provided in what follows.

a: TEXTUAL PROPERTIES

Textual properties – derived from the page source codes – refer to the layout of web pages. In this context, the composition of source codes is analyzed under many different perspectives by considering the entire HTML document as well as the various types of HTML tags used to create a page.

Features are obtained from general and specific properties of the source codes.

General properties are related to the frequency of some tags (e.g., input, iframe), the number of terms appearing in the HTML document and their frequency (see, e.g., [95], [102], [103]). On the contrary, specific properties refer to the content associated with HTML tags, such as title, copyright, form, anchor. For example, in [88] to identify potentially harmful content in a web page, features are derived from the properties of HTML forms and action fields and in particular from the content of these fields and from the keywords related to sensitive information appearing in the login forms. Similarly, starting from the idea that each website has an identity that is difficult to be manipulated or forged, Pan and Ding [79] analyze the inconsistencies between the identity and the properties of web pages. In detail, the identity – extracted by considering keywords associated with specific DOM objects, such as title and copyright – is used to generate features referring to the various components of the page. In some papers (see, e.g., [74], [83]), features – capturing the relationships and consistency between textual content of web pages and the content of the corresponding URLs – are extracted by matching the domain name or the brand name included in the URL with key components of the page source.

To overcome the text and code level obfuscations, in [85] features correspond to the frequency of the keywords appearing in the text obtained – by applying visual analysis and optical character recognition – from page screenshots and from login form regions.

Some works (see, e.g., [67], [81]) investigate the properties of phishing websites hosted on compromised servers by comparing the textual content of the suspicious page with the content of the home page of the compromised website. More precisely, features correspond to the values of the Jaccard index of similarity computed for the text extracted from various tags, such as title, copyright.

HTML source codes have also been analyzed by considering the textual properties of the hyperlinks associated with attributes, such as `href` and `src`, and by applying some heuristics to obtain the features (see, e.g., [80], [103], [104]). For example, some properties refer to the overall link frequency and to the frequencies of null links and of the most common links appearing in the HTML body or in the footer of a web page. Other properties are related to the fraction of links pointing to local or foreign domains. In addition, the presence of anchor links in the HTML body is considered as a relevant property to detect phishing web pages designed with a single background image – resembling the targeted legitimate page – instead of textual content including links.

In [84] links have been examined from a different perspective by analyzing link manipulation patterns exploited by attackers. To model the overall link and network structure of web pages, this work proposes web graphs – built using page linking data collected by crawling the pages corresponding to the links of a web page. Features, such as in-degree and out-degree centrality, graph density, number of strongly connected components, are extracted from the properties of the graphs.

b: VISUAL PROPERTIES

Visual properties – derived from the screenshots of the rendered page – refer to the appearance of a web pages. In this context, the properties of the screenshots of the entire page or of arbitrary or specific regions (e.g., rectangular regions, logo region) are considered. In general, these properties refer to the representations used by computer vision for object detection and image classification. For example, in some papers (see, e.g., [67], [105]) features are extracted from properties, such as Histogram of Oriented Gradients – representing local object appearance in a semi-rigid way through distribution of intensity gradients and edge directions – and color histograms – representing the spatial distribution of colors within an image. Similarly, in [106] features are extracted from color-based visual descriptors, such as Scalable Color Descriptor, Color Layout Descriptor, Fuzzy Color and Texture Histogram.

c: TRAFFIC PROPERTIES

The properties of the traffic generated by page downloads mainly refer to the characteristics of the HTTP response messages received by browsers and of the TLS certificates provided by web servers to establish encrypted connections. For example, the composition of the header sections and the status codes associated with HTTP messages are useful for detecting phishing web pages. In fact, to confuse users, the pages created by attackers often contain broken links and redirections.

As already mentioned, features extracted from traffic properties are typically used in combination with other features. In some works (see, e.g., [71], [80]) features describing broken links and redirections complement the features extracted from the textual properties of hyperlinks.

Similarly, in [78] features extracted from the header section (e.g., number of header fields, number of non-standard header fields) and from the certificates (e.g., validity, issuer name) are complemented by features describing the complexity of the code in terms of functionalities offered to users. On the contrary, the certificates issued by web servers are the only source considered in [68] to derive features. In detail, properties, such as subject name, cryptographic algorithm, certificate version, are analyzed to obtain these features.

In conclusion, from the analysis of the features used in the literature, we can identify the main strengths and weaknesses associated with URL-based and HTML-based features (see Table 5 for a summary). We notice that some features can be easily extracted, thus allowing on-the-fly detection of previously unseen phishing URLs, whereas the extraction of some others require the page download, thus creating safety concerns and slowing down the detection of phishing pages.

TABLE 5. Summary of the main strengths and weaknesses associated with URL-based and HTML-based features.

Features	Properties	Strengths	Weaknesses
URL-based	Lexical & Statistical	- No page download - Fast and safe extraction - Real-time detection	- Vulnerable to URL manipulation
	Network & Reputation	- No page download - Up-to-date insights on domain registration and reputation	- Third-party services - Significant delays in the extraction - Not fully reliable, e.g., compromised websites
HTML-based	Textual & Visual	- Robust to evasion strategies - Robust to obfuscation techniques - Robust to cloaking techniques	- Page download and parsing - Delays in the extraction - Safety and security issues - Storage requirements
	Traffic	- Able to recognize redirections	- Page download - Delays in the extraction - Safety and security issues

B. FEATURE SELECTION

Feature selection refers to the process of choosing – among the features being extracted – the "best" features to be used to classify a page as legitimate or phishing [107]. This process is important to make models parsimonious, avoid overfitting, improve accuracy and lower computation requirements especially when the number of features is big.

Despite the potential benefits of this process, our literature review has highlighted that some papers analyze the importance of the extracted features as part of the training process (see, e.g. [71], [78], [81], [90], [108], [109]), while only few papers analyze features with the objective of retaining the most representative ones (see, e.g., [82], [91], [110], [111], [112], [113], [114], [115]).

To select the optimal subset of features to be used for detecting phishing websites, these papers devise the conventional feature selection methods, namely:

- *Filter methods*, i.e., methods that select features by ranking them according to their relevance independently of the classifier;
- *Wrapper methods*, i.e., methods that select subsets of features by ranking them according to their predictive power for a given classifier.

In detail, in the context of filter methods, features (or subsets of features) are selected according to the rank provided by one or multiple statistical measures, such as Chi-square test, Information Gain score, Correlation-based Feature Selection and Fisher score. For wrapper methods, feature selection is considered as a search problem where the subsets of features – identified by strategies, such as Genetic Algorithm and greedy forward selection – are evaluated through a classifier and selected according to their predictive performance. In this context, the classifier is considered as a perfect black box.

It is interesting to point out that, to make the selection more reliable and robust, in some works the various methods complement each other. For example, filter and wrapper methods are applied in [112] to select – among 177 features derived from the web page URL and source code – the optimal subset of features. A feature selection ensemble is proposed in [116]. According to this ensemble, multiple sets of features are generated by different selection methods, thus promoting diversity and improving generalization. To obtain a compact set of effective features, Chiew et al. [113] address the problem of automatically identifying the optimal feature cut-off rank by exploiting patterns in the distribution of filter measure values. This approach is devised in a feature selection framework based on a hybrid ensemble structure consisting of data perturbation and function perturbation techniques.

C. MACHINE LEARNING MODELS

As already mentioned, phishing detection is seen as a binary classification problem whose main goal is to identify whether a web page is legitimate or phishing. To solve this problem and derive the corresponding models, many diverse supervised learning algorithms have been considered in the literature. These algorithms range from traditional state of the art machine learning classifiers, such as Support Vector Machine, Decision tree, K-Nearest Neighbors, to various types of artificial and deep neural networks (see, e.g., [117], [118], [119], [120]).

Table 6 presents an overview of the ten most popular *traditional machine learning algorithms* applied in the context of phishing detection. As can be seen, Support Vector Machine and Random Forest are the two most common algorithms. In addition, since algorithm performance heavily depends on the features being extracted and on the dataset used, we observe that many papers apply more than one algorithm to identify the best algorithm for their settings. An interesting finding is that – although the results obtained under different settings are not directly comparable – Random Forest tends to outperform the other algorithms.

It is also important to outline that – to improve predictive performance of the models – some papers (see, e.g., [74], [114], [121], [122]) apply ensemble learning methods, that is, they combine multiple predictions obtained from several machine learning models by using various approaches, such as stacking.

Another machine learning approach considered in the literature in the context of phishing detection refers to *online learning* where – unlike batch learning typical of traditional

TABLE 6. Summary of the traditional state of the art machine learning algorithms applied for detecting phishing web pages. Algorithms are listed according to the number of papers using them.

Algorithm	Papers
Support Vector Machine	[121], [111], [113], [67], [70], [90], [98], [71], [72], [75], [101], [76], [77], [79], [80], [81], [92], [82], [122], [83], [123], [84], [88], [124]
Random Forest	[121], [113], [68], [90], [114], [71], [91], [101], [76], [77], [78], [116], [80], [92], [82], [122], [123], [84], [85], [87], [88]
Logistic Regression	[68], [69], [90], [114], [71], [75], [77], [116], [80], [92], [122], [93], [87], [88]
K-Nearest Neighbors	[68], [90], [114], [91], [77], [78], [116], [92], [82], [122], [83], [123], [85]
Decision Tree	[114], [101], [77], [116], [92], [82], [122], [83], [123], [88]
Adaboost	[121], [125], [114], [101], [78], [116], [80], [82], [122], [88]
Naive Bayes	[113], [68], [75], [82], [83], [84], [85], [87]
C4.5	[113], [68], [91], [76], [115], [80], [84], [87], [88]
Gradient Boosting	[121], [114], [77], [116], [122], [83], [123]
XGBoost	[121], [114], [74], [116], [92]

TABLE 7. Overview of the main online learning algorithms applied for detecting phishing web pages. Algorithms are listed according to the number of papers using them.

Algorithm	Papers
Perceptron	[73], [126], [127], [109], [86]
Confidence-Weighted	[66], [73], [126], [127], [86]
Passive-Aggressive	[126], [127], [109], [86]
Adaptive Regularization of Weights	[73], [86]
Logistic Regression with stochastic gradient descent	[126], [127]

algorithms – the model training is performed in an incremental manner by continuously feeding data as it arrives, thus allowing for a real-time detection. For this purpose, various algorithms, such as Perceptron, Confidence-Weighted, are applied (see Table 7 for an overview). These algorithms mainly differ in the way feedback information is used to update models in case of misclassifications.

It is interesting to point out that the pioneer work by Ma et al. [126] has demonstrated that this approach can cope with the size of training datasets – that are generally very large – and the distribution of features – that continuously changes. Nevertheless, despite these potential benefits, a limited number of works focus on this type of approach. In fact, online learning methods could be computationally expensive and cumbersome to control and protect since models might change frequently.

Recently, the research on phishing detection has also focused on *deep learning* approaches because of their ability of identifying patterns and extracting features on their own. Various classes of artificial neural networks, such as Convolutional and Deep Neural Networks, have been adopted (see Table 8 for an overview). These networks mainly differ

TABLE 8. Overview of the main deep learning algorithms applied for detecting phishing web pages. Algorithms are listed according to the number of papers using them.

Algorithm	Papers
Convolutional Neural Network	[128], [129], [130], [131], [132], [133], [134], [135], [136], [137], [138]
Long Short-Term Memory	[108], [130], [139], [133], [140], [134], [135], [136], [137], [138]
Deep Neural Network	[110], [133], [141], [142]
Multilayer Perceptron	[130], [143]

in terms of the number of layers considered and the types of connections among layers. From the table, we notice that approaches based on Convolutional Neural Network and Long Short-Term Memory are the most popular because their characteristics make them particularly suitable for detecting phishing web pages.

An interesting comparison of the performance of several machine learning models is presented in [144]. This investigation focuses on the impact of different feature sets as well as of datasets consisting of different fractions of legitimate and phishing samples. The study demonstrates the importance of taking precisely into account the imbalanced nature of phishing attacks.

D. PERFORMANCE METRICS

Model performance refers to ability of a model to identify classes correctly, thus the evaluation of its performance is a very important task in every machine learning-based setting. This assessment is generally based on standard quantitative metrics.

To analyze the performance at different levels of granularity, our literature review has shown that most papers use multiple metrics. Among these metrics, some of them are widely used, namely:

- *Accuracy*: a measure of the overall performance of a model, that is, the fraction of phishing and legitimate web pages correctly identified;
- *Precision*: a measure of how accurate the phishing identification is, that is, the fraction of phishing web pages correctly identified with respect to the total number of phishing pages;
- *Recall* (or equivalently *True Positive Rate)*: a measure of the coverage of actual phishing web pages, that is, the fraction of web pages actually identified as phishing with respect to the total number of pages that should have been identified as phishing;
- *Specificity* (or equivalently *True Negative Rate)*: a measure of the coverage of actual legitimate web pages, that is, the fraction of web pages actually identified as legitimate with respect to the total number of pages that should have been identified as legitimate;
- *F1-score*: a combination of precision and recall in a single score that assesses their tradeoff.

In some papers, the performance of the models has also been summarized by considering the confusion matrix and the receiver operating characteristic (ROC) curve as well as error rates.

E. DATASETS

To assess the performance of the proposed approaches, most researchers create their own datasets by collecting data from various sources, such as PhishTank archive for phishing websites and Alexa top-ranked domains for legitimate websites. Although datasets often share the data sources, in general their content varies in terms of both websites considered and features chosen for their description. As a consequence, the results obtained by different papers are not directly comparable.

Researchers making the datasets used in their papers available to the community are commendable (see, e.g., [49], [67], [76], [82], [91], [106], [128]). Nevertheless, the availability of these datasets heavily depends on the availability of authors websites where they are published, thus these datasets tend to disappear after some years.

To cope with all these issues, in the years some datasets have been hosted in public repositories. These datasets are of paramount importance for the research community. For example, a popular dataset – known as UCI phishing websites dataset or simply as UCI dataset – was donated in 2015. This dataset – available at the University of California Irvine Machine Learning repository [145] – includes 6,157 phishing and 4,898 legitimate website samples, each described by 30 features extracted from URLs, HTML and JavaScript content and DNS services [96]. Another popular dataset – available from Mendeley Data portal – was published 2018 [146]. This is a balanced dataset containing 5,000 phishing and 5,000 legitimate website samples, each described by 48 features. More recently, Vrbančič [147] published a dataset consisting of 58,000 and 30,647 instances of legitimate and phishing websites, respectively. This dataset – also available from Mendeley Data Portal – includes 111 features mainly referring to various properties of URL components, such as domain, directory, file name [148].

F. DISCUSSION

The analysis of the state of the art has shown that machine learning provides sound phishing detection solutions characterized by the ability of detecting zero-hour attacks and handling efficiently new types of phishing web pages. This analysis has also suggested that machine learning is a very active research field. The many diverse approaches offered in the literature primarily differ in terms of the set of features extracted for describing phishing and legitimate web pages and of the learning algorithms applied to derive the models.

In general, features are extracted by considering multiple complementary properties of the page URL and source codes. In fact, a high quality set of features able to discriminate phishing and legitimate pages is crucial for the effectiveness of a phishing detection solution. This also means that features associated with phishing pages are significantly different from their legitimate counterparts.

In the context of machine learning models, our review has shown a prevalence of models built using traditional state of the art algorithms, although models based on deep learning algorithms have started to appear. In fact, these algorithms avoid the burden of feature extraction and selection even though at the expense of large computation requirements.

The training, testing and validation of machine learning models are frequently based on datasets created by researchers on their own because of the limited number of publicly available datasets. As a consequence, it is difficult to assess the validity of the obtained results.

Our investigation has also highlighted some interesting findings that can be summarized by the following recommendations:

- Features should be chosen by taking into account:
  - Their strengths and weaknesses;
  - Their discriminating power;
  - The attacker strategies;
  - The principle of parsimony.
- Datasets should be diverse, unbiased and include an appropriate number of phishing and legitimate samples;
- Metrics chosen for evaluating model performance should reflect the desired behavior of the detection solution;
- Feature engineering should be part the detection approaches to improve the model predictive power;
- Datasets, model parameters and implementation methods should be disclosed to ensure the reproducibility of the results.

Finally, it should be noted that, although machine learning is a pervasive and powerful form of artificial intelligence, it cannot be used as a black box. In fact, to extract a meaningful set of features, the strategies applied by attackers and their implications on website design have to be taken precisely into account. This means that the machine learning competences have to be coupled with a solid domain knowledge.

VIII. LESSONS LEARNT

As already mentioned, the detection of phishing websites faces several challenges primarily related to the nature of these dangerous security threats. This section summarizes the lessons learnt from the analysis of the state of the art by reviewing the main strengths and weaknesses of list-based, similarity-based and machine learning-based methods.

A first interesting finding refers to list-based detection methods. These methods are generally simple and fast, although not always effective. In fact, their reactive nature – coupled with the delays in identifying new phishing campaigns and updating the lists accordingly – makes them unable to cope with zero-hour attacks. The automatic updates of blacklists – by predicting phishing URLs from blacklisted ones – and the creation of customized whitelists, are solutions

of paramount importance, even though they might fail to predict previously unseen phishing URLs. In general, the usage of list-based methods in isolation is not recommended, instead these methods can be seen as an initial detection step that complements other methods.

Unlike list-based detection methods, similarity-based methods can generally cope with zero-hour attacks, although they are slower and more complex to implement. In fact, the textual or visual comparisons of suspicious pages with their legitimate counterparts are computation-intensive, especially when page screenshots are involved. These methods are also storage-intensive. In fact, to ensure an effective detection, a large number of legitimate web pages has to be stored. Despite these issues, similarity is a robust indicator of phishing web pages since it captures the strategies adopted by attackers for creating phishing web pages, thus metrics, such as similarity scores, are particularly useful for the detection.

Machine learning-based methods are generally fast and effective. They cope well with zero-hour attacks and allow on-the-fly detection of phishing web pages. Nevertheless, the performance of these methods varies and mainly depends on the features extracted from web pages and on the composition of the training datasets. In general, features extracted from page URL are obtained quickly, although they are vulnerable to URL manipulations. On the contrary, features extracted from page source codes are robust to the evasion techniques implemented by attackers, even though page downloads are needed to obtain these features, thus causing delays and safety issues. Despite these issues, machine learning models are a valid and promising solution for detecting phishing web pages.

Lastly, it is worth mentioning that third-party services, such as search engines and DNS, provide useful information for the detection, although they introduce significant overhead that might delay the detection itself, thus the actual benefits of this information have to be carefully assessed.

IX. CONCLUSION

Phishing is a very active and effective security threat that affects individuals as well as the targeted companies and organizations. Despite being around for many years, this threat is still one of the attack vectors most commonly used nowadays. The level of sophistication of the phishing campaigns has increased significantly over the years. Attackers employ numerous social engineering strategies and evasion techniques to make attacks more and more convincing for individuals and more challenging for detection tools. In this context research plays a critical role.

Our survey has shown that many research efforts have been dedicated to the detection of phishing websites. Among the various detection approaches, machine learning-based methods are becoming quite popular because of their ability to detect zero-hour attacks and handle efficiently newly discovered phishing web pages. Nevertheless, to fight phishing more effectively, it is necessary to stay one step ahead of attackers, thus some research gaps need to be filled. In what follows, we discuss the main gaps identified from this survey.

An important research gap refers to the increased use by attackers of URL shortening services that mask the real phishing URLs. These short URLs create an additional challenge to list-based approaches and in particular to the management of blacklists. Similarly, these URLs affect machine learning-based approaches since most URL features suggested in the literature become meaningless in this context, thus making the detection mechanisms fail.

Other open issues associated with features are related to their relationships with the evasion techniques implemented by attackers. In general, it is not sufficient to retrain a machine learning model whenever new data becomes available, instead there is the compelling need to quickly identify the tactics used by these ever-evolving attacks and automatically extract appropriate features. Hence, further research efforts should be dedicated to investigate these issues.

Model explainability is another interesting research direction in the context of machine learning. In fact, understanding the decisions taken by machine learning algorithms, that is, what characteristics make a web page phishing or legitimate, has several important implications for the design of security systems. Adversarial attacks should also be investigated to make the machine learning models more robust.

It is also important to outline that, to ensure the advancement of the state of the art, experiments should be made reproducible. This means that all implementation details should be clearly specified and the datasets used should be made publicly available.

Finally, we believe that the education of individuals – who often represent the weak link of the chain – should be included to some extent in every kind of phishing countermeasure.

ACKNOWLEDGMENT

The authors wish to thank the anonymous reviewers for their constructive suggestions, which helped to improve the clarity of the article.

REFERENCES

[1] P. Zhang, A. Oest, H. Cho, Z. Sun, R. Johnson, B. Wardman, S. Sarker, A. Kapravelos, T. Bao, R. Wang, Y. Shoshitaishvili, A. Doupé, and G.-J. Ahn, "CrawlPhish: Large-scale analysis of client-side cloaking techniques in phishing," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2021, pp. 1109–1124.

[2] ENISA. (2021). *Cybersecurity for SMEs—Challenges and Recommendations*. [Online]. Available: https://www.enisa.europa.eu/publications/enisa-report-cybersecurity-for-smes

[3] Cisco. (2021). *Cyber Security Threat Trends: Phishing, Crypto Top the List*. [Online]. Available: https://umbrella.cisco.com/info/2021-cyber-security-threat-trends-phishing-crypto-top-the-list

[4] M. Alsharnouby, F. Alaca, and S. Chiasson, "Why phishing still works: User strategies for combating phishing attacks," *Int. J. Hum.-Comput. Stud.*, vol. 82, pp. 69–82, Oct. 2015.

[5] Anti-Phishing Working Group—APWG. (2022). *Phishing Activity Trends Report-1Q*. [Online]. Available: https://docs.apwg.org/reports/apwg_trends_report_q1_2022.pdf

[6] T. Berners-Lee, R. Fielding, and L. Masinter. *Uniform Resource Identifier (URI): Generic Syntax*, RFC 3986, Jan. 2005. [Online]. Available: https://www.rfc-editor.org/rfc/rfc3986.txt

[7] K. L. Chiew, K. S. C. Yong, and C. L. Tan, "A survey of phishing attacks: Their types, vectors and technical approaches," *Exp. Syst. Appl.*, vol. 106, pp. 1–20, Sep. 2018.
[8] B. B. Gupta, N. A. G. Arachchilage, and K. E. Psannis, "Defending against phishing attacks: Taxonomy of methods, current issues and future directions," *Telecommun. Syst.*, vol. 67, no. 2, pp. 247–267, 2018.
[9] I. Qabajeh, F. Thabtah, and F. Chiclana, "A recent review of conventional vs. automated cybersecurity anti-phishing techniques," *Comput. Sci. Rev.*, vol. 29, pp. 44–55, Aug. 2018.
[10] E. Benavides, W. Fuertes, S. Sanchez, and M. Sanchez, "Classification of phishing attack solutions by employing deep learning techniques: A systematic literature review," in *Developments and Advances in Defense and Security* (Smart Innovation, Systems and Technologies), vol. 152, A. Rocha and R. P. Pereira, Eds. Berlin, Germany: Springer, 2020, pp. 51–64.
[11] D. Sahoo, C. Liu, and S. C. H. Hoi, "Malicious URL detection using machine learning: A survey," 2017, *arXiv:1701.07179*.
[12] C. M. R. Da Silva, E. L. Feitosa, and V. C. Garcia, "Heuristic-based strategy for phishing prediction: A survey of URL-based approach," *Comput. Secur.*, vol. 88, Jan. 2020, Art. no. 101613.
[13] G. Varshney, M. Misra, and P. K. Atrey, "A survey and classification of web phishing detection schemes," *Secur. Commun. Netw.*, vol. 9, no. 18, pp. 6266–6284, Dec. 2016.
[14] A. Basit, M. Zafar, X. Liu, A. R. Javed, Z. Jalil, and K. Kifayat, "A comprehensive survey of AI-enabled phishing attacks detection techniques," *Telecommun. Syst.*, vol. 76, no. 1, pp. 139–154, Jan. 2021.
[15] A. Das, S. Baki, A. El Aassal, R. Verma, and A. Dunbar, "SoK: A comprehensive reexamination of phishing research from the security perspective," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 671–708, 1st Quart., 2020.
[16] Z. Dou, I. Khalil, A. Khreishah, A. Al-Fuqaha, and M. Guizani, "Systematization of knowledge (SoK): A systematic review of software-based web phishing detection," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2797–2819, 4th Quart., 2017.
[17] A. K. Jain and B. B. Gupta, "Phishing detection: Analysis of visual similarity based approaches," *Secur. Commun. Netw.*, vol. 2017, pp. 1–20, Jan. 2017.
[18] M. Khonji, Y. Iraqi, and A. Jones, "Phishing detection: A literature survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 4, pp. 2091–2121, 4th Quart, 2013.
[19] *Google Safe Browsing*. Accessed: Oct. 10, 2022. [Online]. Available: https://safebrowsing.google.com/
[20] S. Bell and P. Komisarczuk, "An analysis of phishing blacklists: Google Safe Browsing, OpenPhish, and PhishTank," in *Proc. Australas. Comput. Sci. Week Multiconf.*, Feb. 2020, pp. 1–11.
[21] S. Sheng, B. Wardman, G. Warner, L. Cranor, J. Hong, and C. Zhang, "An empirical analysis of phishing blacklists," in *Proc. 6th Conf. Email AntiSpam (CEAS)*, 2009, pp. 1–10.
[22] A. Oest, Y. Safaei, A. Doupé, G.-J. Ahn, B. Wardman, and K. Tyers, "PhishFarm: A scalable framework for measuring the effectiveness of evasion techniques against browser phishing blacklists," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2019, pp. 1344–1361.
[23] A. Oest, Y. Safaei, P. Zhang, B. Wardman, K. Tyers, Y. Shoshitaishvili, and A. Doupé, "PhishTime: Continuous longitudinal measurement of the effectiveness of anti-phishing blacklists," in *Proc. 29th USENIX Secur. Symp.*, 2020, pp. 379–396.
[24] N. A. Azeez, S. Misra, I. A. Margaret, L. Fernandez-Sanz, and S. M. Abdulhamid, "Adopting automated whitelist approach for detecting phishing attacks," *Comput. Secur.*, vol. 108, Sep. 2021, Art. no. 102328.
[25] Y. Cao, W. Han, and Y. Le, "Anti-phishing based on automated individual white-list," in *Proc. 4th ACM Workshop Digit. Identity Manag.*, Oct. 2008, pp. 51–60.
[26] W. Han, Y. Cao, E. Bertino, and J. Yong, "Using automated individual white-list to protect web digital identities," *Exp. Syst. Appl.*, vol. 39, no. 15, pp. 11861–11869, Nov. 2012.
[27] A. K. Jain and B. B. Gupta, "A novel approach to protect against phishing attacks at client side using auto-updated white-list," *EURASIP J. Inf. Secur.*, vol. 2016, no. 1, pp. 1–11, Dec. 2016.
[28] L.-H. Lee, K.-C. Lee, H.-H. Chen, and Y.-H. Tseng, "POSTER: Proactive blacklist update for anti-phishing," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2014, pp. 1448–1450.
[29] P. Prakash, M. Kumar, R. R. Kompella, and M. Gupta, "PhishNet: Predictive blacklisting to detect phishing attacks," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–5.
[30] R. S. Rao and A. R. Pais, "An enhanced blacklist method to detect phishing websites," in *Information Systems Security* (Lecture Notes in Computer Science), vol. 10717, R. K. Shyamasundar, V. Singh, and J. Vaidya, Eds. Berlin, Germany: Springer, 2017, pp. 323–333.
[31] M. Sharifi and S. H. Siadati, "A phishing sites blacklist generator," in *Proc. IEEE/ACS Int. Conf. Comput. Syst. Appl.*, Mar. 2008, pp. 840–843.
[32] C. Whittaker, B. Ryner, and M. Nazif, "Large-scale automatic classification of phishing pages," in *Proc. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, 2010, pp. 1–14.
[33] G. Xiang, B. A. Pendleton, J. Hong, and C. P. Rose, "A hierarchical adaptive probabilistic approach for zero hour phish detection," in *Computer Security—ESORICS* (Lecture Notes in Computer Science), vol. 6345, D. Gritzalis, B. Preneel, and M. Theoharidou, Eds. Berlin, Germany: Springer, 2010, pp. 268–285.
[34] G. Sonowal and K. S. Kuppusamy, "PhiDMA—A phishing detection model with multi-filter approach," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 32, no. 1, pp. 99–112, 2020.
[35] PhishTank. Accessed: Nov. 4, 2022. [Online]. Available: https://www.phishtank.org
[36] Curlie. Accessed: Nov. 4, 2022. [Online]. Available: https://curlie.org/
[37] S. Afroz and R. Greenstadt, "PhishZoo: Detecting phishing websites by looking at them," in *Proc. IEEE 5th Int. Conf. Semantic Comput.*, Sep. 2011, pp. 368–375.
[38] J.-L. Chen, Y.-W. Ma, and K.-L. Huang, "Intelligent visual similarity-based phishing websites detection," *Symmetry*, vol. 12, no. 10, Oct. 2020, Art. no. 1681.
[39] K. T. Chen, J. Y. Chen, C. R. Huang, and C. S. Chen, "Fighting phishing with discriminative keypoint features," *IEEE Internet Comput.*, vol. 13, no. 3, pp. 56–63, May 2009.
[40] T.-C. Chen, S. Dick, and J. Miller, "Detecting visually similar web pages: Application to phishing detection," *ACM Trans. Internet Technol.*, vol. 10, no. 2, pp. 1–38, May 2010.
[41] J. Chen and C. Guo, "Online detection and prevention of phishing attacks," in *Proc. 1st Int. Conf. Commun. Netw. China*, Oct. 2006, pp. 1–7.
[42] K. L. Chiew, E. H. Chang, S. N. Sze, and W. K. Tiong, "Utilisation of website logo for phishing detection," *Comput. Secur.*, vol. 54, pp. 16–26, Oct. 2015.
[43] M. Dunlop, S. Groat, and D. Shelly, "GoldPhish: Using images for content-based phishing analysis," in *Proc. 5th Int. Conf. Internet Monitor. Protection*, 2010, pp. 123–128.
[44] A. Y. Fu, L. Wenyin, and X. Deng, "Detecting phishing web pages with visual similarity assessment based on Earth mover's distance (EMD)," *IEEE Trans. Dependable Secure Comput.*, vol. 3, no. 4, pp. 301–311, Oct. 2006.
[45] M. Hara, A. Yamada, and Y. Miyake, "Visual similarity-based phishing detection without victim site information," in *Proc. IEEE Symp. Comput. Intell. Cyber Secur.*, Mar. 2009, pp. 30–36.
[46] C.-Y. Huang, S.-P. Ma, W.-L. Yeh, C.-Y. Lin, and C.-T. Liu, "Mitigate web phishing using site signatures," in *Proc. TENCON IEEE Region Conf.*, Nov. 2010, pp. 803–808.
[47] W. Khan, A. Ahmad, A. Qamar, M. Kamran, and M. Altaf, "SpoofCatch: A client-side protection tool against phishing attacks," *IT Prof.*, vol. 23, no. 2, pp. 65–74, Mar. 2021.
[48] I. F. Lam, W. C. Xiao, S. C. Wang, and K. T. Chen, "Counteracting phishing page polymorphism: An image layout analysis approach," in *Advances in Information Security and Assurance* (Lecture Notes in Computer Science), vol. 5576, J. H. Park, H. H. Chen, M. Atiquzzaman, C. Lee, T. Kim, and S. S. Yeo, Eds. Berlin, Germany: Springer, 2009, pp. 270–279.
[49] Y. Lin, R. Liu, D. M. Divakaran, J. Ng, Q. Chan, Y. Lu, Y. Si, F. Zhang, and J. Dong, "Phishpedia: A hybrid deep learning based approach to visually identify phishing webpages," in *Proc. 30th USENIX Secur. Symp.*, 2021, pp. 3793–3810.
[50] W. Liu, X. Deng, G. Huang, and A. Y. Fu, "An antiphishing strategy based on visual similarity assessment," *IEEE Internet Comput.*, vol. 10, no. 2, pp. 58–65, Mar. 2006.
[51] J. Mao, P. Li, K. Li, T. Wei, and Z. Liang, "BaitAlarm: Detecting phishing sites using similarity in fundamental visual features," in *Proc. 5th Int. Conf. Intell. Netw. Collaborative Syst.*, Sep. 2013, pp. 790–795.

[52] J. Mao, W. Tian, P. Li, T. Wei, and Z. Liang, "Phishing-alarm: Robust and efficient phishing detection via page component similarity," *IEEE Access*, vol. 5, pp. 17020–17030, 2017.

[53] E. Medvet, E. Kirda, and C. Kruegel, "Visual-similarity-based phishing detection," in *Proc. 4th Int. Conf. Secur. Privacy Commun. Netowrks*, Sep. 2008, pp. 1–6.

[54] A. P. E. Rosiello, E. Kirda, C. Kruegel, and F. Ferrandi, "A layout-similarity-based approach for detecting phishing pages," in *Proc. 3rd Int. Conf. Secur. Privacy Commun. Netw. Workshops*, 2007, pp. 454–463.

[55] N. M. Shekokar, C. Shah, M. Mahajan, and S. Rachh, "An ideal approach for detection and prevention of phishing attacks," *Proc. Comput. Sci.*, vol. 49, pp. 82–91, Jan. 2015.

[56] W. Zhang, H. Lu, B. Xu, and H. Yang, "Web phishing detection based on page spatial layout similarity," *Informatica*, vol. 37, no. 3, pp. 231–244, 2013.

[57] Y. Zhang, J. I. Hong, and L. F. Cranor, "Cantina: A content-based approach to detecting phishing web sites," in *Proc. 16th Int. Conf. World Wide Web*, May 2007, pp. 639–648.

[58] Y. Zhou, Y. Zhang, J. Xiao, Y. Wang, and W. Lin, "Visual similarity based anti-phishing with the combination of local and global features," in *Proc. IEEE 13th Int. Conf. Trust, Secur. Privacy Comput. Commun.*, Sep. 2014, pp. 189–196.

[59] T. A. Phelps and R. Wilensky, "Robust hyperlinks: Cheap, everywhere, now," in *Digital Documents: Systems and Principles* (Lecture Notes in Computer Science), vol. 2023, P. King and E. V. Munson, Eds. Berlin, Germany: Springer, 2004, pp. 28–43.

[60] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. London, U.K.: Pearson, 2018.

[61] R. Szeliski, *Computer Vision: Algorithms and Applications*. Berlin, Germany: Springer, 2010.

[62] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Feb. 2004.

[63] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Proc. 9th Eur. Conf. Comput. Vis.* (Lecture Notes in Computer Science), vol. 3951. Berlin, Germany: Springer, May 2006, pp. 404–417.

[64] C.-R. Huang, C.-S. Chen, and P.-C. Chung, "Contrast context histogram—An efficient discriminating local descriptor for object recognition and image matching," *Pattern Recognit.*, vol. 41, no. 10, pp. 3071–3077, Oct. 2008.

[65] Y. Rubner, C. Tomasi, and L. J. Guibas, "The Earth mover's distance as a metric for image retrieval," *Int. J. Comput. Vis.*, vol. 40, no. 2, pp. 99–121, Nov. 2000.

[66] A. Blum, B. Wardman, T. Solorio, and G. Warner, "Lexical feature based phishing URL detection using online learning," in *Proc. 3rd ACM Workshop Artif. Intell. Secur.*, Oct. 2010, pp. 54–60.

[67] I. Corona, B. Biggio, M. Contini, L. Piras, R. Corda, M. Mereu, G. Mureddu, D. Ariu, and F. Roli, "DeltaPhish: Detecting phishing webpages in compromised websites," in *Computer Security—ESORICS* (Lecture Notes in Computer Science), vol. 10492, S. N. Foley, D. Gollmann, and E. Snekkenes, Eds. Berlin, Germany: Springer, 2017, pp. 370–388.

[68] Z. Dong, A. Kapadia, J. Blythe, and L. J. Camp, "Beyond the lock icon: Real-time detection of phishing websites using public key certificates," in *Proc. APWG Symp. Electron. Crime Res. (eCrime)*, May 2015, pp. 1–12.

[69] S. Garera, N. Provos, M. Chew, and A. D. Rubin, "A framework for detection and measurement of phishing attacks," in *Proc. ACM Workshop Recurring Malcode*, Nov. 2007, pp. 1–8.

[70] R. Gowtham and I. Krishnamurthi, "A comprehensive and efficacious architecture for detecting phishing webpages," *Comput. Secur.*, vol. 40, pp. 23–37, Feb. 2014.

[71] A. K. Jain and B. B. Gupta, "Towards detection of phishing websites on client-side using machine learning based approach," *Telecommun. Syst.*, vol. 68, no. 4, pp. 687–700, Aug. 2018.

[72] M.-Y. Kan and H. O. N. Thi, "Fast webpage classification using URL features," in *Proc. 14th ACM Int. Conf. Inf. Knowl. Manag.*, Oct. 2005, pp. 325–326.

[73] A. Le, A. Markopoulou, and M. Faloutsos, "PhishDef: URL names say it all," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 191–195.

[74] Y. Li, Z. Yang, X. Chen, H. Yuan, and W. Liu, "A stacking model using URL and HTML features for phishing webpage detection," *Future Gener. Comput. Syst.*, vol. 94, pp. 27–39, May 2019.

[75] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: Learning to detect malicious web sites from suspicious URLs," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jun. 2009, pp. 1245–1254.

[76] S. Marchal, J. Francois, R. State, and T. Engel, "PhishStorm: Detecting phishing with streaming analytics," *IEEE Trans. Netw. Service Manag.*, vol. 11, no. 4, pp. 458–471, Dec. 2014.

[77] T. Nagunwa, S. Naqvi, S. Fouad, and H. Shah, "A framework of new hybrid features for intelligent detection of zero hour phishing websites," in *Proc. Int. Joint Conf., 12th Int. Conf. Comput. Intell. Secur. Inf. Syst. (CISIS) 10th Int. Conf. Eur. Transnational Educ.*, vol. 951, F. Martinez Álvarez, A. Troncoso Lora, J. A. Saez Muñoz, H. Quintián, and E. Corchado, Eds. Berlin, Germany: Springer, 2020, pp. 36–46.

[78] A. Niakanlahiji, B.-T. Chu, and E. Al-Shaer, "PhishMon: A machine learning framework for detecting phishing webpages," in *Proc. IEEE Int. Conf. Intell. Secur. Informat. (ISI)*, Nov. 2018, pp. 220–225.

[79] Y. Pan and X. Ding, "Anomaly based web phishing page detection," in *Proc. 22nd Annu. Comput. Secur. Appl. Conf. (ACSAC)*, Dec. 2006, pp. 381–392.

[80] R. S. Rao and A. R. Pais, "Detection of phishing websites using an efficient feature-based machine learning framework," *Neural Comput. Appl.*, vol. 31, no. 8, pp. 3851–3873, Aug. 2019.

[81] R. S. Rao, A. R. Pais, and P. Anand, "A heuristic technique to detect phishing websites using TWSVM classifier," *Neural Comput. Appl.*, vol. 33, no. 11, pp. 5733–5752, Jun. 2021.

[82] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine learning based phishing detection from URLs," *Exp. Syst. Appl.*, vol. 117, pp. 345–357, Mar. 2019.

[83] H. Shirazi, B. Bezawada, and I. Ray, "'Kn0w thy Doma1n name': Unbiased phishing detection using domain name based features," in *Proc. 23rd ACM Symp. Access Control Models Technol.*, Jun. 2018, pp. 69–75.

[84] C. L. Tan, K. L. Chiew, K. S. C. Yong, S. N. Sze, J. Abdullah, and Y. Sebastian, "A graph-theoretic approach for the detection of phishing webpages," *Comput. Secur.*, vol. 95, Aug. 2020, Art. no. 101793.

[85] K. Tian, S. T. K. Jan, H. Hu, D. Yao, and G. Wang, "Needle in a haystack: Tracking down elite phishing domains in the wild," in *Proc. Internet Meas. Conf.*, Oct. 2018, pp. 429–442.

[86] R. Verma and A. Das, "What's in a URL: Fast feature extraction and malicious URL detection," in *Proc. 3rd ACM Int. Workshop Secur. Privacy Anal.*, Mar. 2017, pp. 55–63.

[87] R. Verma and K. Dyer, "On the character of phishing URLs: Accurate and robust statistical learning classifiers," in *Proc. 5th ACM Conf. Data Appl. Secur. Privacy*, Mar. 2015, pp. 111–122.

[88] G. Xiang, J. Hong, C. P. Rose, and L. Cranor, "CANTINA+: A feature-rich machine learning framework for detecting phishing web sites," *ACM Trans. Inf. Syst. Secur.*, vol. 14, no. 2, pp. 1–28, Sep. 2011.

[89] D. K. McGrath and M. Gupta, "Behind phishing: An examination of phisher modi operandi," in *Proc. 1st USENIX Workshop Large-Scale Exploits Emergent Threats (LEET)*, 2008.

[90] B. B. Gupta, K. Yadav, I. Razzak, K. Psannis, A. Castiglione, and X. Chang, "A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment," *Comput. Commun.*, vol. 175, pp. 47–57, Jul. 2021.

[91] M. Mamun, M. A. Rathore, A. H. Lashkari, N. Stakhanova, and A. A. Ghorbani, "Detecting malicious URLs using lexical analysis," in *Network and System Security* (Lecture Notes in Computer Science), vol. 9955, J. Chen, V. Piuri, C. Su, and M. Yung, Eds. Berlin, Germany: Springer, 2016, pp. 467–482.

[92] R. S. Rao, T. Vaishnavi, and A. R. Pais, "CatchPhish: Detection of phishing websites by inspecting URLs," *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 2, pp. 813–825, Feb. 2020.

[93] H. Tupsamudre, A. K. Singh, and S. Lodha, "Everything is in the name—A URL based approach for phishing detection," in *Cyber Security Cryptography and Machine Learning* (Lecture Notes in Computer Science), vol. 11527, S. Dolev, D. Hendler, S. Lodha, and M. Yung, Eds. Berlin, Germany: Springer, 2019, pp. 231–248.

[94] M. M. Yadollahi, F. Shoeleh, E. Serkani, A. Madani, and H. Gharaee, "An adaptive machine learning based approach for phishing detection using hybrid features," in *Proc. 5th Int. Conf. Web Res. (ICWR)*, Apr. 2019, pp. 281–286.

[95] H. Choi, B. B. Zhu, and H. Lee, "Detecting malicious web links and identifying their attack types," in *Proc. 2nd USENIX Conf. Web Appl. Develop.*, 2011.

[96] R. M. Mohammad, F. Thabtah, and L. McCluskey, "An assessment of features related to phishing websites using an automated technique," in *Proc. Int. Conf. Internet Technol. Secured Trans.*, 2012, pp. 492–497.

[97] M. A. Adebowale, K. T. Lwin, E. Sánchez, and M. A. Hossain, "Intelligent web-phishing detection and protection scheme using integrated features of images, frames and text," *Exp. Syst. Appl.*, vol. 115, pp. 300–313, Jan. 2019.
[98] M. He, S.-J. Horng, P. Fan, M. K. Khan, R.-S. Run, J.-L. Lai, R.-J. Chen, and A. Sutanto, "An efficient phishing webpage detector," *Exp. Syst. Appl.*, vol. 38, no. 10, pp. 12018–12027, Sep. 2011.
[99] Y.-T. Hou, Y. Chang, T. Chen, C.-S. Laih, and C.-M. Chen, "Malicious web content detection by machine learning," *Exp. Syst. Appl.*, vol. 37, no. 1, pp. 55–60, Jan. 2010.
[100] D.-J. Liu, G.-G. Geng, X.-B. Jin, and W. Wang, "An efficient multistage phishing website detection model based on the CASE feature framework: Aiming at the real web environment," *Comput. Secur.*, vol. 110, Nov. 2021, Art. no. 102421.
[101] J. Mao, J. Bian, W. Tian, S. Zhu, T. Wei, A. Li, and Z. Liang, "Phishing page detection via learning classifiers from page layout feature," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, pp. 1–14, Dec. 2019.
[102] H. Zhang, G. Liu, T. W. S. Chow, and W. Liu, "Textual and visual content-based anti-phishing: A Bayesian approach," *IEEE Trans. Neural Netw.*, vol. 22, no. 10, pp. 1532–1546, Oct. 2011.
[103] S. Marchal, G. Armano, T. Gröndahl, K. Saari, N. Singh, and N. Asokan, "Off-the-hook: An efficient and usable client-side phishing prevention application," *IEEE Trans. Comput.*, vol. 66, no. 10, pp. 1717–1733, Oct. 2017.
[104] A. K. Jain and B. B. Gupta, "A machine learning based approach for phishing detection using hyperlinks information," *J. Ambient Intell. Humanized Comput.*, vol. 10, no. 5, pp. 2015–2028, May 2019.
[105] A. S. Bozkir and M. Aydos, "LogoSENSE: A companion HOG based logo detection scheme for phishing web page and E-mail brand recognition," *Comput. Secur.*, vol. 95, Aug. 2020, Art. no. 101855.
[106] F. C. Dalgic, A. S. Bozkir, and M. Aydos, "Phish-IRIS: A new approach for vision based brand prediction of phishing web pages via compact visual descriptors," in *Proc. 2nd Int. Symp. Multidisciplinary Stud. Innov. Technol. (ISMSIT)*, Oct. 2018, pp. 1–8.
[107] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, May 2003.
[108] A. C. Bahnsen, E. C. Bohorquez, S. Villegas, J. Vargas, and F. A. Gonzalez, "Classifying phishing URLs using recurrent neural networks," in *Proc. APWG Symp. Electron. Crime Res. (eCrime)*, Apr. 2017, pp. 1–8.
[109] F. Sadique, R. Kaul, S. Badsha, and S. Sengupta, "An automated framework for real-time phishing URL detection," in *Proc. 10th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2020, pp. 335–341.
[110] W. Ali and A. A. Ahmed, "Hybrid intelligent phishing website prediction using deep neural networks with genetic algorithm-based feature selection and weighting," *IET Inf. Secur.*, vol. 13, no. 6, pp. 659–669, Nov. 2019.
[111] M. Babagoli, M. P. Aghababa, and V. Solouk, "Heuristic nonlinear regression strategy for detecting phishing websites," *Soft Comput.*, vol. 23, no. 12, pp. 4315–4327, Jun. 2019.
[112] R. B. Basnet, A. H. Sung, and Q. Liu, "Feature selection for improved phishing detection," in *Advanced Research in Applied Artificial Intelligence* (Lecture Notes in Artificial Intelligence), vol. 7345, H. Jiang, W. Ding, M. Ali, and X. Wu, Eds. Cham, Switzerland: Springer, 2012, pp. 252–261.
[113] K. L. Chiew, C. L. Tan, K. Wong, K. S. C. Yong, and W. K. Tiong, "A new hybrid ensemble feature selection framework for machine learning-based phishing detection system," *Inf. Sci.*, vol. 484, pp. 153–166, May 2019.
[114] P. L. Indrasiri, M. N. Halgamuge, and A. Mohammad, "Robust ensemble machine learning model for filtering phishing URLs: Expandable random gradient stacked voting classifier (ERG-SVC)," *IEEE Access*, vol. 9, pp. 150142–150161, 2021.
[115] R. M. Mohammad, F. Thabtah, and L. McCluskey, "Intelligent rule-based phishing websites classification," *Inf. Secur., IET*, vol. 8, no. 3, pp. 153–160, May 2014.
[116] A. V. Ramana, K. L. Rao, and R. S. Rao, "Stop-Phish: An intelligent phishing detection method using feature selection ensemble," *Social Netw. Anal. Mining*, vol. 11, no. 1, Dec. 2021, Art. no. 110.
[117] T. M. Mitchell, *Machine Learning*. New York, NY, USA: McGraw-Hill, 1997.
[118] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012.
[119] S. C. H. Hoi, D. Sahoo, J. Lu, and P. Zhao, "Online learning: A comprehensive survey," *Neurocomputing*, vol. 459, pp. 249–289, Oct. 2021.
[120] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[121] M. Al-Sarem, F. Saeed, Z. G. Al-Mekhlafi, B. A. Mohammed, T. Al-Hadhrami, M. T. Alshammari, A. Alreshidi, and T. S. Alshammari, "An optimized stacking ensemble model for phishing websites detection," *Electronics*, vol. 10, no. 11, May 2021, Art. no. 1285.
[122] M. Sameen, K. Han, and S. O. Hwang, "PhishHaven—An efficient real-time AI phishing URLs detection system," *IEEE Access*, vol. 8, pp. 83425–83443, 2020.
[123] H. Shirazi, S. R. Muramudalige, I. Ray, and A. P. Jayasumana, "Improved phishing detection algorithms using adversarial autoencoder synthesized data," in *Proc. IEEE 45th Conf. Local Comput. Netw. (LCN)*, Nov. 2020, pp. 24–32.
[124] W. Zhang, Q. Jiang, L. Chen, and C. Li, "Two-stage ELM for phishing web pages detection using hybrid features," *World Wide Web*, vol. 20, no. 4, pp. 797–813, Jul. 2017.
[125] Y. A. Alsariera, V. E. Adeyemo, A. O. Balogun, and A. K. Alazzawi, "AI meta-learners and extra-trees algorithm for the detection of phishing websites," *IEEE Access*, vol. 8, pp. 142532–142542, 2020.
[126] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Identifying suspicious URLs: An application of large-scale online learning," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, Jun. 2009, pp. 681–688.
[127] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Learning to detect malicious URLs," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–24, Apr. 2011.
[128] S. Abdelnabi, K. Krombholz, and M. Fritz, "VisualPhishNet: Zero-day phishing website detection by visual similarity," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2020, pp. 1681–1698.
[129] A. Aljofey, Q. Jiang, Q. Qu, M. Huang, and J.-P. Niyigena, "An effective phishing detection model based on character level convolutional neural network from URL," *Electronics*, vol. 9, no. 9, Sep. 2020, Art. no. 1514.
[130] Y. Huang, Q. Yang, J. Qin, and W. Wen, "Phishing URL detection via CNN and attention-based hierarchical RNN," in *Proc. 18th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun./13th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, Aug. 2019, pp. 112–119.
[131] H. Le, Q. Pham, D. Sahoo, and S. C. H. Hoi, "URLNet: Learning a URL representation with deep learning for malicious URL detection," 2018, *arXiv:1802.03162*.
[132] C. Opara, B. Wei, and Y. Chen, "HTMLPhish: Enabling phishing web page detection by applying deep learning techniques on HTML analysis," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–8.
[133] M. Somesha, A. R. Pais, R. S. Rao, and V. S. Rathour, "Efficient deep learning techniques for the detection of phishing websites," *Sādhanā*, vol. 45, no. 1, pp. 1–18, Dec. 2020.
[134] W. Wang, F. Zhang, X. Luo, and S. Zhang, "PDRCNN: Precise phishing detection with recurrent convolutional neural networks," *Secur. Commun. Netw.*, vol. 2019, pp. 1–15, Oct. 2019.
[135] W. Wei, Q. Ke, J. Nowak, M. Korytkowski, R. Scherer, and M. Wozniak, "Accurate and fast URL phishing detector: A convolutional neural network approach," *Comput. Netw.*, vol. 178, Sep. 2020, Art. no. 107275.
[136] X. Xiao, W. Xiao, D. Zhang, B. Zhang, G. Hu, Q. Li, and S. Xia, "Phishing websites detection via CNN and multi-head self-attention on imbalanced datasets," *Comput. Secur.*, vol. 108, Sep. 2021, Art. no. 102372.
[137] X. Xiao, D. Zhang, G. Hu, Y. Jiang, and S. Xia, "CNN–MHSA: A convolutional neural network and multi-head self-attention combined approach for detecting phishing websites," *Neural Netw.*, vol. 125, pp. 303–312, May 2020.
[138] P. Yang, G. Zhao, and P. Zeng, "Phishing website detection based on multidimensional features driven by deep learning," *IEEE Access*, vol. 7, pp. 15196–15209, 2019.
[139] Y. Liang, J. Deng, and B. Cui, "Bidirectional LSTM: An innovative approach for phishing URL Identification," in *Innovative Mobile and Internet Services in Ubiquitous Computing* (Advances in Intelligent Systems and Computing), vol. 994, L. Barolli, F. Xhafa, and O. K. Hussain, Eds. Cham, Switzerland: Springer, 2020, pp. 326–337.
[140] S. Wang, S. Khan, C. Xu, S. Nazir, and A. Hafeez, "Deep learning-based efficient model development for phishing detection using random forest and BLSTM classifiers," *Complexity*, vol. 2020, pp. 1–7, Sep. 2020.
[141] S. Mahdavifar and A. A. Ghorbani, "DeNNeS: Deep embedded neural network expert system for detecting cyber attacks," *Neural Comput. Appl.*, vol. 32, no. 18, pp. 14753–14780, 2020.

[142] B. Wei, R. A. Hamad, L. Yang, X. He, H. Wang, B. Gao, and W. L. Woo, "A deep-learning-driven light-weight phishing detection sensor," *Sensors*, vol. 19, no. 19, Sep. 2019, Art. no. 4258.
[143] S. Al-Ahmadi and T. Lasloum, "PDMLP: Phishing detection using multilayer perceptron," *Int. J. Netw. Secur. Appl.*, vol. 12, no. 3, pp. 59–72, May 2020.
[144] A. El Aassal, S. Baki, A. Das, and R. M. Verma, "An in-depth benchmarking and evaluation of phishing detection research for security needs," *IEEE Access*, vol. 8, pp. 22170–22192, 2020.
[145] D. Dua and C. Graff, "UCI machine learning repository," School Inf. Computer Sci., Univ. California, Irvine, 2019. [Online]. Available: http://archive.ics.uci.edu/ml
[146] C. L. Tan. (2018). *Phishing Dataset for Machine Learning: Feature Evaluation*. [Online]. Available: https://data.mendeley.com/datasets/h3cgnj8hft/1
[147] G. Vrbančič. (2020). *Phishing Websites Dataset*. [Online]. Available: https://data.mendeley.com/datasets/72ptz43s9v/1
[148] G. Vrbančič, I. Fister, and V. Podgorelec, "Datasets for phishing websites detection," *Data Brief*, vol. 33, Dec. 2020, Art. no. 106438.

RASHA ZIENI received the master's degree in computer engineering from the University of Pavia, Italy, where she is currently pursuing the Ph.D. degree with the Department of Electrical, Computer and Biomedical Engineering. Her research interests include cybersecurity focusing on phishing detection and machine learning.

LUISA MASSARI received the Ph.D. degree in computer engineering from the University of Pavia, Italy, where she is currently an Assistant Professor of computer engineering with the Department of Electrical, Computer and Biomedical Engineering. Her research interests include performance evaluation and workload characterization of computer networks, systems and services, and of online social networks.

MARIA CARLA CALZAROSSA (Senior Member, IEEE) received the Laurea degree in mathematics from the University of Pavia, Italy, where she is currently a Professor of computer engineering with the Department of Electrical, Computer and Biomedical Engineering. Her research interests include performance evaluation and workload characterization of complex systems and services, cloud computing, benchmarking, and social networks.

Open Access funding provided by 'Università degli Studi di Pavia' within the CRUI CARE Agreement