A Review of Human- and Computer-Facing URL Phishing Features

A Review of Human- and Computer-Facing URL Phishing Features

Kholoud Althobaiti*[†] k.althobaiti@sms.ed.ac.uk

Ghaidaa Rummani[‡] gr15@hood.edu

Kami Vaniea* kvaniea@inf.ed.ac.uk

*University of Edinburgh, Edinburgh, UK
[†]Taif University, Taif, KSA
[‡]Hood College, Frederick, Maryland, USA

***Abstract*—When detecting phishing websites, both humans and computers rely on aspects of the website (features) to aid in their decision making. In this work, we conduct a review of URL-based phishing features that appear in publications targeting human-facing and automated anti-phishing approaches. We focus on both humans and computers to obtain a more comprehensive feature list and create a cross-community foundation for future research. We reviewed 94 papers and categorise their features into: lexical, host, rank, redirection, certificate, search engine, and black/white lists. We find that research on automation has used all feature categories but several, such as host-based features (e.g. DNS), are minimally explored in human-facing anti-phishing research.**

***Index Terms*—Phishing, Phishing features, Phishing Education, Usable security**

I. Introduction

Phishing, where users are tricked into giving away valuable data, is not only expensive [1], it is also hard for both humans and computers to detect accurately [2, 3]. After all, the goal of a phisher is to first get their message to users by bypassing automated detection systems, and then deceive users into interacting with the message. However, while phishers can manipulate many aspects of their communications, there are a few aspects that are very challenging for them to fully hide, such as the destination of URLs (Universal Resource Locators). In this paper, we review phishing research and catalogue URL-based anti-phishing features aimed at both humans and automated systems. Our aim is to create a foundation for future research to improve the state of human-facing support.

Ideally, all phishing detection, URL or otherwise, would be done automatically without human involvement. But there are two major challenges to doing so. First, while automatic detection is impressively accurate with classification rates as high as 99% [4] and the preferred first line of defence for most users [5], the remaining 1% is highly problematic and potentially very damaging [6]. Second, humans are needed to report and annotate new phishing attacks so that automated systems can in turn be updated to detect the latest threats. Effectively, humans label phishing, which is then used to train automated systems, which in turn causes phishers to change tactics [7], leading to undetected phishing, which is then reported by humans, starting the whole cycle over again.

Automatic phishing detection of URLs comes down to deciding if a URL's destination, is "bad" or not. For a human, "bad" can be defined as any website other than the one they intend on visiting. But computers lack users' understanding of context, so they must instead define "bad" based on pre-labeled lists (/white lists), heuristics (rule-based) [8], and building machine learning classifiers using labelled examples [9]. This difference means that humans and computers likely find different features more or less useful when making phishing judgements. Park et al. conducted a lab study to compare the abilities of machines and humans to detect phishing emails [2]. They found that humans are as good as machines in labelling legitimate emails. For phishing emails, some emails were easy to spot for humans but not machines while others were easier for machines to detect than humans. They concluded that a collaboration between machines and humans is needed to reach an optimal solution to combating phishing.

Since humans are not naturally skilled at detecting phishing, *education* and *support* are used to help them accurately detect it. Education approaches attempt to train users to look at specific features of the URL or communication. Examples include sending fake phishing emails to employees with targeted training [10], training via games [11, 12], dedicated upfront training, and online advice pages [13]. Some trainings also include guidance on how to use phishing features to differentiate between a safe and malicious page. Educating users takes time, and providing updates to that education is also very expensive, so theoretically there is a natural bias towards teaching features that are easier for humans to understand and that are stable across time.

However, some URLs are impossible for people to read even if they have high awareness. Punycode (RFC 3492) URLs, for example, allow Unicode characters to be encoded using ASCII such that there is no human-visible difference between the real URL and the malicious one even though the computer would see a difference. Detecting such problems requires *support* systems where the computer extracts and highlights feature data to support the human in making a decision. Two recent examples from research are TORPEDO [14] and Faheem [15]. Both of which provide the user with just-in-time information (features) with the goal of supporting users' decisions.

In this paper, we explore the literature to answer two questions: (1) What phishing URL features are used in existing research? (2) Are the features used in the automated detection research also explored in human-facing research?

To our knowledge, no prior literature review has considered

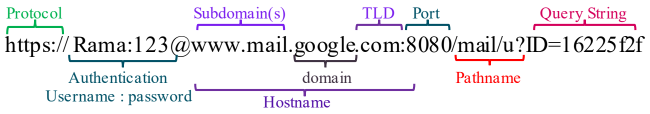

Figure 1. Example URL with the standard components labelled

URL-based phishing features in reference to both humans and computers. Several works have compared automated and human training approaches [5, 16]–[19]. Two general surveys looked at automated web phishing detection [20, 21]. Phishing features used in machine learning solutions were previously reviewed in [9] (2017) and [22] (2015). Reviews of feature usage in web content [23, 24] and DNS [25] also exist.

We review of phishing literature from three libraries, compiled a list of phishing features, and then group those features into categories. We find that there are a very large number of features and that all feature-types have been tried in the automated detection literature. However, several categorizations of features have minimal exploration in the human-facing work. Examples include, host features (i.e. DNS) and page popularity (i.e. PageRank). We also find that the domain of the URL is heavily used in human-facing work, but minimally used (beyond blacklists) in automated work.

II. Uniform Resource Locator (URL)

As shown in Figure 1, a URL is made up of a protocol, authentication, hostname, port, pathname and query. These are in turn made up of smaller components. Hostname, for example, is made up of the subdomains, domain, and top level domain (TLD). Only the protocol, and TLD are strictly necessary to create a working URL, though in practice the domain is also required. Generally to resolve a URL, the browser uses the Domain Name System (DNS) to locate the hostname's IP Address, then contacts the server using the protocol and port, it also provides the path, query, and authentication so the server can "locate" the requested resource.

The phisher wants users to load a page under the phisher's control. To do so, the phisher must, for a domain they control, accurately state protocol, domain, and TLD. This domain/TLD could be similar to an organisation they wish to impersonate, but cannot be identical. Subdomains are controlled by the domain owner, so a phisher can create arbitrary subdomains for their domain with virtually no oversight. The other elements (authentication, port, path, and query) can be ignored by the malicious server, and therefore can contain any syntactically valid information the phisher wants. The limits placed on the URL, mean that the domain/TLD is the most accurate in terms of where the URL will lead, but a phisher can select the domain to be confusing or put valid-looking URL elements into the other elements to confuse humans readers [26].

III. Methodology

Our research goal is to create a representative list of URL phishing features that have been tested with machines and/or with humans by reviewing past research papers.

A. Procedure

Literature was collected first using Google Scholar (October 2018), and then ACM Digital Library and the IEEE Xplore Digital Library (December 2018).

We searched Google Scholar with the keyword "phishing URL" to look for phishing features. One researcher started with the publications rated most relevant and reviewed the title and abstract to make sure the paper matched the inclusion criteria (Section III-B). They then reviewed the content, marking any sections that discussed phishing features. Identified features were then included in a spreadsheet used to track reviewed papers. They kept reviewing till new papers were no longer adding meaningfully different phishing features. For example, the features "count of subdomains" and "count of dots in hostname" effectively measure the same thing and were not considered meaningfully different. After completing the procedure for the first database, the researcher reviewed the second, and then third databases. A second researcher then went through each annotated paper and verified that features had been accurately identified in the paper and the spreadsheet.

Next, we grouped the features into categories. Where possible, categories were formed and named using common conventions from reviewed papers. "Lexical Features", for example, appears in several papers and always refers to features extracted from URL text. For each category shown in Table I, we also summarise key aspects of features such as how they are used (automatic, human) and limitations to their use (time, storage, and dependencies).

B. Inclusion/exclusion criteria

We included papers whose primary focus was the determination of whether a URL would lead to a phishing page without requiring the loading of the full page content. We focused on features that detected phishing rather than other attacks, such as XSS. We excluded papers which were not focused on URLs, such as those only looking at email content without also analysing URLs. Poster papers, extended abstracts, theses, and technical reports were also excluded due to lack or limited peer-review. We also excluded content-based features that required the full page to be loaded as our focus is preload. Due to the limited number of papers related human-facing features, we included any paper that studied people's susceptibility to phishing and otherwise met the above criteria.

C. Limitations

Stopping reviewing at saturation limits the scope of the work and likely results in some features not being included. We decided to stop at saturation anyway because: 1) there are a very large number of URL phishing-related papers (∼26,400 Google Scholar results), 2) papers tend to have high overlap in features used, and 3) the more effective features tend to appear in multiple papers. However, the result is still a bias toward well cited papers that closely match our search keywords.

IV. Phishing Features

Our review identified 94 papers, 58 of which were from Google Scholar with an overlap of 24 papers in IEEE or ACM. We categorise identified features into: lexical features, host features, rank features, redirection features, certificate features, search engine features, and black/white list features. In this section, we discuss the primary features for each category and their use in automated, and human-facing detection methods in detail. For the features with more than 4 citations we give an example of the papers use stated those features.

A. URL Lexical Features

Parsing the URL string itself and using the resulting components as features is very popular and reliable. Lexical features are attractive because they require low processing time, low amounts of data storage, and can be processed without having to call out other services [27], which is also a nice privacy feature. As a result, they have high real-time efficiency [28]. Since URLs are unique to a site, they are also impossible to fully spoof, so while it is possible to create a similar-looking URL (i.e. `pavpal.com` or `evil.com/paypal`) it is not possible to use the correct URL domain (i.e. `paypal.com`) in a phishing URL without first compromising the domain.

Although the use of URL lexical features alone has been shown to result in high accuracy (∼97%) [29, 30], phishers have learned how to make predicting a URL destination difficult by carefully manipulating the URL to evade detection [30]–[32]. Therefore, combining these features with others, such as host, is the most effective approach [33].

Domain. The domain is a prevalent feature in anti-phishing, likely because while phishers can register new domains, they are not generally able to attack a user visiting a legitimate domain. Extracting the domain is also easy, requiring only simple URL parsing. But using it alone to classify URLs is difficult because context, such as where the user wants to go, is missing. Instead it is combined with other features such as comparing it to the page title or meta-data [34, 35].

Human education papers commonly teach users how to parse out the domain as a way of enabling them to compare the domain to the one they expect to be visiting [12, 36]–[38]. However, people struggle to retain these skills [36].

For the papers using a human-support approach, such as those discussing SpoofStick and Netcraft, these tools used the hostname to help users correctly identify the sites they visit. In other tools, the domain part was pointed out as the destination [15] or highlighted once the mouse hovered over the link [14]. However, human-support only works if users are aware of what the correct domain is. Domains that do not directly line up with a recognized brand name, such as `www.nytimes.com`, can still confuse users even if they can parse the domain out correctly [15].

Other URL components. As shown in, Figure 1, the URL standard defines multiple components [39], and while the hostname is the most commonly used feature, the other components are also commonly used (e.g. [40]–[43]).

The authentication components, identified by the presence of '@', appears right after the protocol, making it an easy place to put a brand name and fool users (i.e. `http://bank.com@evil.com`). Authentication components are rare in legitimate URLs, so nearly all commercial modern automatic filters use it as a feature (e.g. [4, 44]–[46]).

Some automated detection papers use the existence of non-standard port numbers as a feature, where standard port numbers are either a common port number of the associated number with the protocol (e.g. [47]–[50]) because phishers use different port number to escape the detection [49]. However, port numbers are generally rare in URLs (0% in legitimate vs. 0.01% in phishing) [41].

"Non-standard" TLDs are also used as features but there is no consensus on the definition. Specific country-code TLDs (ccTLDs), such as '.cn' and '.ru', are used as features [51] while others focus on whether the TLD is a ccTLDs or a generic TLD such as '.net' and '.com' is used [52]–[54] and it is found to be a strong feature for classification (2017) [54]. In [55], ccTLDs are compared to host locations to see if the owner is located in the same country. Although ccTLDs are cheaper to obtain and sometimes used in phishing URLs, '.org' was found to be the most popular TLD for phishing websites in 2014 [7].

Other works apply weights to TLDs based on their training set [56, 57]. Weighting the features results in TLDs like '.info' and '.kr' being in the top phishing features while '.gov' and '.edu' being in the top legitimate features [57]. A set of 5 TLDs including '.com', '.net', and '.org' are also used [58].

Human-facing approaches have tried teaching users about URL structure components, such as TLD and authentication, to enable them to differentiate between the hostname and other URL components [11, 12, 15, 36]. Faheem [15] warns users about non-standard port numbers.

Special Characters. The presence of special characters such as '/', '=', and '_' is an aspect that has been used in many papers (e.g. [23, 58]–[60]) along with the frequency of their appearance [61]. Based on analysis of PhishTank URLs, 77% of phishing hostnames contain special characters [59].

Hyphens are one of the most commonly used features and the existence of a hyphen symbol, especially in the domain, is a phishing feature in automated and human-support methods (e.g. [12, 15, 61, 62]). Hyphens appear in legitimate URLs as well (2% in legitimate vs. 9% in phishing [40]), so they cannot be used as an indicator in isolation [28, 59]. Phishing websites tend to use the hyphen commonly to separate the brand name from the suffix (TLD) or prefix (i.e. `www-paypal.com`) [63], signifying the existence of a hyphen and suffix/prefix in the domain is a compelling phishing indicator. Other researchers included the number of hyphens as a feature (e.g. [47, 48, 64, 65]). The maximum number of hyphens in legitimate URLs hostnames is one while in phishing it is two or more [59]. Interestingly, the feature was one of the insignificant features in their classifier performance [54].

Dots and slashes are special characters that delineate components. Hence, the number of dots is linked to the number

Table I
SUMMARY OF IDENTIFIED FEATURES

Feature Category	Feature Subcategory	Most popular feature	Use of the features			Criteria		
			Automated	*Human education*	*Human support*	*Time*	*Storage*	*Dependency*
Lexical	Domain	Domain	Low	High	High	Low	Low	No
	Other URL components	Authentication	High	Mid	Low	Low	Low	No
	Special Characters	Number of dots	High	Low	Low	Low	Low	No
	Length	Length of URL	High	NA	NA	Low	Low	No
	Numeric Representation	Raw IP address	High	High	Mid	Low	Low	No
	Tokens & Keywords	Phishing keywords	High	Low	NA	Mid	Mid	No
	Deviated domains	Similarity with PhishTank	High	High	High	Mid	Mid	No
	Embedded URL		Low	NA	Low	Low	Low	Maybe
Host	Whois	Domain age	Mid	NA	Low	Mid	Low	Yes
	DNS	No records	Mid	NA	NA	Mid	Low	Yes
	Connection	Connection speed	Mid	NA	NA	Mid	Low	Yes
Rank	Domain Popularity	Alexa Rank	High	NA	Low	Mid	Low	Yes
	PageRank	Google PageRank	High	NA	NA	Mid	Low	Yes
Redirection		No. of Redirections	Mid	NA	Low	Mid	Mid	No
Certificate	Encryption	Is it HTTPS?	High	Mid	Low	Low	Low	No
	Certificate values	Is EV?	Low	NA	Low	Low	Low	Maybe
Search Engines		Query the Full URL	Mid	High	Low	Mid	Low	Yes
Black/White lists	Simple List	PhishTank	High	NA	Mid	Low	Low	Yes
	Proactive List	Blacklisting the IP	Mid	NA	Low	Mid	High	Yes

of subdomains and is a strong commonly used indication of phishing (e.g. [55, 66]–[68]). Analysis of phishing and legitimate URLs in [40, 41] found the number of dots in legitimate URL hosts ranges from 1 to 5, where 5 is rarely found in legitimate URLs, while in phishing URLs it ranges from 0 to 30. Some papers mark URLs with 3 or more dots as suspicious [65]. Therefore, the more dots, the more suspicious the URL [65, 68]. The number of slashes is also a phishing feature [27, 55, 69] with a threshold of five in some research [34, 59]. Having a hostname with no dots, consisting of only a single TLD, was also used as a feature [59].

The hyphen is the only special character used in human education [11, 12, 62] and human-support [15]. However, since hyphens only indicate phishing if there are too many of them, and "too many" is not well defined [41], these features may not be a good match for future human-facing research.

Length. Attackers tend to use long complex URLs as another way of hiding the true destination. Length-type (character count) features are commonly used to detect phishing URLs. Though, shortened and simple URLs can also be misclassified based on it [70].

One common feature is the length of the full URL (e.g. [59, 65, 71, 72]). The URL length is one of the features that contributed best to the classifier performance of [66, 72]. Taking dataset bias into consideration, phishing URLs are typically longer in publicly available blacklists than non-phishing URLs which are usually Alexa top sites [73].

Length of other URL components is also used as a feature, such as the length of the hostname (e.g. [56]–[58, 72]) – on average 20 characters in legitimate URLs [59], subdomain [48, 55], domain (e.g. [60, 64, 71, 72]), path (e.g. [41, 54, 71, 74]), or query [75]. The hostname's length (max 240 in phishing vs. 70 characters in legitimate) was the most useful as compared to the full URL and path's lengths [40]. Other features include average and longest domain and path token length, domain and path token count (e.g. [53, 54, 56, 64]), length of max-length in domain name (e.g. [58, 61, 67, 71]).

Human-facing methods in our dataset did not consider the URL length as a feature; nevertheless, participants in [38] assumed the longer the URL the less secure it was. Alsharnouby et al. [26] also found that people without training tend to classify URLs based on their perceived simplicity.

Numeric host representation. Legitimate URLs primarily use the registered hostname of the website, while phishers sometimes use different representations of the hostname to hide the destination. Examples include: IP addresses (i.e. `http://216.58.204.46`) (e.g. [68, 76]–[78]), dotless IP address (i.e. `http://3627733550/`) [69], encoded IP address hex value: (i.e. `http://0xd83acc2e`) [23, 40, 41, 64, 79], or even encode the hostname or part of it as Unicode (i.e. `http://%63%6E%2E%63%6F%6D`) (e.g. [59, 67, 78, 80]) to make the URL text difficult to understand [79]. In [59], 65.16% of phishing URLs contained Unicode. IP addresses are the most common feature in automated detection and the only numeric feature used in human-facing detection [12, 14, 15, 62].

Tokens and Keywords. Many automated detection papers tried tokenizing the URL and treating it as either a bag of words (e.g. [29, 30, 70, 81]), an N-gram [32, 82], a combination of tokens and bi-grams [83], or character frequency [46, 47]. The bag of words approach is effective, but the models are unstable over time and require frequent updating [53, 64].

Common keywords are also looked for in phishing URLs, such as "secure", "account", or "confirm" (e.g. [27, 40, 69, 79]). Similarly, human education has tried teaching users not to click on URLs with security-related keywords [12, 62]. However, keywords are unstable over time because attackers adapt and change words [28, 73]. Sananse et al. argue these features appear in both legitimate and phishing websites [84].

Number and average of terms are used as a feature [54, 59, 66, 72], with >4 terms in the host indicating phishing [59].

Path extension such as '.txt' [52, 53, 85] – attackers can add scripts to benign websites making '.js' pages more dangerous.

Specific out-of-place URL components can also be a feature. For example, the presence of two HTTP or HTTPS in the URL [27, 65], presence of TLD in the domain, subdomain or path position, such as "`cnn.com.malicious.org`" (e.g. [28, 65, 78, 86]), or a prefix (i.e. `www-chase.com`) [8, 48]. Out-of-place brand names can also be features, such as in the subdomain or path (e.g. [4, 62, 87, 88]). The NoPhish education game [77] added the brand name in the subdomain to help users understand phisher tactics. Similarly, in [62], users learn not to click on long hostnames if they contain part of well-known brand name. Providing correct parsing of URLs can also help users learn to read them [15].

Although phishing education research teaches users not to click on URLs if the domain has unknown terms (unrelated words) [11, 77], it is challenging to identify arbitrary words automatically [83]. Some papers attempt to detect random strings, using methods such as comparing URL tokens to proper or common nouns [4], or by calculating the string's entropy [48, 53]. Nonetheless, URLs are not necessarily constructed from proper nouns, as is the case with the New York Times "`www.nytimes.com`". A limitation of this approach is Internationalized Domain Names (IDNA), which cannot be detected with these features. Digits in the URL hostname may indicate randomness and are common in the host of phishing URLs (30%) compared to trustworthy URLs (3%) [40, 41]. The number or continuity of digits is also looked for in the host (e.g. [8, 58, 73, 89]) and other URL components [54]. Or even, the continuity of characters such as letters, digits and symbols and the number of each [53].

Deviated Domains. Construction of domain names to mimic legitimate ones is another lexical trick. An approach is to replace the TLD with a different TLD [88]. UTF8 encoding can also be used to produce identical-looking characters from different languages and alphabets, such as replacing the English 'b' with the Russian 'b' [90] or using confusing character combinations such as 'rn' for 'm' [69].

Prior work computed the similarity of domains and precomputed whitelisted domains [55], the target domains provided by PhishTank (e.g. [45, 60, 72, 91]), Vulnerable Sites List [60], top Alexa domains [31], and dictionary words [58]. In addition, Garera et al. [79] looked for the brand name in the domain concatenating with other characters. [66, 72] looked at if the starting/landing domain appears in, or is a substring of, the title in full or part. Verma and Dyer [61] also used features like Euclidean Distance to find deviated domains.

Training approaches teach users to check for spelling mistakes letter by letter [11, 62, 77]; however, users are not good at identifying visually deceptive domain names [12, 26, 36, 37]. For Unicode, human detection games excluded this feature due to the limitations of humans ability to recognise the subtle differences [11, 77]. Human-support solutions, such as Faheem [15], show that providing the user with assistance by automatically looking for similar popular domains and unexpected Unicode characters can be quite effective.

Embedded URLs. The query string can also contain a request for the destination site to forward the user on to another site. The occurrence of '//' in the query string is used as a feature in automated detection, however, it was not a top 5 feature [65]. Number of domains combined with TLD is also a phishing feature [28, 80]. No human education covered embedded URLs but human-support (TORPEDO) does give redirection information on mouse over [14].

B. Host Features

Querying community managed data sets, such as DNS, or reading HTTP headers, can provide many features; such as domain registration date, where it is hosted and who owns it.

Host-based features increase the overall accuracy [86]. And comparisons between lexical, host, and rank features found that host features contribute the most to classification performance [56, 83]. However, connecting to DNS or Whois also requires on average a 1.6 second delay [64] which is time expensive. Phishers can also avoid presenting accurate host features by using link shortners, web hosting services [92], or using compromised accounts so that registrations appear associated with the compromised account owner and not the phisher [93]. Some of these avoidances can be themselves used as features, for example, identifying if the host information is hosting provider or a link shortening service [92].

Whois Features. Whois is a query protocol that provides 48 features relating to websites [84].

Phishing websites, if they have details at all [34], generally have recent Whois registration dates, near future expiration dates, or recent update date (e.g. [51, 57, 60, 89]). While these three dates are commonly used in research, the domain age is the most commonly used, with a range of definitions for "recent" – 2 [93], 3 [94], or 6 months [65, 95]. Fang et al, [95] found that approximately 95% of the phishing URLs in their dataset were less than six months old and Hao et al. [96] found 55% of domains appeared the day after they were registered.

Other record information includes geolocation-based features, such as the timezone, netspeed [51], physical location of the country/city [7], or the IP geolocation [57, 71, 97]. Also, the existence of the domain in Whois [65, 89], the alignment between the URL domain and the domain registered in Whois [65, 78, 84], the registrars or registrants [60, 64, 67, 71, 86, 97]. Finally, the domain match between Registrar URL and Registrar Whois Server [84].

Some registrars also operate as hosting providers, some of which regularly scan their hosted domains for malicious pages (e.g. GoDaddy [84]) and remove them, while others do not. One feature is the historical reputation of the hosting provider associated with the URL.

The Whois based features used in human-support systems are: the server location by Netcraft [98], site country origin and length of registration by CallingID [99].

DNS features. Human-friendly hostnames are converted into IP addresses using DNS, so one common tactic of anti-phishing groups is to remove records of known phishing sites. A missing DNS record is a strong phishing feature (e.g. [42, 63, 94, 100]). It maintains information that is used as features, such as associated IP addresses (host, mail exchange, name server), Autonomous System number, domain name, sender policy framework, associated BGP and country code (e.g. [33, 54, 55, 64]), IP address segments [49], time to live (TTL) [55, 57, 69, 71], the number of resolved IP addresses [56, 87], number of name server and the number of IPs name servers associated with [87]. Additionally, the ratio of malicious Autonomous System numbers for the resolved IP address and Name servers associated with the resolved IPs [87].

DNS record information is used to ensure that sites are not hosted in a portion of the Internet that is considered disreputable or known for hosting phishing websites (e.g. [64, 83, 96, 97]). Temporary phishing websites also tend to not have PTR record values [55, 56].

Although DNS provides helpful information, DNS fluxing is used to hide the attacker's identity in an ever-changing network. To avoid Fluxing, Veni et al. look up the domain name of a URL and repeat the DNS lookup after TTL [87]. Notably, while using DNS server records is expensive and may face performance and resource strains [98], requesting the website after TTL period is prohibitive for real time detection.

No DNS features appeared in the human-facing papers.

Connection Features. Although we exclude full page download features, features regarding the connection to the website can contain useful information about the server, such as the http headers, without requiring a page download. Fields in the HTTP response headers contain information such as HTTP status [50, 80], content- type [50], content-length [50, 87] – negative in some phishing websites, and cookies [34, 50] – some phishing websites store cookies on foreign servers.

Connection speeds [57, 86, 87, 97] are faster on reputable websites, and also domain lookup tends to be quicker as popular websites tend to have a local DNS server.

Human-facing papers did not contain connection features.

C. Rank-based Features

Because they are not real websites, phishing sites tend to have a lower visitor count, and are not commonly linked to by other sites, resulting in low popularity and a low PageRank.

Domain popularity. Domain popularity is used in several automated detection systems [66, 101]. For example, Alexa's rank is a common feature (e.g. [4, 48, 55, 67]). Alexa produces this value based on the relative popularity of URLs throughout the previous three months [43], the threshold for legitimate URLs is 150k [63, 84], or 300k [43]. Alexa also provides rank reputation [43, 55]. However, a side-effect of this ranking system is that it is domain based; therefore, URLs from services such as link shorteners, and web hosting websites can still achieve a high popularity [89], shielding malicious agents using these services. Another metric used is webtraffic [42] or the popularity ranking of Netcraft [57].

Looking at human-support systems such as Netcraft and CallingID, we see that they provide site rank based on the hostname popularity [99]. In our opinion, providing the hostname popularity will mitigate the problem of free web hosting domains provided the web host gives each website its own subdomain thereby creating unique hostnames for each site. Yang et al. [101] also designed a warning to tell users about abnormally low website traffic ranks. They argued that including the concept in the warning design reduces the click-through rate in automated detection systems.

Page Popularity. Roughly, PageRank is a weighted count of how many other pages link to this web page, where the weights are the other pages' PageRanks. Intuitively, it measures how many other well-known pages link to this one.

Google's PageRank is successfully used as a feature in automated detection (e.g. [28, 42, 43, 79]) where URLs that are ranked less than 5 are classified as phishing in [84]. Veni et al. [87] combined the PageRank results from AltaVista, AllTheWeb, Google, Yahoo, and Ask to get a more accurate PageRank. Garera et al. [79] used a number of PageRank features such as, the URL and hostname PageRank, the page presence in the index of a crawler dataset index [79], or in Google index [42, 57].

The PageRank is a robust feature since Google updates it frequently; however, it still produces false positives, and is recommended to be used in conjunction with other features [43, 56]. Another problem with the PageRank is link-farming where attackers manipulate the rank by increasing the number of websites that link to the URL [56, 87]. To evade farming problems, Veni et al. [87] added more features such as the number of the different links that link to the page, and whether it is linked to other malicious URLs.

D. Search Engine features

Search engines optimise for finding the website that the user most wants or expects, given only a small set of keywords. This behaviour makes search engines an excellent proxy for what web site a user might expect to see given some contextual keywords and allows automated systems to identify what website a phishing attack is trying to mimic, because a search for the website title will often bring up the real site [43, 55, 68]. Search engines can also be used to spell check a domain and see if it is a short edit distance from a real one.

Automated detection systems query search engines using the full URL, domain name, page title with domain name, or domain name with TLD (e.g. [41, 45, 69, 98]). Varshney et al. compared the search results of queries containing domain name, page title, description, and domain name with the page title [98] and found the page title with the domain name gave the highest accuracy. Both page title and domain name can be fetched without needing to load the entire page. A website is considered to not be phishing if it appears in the top 30 [28] or top 10 [34, 69] results in a search for its own URL. In 2014, Basnet et al. [40, 41] compared the results of searching Google, Yahoo and Bing for either the URL or the domain. They found Google to have the highest accuracy; however, they still used all three in their phishing classifiers. While the complexity of querying a search engine is lower than querying the DNS [98], querying three of them in real time may be too time intensive.

Google spelling suggestions are used to detect the similarity between potential phishing domains and popular domains [43, 55, 68]. Another usage of search engines is finding intra-URL relatedness features, the relatedness between domain with TLD and the rest of the words in the URL, are used in [45, 102] using Google Trends and Yahoo Clues. The only limitation in this feature is the limited number of returned results.

Search engine features are not always consistent and can be different based on the location of the searcher [84]. This issue is particularly problematic for automated systems where the server and the user may not be in the same country.

Human-facing systems tend to advise users that when they are uncertain about a destination, they should search for it and select one of the top results as a way of finding the "correct" website [12, 14, 15]. Using this advice, users are able to accurately classify URLs [26].

E. Redirection-based Features

Redirection can be identified using the HTTP status code, page meta-refresh tag, or JavaScript. The latter occurs with shortened links [7], where URLs with a small number of characters redirect to longer URLs. These are common in Twitter phishing URLs [103].

Automated systems can also follow redirection links providing two features: the initial URL, and the landing URL [104, 105]. However, phishers will sometimes cloak URL redirects, by checking for features of the user's system first, and then deciding which landing page to send them to. Therefore, other features must also be used. For example, the number of different domains and IP addresses in the chain [106]. The number of redirections also indicates phishing URLs [66, 72, 89, 106] and has been used as a phishing feature [106, 107]. URLs with <2 redirections were found to be legitimate, $2-4$ were suspicious, and >4 were considered phishing [107]. The similarity between the hostname in the URLs in the redirect chains is also a feature since legitimate URLs typically redirect to same-domain URLs [66, 72].

For shortened links, Gupta et al. [104] analysed blacklisted Bitly shortened links. They found several features such as the time between the shortening and the domain creation, or the time between the shortening and using the link. The numbers of redirections has been also shown as a feature in nested shortened URLs, where 80% of the phishing tweets have at least one redirection [103].

Humans are unable to identify redirection prior to clicking without machine support, and even after clicking redirection can happen so fast that a user cannot see it happening. Some human-support systems detect redirection for users and show them the final destination before clicking [14, 15].

F. Certificate-based Features

SSL/TLS is a protocol commonly used for encrypting web traffic. An initial step of the protocol is for the client's browser to fetch the public key certificate from the server. The certificate is used to validate the pubic key, which in turn is then used to setup an encrypted connection.

Encryption. Early automation work found that legitimate URLs usually supported encryption, while phishing URLs generally did not (e.g. [55, 74, 75, 89]). Since obtaining a valid certificates cost money, it made some sense that legitimate sites would be more likely to have them. However, after the introduction of LetsEncrypt, which provides free certificates to websites, support for encryption is no longer a significant phishing feature as both legitimate and phishing sites now have valid certificates [8, 90].

Similarly, prior advice to end-users was to "look for the lock icon" which signalled encryption. This is still good security advice [11, 15], and can impact user decision making [14, 26, 76], but it no longer helps detect phishing.

Certificates values. Values found in the certificate fields, beyond setting up the encryption, are also used as features. Torroledo et al. [108] used ~40 TLS features to classify URLs, such as the validation level, issuer location, or if it is paid or free. The certificate start and end date are used in [50, 108]. Trusted certificate authorities, such as Comodo, Symantec, GoDaddy, GlobalSign and DigiCert [50, 94] are used to judge the certificate trustworthiness.

Public key certificates can be verified at one of three levels which range from a simple check that the domain is controlled by the certificate requester (domain-validate) [50], to the Extended Validation (EV) certificate which requires the issuer to perform extensive checks of the identity of the organisation the certificate is being given to [108]. EV certificates are effective at proving identity, but they also expensive to obtain, and many sites do not have them [38, 90].

Most modern browsers show EV certificate information to end-users by providing the validated organisation's name in a green box next to the domain. In a lab study, 19% of participants referred to the certificate when deciding safely [26]. However, in our paper set, all certificate-based features shown to humans required that the page be loaded first except for Netcraft which provide information on request [38].

G. Black/White List Features

When a URL is labelled as phishing, it is typically added to a publicly visible blacklist so that other anti-phishing tools can quickly block it [88].

Simple List. Lists are used in automated detection as a strong feature due to their low false positive rate – almost 0% for newly observed phishing URLs [7]. Blacklists are also very efficient. It may take humans a while to label the URLs as phishing, but once labelled, computers can easily compare URLs against common blacklists [86, 88], such as PhishTank [84, 85], Google Safe Browsing (e.g. [69, 85, 104, 109]), VirusTotal [109, 110] or Anubis [109], which can be accessed via API or even downloaded locally.

Proactive list. Unfortunately, while blacklists are very accurate, they are insufficient to detect all phishing websites [37], likely because blacklists only contain previously seen URLs.

Prior work utilized the lists to proactively discover unreported phishing URLs and trustworthy ones [88, 100].

Several approaches have proposed features derived from blacklists, such as marking a URL as malicious if its domain matches malicious domains [40, 109, 111]. Or even if its IP address is on a blacklist [40, 57, 95, 97]. However, blocking a IP addresses runs the risk of inadvertently blocking good sites if the phishing site is using a free hosting service, so features were proposed that compared the URL's domain to a pre-computed list of web hosting services; therefore, Prakash et al. blacklisted an IP address depending on the number of phishing URLs associated with it [88].

Attackers often reuse phishing kits, therefore, a URL could have similar pathname (directory) to previously blacklisted URLs [88, 90, 91, 112]. They also reuse their redirection servers [106], therefore, another feature is to expand shortened URLs before adding to the list [40, 106] and also include URLs in the redirection chain, including HTTP, meta and JavaScript redirection [88, 100], in the blacklists.

Proactive detection of phishing URLs is computationally expensive since it requires storage space in the majority of the features and more time to compare the downloaded list.

Creating whitelists of validated websites is more complicated. Automated systems typically use high-profile popular sites such as Alexa's top sites [79] or a customised whitelist of domains often targeted in phishing attacks [79]. However, maintaining a comprehensive list of validated URLs is intractable [37]. An alternative solution is to add URLs to a local whitelist after users visit it [101, 113].

Blacklists are heavily used in human-support systems with most modern web browsers actively blocking URLs that appear on popular blacklists. Plugins like Netcraft, as a first step of defence, also block reported and verified phishing websites. Similar to CallingID, it also shows the users risk scores on a coloured scale along with the other phishing indicators discussed previously [99]. Human-support systems also leverage user's awareness of phishing indicators and allow them to report phishing URLs for labelling and potential inclusion on blacklists. Cloudmark Anti-Fraud relies solely on users' reports and verification to block phishing URLs [99].

V. Discussion

The above described research provides ample opportunities to improve human-facing approaches, particularly human-support systems which have the technical ability to leverage many of the features used by automated approaches and provide that data to humans in a meaningful way. Below we discuss some of the more thought-provoking issues.

Shifting effectiveness of features. While many automated detection papers discussed the effectiveness (weights) of their features, comparing results is challenging. Effectiveness was evaluated differently across papers, such as statistically by comparing the feature prevalence in benign and phishing URLs [40] or by comparing the classifier accuracy between different groups of features [56]. Feature effectiveness also changes based on the data set and the domain [114]. Link shorteners, for example, are common on social media, but less so in email communication. So, different features work better in different situations.

Effectiveness also changed over time as phishing tactics themselves changed. For example, lexical features like the number of subdomains are subject to both the current website design trends and phisher behaviour. The introduction of free signed certificates by LetsEncrypt also impacted the lock icon (encryption) phishing feature used by many people.

Balancing "safe" and "phish" data sets. A serious methodology problem we kept running into was the unbalanced selection of data sets to represent safe and phishing URLs, also noted by [73] who points to [72] as being one of the only papers to use balanced data sets.

Papers commonly use repositories like PhishTank or Google Safe Browsing to get phishing URLs. These provide a realistic view of what users are seeing. PhishTank, in particular, provides the full raw URL as it was reported. Finding safe URLs that are representative of a "normal" URL is more challenging. Many papers use repositories such as Alexa's top sites or Open Directory Project (DMOZ). The problem with these data sets is that they are taken from directory listings and therefore may not represent the composition of a typical safe URL. For example, query strings are rarely included in directory listings, but are very common in say Amazon.com URLs for products. Essentially, the safe and phish datasets tend to be drawn from different distributions which brings the true effectiveness of features, such as length, into question because the source of the difference is unclear.

Host-obscuring tactics. Many features are dependent on the ability to accurately extract the URL's hostname, such as lexical features and host-based features. Phishers can obscure the hostname by using link shortening services or redirection which hide the final destination URL. These tactics impact the ability of both humans and computers since people can only see the initial URL and computers must take the extra step of resolving the URL to get the final destination before trying to make any predictions. Some research exists on how to detect and resolve these URLs technically, but minimal research looks at how people think about redirects, or how to meaningfully present the information about them.

Exploring human-facing features. Human-education approaches try to help the user learn to extract phishing features on their own and interpret them. But humans take time to learn and teaching them new things is challenging, so it is vital that human-education approaches require minimal prior knowledge and be robust with a few false positives. For example, long URLs and hyphens are bad human-facing features because they are both found in legitimate and phishing URLs which may confuse users. Conversely, domains are a good choice because how the user interacts with them does not change quickly and they can leverage their knowledge.

We recommend that future work explore the use of automation features in human-support. While many features cannot be understood by humans unaided, there is great potential for human-support soloutions to use them.

VI. Conclusion

With the aim of providing a foundation for future research into human-facing phishing support, we reviewed features used in the automated, human-education and human-support phishing detection systems. In total we reviewed 94 papers and grouped the resulting features into 7 categories including: lexical, host, rank, redirection, certificate, search engines, and black\white lists. We found that all feature categories were used by automated phishing detection, but human-facing approaches have only evaluated some of them.

Acknowledgement

We would like to thank everyone in the TULiPS group at the University of Edinburgh for helpful discussions and feedback. This research is funded in part by a Google Research Award.

References

[1] "2018 Data Breach Investigations Report," Verizon, Tech. Rep., 2018, https://vz.to/2Rzk8Zw.

[2] G. Park *et al.*, "Comparing machine and human ability to detect phishing emails," in *IEEE SMC*, 2014.

[3] K. R. Sahu *et al.*, "A Survey on Phishing Attacks," *Journal of Computer Applications*, 2014.

[4] P. K. V *et al.*, "Performance study of classification techniques for phishingURLdetection," in *6th ICoAC*, 2014.

[5] A. Tewari *et al.*, "Recent survey of various defense mechanisms against phishing attacks," *Journal of Information Privacy and Security*, 2016.

[6] NCSC-NCA, "Cyber security breaches survey," Tech. Rep., 2019, https://bit.ly/2K2eQq7.

[7] S. Gupta *et al.*, "Emerging phishing trends and effectiveness of the anti-phishing landing page," in *APWG eCrime*, 2014.

[8] M. Al-Janabi *et al.*, "Using Supervised Machine Learning Algorithms to Detect SuspiciousURLsin Online Social Networks," in *IEEE/ACM Advances in Social Networks Analysis and Mining*, 2017.

[9] D. Sahoo *et al.*, "MaliciousURLDetection using Machine Learning: A Survey," *arXiv preprint*, 2017.

[10] P. Kumaraguru *et al.*, "School of Phish: A Real-world Evaluation of Anti-phishing Training," in *5th SOUPS '09*, 2009.

[11] G. Canova *et al.*, "Learn to spot phishingURLswith the android nophish app," in *IFIP Advances in Information and Communication Technology*, 2015.

[12] S. Sheng *et al.*, "Anti-Phishing Phil: The Design and Evaluation of a Game That Teaches People Not to Fall for Phish," in *SOUPS '07*, 2007.

[13] R. Wash *et al.*, "Who Provides Phishing Training? Facts, Stories, and People Like Me," in *CHI Conf.*, 2018.

[14] M. Volkamer *et al.*, "User experiences of TORPEDO: TOoltip-poweRed Phishing Email DetectiOn," *Computers and Security*, 2017.

[15] K. Althobaiti *et al.*, "Faheem : ExplainingURLsto people using a Slack bot," in *AISB*, 2018.

[16] A. N. Shaikh *et al.*, "A literature review on phishing crime, prevention review and investigation of gaps," in *10th Software, Knowledge, Information Management & Applications*, 2016.

[17] S. Gupta *et al.*, "A literature survey on social engineering attacks: Phishing attack," in *Proceeding of ICCCA*, 2016.

[18] V. Suganya, "A Review on Phishing Attacks and Various Anti Phishing Techniques," *Journal of Computer Applications*, 2016.

[19] S. Purkait, "Phishing counter measures and their effectiveness literature review," *Information Management & Computer Security*, 2012.

[20] Z. Dou *et al.*, "Systematization of knowledge (sok): A systematic review of software-based web phishing detection," *IEEE Communications Surveys Tutorials*, Fourthquarter 2017.

[21] G. Varshney *et al.*, "A survey and classification of web phishing detection schemes," *Security and Communication Networks*, 2016.

[22] D. R. Patil *et al.*, "Survey on Malicious Web pages Detection Techniques," *Journal of u- and e-Service, Science and Technology*, 2015.

[23] S. B. Rathod *et al.*, "A comparative performance evaluation of content based spam and maliciousURLdetection in E-mail," in *IEEE CGVIS*, 2015.

[24] I. Qabajeh *et al.*, "A recent review of conventional vs. automated cybersecurity anti-phishing techniques," *Computer Science Review*, 2018.

[25] Y. Zhauniarovich *et al.*, "A Survey on Malicious Domains Detection through DNS Data Analysis," *CoRR*, 2018.

[26] M. Alsharnouby *et al.*, "Why phishing still works: User strategies for combating aphishing attacks," *Journal of Human Computer Studies*, 2015.

[27] A. Haider *et al.*, "PhishingURLDetection using Neural Network Optimized by Cultural Algorithm," *Journal of Computer Sciences and Engineering*, 2018.

[28] G. Xiang *et al.*, "CANTINA+: A Feature-Rich Machine Learning Framework for Detecting Phishing Web Sites," *ACM Transactions on Information and System Security*, 2011.

[29] M. Khonji *et al.*, "LexicalURLanalysis for discriminating phishing and legitimate e-mail messages," in *Internet Technology and Secured Transactions*, 2011.

[30] A. Blum *et al.*, "Lexical Feature Based PhishingURLDetection Using Online Learning," in *3rd ACM AISec*, 2010.

[31] M. Darling *et al.*, "A lexical approach for classifying malicious urls," in *HPCS*, 2015.

[32] A. Y. Daeef *et al.*, "Wide scope and fast websites phishing detection usingURLslexical features," in *3rd ICED*, 2016.

[33] M. N. Feroz *et al.*, "PhishingURLDetection UsingURLRanking," in *IEEE Congress on Big Data*, 2015.

[34] V. S. Lakshmi *et al.*, "Efficient prediction of phishing websites using supervised learning algorithms," in *Communication Technology and System Design*, 2012.

[35] R. Srinivasa Rao *et al.*, "Detecting Phishing Websites using Automation of Human Behavior," in *3rd ACM Workshop on CPSS '17*, 2017.

[36] G. Canova *et al.*, "NoPhish App Evaluation : Lab and Retention Study," in *Workshop on Usable Security and Privacy*, 2015.

[37] J.-P. Erkkilä, "Why we fall for Phishing," in *CHI*, 2011.

[38] A. Xiong *et al.*, "Is Domain Highlighting Actually Helpful in Identifying Phishing Web pages?" *HUMAN FACTORS*, 2017.

[39] T. Berners-Lee *et al.*, "RFC1738: Uniform Resource Locators (URL)," https://www.w3.org/Addressing/rfc1738.txt, 1994, wWW Consortium Request for Comments.

[40] R. B. Basnet *et al.*, "Towards Developing a Tool to Detect Phishing urls: A Machine Learning Approach," in *IEEE Computational Intelligence Communication Technology*, 2015.

[41] ——, "Learning to detect phishing URLs," *IJRET: Journal of Research in Engineering and Technology*, 2014.

[42] Y. Mourtaji *et al.*, "Perception of a new framework for detecting phishing web pages," in *Mediterranean Symposium on Smart City Application '17*, 2017.

[43] L. A. T. Nguyen *et al.*, "A novel approach for phishing detection using URL-based heuristic," *ComManTel*, 2014.

[44] K. L. Chiew *et al.*, "Utilisation of website logo for phishing detection," *Computers and Security*, 2015.

[45] H. Y. Abutair *et al.*, "Using Case-Based Reasoning for Phishing Detection," *Procedia Computer Science*, 2017.

[46] C. Liu *et al.*, "Finding effective classifier for maliciousURLdetection," in *2nd ICMSS*, 2018.

[47] S. Gupta, "Efficient malicious domain detection using word segmentation and BM pattern matching," in *ICRAIE*, 2016.

[48] W. Chen *et al.*, "Phishing Detection Research Based on LSTM Recurrent Neural Network," in *Advances in Internet, Data & Web Technologies*, 2018.

[49] G. Tan *et al.*, "Adaptive MaliciousURLDetection: Learning in the Presence of Concept Drifts," in *17th IEEE on TrustCom and 12th IEEE On BigDataSE*, 2018.

[50] S. N. Bannur *et al.*, "Judging a Site by its Content : Learning the Textual , Structural , and Visual Features of Malicious Web pages," in *4th ACM Workshop on Security and Artificial Intelligence*, 2011.

[51] D. Canali *et al.*, "Prophiler: A Fast Filter for the Large-scale Detection of Malicious Web pages," in *WWW*, 2011.

[52] H. K. Pao *et al.*, "MaliciousURLdetection based on Kolmogorov complexity estimation," in *IEEE/WIC/ACM Joint Web Intelligence and Intelligent Agent Technology*, 2012.

[53] M. Lin *et al.*, "MaliciousURLfiltering ? – A big data application," in *IEEE Big Data*, 2013.

[54] G. Chakraborty *et al.*, "AURLaddress aware classification of malicious websites for online security during web-surfing," in *IEEE ANTS*, 2017.

[55] J.-l. Lee *et al.*, “Heuristic-based Approach for Phishing Site Detection UsingURLFeatures,” *Advances in Computing, Electronics and Electrical Technology*, 2015.

[56] H. Liu *et al.*, “Learning based Malicious Web Sites Detection using Suspicious URLs,” in *Software Engineering.*, 2016.

[57] J. Ma *et al.*, “Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs,” in *ACM KDD ’09*, 2009.

[58] R. Kumar *et al.*, “MaliciousURLdetection using multi-layer filtering model,” in *14th ICCWAMTIP*, 2017.

[59] S. C. Jeeva *et al.*, “Intelligent phishingURLdetection using association rule mining,” *Human-centric Computing and Information Sciences*, 2016.

[60] Y. Xue *et al.*, “Phishing sites detection based onURLCorrelation,” in *4th CCIS*, 2016.

[61] R. Verma *et al.*, “On the Character of Phishing urls: Accurate and Robust Statistical Learning Classifier,” in *5th ACM CODASPY ’15*, 2015.

[62] N. A. G. Arachchilage *et al.*, “Phishing threat avoidance behaviour: An empirical investigation,” *Computers in Human Behavior*, 2016.

[63] R. M. Mohammad *et al.*, “Intelligent rule-based phishing websites classification,” *IET Information Security*, 2014.

[64] A. Le *et al.*, “PhishDef:URLnames say it all,” in *Proc. of IEEE INFOCOM*, 2011.

[65] A. K. Jain *et al.*, “PHISH-SAFE:URLfeatures-based phishing detection system using machine learning,” in *Advances in Intelligent Systems and Computing*, 2018.

[66] S. Marchal *et al.*, “Know Your Phish: Novel Techniques for Detecting Phishing Sites and Their Targets,” in *Distributed Computing Systems*, 2016.

[67] S. Wen *et al.*, “Detecting Malicious Websites in Depth through Analyzing Topics and Web-pages,” in *2nd ICCSP*, 2018.

[68] F. Kausar *et al.*, “Hybrid Client Side Phishing Websites Detection Approach,” *IJACSA*, 2014.

[69] V. R. Hawanna *et al.*, “A novel algorithm to detect phishing urls,” *ICACDOT*, 2016.

[70] K. Su *et al.*, “SuspiciousURLFiltering Based on Logistic Regression with Multi-view Analysis,” in *8th Asia Joint Information Security*, 2013.

[71] A. R. Nagaonkar *et al.*, “Finding the maliciousURLsusing search engines,” in *3rd INDIACom*, 2016.

[72] S. Marchal *et al.*, “Off-the-hook: An efficient and usable client-side phishing prevention application,” *IEEE Transactions on Computers*, 2017.

[73] H. Shirazi *et al.*, “”Kn0W Thy Doma1N Name”: Unbiased Phishing Detection Using Domain Name Based Features,” in *Access Control Models and Technologies*, 2018.

[74] S. Gupta *et al.*, “Invitation or Bait? Detecting MaliciousURLsin Facebook Events,” in *11th IC3*, 2018.

[75] M. Moghimi *et al.*, “New rule-based phishing detection method,” *Expert Systems with Applications*, 2016.

[76] J. S. Downs *et al.*, “Decision strategies and susceptibility to phishing,” in *SOUPS ’06*, 2006.

[77] G. Canova *et al.*, “NoPhish: An Anti-Phishing Education App,” in *10th Security and Trust Management*, 2014.

[78] D. Zhang *et al.*, “A domain-feature enhanced classification model for the detection of Chinese phishing e-Business websites,” *Information and Management*, 2014.

[79] S. Garera *et al.*, “A Framework for Detection and Measurement of Phishing Attacks,” in *ACM Workshop on Recurring Malcode*, 2007.

[80] N. A. Azeez *et al.*, “CyberProtector: Identifying CompromisedURLsin Electronic Mails with Bayesian Classification,” in *CSCI*, 2016.

[81] B. Gyawali *et al.*, “Evaluating a Semisupervised Approach to PhishingURLIdentification in a Realistic Scenario,” in *8th CEAS*, 2011.

[82] R. Verma *et al.*, “What’s in a url: Fast Feature Extraction and MaliciousURLDetection,” in *3rd ACM on Workshop on Security And Privacy Analytics*, 2017.

[83] M. N. Feroz *et al.*, “Examination of data, rule generation and detection of phishingURLsusing online logistic regression,” in *Proc. of IEEE Big Data*, 2014.

[84] B. E. Sananse *et al.*, “PhishingURLDetection: A Machine Learning And Web Mining-Based Approach,” *Journal of Computer Applications*, 2015.

[85] H. Yao *et al.*, “Towards preventing QR code based attacks on android phone using security warnings,” in *8th ACM ASIA CCS ’13*, 2013.

[86] J. Ma *et al.*, “Identifying suspicious urls,” in *26th Annual Conf. ICML ’09*, 2009.

[87] R. H. Veni *et al.*, “Identifying Malicious Web Links and Their Attack Types in Social Networks,” *IJSRCSEIT*, 2018.

[88] P. Prakash *et al.*, “PhishNet: Predictive blacklisting to detect phishing attacks,” in *IEEE INFOCOM*, 2010.

[89] G. Bottazzi *et al.*, “MP-shield: A framework for phishing detection in mobile devices,” in *IEEE 15th on CIT, 14th on IUCC, and 13th on Dependable, Autonomic and Se*, 2015.

[90] A. Oest *et al.*, “Inside a phisher’s mind: Understanding the anti-phishing ecosystem through phishing kit analysis,” in *APWG eCrime*, 2018.

[91] H. Bo *et al.*, “A Hybrid System to Find & Fight Phishing Attacks Actively,” in *IEEE/WIC/ACM Web Intelligence and Intelligent Agent Technology*, 2011.

[92] J. Zhang *et al.*, “URL based Gateway Side Phishing Detection Method,” in *IEEE Trustcom/BigDataSE/ISPA*, 2016.

[93] B. Alghamdi *et al.*, “Toward detecting malicious links in online social networks through user behavior,” in *IEEE/WIC/ACM WIW*, 2016.

[94] A. K. Jain *et al.*, “Comparative Analysis of Features Based Machine Learning Approaches for Phishing Detection,” in *3rd INDIACom*, 2016.

[95] L. Fang *et al.*, “A proactive discovery and filtering solution on phishing websites,” in *Proc. of IEEE Big Data*, 2015.

[96] S. Hao *et al.*, “Monitoring the Initial DNS Behavior of Malicious Domains,” in *ACM SIGCOMM - IMC ’11*, 2011.

[97] J. Ma *et al.*, “Learning to Detect Malicious urls,” *ACM Transactions on Intelligent Systems and Technology*, 2011.

[98] G. Varshney *et al.*, “A phish detector using lightweight search features,” *Computers and Security*, 2016.

[99] Y. Zhang *et al.*, “Phinding Phish: Evaluating Anti-Phishing Tools,” in *14th annual symposium NDSS*. Citeseer, 2007.

[100] L. H. Lee *et al.*, “POSTER: Proactive Blacklist Update for Anti-Phishing,” in *ACM SIGSAC on CCS ’14*, 2014.

[101] W. Yang *et al.*, “Use of phishing training to improve security warning compliance: evidence from a field experiment,” in *HoTSoS*, 2017.

[102] S. Marchal *et al.*, “PhishStorm: Detecting phishing with streaming analytics,” *IEEE Transactions on Network and Service Management*, 2014.

[103] C. Grier *et al.*, “@Spam: The Underground on 140 Characters or Less,” in *ACM on CCS*, 2010.

[104] N. Gupta *et al.*, “bit.ly/malicious: Deep dive into shortURLbased e-crime detection,” in *APWG on eCrime*, 2014.

[105] N. S. Gawale *et al.*, “Implementation of a system to detect maliciousURLsfor Twitter users,” in *ICPC*, 2015.

[106] S. Lee *et al.*, “WarningBird: A Near Real-Time Detection System for SuspiciousURLsin Twitter Stream,” *IEEE Transactions on Dependable and Secure Computing*, 2013.

[107] P. Singh *et al.*, “Phishing Websites Detection through Supervised Learning Networks,” *Proc. of ICCCT*, 2015.

[108] I. Torroledo *et al.*, “Hunting Malicious TLS Certificates with Deep Neural Networks,” in *11th ACM AISec ’18*, 2018.

[109] K. Tsyganok *et al.*, “Development the Method of Detection the Malicious pages Interconnection in the Internet,” in *6th Security of Information and Networks*, 2013.

[110] O. Catakoglu *et al.*, “Automatic Extraction of Indicators of Compromise for Web Applications,” in *25th WWW ’16*, 2016.

[111] M. Akiyama *et al.*, “Searching Structural Neighborhood of MaliciousURLsto Improve Blacklisting,” in *IEEE/IPSJ*, 2011.

[112] C. H. Hsu *et al.*, “Identify Fixed-path Phishing Attack by STC,” in *CEAS ’11*, 2011.

[113] Y. Cao *et al.*, “Anti-phishing based on automated individual white-list,” in *DIM ’08*, 2008.

[114] H. Zuhair *et al.*, “Feature Selection for Phishing Detection: A Review of Research,” *Journal of Intelligent Systems Technologies and Applications*, 2016.