SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks 2016; **9**:6266–6284
Published online 26 October 2016 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.1674

REVIEW ARTICLE

A survey and classification of web phishing detection schemes

Gaurav Varshney[1]*, Manoj Misra[1] and Pradeep K. Atrey[2]

[1] Department of Computer Science and Engineering, Indian Institute of Technology Roorkee, Roorkee, Uttarakhand, India
[2] Department of Computer Science, State University of New York (SUNY), Albany, NY, U.S.A.

ABSTRACT

Phishing is a fraudulent technique that is used over the Internet to deceive users with the goal of extracting their personal information such as username, passwords, credit card, and bank account information. The key to phishing is deception. Phishing uses email spoofing as its initial medium for deceptive communication followed by spoofed websites to obtain the needed information from the victims. Phishing was discovered in 1996, and today, it is one of the most severe cybercrimes faced by the Internet users. Researchers are working on the prevention, detection, and education of phishing attacks, but to date, there is no complete and accurate solution for thwarting them. This paper studies, analyzes, and classifies the most significant and novel strategies proposed in the area of phished website detection, and outlines their advantages and drawbacks. Furthermore, a detailed analysis of the latest schemes proposed by researchers in various subcategories is provided. The paper identifies advantages, drawbacks, and research gaps in the area of phishing website detection that can be worked upon in future research and developments. The analysis given in this paper will help academia and industries to identify the best anti-phishing technique. Copyright © 2016 John Wiley & Sons, Ltd.

KEYWORDS
phishing; deception; search engine

***Correspondence**
Gaurav Varshney, Department of Computer Science and Engineering, Indian Institute of Technology Roorkee, Roorkee, Uttarakhand, India.
E-mail: gauravdtsi@gmail.com

1. INTRODUCTION TO WEB PHISHING

1.1. Web phishing attacks

Phishing is a fraudulent act that is used to deceive users over the Internet with the goal of obtaining their personal information [1,2]. The attackers who plan phishing attacks are commonly termed as phishers. Phishing became a serious cyber threat in 1996 when phishers stole the user names and passwords of AOL users [3,4]. In most cases, a successful phishing attack is accomplished using email spoofing [5] and website spoofing techniques [6,7], as shown in Figure 1.

Phishing starts with attackers sending spoofed emails [8] to target victims over the Internet and making their communication look as though it is coming from authentic entities such as banks, credit card companies, and government agencies [9,10]. The email addresses are spoofed as their source addresses are made to look similar to that of an email coming from an authentic source. For example, if a bank manager of bank "xyz" has the email address: bankmanager@xyz.co.in, then the attacker will try to spoof his email with the address bankmanager@xyz.co.in so that the user believes in the authenticity of the email and performs the steps asked by the phisher. The actions requested in the email are typically opening web links and providing their identity or banking information either on the web or as a reply to the email. Open SMTP (Simple Mail Transfer Protocol) servers are utilized for email spoofing, which allows attackers to send spoofed emails to victims. As many users do not reveal their personal information in reply to an email, another deceptive tactic is implemented by creating phished websites that are identical or very similar in their look and feel to the targeted authentic websites [11–13].

Once the user clicks the web link provided in the spoofed email, he is directed to the phished website created by the phisher. As this phished website looks similar to the original website, a novice user often fails

Figure 1. Successful phishing: A combination of phished email and websites.

to identify it as malicious and enters the requested information, resulting in a successful phishing attempt. Apart from emails, attackers can lead users to malicious links by publicizing them as advertisement links on authentic websites. Furthermore, in some cases, an infected DNS can result in user redirection to abnormal and phished websites. Other well-known ways used by phishers for carrying out web phishing attacks are as follows:

(1) Phisher creates informational websites that provide valuable information to the user. These websites also provide links to connect to Facebook, Gmail, and Twitter. These links redirect the user to phished websites instead of the authentic websites. To maintain the user's confidence in deceptive links, phishers generally use link manipulation techniques.
(2) In a covert phishing attack, a compromised real website can be used by an attacker to create a login pop up asking for a user's personal information [7].
(3) A compromised DNS may be used by phishers to redirect the user's request for benign URLs to their phished ones [14].

1.2. A practical phishing scenario

Figure 2 shows a phishing scenario that uses spoofed email communication.

The steps followed by a phisher for a typical phishing attempt are as follows:

(1) Phisher runs a spoofed version of the targeted website over a web server and sends spoofed emails to the target users.
(2) The email generally contains a message stating an emergency which requires immediate action. For example, it may ask the user to login into his bank account and provide some information, failing which his bank account will be closed.
(3) The spurious link in the email directs the user to the web server on which the phisher has hosted the similar looking login page of the targeted website.
(4) The user supplies his credentials on the spoofed website, which is then stored by the phisher.
(5) This user's information is then utilized by the phisher to carry out fraud such as using credit card information for online purchases or for carrying out fraudulent transactions using the user's bank credentials.

1.3. Web phishing tactics

A special form of phishing known as spear phishing targets a specific individual or a specific company [15,16]. When phishing targets high-profile users, it is termed as Whaling [7]. A variant of phishing is Tabnabbing [17]

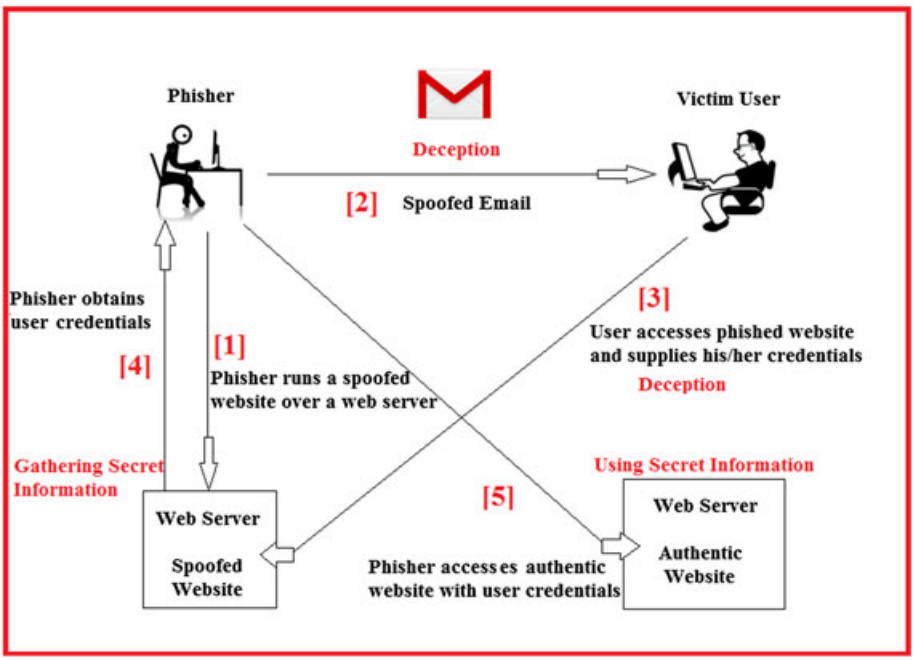

Figure 2. A phishing scenario.

wherein a tab switch performed by the user is used by the malicious websites to load a phishing webpage. When the user switches back the tab, assuming that it as an authentic website left opened by him or her, the user enters his or her credentials to this phishing webpage. Phishing uses deception techniques for its success. Figure 3 shows major deception techniques used by phishers.

Advanced Internet fraud techniques such as email spoofing, website spoofing, and exploitation of browser and web technology vulnerabilities are used to deceive/lure/redirect the targeted users to a phished website [6,18]. Phishers use various tricks to carry out a successful deception. These tricks include the following: (i) link manipulation (the contents of $< A >$ tag content are made to display a web link going to an authentic URL, where as in the background it actually goes to a phished or malicious URL); (ii) evading phishing detection filters [19] (with the use of images instead of text that can remain undetected by many phishing filters [20]); (iii) malicious use of web scripting languages (using Java script to hide browser address bar and create a custom address bar displaying a hard coded authentic URL to the user); (iv) using pop-up windows to ask user names and passwords; and (iv) utilizing browser vulnerabilities (e.g., Tabnabbing) [11].

Figure 3. Web phishing tactics.

1.4. Current phishing statistics

Starting from 1996 to the date of writing this paper, there have been a significant number of phishing attacks recorded and analyzed by anti-phishing organizations such as APWG and Phishtank. According to the APWG Phishing Trends Report, 158,544 unique phishing websites were recorded in the fourth quarter of 2015. In most of the cases, financial and online payment companies were targeted, and the majority of these phishing websites are hosted on a server located in the USA. Table I.I describes the current trends (first Quarter 2014 to fourth Quarter of 2015) [21] of phishing attacks in terms of "Number of Phishing Websites," "Number of Phishing Emails," "Top Country Hosting Phishing Sites," "Most Affected Services," and "Most Targeted Top Level Domain (TLD)."

As demonstrated by Table I.I, there is no significant decrease in the number of phishing incidents, despite various phishing detection and prevention schemes proposed by researchers. Hence, there is an immense need for research and development on security solutions to prevent or detect phishing attacks over the Internet and to safeguard novice users and online transactions [22].

1.5. Anti-phishing solutions

There has been a great deal of research in the area of phishing prevention and detection. The proposed solutions include training users on phishing-related activities; phishing detection and prevention; use of anti-phishing software; browser extensions and toolbars [23]; DNS and WhoIs information of URLs; new measures of user authentication; filtering phishing emails [24] and websites; real time, proactive detection, monitoring and shutting down of phishing websites; two factor authentication schemes; disabling malicious Java scripts; secure browser developments; and so on. Anti-phishing solutions can be majorly categorized as phishing prevention solutions, user training solutions, and phishing detection solutions. Figure 4 shows the classification.

Table I.I. Current phishing attack trends.

Quarter/parameters	Q1 2014	Q2 2014	Q3 2014	Q4 2014	Q1 2015	Q2 2015	Q3 2015	Q4 2015
Number of phishing websites	126215	128378	92473	47094	136,347	253007	241140	158544
Number of phishing emails	171792	171801	163333	197252	221211	417472	395015	380280
Top country hosting phishing sites	USA	USA	USA	USA	USA	USA	USA	USA
Most affected services	Paym. 46.57%	Paym. NA	Paym. 32.06%	Retail 29.37%	ISP 26.24%	ISP 25.34%	ISP 25.34%	Retail 24.03%
Most targeted TLD	. COM 46%	. COM 51%	.COM 55%	.COM 46%	NA	NA	NA	NA

Note: Paym, payment services; NA, not available; QX, quarter number X.

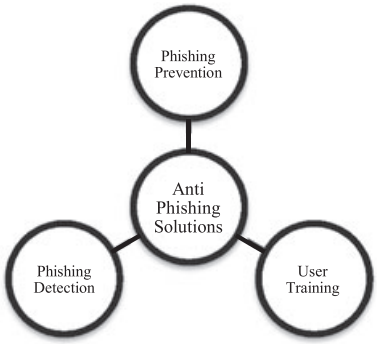

Figure 4. A classification of anti-phishing solutions.

1.5.1. Phishing prevention schemes

Phishing prevention schemes try to prevent phishing attacks by providing an extra layer of security to the authentication schemes and user interaction platforms (via two factor authentication and two-way authentication). This reduces the probability of a user being deceived by an attacker's phishing website. Phishing prevention techniques can be further classified as watermarking based [25], RFID based [26], external authentication devices based [27], picture password based [28], dynamic security skin based [18,29], smart card based [30], and QR Code based techniques [31], and so on. [12]. These techniques can prevent most phishing attacks, but they require changes and support on the website's side and cooperation and understanding on the user's side for their success. Furthermore, these solutions may lead to complex user interfaces, may incur extra cost for the computation of each authentication, and may also require users to keep extra authentication devices, making it cumbersome to implement and use. Figure 5 shows the aforementioned phishing prevention schemes.

Figure 5. Major phishing prevention solutions.

1.5.2. User training schemes

User training schemes try to educate users through emails and other mediums so that users themselves can identify phishing attempts targeted at them. Although the idea is good, it does not provide a fool-proof solution, as a large population of users are novice who may not understand how SSL, certificates, and URLs of the websites can be checked for their authenticity, even after training. Therefore, a complete reliance on such ineffective solutions can be disastrous [32–37].

1.5.3. Phishing detection schemes

Phishing detection schemes [38–45] that detect a phishing website, either through a web browser on the client side or via specific software at the host, or solutions that detect phishing at the server side are better than phishing prevention and user training schemes. This is because they have a minimal reliance on novice Internet users. When a website is detected as a phishing or probable phishing website, access to the website is blocked or the user is notified that the website may not be authentic. This method required minimal user training and does not require any changes to the existing authentication schemes used by a website. The accuracy of the detection schemes is measured in terms of the following parameters:

- Number of True Positives (TP): The number of phishing websites correctly labeled as phishing.
- Number of True Negatives (TN): The number of legitimate websites correctly labeled as legitimate.
- Number of False Positives (FP): The number of legitimate websites incorrectly labeled as phishing.
- Number of False Negatives (FN): The number of phishing websites incorrectly labeled as legitimate.

The accuracy of phishing detection schemes is normally evaluated using a set of benchmark datasets. The popular normal dataset and phishing dataset are given as follows:

I. Normal Dataset
   Alexa Dataset [46]: Alexa dataset is used as a benchmark dataset of benign and normal websites. Alexa is a commercial company that performs the task of web traffic data analytics over the web. It obtains users' browsing patterns from various sources and critically analyzes them for web traffic reporting and ranking of URLs on the Internet. The rankings provided by Alexa are used by researchers to accumulate a set of highly ranked websites as a normal dataset for testing and identifying TNR. Alexa provides the dataset of normal websites in the form of a raw text file in which each line mentions the rank of a website and its domain name in ascending order.
II. Phishing Dataset
   Phishtank Dataset [47]: Phishtank dataset is used as a benchmark dataset of phishing websites. Phishtank is a community-based system that verifies phishing websites. A variety of users and third parties submit

suspected phishing sites that are eventually voted on by a set of users for their availability as a valid phish. Phishtank thus provides a real-time dataset of phishing websites. Phishing websites provided by Phishtank are used by researchers to create a dataset of phishing websites for testing and identifying the TPR. Phishtank dataset is available in CSV file format, and each line of the file contains details of a unique phish reported over Phishtank. The details include phish ID, phish URL, phish detail URL, submission time, verified status, verification time, online status, and target URL.

This paper reviews the work in the area of web phishing detection schemes. The review identifies the pros and cons of the existing schemes that can help academia and industry to identify the best technique to be utilized for phishing detection at their end. This paper is organized as follows: this section has introduction, and the next section provides an overview of the current proposals in the area of phishing detection schemes followed by Section 3, which concludes the paper.

2. PHISHING DETECTION SCHEMES

While there are many proposals that detect phishing websites, this paper describes and analyzes only the most recent phishing detection proposals. The aim is to provide an overall picture of the state of the art in the area of phishing detection, which can help industries, researchers, and academia to review latest schemes with their pros and cons and find the most suitable scheme for phishing detection at their end.

2.1. Classification of web phishing detection schemes

A broad classification of phishing detection schemes based on the underlying technique utilized for phished website identification is shown in Figure 6. The techniques are majorly classified as search engine based (SEB), heuristics and machine learning based (HMLB), phishing blacklist and whitelist based (PBWB), visual similarity based (VSB), DNS based (DNSB), and proactive phishing URL detection-based (PPUDB) schemes. Figure 6 shows the categorization.

(1) Search engine based
Search engine-based techniques extract features such as text, images, and URLs from websites, then search for them using single or multiple search engines and collect the findings. The assumption when detecting a normal website is that it will be among the top search results, as normal websites typically have a higher index than phishing webpages, which remain active for a very short time.

(2) Heuristics and machine learning based
These techniques extract a set of features of either text, image, or URL-specific information from normal or abnormal websites. A set of heuristics is utilized, and the thresholds or rules obtained from the learning algorithms are used for anomaly detection.

(3) Phishing blacklist and whitelist based
The methods in this category utilize the whitelist of normal websites and the blacklist containing anomalous websites to detect phishing. The blacklist is obtained either by user feedback or via reporting by the third parties who perform phishing URL detection using one of the other phishing detection schemes.

(4) Visual similarity based
The technique utilizes the visual similarity between webpages to detect phishing. When phishing web sites are matched in terms of their visual characteristics with the authentic websites, it checks whether the URL is on the authentic domain URL list. If not, the website is marked as a phishing website.

(5) DNS based
DNS is used to validate the IP address of a phishing website. For example, DNS will identify whether the IP address over which the phishing website is running is on the list of authentic website IPs. If it is not, the website is marked as phishing. DNS can also be utilized by these techniques in other ways, based on the needs of the user.

(6) Proactive phishing URL detection based
This scheme detects probable phishing URLs by generating different combinatorial URLs from existing authentic URLs and determining whether they exist and are involved in phishing-related activities on the web.

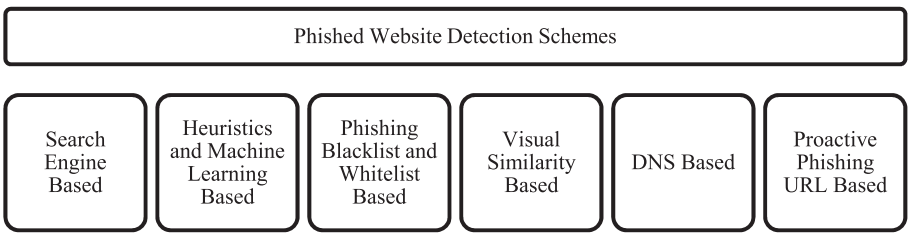

Figure 6. A classification of phished website detection schemes.

The identified pros and cons resulting from the study and analysis of the aforementioned schemes are given in Table II.I.

An analysis of these techniques in terms of various parameters such as availability as a client side implementation, real-time phishing attack detection capability, the need for training and updates, computational complexity, storage complexity, and communication cost is shown in Table II.II.

2.2. Literature review of phishing detection proposals

Latest phishing detection proposals in various categories are as follows:

2.2.1. Search engine-based techniques

All SEB techniques extract and use webpage text, images, or URLs as a search string to determine the popularity of a website using search engines to detect phishing. The techniques are different, however, in terms of (i) type and number of features extracted (text, URL, or images) from a webpage; (ii) number of search engines used to determine webpage popularity, (iii) number of top results used for matching; (iv) the underlying decision making algorithm; and (v) additional use of logic from other anti-phishing schemes. A brief description of these schemes is as follows:

Varshney *et al.* [48] focused on the need of lightweight phishing detection approach using search engines. Authors identified the lightest possible features (page title and domain name) that can be extracted from a webpage without a complete webpage loading. Based on this, authors developed an intelligent anti-phishing chrome extension named lightweight phish detector (LPD). LPD not only detects but also suggests the authentic webpage to the user when a user reaches a deceptive or phishing page on the browser. Ramesh *et al.* [49] proposed a technique that collects and matches a group of domains having direct and indirect association with the domain of the suspicious webpage to detect phishing. Jun *et al.* [38] proposed a scheme wherein the URL of a website is searched using popular search engines such as Google, Bing, and Yahoo. The number of search results obtained and their rankings are then used for classification. Hung *et al.* [50] proposed an approach that captures a screenshot of the webpage and extracts the website logo, which is then searched using Google image search. The returned keywords are then fed to Google text search, and if current domain name does not match any of the top 30 domain names returned in the search results, then the website is identified as phishing. Xiang *et al.* [51] proposed a

Table II.I. Pros and cons of various phished website detection schemes.

Scheme	Pros	Cons
SEB	Lightweight and can be implemented within browsers or as add-ons; highly accurate and, works in real time without any updates; low complexity which is equivalent to single or multiple web search and string matching operations per URL; low-storage complexity, nothing gets stored at client side.	Can fail in rare cases if search engines are subverted to rank phishing webpages at the top; accuracy and efficiency depend upon the features and the underlying algorithm used for detection.
HMLB	The use of new classification features and machine learning algorithms can improve accuracy, making the scheme adaptable.	Needs computational resources and time for training; cannot be solely implemented at client side because of resource limitations over browsers; frequent need for training of new features if phishers start bypassing them.
PBWB	Lightweight and can be implemented within the browser; low Complexity which is equivalent to matching the requested URL with the existing blacklist or whitelist of URLs from the centralized database.	Frequent updates of blacklist or whitelist are needed for detecting new phishing websites; extra cost of remote querying and matching to monitor each URL.
VSB	Can detect phishing attacks such as Tabnabbing; can be implemented at client side but needs support from the operating system and softwares other than the client browser. Ex: OCR.	Computation and storage intensive as matching of images is required. Can increase page loading time if implemented as a browser add-on; does not greatly improve the accuracy of existing techniques.
DNSB	Lightweight; requires less feature extraction from websites in the form of texts or images; low-storage complexity at client and server; client side implementation possible.	Can fail in case of DNS poisoning; High communication cost and extra load on DNS. Every single URL monitored needs a query to the DNS server; client side implementation causes delay in page loading and is dependent on DNS response time.
PPUDB	Helps in detecting phishing websites before they reach users for carrying out fraud activities.	Computationally intensive, needs a random combination of URL's to be generated and checked on the web on a regular basis to detect possible phishing attacks on specific targeted domains; cannot be implemented on client side and needs central or distributed server implementation, adding cost.

SEB, search engine based; HMLB, heuristics and machine learning based; PBWD, phishing blacklist and whitelist based; VSB, visual similarity based; DNSB, DNS based; PPUDB, proactive phishing URL detection based.

Table II.II. Phishing attack detection capabilities.

Type of solution	Client side implementation	Real-time detection	Require training	Require updates	Computational/ communication cost	Storage complexity
SEB	Yes	Yes	No	No	Low/High	Low
HMLB	No	Yes	Yes. Need frequent training over new features with time	Yes.	High/Medium	High. Need to store and update training dataset.
PBWB	Yes	No	No	Yes. Need frequent updates of blacklist and whitelist.	Low/High	Medium. Blacklist is stored at central server.
VSB	Yes	Yes	Yes		High/Medium	High
DNSB	Yes	Yes	No	No	Low/High	Low
PPUDB	No	Yes	No	No	High/High	High. Need to store probable phishing URL's.

SEB, search engine based; HMLB, heuristics and machine learning based; PBWD, phishing blacklist and whitelist based; VSB, visual similarity based; DNSB, DNS based; PPUDB, proactive phishing URL detection based.

technique that uses "site: declared brand domain 'page domain' " as a Google search engine query and checks whether the returned results indicate the same domain name or not. If the returned results do not indicate the same domain name, keywords from the webpage visited by the user are extracted and searched. If the domain name does not appear in the top N search results, the URL is declared as phishing. They also proposed that before using the Google search engine query, the URL should be searched on the whitelist and the page should be passed through a login form filter. If the URL is on the whitelist or if the page does not contain any login form, it is declared as normal, and further processing is not carried out. Dunlop *et al.* [52] proposed a technique where an IE toolbar takes a snapshot of the current page and the image contents, including logos. The image contents and logos are converted to text, which is searched using the Google text search. The top level and second-level domains are matched with the top four links obtained from the Google search to detect phishing. An analysis of these schemes in terms of novelty, dataset, accuracy, and drawbacks is given in Table II.III.

Research gaps and future needs: Search engine-based techniques are new in the area of phishing detection and are gaining popularity because of their lightweight implementations and client side deployments. However, there are a few points that still require further research, such as the following:

(1) There is a need for research and development to identify long-term benign domains that suddenly start phishing activities. As these domains have been running for a long time, they appear in top search results even when they are carrying out phishing.
(2) There is a need to reduce the number of false positives for benign domains running on the web with a very short lifetime, which results in them not appearing in top search results.

2.2.2. Heuristics and machine learning-based techniques

All techniques in this category extract a set of features such as webpage content, URL, and/or network features and a set of machine learning or classification techniques, which are used to create a model for classification. These techniques differ in terms of (i) the type and number of features extracted; (ii) the algorithms used to identify the best feature sets and assign weights; (iii) the type and number of machine learning or classification algorithms applied to these features; (iv) the use of optimization algorithms, and (v) the use of logic from other anti-phishing techniques. A brief description of these schemes is as follows:

Moghimi *et al.* [53] proposed the use of Levenshtein Distance for string matching to find the relationship between the content and the URL of a webpage and used SVM for classification. Singh *et al.* [54] proposed training via Adaline network as more efficient and accurate for neural network for phishing detection. Mohammad *et al.* [55] proposed the idea of self-structuring neural network. Mohammad *et al.* [56] also proposed the idea of neural network training with back propagation training for phishing website classification. Tuan *et al.* [57] proposed six heuristics: primary domain, sub domain, path, domain, page rank, Alexa rank, and Alexa reputation. Each heuristic is assigned a weight during experiments, and based on the accuracy of the results, the optimal weight allocation and threshold for phishing detection are identified. Abdelhamid *et al.* [58, 59] proposed a multi-label classifier-based associative classification approach for phishing detection. In the approach of Mohammad *et al.* [60], important phishing detection features were assessed using rule-based data mining techniques for phishing detection. The authors demonstrated that the C4.5 classification algorithm performs better than the RIPPER, PRISM, and CBA algorithms in terms of accuracy. Kausar *et al.* [61] proposed an approach that uses a combination of heuristic and Naive Bayes classifiers. The authors used the Tuan's

Table II.III. Analysis of search engine-based techniques.

Authors	Novelty	Dataset/accuracy	Drawbacks
Varshney *et al.* [48]	Proposed optimization for search engine based detection approaches such as [37,47,49]. Used page title and URL alone to create an effective search string to uniquely identify phishing websites. Identified optimized search results for comparison and developed a working prototype over Google chrome (LPD).	Dataset: Phishtank 500 URLs, Alexa 0-500500 URLs in groups of 1000 URLs; TPR = 99.5 % TNR = 100% (Alexa to 500 URLs) and 92.4% (Alexa 0-500500 URLs)	LPD may give false positives over recently launched benign websites being lightweight and dependent on only two features. The authors proposed inclusion of other features in future work to decrease FPR yet maintaining the resource effectiveness, which is the key idea of LPD proposal.
Ramesh *et al.* [49]	Unlike other schemes [38] which only utilize SE results, it also uses hyperlinks from HTML sources and DNS lookups to identify the target domain set for matching..	Dataset: Phishtank database; TPR = 99.67%, TNR = 99.5%; Acc. = 99.62%.	DNS requirement makes it system dependent and complex; Finding direct and indirect association for every webpage visited by a user is time consuming and tedious.
Jun *et al.* [38]	Does not need webpage content analysis to extract keywords for creating search engine query for phishing detection as used in the approaches of Ramesh [49] and Xiang *et al.*[51].	TPR = 98%, FNR, FPR = 2% with KNN.	Uses a complex learning and classification approach (Such as KNN, SVM, and NB to make a decision from inputs of all search engines; three web searches to check a URL increase the detection and response time and cost.
Hung *et al.* [50]	Obtains keywords for the search string by searching the logo of the webpage using Google image search. This reduces the need for webpage content analysis as needed in approaches such as [49].	Dataset: 400 phishing websites from Phishtank and 50 websites from Alexa; TPR = 92.5% and TNR = 100%.	The assumption of the logo being on the top left part of the webpage and being included in the either 1×2 or 2×2 or 3×3 size segment of the complete webpage might not be true in all cases; The use of two search engine queries per URL increases communication and response time.
Dunlop *et al.* [52]	Concept of converting logos and images obtained via snapshot to text using OCR, thus reducing the two queries approach (as used by Hung *et al.* [50]) to one query.	Dataset: Phishtank database, 100 phishing sites; TPR = 98%.	Delays the rendering of a webpage due to the overhead of image conversion to text; Platform dependent as it needs OCR processing on the client machine.
Xiang *et al.* [51]	Use of Google site operator engine query, that is, "site: declared brand domain page domain." Unlike [38,49], it uses login form filter and whitelist before search engine query, thus reducing required resources and time for detection.	Dataset: 7906 phishing pages and 3543 normal webpages; TPR = 90.06%, FPR = 1.95%.	The use of two search engine queries increases communication cost and response time; maintaining the whitelist is tedious.

approach [62] and added the first phase of Gu's [63] approach to increase accuracy. The approach of Barraclough *et al.* [64] uses five inputs in their neuro fuzzy approach, including legitimate site rules, user behavioral profile of interaction with phishing and legitimate websites, URL information from Phishtank, and so on. then, if then rules are generated via the neuro fuzzy approach to detect phishing. In the approach of Li *et al.* [65], the authors used both application layer contents such as the information available in the URL, HTTP header, host and web content, and the network layer features such as remote server attributes, crawler-server communication attributes, and DNS information, and aggregated these features to train a variety of classification models to detect phishing websites. Birhanu [66] proposed combining static analysis approaches that obtained features from URL and HTML content, and dynamic approaches that test the dynamics of a webpage during execution, such as proxy level, sandboxing, and honey clients with learning algorithms to detect phishing. Shahriar *et al.* [67] proposed the creation of a finite state model to check the responses of websites to random inputs. Then, a set of heuristics was utilized to match this behavior with the behavior of phishing websites. In the approach of Weiwei *et al.* [68], the authors used 10 features including title, keywords, and link text information to build various heterogeneous classifiers and used an ensemble classification to combine the results of trained classifiers with a hierarchical clustering algorithm for categorizing phishing websites. The approach of Xiang *et al.* [42] uses a set of webpage and

URL features that utilize the HTML DOM, search engines, and third party services and finally machine learning techniques for classification of normal and phishing sites. To increase the amount of true positives and decrease the amount of false positives, the authors used two filters; one identifies near duplicate phish with the help of hashing and the second is the login form filter that classifies web sites with no login page as normal websites. He *et al.* [69] identified 12 important features and used the TF-IDF approach to detect phishing. SVM classifier was trained over normal and phishing webpages to detect phishing. The scheme of Aburrous *et al.* [70] uses five data mining algorithms, namely, C4.5, Ripper, Part, Prism, and CBA to identify the best feature sets for creating fuzzy rules. The authors divided the features into six categories: URL and domain identity, security and encryption, source code and Java script, page style and contents, web address bar and social human factor. Weights are assigned to each feature set, and fuzzy rules are created for phishing website detection. Justin *et al.* [71] proposed a scheme wherein the lexical features of the URL and the host-based features were extracted for phishing classification using a logistic regression classifier. An analysis of these schemes in terms of novelty, dataset, accuracy, and drawbacks is given in Table II.IV.

Research gaps and future needs: As machine learning has a high-computational cost and requires training datasets, it is impractical as a client side deployment solution as a browser add-on or a lightweight IDS. Hence, there is a need to

(1) Find alternative and lightweight machine learning techniques that require less training, thus demanding less resources while providing comparable accuracy. While it is true that in the era of cloud computing, increased need for resources is not a constraint, hiring special resources for running phishing detection solutions adds cost. However, if the same solution can be made to run on the client side, such extra resources can be saved.
(2) Develop scalable and dynamic solutions that are adaptable to the changes in the environment. For example, designing a solution that takes into account that phishers frequently bypass the text and visual features considered by phishing detection schemes and dynamically identify and add new features based on self-learning.

2.2.3. Phishing blacklist and whitelist-based techniques

All techniques in this category are similar in that they all utilize a blacklist containing URLs of phishing websites or a whitelist containing URLs of normal websites at the client or at a remote server. These URLs are compared with the URL being visited by the user to detect phishing. The techniques are different in terms of (i) how the blacklist or whitelist is created, stored, and accessed; (ii) whether whitelist, blacklist, or both are employed for detection, and so on. A brief description of the schemes is as follows:

In the anti-phishing approach used by the Google Chrome browser [43], each URL opened by the user is checked against the Google's blacklist of phishing websites using Google Safe Browsing API. For each such URL, an HTTP API request is sent, and the response to this request is used to detect phishing. The Firefox web browser [75] also uses the Google Safe Browsing API for phishing detection. It alerts the user if API results indicate phishing. In the approach of Li *et al.* [76], an analysis of blacklist and whitelist-based anti-phishing tools incorporated into a browser was presented. The scheme of Krishnamurthy *et al.* [77] classifies Internet domains/target domains as legitimate or phished and creates a whitelist of URLs and a blacklist of URLs. URLs are first compared with the URLs in the whitelist and if there is no match, then they are compared with the URLs in the blacklist. Finally, the URLs in the filtered set are those which are closest to phishing URLs. An analysis of these schemes in terms of novelty, dataset, accuracy, and drawbacks is given in Table II.V.

Research gaps and future needs: Phishing blacklist and whitelist-based techniques suffer from a common drawback in that they both require time for a list update to detect new phishing URLs. There is a definite amount of time after which the list updates either on the client side or at the central server end. Until the list is updated, new URLs will remain undetected by this solution. Hence, there is a need for research and development in identifying

(1) Efficient, speedy, and more proactive mechanisms for list updates so that the new phishing URLs can be detected with almost no delay.
(2) Whether the solution is running and storing URLs at the client's end, in which case it will require a substantial amount of storage to store all blacklist URLs. Moreover, if it stores whitelist URLs, then it will require even more storage. Solutions are required to efficiently manage the URLs stored in cases of client side browser-based solutions, such as browser add-ons. Schemes to reduce communication cost are required if the list is stored on remote central servers. Methods to reduce communication cost during each check by either using caches or some other temporary storage should be studied.

2.2.4. Visual similarity-based techniques

All techniques in this category use visual similarity between authentic and phishing webpages and their visual features to detect phishing. They are, however, different in terms of (i) the visual features extracted to identify similarity; (ii) the visual matching algorithm used; and (iii) the use of the logic of other anti-phishing techniques. Most of the VSB techniques are capable of detecting a specific type of phishing attack called Tabnabbing [78].

Chiew *et al.* [79] proposed a logo extraction and image search-based phishing detection scheme. The scheme of Teh-Chung *et al.* [80] calculates normalized compression

Table II.IV. Analysis of heuristics and machine learning-based techniques.

Authors	Novelty	Dataset/accuracy	Drawbacks
Moghimi and Varjani [53]	Similar to [54], authors used SVM as classification over 17 features. Proposed eight new features that identify the relationship between content and URL of a webpage using approximate string matching algorithm (Levenshtein Distance).	Dataset: 3066 pages from Phishtank database and 686 legitimate websites from yahoo directory; Acc. = 99.14%	If a webpage is carefully designed by Phisher the extracted features might not give enough information to detect Phishing. Similar case with the use of images and flash media instead of text by the Phishers.
Singh *et al.* [54]	Authors tested Adaline and back propagation training of neural network on top of SVM for classification of phishing websites over 15 features. Claimed Adaline as more effective and accurate for classification.	Dataset: 179 phished websites from Phishtank database, 179 legitimate websites from Alexa; Acc. = 99.14%	Multistage classification via SVM and neural network need extra resources.
Mohammad *et al.* [55]	Authors proposed self-structuring neural network for classification on top of the 17 features identified in [60] for phishing detection. Authors claim the automation of structuring the neural network has also acceptable generalization ability.	Dataset: 600 legitimate websites from yahoo directory, 800 phishing sites from Phishtank and Millersmiles archive; Acc = 92.18 over 1000 Epochs of NN.	Use of 17 features and incorporation of self-structuring neural network makes it only a server side implementation choice instead as a client solution. Accuracy achieved is a bit lower than other existing schemes.
Tuan *et al.* [57]	The novelty is in the identification of six minimal features claimed to provide a high accuracy.	Dataset: Phishtank database, 11660 phishing sites; Acc. = 97.16%.	Complex as it employs various heuristics, making its deployment difficult as a real-time client side tool; heavily dependent on third parties for its operations.
Mohammad *et al.* [60]	The accuracy of various data mining techniques has been studied for phishing detection, and CBA has been identified as the best performing one.	Lowest error rate of 4.5% was obtained using CBA.	A practical implementation as an anti-phishing tool and/or its effectiveness and cost benefit analysis is missing.
Kausar *et al.* [61]	Experimental validation and identification of the best combination of phases from the two approaches [62,63] taken to improve detection accuracy.	Dataset: 89 phishing websites and 71 legitimate websites; Acc. = 87.5%.	Adds little in terms of proposing a new scheme for phishing detection. Work is more towards study of combining the phases from Tuan [62] and Gu [63] approach.
Barraclough *et al.* [64]	Unlike others [57,61,65], the novelty is the addition of neuro fuzzy approach and the inclusion of user behavioral profile of website interaction as input for phishing detection.	Acc. = 98.5%; claims that it provides better results than Netcraft [72], CANTINA [73], and SpoofGuard [74].	Complex and highly dependent on inputs; creating user profile of interaction with a set of websites is tedious, time consuming and resource intensive.
Li *et al.* [65]	Similar to [57] the novelty of the scheme is in the identification of the best application and network features.	Crawled a set of normal and malicious websites; applied the obtained features to different classifiers (NB, LR, SVM, and J48) for learning and detection; Acc. = 99.8%.	Computationally complex; average detection time of 4.9 s that makes client side implementation slow and non-real time; needs noticeable amount of time and resources to crawl websites for detection.
Birhanu [66]	Concept of combining static and dynamic analysis approaches with machine learning. Unlike others [64,65], used evolutionary search and optimization algorithm for better accuracy.	Exists as a proposal.	Not enough details of testing and validation are available to comment on the effectiveness of the proposed work and its benefits.

(*Continues*)

Table II.IV. (Continued)

Authors	Novelty	Dataset/accuracy	Drawbacks
Shahriar *et al.* [67]	Unlike most of the schemes in this category, the proposal is of a finite state model for phishing detection.	Dataset: 33 phishing and 19 legitimate websites; FPR = 0% and FNR = 0%.	Constructing behavioral model is tedious, computation intensive, increases the response time, and needs frequent updates.
Weiwei *et al.* [68]	Similar to [57,65], the novelty is the identification of best features and the incorporation of the heterogeneous and ensemble classification approach.	Compared their solution with Kaspersky, Netcraft [72], and site advisor; obtained higher precision and recall = ~98%.	Complex in operation and training as it uses multiple classifiers; cannot detect new attacks as phishers change the features they exploit to do phishing daily.
Xiang *et al.* [42]	Additional use of hashing to identify near duplicate phishes, unlike [68,70] inclusion of login page filter is also a novel achievement.	Dataset: Experimental corpora of 8118 phishing and 4883 legitimate websites; Acc. = 92%.	Complex due to the addition of machine learning, search engines and noticeable feature extraction;
He *et al.* [69]	The use of TF-IDF approach for identification of best features, as in [57,68]	Dataset:100 login pages of targeted legitimate sites from millers miles report, 375 phishing pages from Phishtank and Castle Cops; TPR = 97.33%, FPR = 1.45%.	Resembles CANTINA which uses a comprehensive set of features for phishing detection; TF-IDF approach limits phishing detection over English language webpages.
Aburrouset al. [70]	Use of comprehensive set of features and application of fuzzy rules, similar to [64]	Dataset information not given.	Complex to implement and practical use needs multiple classifiers.
Justin *et al.* [71]	Use of LR classification approach to improve the accuracy, as in [66].	Dataset: Normal dataset from DMOX open directory project and yahoo directory. Phishing dataset from Phishtank and Spam scatter; Acc. = 95–99%.	Extraction of lexical and host-based features such as querying WHO IS, and domain name properties for each URL is computationally expensive.

Table II.V. Analysis of phishing blacklist and whitelist based techniques.

Authors	Novelty	Dataset/accuracy	Drawbacks
Chrome anti-phishing [43]	Phishing detection using blacklist inside the browser's operational mechanism during a user's visit to a URL.	Real-time implementation is available in the form of the Google Chrome web browser.	Lightweight solution, but cannot detect real-time phishing attacks if the blacklist is not updated with new phishing URLs.
Firefox anti-phishing [75]	Similar to [43], it also uses Google safe browsing API, but the results demonstrate that it gives faster response time.	Firefox web browser implementation with anti-phishing is available publicly.	Relies on Google Safe Browsing API for phishing detection and cannot detect real-time phishing attacks if the blacklist is not up to date.
Li *et al.* [76]	The work is novel as it studies the use of blacklist and whitelist in terms of accuracy and applicability and gives a set of suggestions to implementers about their use for phishing detection, for example, both blacklist and whitelist can be used for higher accuracy.	Authors identified that there is no difference in toolbar detection accuracy whether a black list or white list is used for detection.	Suggested that security pop ups must contain only limited information for inexperienced users.
Krishnamurthy *et al.* [77]	Instead of using existing blacklists and whitelists as used by [43,75], the authors created their own blacklist and whitelist using a set of regular expressions to identify unacceptable and acceptable internet domain names of a target organization.	Currently available as a proposal, no test results are available.	Multilist detection increases both computational and storage complexity

distance between a user's webpage image and the cached image of a legitimate site. This is then passed to a classifier that finally alerts the user. Sarika and Paul [81] proposed a framework that contains three levels of agents. Level 1 contains the URL agent and Tabnab agent. The URL agent checks URL obfuscations, and the Tabnab agent checks layout changes. The level 1 agent conveys the message to the level 2 agent, which makes a decision and conveys it to the level 3 agent. The level 3 agent warns and alerts the user about phishing. Singh and Tripathy [40] proposed a scheme that compares the URL of the webpage with a whitelist of URLs. If a match is not found, the SHA-1 digest of the page is calculated. When the page is refocused, this digest is compared with the new digest calculated. If the digests do not match, then the page is declared as legitimate only if no login forms are present. Although this scheme calculates the SHA hash of the source of the webpage, we considered this scheme under this category because it solves the problem of Tabnabbing, which has been solved mostly by VSB schemes. Jian *et al.* [82] compared the CSS layout of the suspicious webpage with the CSS layout of the victim page. If the similarity exceeds a threshold and the URLs of the two pages are different, then the current page is marked as phishing. Phillipe *et al.* [83] proposed a scheme, TabShots, that takes screenshot of tabs at regular intervals with chrome API. Whenever a tab is switched back, it compares the favicon and visual similarity of the current screenshot (by comparing 10 * 10 tiles) with the screenshot saved earlier. Changed tiles are marked with a separate color to alert the user. Unlu and Bicakci's [17] scheme tracks title, favicon, and layout changes for each tab and compares the old and the new state for any radical changes. Comparison is performed when the tab is focused again or the page refreshes itself or redirects to a new URL. Lam *et al.* [84] proposed a scheme based on layout similarity analysis instead of HTML or webpage content analysis. The phishing webpages are divided into blocks and are matched to the original pages by image comparison. Image processing techniques were utilized for matching blocks of the images. An analysis of these schemes in terms of novelty, dataset, accuracy, and drawbacks is given in Table II.VI.

Research gaps and future needs: Because of their need for extra resources such as OCR, pixel matching API's and others, VSB solutions are complex and dependent on specific platforms. For example, Google pixel matching API's might not be present on other browsers such as Firefox. OCR reader, available on Windows, may not be available on operating systems such as Linux. The required features and CSS layouts might not be present on all websites. Websites may use normal HTML with or without special visual appearances and features. In order to make this scheme a viable anti-phishing solution that can be widely used by a variety of users and organizations, we need that the following:

(1) A set of globally accepted visual features that can be expected on a majority of websites would need to be identified.
(2) A set of visual matching techniques that are platform independent would need to be developed.
(3) As visual similarity-based techniques are more complex than text matching techniques, more efficient visual matching techniques would need to be developed in order to make the solution a viable and competitive proposal.

2.2.5. DNS-based techniques

All techniques use DNS information to identify the authenticity of domain names and IP addresses associated with it to detect phishing. The techniques differ in the way they use DNS to obtain the required information to identify phishing. This can vary from obtaining IP addresses of domain names to obtaining domain query logs to identify frequently visited hosts.

Chen *et al.* [85] proposed a scheme where a page signature extractor module on the client side obtains the signature of the current webpage. This is sent as a DNS query to a remote server for comparison with signatures of phishing webpages. After receiving the response, policy enforcer module on the client side, takes responsive actions. Prevost *et al.* [86] proposed a scheme in which the domain name of the URL visited by the user is sent to two DNS servers: the default and third party DNS servers. If the default IP address is in the IP addresses returned by the third party DNS server, then the current website is considered legitimate. If not, then a webpage content similarity analysis of the visited webpage and reference webpage (obtained from the IP addresses returned by the third party DNS server) is performed to detect phishing. In the approach of Bo *et al.* [39], recursive DNS query logs are used to find all living hosts visited by a user to find suspicious phishing hosts. Known phishing URLs are also used to identify frequent phishing paths and eventually an active phishing webpage. Bin *et al.* [87] proposed a scheme in which an information server stores information such as the bank name, range of bank card numbers issued, the DNS server IP address of the bank, and a list of other DNS server IP addresses allowed to take the card numbers as input for login. Packets are sniffed to check if a card number is being entered on a website that is not in the DNS server IP address range to alert the user of phishing. An analysis of these schemes in terms of novelty, dataset, accuracy, and drawbacks is given in Table II.VII.

Research gaps and future needs: While the use of DNS information in identifying a phishing website is a viable idea, it adds a burden on DNS servers as they have to handle queries from the client or central server. Research and development are needed to reduce the amount of communication cost. The use of caching and other smart storage techniques may reduce the network communication cost, delay, and burden on DNS and the network.

2.2.6. Proactive phishing URL detection-based techniques

All techniques in this category use a mechanism to generate or identify a set of probable phishing URLs.

Table II.VI. Analysis of visual similarity-based techniques.

Authors	Novelty	Dataset/accuracy	Drawbacks
Chiew *et al.* [79]	Instead of matching the complete webpage or a set of images from the webpage, only the logo is extracted via machine learning techniques. Logo is searched via Google image search and returned domain is matched to identify phishing site.	Dataset: 500 Phishing pages from Phishtank and 500 legitimate sites from Alexa. TPR = 99.8%, TNR = 87.0%.	TNR is very low for this scheme. Currently, it needs to download images and do logo identification that can be carried out more efficiently via taking the complete screen shot of the webpage.
Teh-Chung *et al.* [80]	The use of NCD for image matching gives high accuracy.	TPR = 99.99%, FPR <0.01%, obtained over a test data set.	Visual similarity calculation is complex and cannot be implemented to work in real time over browsers; secures only known sites from phishing but phishing on sites which are not in the list will never be detected.
Sarika and Paul [81]	Unlike others [80,82], this scheme used a multilevel anti-phishing approach to solve the problem of phishing and Tabnabbing.	Testing data and validation were not available.	Authors do not mention how often the content change is monitored; many websites such as news websites continuously update their content; overhead increases drastically if comparison is required at frequent intervals.
Singh and Tripathy [40]	Uses whitelist and webpage digest matching to detect Tabnabbing, similar to [83].	Exists in the form of a client side browser extension.	For legitimate websites such as news websites, content is updated very frequently (hash will change), and login forms are also present. In such cases, FPR will be high.
Jian *et al.* [82]	Uses CSS layout features for finding the similarity between the suspicious and authentic webpages.	Dataset: Phishtank, 300 phishing sites; Acc. = 100%, FPR = 0%.	The method used by authors to identify the victim page of a current webpage loaded in the browser is not specified; Unclear how it will be identified that the visual similarity of the current user page needs to be matched with a specific target page.
Philippe *et al.* [83]	Visual comparison of screenshots in a tiled fashion. Alerts user by highlighting the tiles which are changed after switching.	TabShots exists as a Google chrome extension; testing or validation data was not available.	Heavy in terms of storage cost as it has to store the screenshots of all tabs in all the browser windows of the Google Chrome browser; matching operations can be costly as they need to be performed during every tab switch for all tabs.
Unlu and Bicakci [2010] [17]	Webpage title is also matched in addition to finding favicon similarity proposed in [83] and layout similarity proposed in [82].	Implementation exists as a Firefox add-on, but no results are given in the paper.	Resistant to minimal layout changes; re-sizing of browser may change the layout drastically, so there will be high-false positive results; scheme fails if transparent elements are placed on the page that allow user interaction to pass through them.
Lam *et al.* [84]	Layout similarity technique similar to [82] and [17],but instead of matching CSS layout features, it performs visual matching of the complete webpage .	Dataset: Phishtank and APWG; 6750 phishing, 149 mimicked targets; Acc. = 99.6%, FPR = .028%, FNR = .003%	Image processing techniques are computationally complex; its efficiency as a browser add-on or client side anti-phishing tool with resource constraints needs to be identified.

These URLs are mined or reported over web to proactively detect them before they are detected or reported by users or other phishing detection systems. The schemes are different in terms of (i) the specific approach or technique used to generate probable phishing URLs and (ii) the technique used to mine the web or search for probable URLs on the web.

He *et al.* [88] proposed an approach to proactively identify newly registered malicious domain names. It is based on the observation that legitimate domain names

Table II.VII. Analysis of DNS-based techniques.

Author	Novelty	Dataset/accuracy	Drawback
Chen *et al.* [85]	Unlike others [86,87], DNS protocol is used to send a query containing the webpage signature to match against the phishing webpage signature at a remote server.	Testing and validation results were not made available.	Communication intensive solution as it requires communication with remote server and DNS.
Prevost *et al.* [86]	The concept of verification of a website domain name from multiple DNS servers and incorporation of web content analysis as the second phase of the detection process to improve accuracy.	Testing results of the use of DNS servers are not available; Avg. similarity using web content analysis: 31.9% (word similarity approach) and 80.5% (character similarity)	Adds complexity via the use of two DNS servers; the utilized webpage content analysis scheme gives a low average similarity rate.
Bo *et al.* [39]	Introduced the concept of analyzing DNS query logs that do not require active use of DNS servers, as seen in [86,87].	Dataset: real-time experiment on everyday URLs, Acc. = 100%.	Computationally complex as it is based on a recursive query of a user's DNS query logs, which requires heavy resources on the client side, resulting in performance degradation.
Sun *et al.* [87]	Introduced the novel concept of sniffing packets and identifying important information being leaked to websites not in the DNS server IP address range of authentic websites.	Testing and validation results are not available.	The author's claim that their scheme avoids sending user credentials to phishing sites needs further justification, as the scheme sniffs packets to detect phishing when the information is already sent from the client machine and traveling on the wire; for pages using HTTPs, sniffing and matching plain text from web traffic is not feasible.

are made up of meaningful English words. The second-order Markov model identifies useful features, and random forest classification is applied for detection. Ferolin *et al.* [89] proposed a scheme in which phishing emails are classified, and weblinks inside the email are reverse looked up using WhoIs query to find the location, IP address, and host information of the server hosting the phishing weblinks. Then, a proactive warning notification is sent to the administrator of the server hosting the weblinks to take proper action. Marchal *et al.* [41] proposed a scheme which takes URL addresses as input and breaks them down into top-level domain (TLD) and second-level domain (main domain). The main domain is then divided into meaningful words and Markov chain, and other models are used to compute malicious domain names from these words using probabilistic transition of words. The semantic extensions create a list of possible words and newer URLs that might be performing potential phishing activities. These URLs are checked online to proactively identify phishing. An analysis of these schemes in terms of novelty, dataset, accuracy, and drawbacks is given in Table II.VIII.

Table II.VIII. Analysis of proactive phishing URL detection-based techniques.

Author	Novelty	Dataset/accuracy	Drawback
Ferolin *et al.* [89]	Unlike [41,88], proactive detection and removal of phished links during the detection of phishing emails.	Removal success rate of phishing websites = 81.81%. Phishtank archive: 1000 samples	A phishing email might also contain a set of legitimate weblinks, causing increased FPR.
Marchal *et al.* [41]	Unlike regular monitoring of URLs, as seen in [88], this method uses the Markov chain model to generate probable phishing URLs.	Testing and validation results were not made available.	Checking URLs generated by this scheme for phishing on a daily basis is computationally intensive because of the complexity of the operations involved.
He *et al.* [88]	Regularly monitors newly registered domain names and detects malicious domain names as soon as they are registered.	Dataset: 319526 domain names registry information; TPR = 100%.	Can be bypassed if attacker makes use of meaningful URLs for phishing; Computationally intensive as it requires feature extraction and classification over all newly registered domain names on a regular basis.

Table II.IX. Analysis of mobile phone-specific web phishing detection schemes.

Author	Novelty	Dataset/accuracy	Drawbacks
Wu *et al.* [90,91]	1. MobiFish is an anti-phishing tool for Android. It consists of two modules – WebPhish and AppPhish. 2. WebFish matches the domain name of the URL with the whitelist. If a match is not found, then the webpage is searched for login form, and if the form is found, then OCR is used to extract text from the snapshot of the page, and if it contains the second-level domain of the URL, then the page is legitimate. In a similar way, AppFish is developed for phishing detection via mobile apps.	Dataset: 100 Phishing URLs from Phishtank.com. TPR = 100%. TNR = NA.	1. Modified version of Android for capturing screenshots and maintaining a suspicious app list is needed. SAS requires maintaining of official servers of the apps and the list needs to be updated. 2. Requires white listing of legitimate domains.
Hou and Yang [92]	1. Contrast to [90,91] approach, Hou and Yang proposed a warning mechanism to warn the user when the sensitive information is entered in apps and on the websites not in the whitelist. 2. The warning mechanism has a key logger that intercepts the keys typed on the system keyboard. A white list of apps and websites is maintained along with their corresponding usernames. If the user enters the username on a website or in an app not in the white list the system alerts the user.	It was implemented on iOS. No false negatives as it will always warn for non-white list apps and websites.	1. Limitations are that it uses a modified version of iOS and the users have to maintain a white list of all the apps and websites to map their usernames. 2. It is only a warning mechanism instead of classification.

Research gaps and future needs: The solutions in this category are generally avoided because they require heavy amount of resources for web mining to identify phishing URLs. Also, they use computationally expensive algorithms for creating probable phish URLs. Therefore, there is a need to address these issues in future research.

2.3. Web phishing detection schemes for mobile phones

Phishing detection on mobile phones has the following additional issues in comparison with phishing detection on desktop machines:

(1) Mobile phones have limited resources. Therefore, a complex anti-phishing solution cannot run on the mobile phones.
(2) It is difficult to detect phishing websites from their visual appearance or via security indicators on mobile phones due to their small screen size.
(3) Browsers used over mobile phones are lightweight and have reduced security capabilities that can exist or can be developed via third party security companies. For example, an anti-phishing module cannot be developed for chrome browser running on Android as it does not support extensions over mobile phones.

Because of these and many such difficulties, specific anti-phishing techniques are needed in this area. Solutions proposed by some of the researchers to detect web phishing over mobile phones are discussed in Table II.IX.

Research gaps and future needs: There are a very few web phishing detection schemes for mobile phones in comparison with the number of phishing prevention schemes. The reason might be the fact that the detection schemes require more resources in comparison with the phishing prevention schemes (security indicators, secure authentication schemes, and so on). Researchers are working on lightweight phishing detection so that real-time detection of phishing URLs can be made possible with the limited resources available over mobile platform.

3. CONCLUSIONS

In this paper, an analysis of the techniques proposed for phishing detection has been performed. The paper focuses on the fact that phishing detection schemes perform better than phishing prevention and user training solutions because they do not require changes in authentication platforms and do not rely on the user's ability to detect phishing. Furthermore, phishing detection solutions are cheaper than the phishing prevention solutions in terms of the extra hardware required and password management. The paper categorizes phishing detection solutions into six

categories and outlines the advantages and drawbacks of using each one of them. This is accompanied by a description of the popular proposals in each category with their individual pros and cons. We have identified that search engine-based techniques are the lightest possible solutions for phishing detection as they only require a single search engine query result with its underlying algorithm to detect phishing websites at the user's end. It can be deployed both at the client side or on the server side (network periphery servers). SEB techniques require neither machine learning nor training. They are platform independent and can be deployed over any browser and over any operating system as a browser add-on. However, there remain many challenges in the area of search engine-based phishing detection, such as (i) improving phishing detection accuracy when a long-term benign domain decides to begin carrying out malicious phishing activity; and (ii) reducing the number of false positives for benign domains that are running for a very short period of time and are therefore not displayed among the top search results.

REFERENCES

1. Lastdrager EE. Achieving a consensual definition of phishing based on a systematic review of the literature. *Crime Science* 2014; **3**(1):1–10.
2. Mohammad RM, Thabtah F, McCluskey L. Tutorial and critical analysis of phishing websites methods. *Computer Science Review* 2015; **17**:1–24.
3. Garera S, Provos N, Chew M, Rubin AD. A framework for detection and measurement of phishing attacks. In Proceedings of the 2007 ACM workshop on Recurring malcode. Alexandria, Virginia, USA, 2007; 1–8.
4. Khonji M, Iraqi Y, Jones A. Phishing detection: a literature survey. *IEEE Communications Surveys & Tutorials* 2013; **15**(4):2091–2121.
5. Pandove K, Jindal A, Kumar R. Email spoofing. *International Journal of Computer Applications* 2010; **5**(1):27–30.
6. Varshney G, Sardana A, Joshi RC. Secret information display based authentication technique towards preventing phishing attacks. In Proceedings of the International Conference on Advances in Computing, Communications and Informatics, Chennai, India, 2012; 602–608.
7. Hong J. The state of phishing attacks. *Communication of the ACM* 2012; **55**(1):74–81.
8. Drake CE, Oliver JJ, Koontz EJ. Anatomy of a phishing email. In CEAS, 2004.
9. Jagatic TN, Johnson NA, Jakobsson M, Menczer F. Social phishing. *Communication of the ACM* 2007; **50**(10):94–100.
10. Almomani A, Gupta B, Atawneh S, Meulenberg A, Almomani E. A survey of phishing email filtering techniques. *IEEE Communications Surveys & Tutorials* 2013; **15**(4):2070–2090.
11. Varshney G, Joshi R, Sardana A. Personal secret information based authentication towards preventing phishing attacks. In Advances in Computing and Information Technology, 2012; 31–42.
12. Gupta S,Kumar P. A desktop notification based scheme for preventing online frauds attempts to cloud users. In Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on, 2013; 255–260.
13. Workman M. Wisecrackers: a theory-grounded investigation of phishing and pretext social engineering threats to information security. *Journal of the Association for Information Science and Technology* 2008; **59**(4):662–674.
14. Kim H, Huh JH. Detecting DNS-poisoning-based phishing attacks from their network performance characteristics. *Electronics Letters* 2011; **47**(11):656–658.
15. Magdalin V. Securing networks against spear phishing attacks,to Google Patents, 2015.
16. Bradley T. Epsilon data breach: expect a surge in spear phishing attacks. 2012; http://www.pcworld.com/article/224192/epsilon_data_breach_expect_a_surge_in_spear_phishing_attacks.html.
17. Unlu SA, Bicakci K. Notabnab: protection against the "tabnabbing attack". In eCrime Researchers Summit (eCrime), 2010, 2010; 1–5.
18. Dhamija R,Tygar JD. The battle against phishing: dynamic security skins. In Proceedings of the 2005 symposium on Usable privacy and security, Pittsburgh, Pennsylvania, USA, 2005; 77–88.
19. Bergholz A, De Beer J, Glahn S, Moens M-F, Paaß G, Strobel S. New filtering approaches for phishing email. *Journal of Computer Security* 2010; **18**(1):7–35.
20. Islam R, Abawajy J. A multi-tier phishing detection and filtering approach. *Journal of Network and Computer Applications* 2013; **36**(1):324–335.
21. APWG. APWG phishing attacks trends report. 2014; http://www.antiphishing.org/resources/apwg-reports/.
22. APWG. Phishing activity treds report. 2014; http://docs.apwg.org/reports/apwg_trends_report_q2_2014.pdf
23. Zhang Y, Egelman S, Cranor L, Hong J. Phinding phish: evaluating anti-phishing tools. In 14th Annual Network and Distributed System Security Symposium (NDSS 2007}, 2006.
24. I. Fette, N. Sadeh, and A. Tomasic, Learning to detect phishing emails. In Proceedings of the 16th

international conference on World Wide Web, 2007; 649–656.
25. Singh A, Kumar V, Sengar S, Wairiya M. In *Detection and Prevention of Phishing Attack Using Dynamic Watermarking*. Information Technology and Mobile Communication, Communications in Computer and Information Science, Das V, Thomas G, Lumban Gaol F (eds). Springer Berlin Heidelberg: Nagpur, Maharashtra, India, 2011; 132–137.
26. Liou JC, Egan G, Patel JK, Bhashyam S. A sophisticated RFID application on multi-factor authentication. In Information Technology: New Generations (ITNG), 2011 Eighth International Conference on, 2011; 180–185.
27. Parno B, Kuo C, Perrig A. In *Phoolproof Phishing Prevention*. Financial Cryptography and Data Security, Lecture Notes in Computer Science, Di Crescenzo G, Rubin A (eds). Springer Berlin Heidelberg: Anguilla, British West Indies, 2006; 1–19.
28. Fraser N. The usability of picture passwords. Tricerion Group plc, 2006.
29. Ross B, Jackson C, Miyake N, Boneh D, Mitchell JC. Stronger password authentication using browser extensions. In Usenix security, 2005; 17–32.
30. Pippal R, Jaidhar CD, Tapaswi S. Robust smart card authentication scheme for multi-server architecture. *Wireless Personal Communications* 2013; **72**(1):729–745.
31. Kieseberg P, Leithner M, Mulazzani M, *et al.* QR code security. In Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia, Paris, France, 2010; 430–435.
32. Kumaraguru P, Cranshaw J, Acquisti A, *et al.* School of phish: a real-world evaluation of anti-phishing training. In Proceedings of the 5th Symposium on Usable Privacy and Security, Mountain View, California, USA, 2009; 1–12.
33. S. Sheng, B. Magnien, P. Kumaraguru, *et al.* Anti-phishing phil: the design and evaluation of a game that teaches people not to fall for phish In Proceedings of the 3rd symposium on Usable privacy and security, Pittsburgh, Pennsylvania, USA, 2007; 88–99.
34. Dodge RC Jr, Carver C, Ferguson AJ. Phishing for user security awareness. *Computers & Security* 2007; **26**(1):73–80.
35. Kumaraguru P, Rhee Y, Acquisti A, Cranor LF, Hong J, Nunge E. Protecting people from phishing: the design and evaluation of an embedded training email system. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, San Jose, California, USA, 2007; 905–914.
36. Kirlappos I, Sasse MA. Security education against phishing: a modest proposal for a major rethink. *IEEE Security & Privacy* 2012; **10**(2):24–32.
37. Kumaraguru P, Sheng S, Acquisti A, Cranor LF, Hong J. Lessons from a real world evaluation of anti-phishing training. In eCrime Researchers Summit, 2008, 2008; 1–12.
38. Jun Ho H, Hyoungshick K. Phishing detection with popular search engines: simple and effective. In Proceedings of the 4th Canada-France MITACS conference on Foundations and Practice of Security, Paris, France, 2013.
39. H. Bo, W. Wei, W. Liming, *et al.* A hybrid system to find & fight phishing attacks actively. In Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on, 2011; 506–509.
40. Singh A, Tripathy S. TabSol: an efficient framework to defend Tabnabbing. In Information Technology (ICIT), 2014 International Conference on, 2014; 173–178.
41. Marchal S, François J, Engel T. *Proactive Discovery of Phishing Related Domain Names*. Research in Attacks, Intrusions, and Defenses. Springer: Amsterdam, The Netherlands, 2012; 190–209.
42. Xiang G, Hong J, Rose CP, Cranor L. CANTINA+: a feature-rich machine learning framework for detecting phishing web sites. *ACM Transactions on Information and System Security (TISSEC)* 2011; **14**(2):21.
43. Developers G. Safe browsing API-developer guide V3. 2014; https://developers.google.com/safe-browsing/developers_guide_v3
44. Neupane A, Rahman ML, Saxena N, Hirshfield L. A multi-modal neuro-physiological study of phishing detection and malware warnings. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, Colorado, USA, 2015; 479–491.
45. Wardman B, Stallings T, Warner G, Skjellum A. High-performance content-based phishing attack detection. In eCrime Researchers Summit (eCrime), 2011, 2011; 1–9.
46. Alexa. The top 500 sites on the web. 2015; http://www.alexa.com/topsites
47. Phishtank. Developer Information. 2015; https://www.phishtank.com/developer_info.php
48. Varshney G, Misra M, Atrey PK. A phish detector using lightweight search features. *Computers & Security* 2016; **62**:213–228.
49. Ramesh G, Krishnamurthi I, Kumar K. An efficacious method for detecting phishing webpages through target domain identification. *Decision Support Systems* 2014; **61**:12–22.
50. Ee Hung C, Kang Leng C, San Nah S, Wei King T. Phishing detection via identification of website identity. In IT Convergence and Security (ICITCS), 2013 International Conference on, 2013; 1–4.

51. Xiang G, Hong JI. A hybrid phish detection approach by identity discovery and keywords retrieval. In Proceedings of the 18th international conference on World wide web, Madrid, Spain, 2009; 571–580.
52. Dunlop M, Groat S, Shelly D. GoldPhish: using images for content-based phishing analysis. In Internet Monitoring and Protection (ICIMP), 2010 Fifth International Conference on, 2010; 123–128.
53. Moghimi M, Varjani AY. New rule-based phishing detection method. *Expert Systems with Applications* 2016; **53**:231–242.
54. Singh P, Maravi YPS, Sharma S. Phishing websites detection through supervised learning networks. In Computing and Communications Technologies (ICCCT), 2015 International Conference on, 2015; 61–65.
55. Mohammad RM, Thabtah F, McCluskey L. Predicting phishing websites based on self-structuring neural network. *Neural Computing and Applications* 2014; **25**(2):443–458.
56. Mohammad RM, Thabtah F, McCluskey L. Predicting phishing websites using neural network trained with back-propagation. In Proceedings on the International Conference on Artificial Intelligence (ICAI) (p. 1). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).
57. Luong Anh Tuan N, Ba Lam T, Huu Khuong N, Minh Hoang N. A novel approach for phishing detection using URL-based heuristic. In Computing, Management and Telecommunications (ComManTel), 2014 International Conference on, 2014; 298–303.
58. Abdelhamid N, Ayesh A, Thabtah F. Phishing detection based associative classification data mining. *Expert Systems with Applications* 2014; **41**(13):5948–5959.
59. Abdelhamid N. Multi-label rules for phishing classification. *Applied Computing and Informatics* 2015; **11**(1):29–46.
60. Mohammad RM, Thabtah F, McCluskey L. Intelligent rule-based phishing websites classification. *IET Information Security* 2014; **8**(3):153–160.
61. Kausar F, Al-Otaibi B, Al-Qadi A, Al-Dossari N. Hybrid client side phishing websites detection approach. *International Journal of Advanced Computer Science and Applications (IJACSA)* 2014; **5**(7):132–140.
62. Luong Anh Tuan N, Ba Lam T, Huu Khuong N, Minh Hoang N. Detecting phishing web sites: a heuristic URL-based approach. In Advanced Technologies for Communications (ATC), 2013 International Conference on, 2013; 597–602.
63. Gu X, Wang H, Ni T. An efficient approach to detecting phishing web⋆. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)* 2013; **3**(5).
64. Barraclough PA, Hossain MA, Tahir MA, Sexton G, Aslam N. Intelligent phishing detection and protection scheme for online transactions. *Expert Systems with Applications* 2013; **40**(11):4697–4706.
65. Li X, Zhenxin Z, Shouhuai X, Keying Y. Cross-layer detection of malicious websites. In Proceedings of the third ACM conference on Data and application security and privacy, San Antonio, Texas, USA, 2013.
66. Birhanu E. Effective analysis, characterization, and detection of malicious web pages. In Proceedings of the 22nd international conference on World Wide Web companion, Rio de Janeiro, Brazil, 2013.
67. Shahriar H, Zulkernine M. Trustworthiness testing of phishing websites: a behavior model-based approach. *Future Generation Computer Systems* 2012; **28**(8):1258–1271.
68. Weiwei Z, Qingshan J, Tengke X. An intelligent anti-phishing strategy model for phishing website detection. In Distributed Computing Systems Workshops (ICDCSW), 2012 32nd International Conference on, 2012; 51–56.
69. He M, Horng S-J, Fan P, *et al.*. An efficient phishing webpage detector. *Expert Systems with Applications* 2011; **38**(10):12018–12027.
70. Aburrous M, Hossain MA, Dahal K, Thabtah F. Intelligent phishing detection system for e-banking using fuzzy data mining. *Expert Systems with Applications* 2010; **37**(12):7913–7921.
71. Justin M, Lawrence KS, Stefan S, Geoffrey MV. Beyond blacklists: learning to detect malicious web sites from suspicious URLs. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, Paris, France, 2009.
72. Netcraft. Netcraft extension. 2015; http://toolbar.netcraft.com/
73. Zhang Y, Hong JI, Cranor LF. Cantina: a content-based approach to detecting phishing web sites. In Proceedings of the 16th international conference on World Wide Web, Banff, Alberta, Canada, 2007; 639–648.
74. Chou N, Ledesma R, Teraguchi Y, Mitchell JC. Client-side defense against web-based identity theft. In NDSS, 2004.
75. Firefox M. How does built-in phishing and malware protection work?. 2014; https://support.mozilla.org/en-US/kb/how-does-phishing-and-malware-protection-work
76. Li L, Berki E, Helenius M, Ovaska S. Towards a contingency approach with whitelist- and blacklist-based anti-phishing applications: what do usability tests indicate? *Behaviour & Information Technology* 2014; **33**(11):1136–1147.

77. Krishnamurthy B, Spatscheck O, Van Der Merwe J, Ramachandran A. Method and apparatus for identifying phishing websites in network traffic using generated regular expressions, to Google Patents, 2009.
78. Raskin A. Tabnabbing: a new type of phishing attack. 2014; http://www.azarask.in/blog/post/a-new-type-of-phishing-attack/
79. Chiew KL, Chang EH, Sze SN, Tiong WK. Utilisation of website logo for phishing detection. *Computers & Security* 2015; **54**:16–26.
80. Teh-Chung C, Torin S, Scott D, James M. An anti-phishing system employing diffused information. *ACM Transaction on Information and System Security* 2014; **16**(4):1–31.
81. Sarika S, Paul V. An anti-phishing framework to defend Tabnabbing attack. In International Conference on Security and Authentication, 2014; 132–135.
82. Jian M, Pei L, Kun L, Tao W, Zhenkai L. BaitAlarm: detecting phishing sites using similarity in fundamental visual features. In Intelligent Networking and Collaborative Systems (INCoS), 2013 5th International Conference on, 2013; 790–795.
83. Philippe De R, Nick N, Lieven D, Wouter J. TabShots: client-side detection of tabnabbing attacks. In Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security, Hangzhou, China, 2013.
84. Lam I-F, Xiao W-C, Wang S-C, Chen K-T. *Counteracting phishing page polymorphism: an image layout analysis approach*. Advances in Information Security and Assurance. Springer: Seoul, Korea, 2009; 270–279.
85. Chen CS, Su SA, Hung YC. Protecting computer users from online frauds,to Google Patents, 2011.
86. Gastellier-Prevost S, Granadillo GG, Laurent M. A dual approach to detect pharming attacks at the client-side.In New Technologies, Mobility and Security (NTMS), 2011 4th IFIP International Conference on, 2011; 1-5.
87. Sun B, Wen Q, Liang X. A DNS based anti-phishing approach. In Networks Security Wireless Communications and Trusted Computing (NSWCTC), 2010 Second International Conference on, 2010; 262–265.
88. He Y, Zhenyu Z, Krasser S, Tang Y. Mining DNS for malicious domain registrations. In Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2010 6th International Conference on, 2010; 1-6.
89. Ferolin RJ,Kang C-U. Phishing attack detection, classification and proactive prevention using fuzzy logic and data mining algorithm. onlinepresent.org, **12**, 2012, 2012.
90. Wu L, Du X, Wu J. Effective defense schemes for phishing attacks on mobile computing platforms. *IEEE Transactions on Vehicular Technology* 2016; **65**(8):6678–6691.
91. Wu L, Du X, Wu J. MobiFish: a lightweight anti-phishing scheme for mobile phones. In Computer Communication and Networks (ICCCN), 2014 23rd International Conference on, 2014; 1–8.
92. Hou J, Yang Q. Defense against mobile phishing attack. Computer Security Course Project, http://wwwpersonal. umich. edu/yangqi/pivot/mobile phishing defense. pdf, 2012.