Telecommunication Systems (2018) 68:687–700
https://doi.org/10.1007/s11235-017-0414-0

Towards detection of phishing websites on client-side using machine learning based approach

Ankit Kumar Jain[1] · **B. B. Gupta**[1]

Published online: 26 December 2017

Abstract
The existing anti-phishing approaches use the blacklist methods or features based machine learning techniques. Blacklist methods fail to detect new phishing attacks and produce high false positive rate. Moreover, existing machine learning based methods extract features from the third party, search engine, etc. Therefore, they are complicated, slow in nature, and not fit for the real-time environment. To solve this problem, this paper presents a machine learning based novel anti-phishing approach that extracts the features from client side only. We have examined the various attributes of the phishing and legitimate websites in depth and identified nineteen outstanding features to distinguish phishing websites from legitimate ones. These nineteen features are extracted from the URL and source code of the website and do not depend on any third party, which makes the proposed approach fast, reliable, and intelligent. Compared to other methods, the proposed approach has relatively high accuracy in detection of phishing websites as it achieved 99.39% true positive rate and 99.09% of overall detection accuracy.

Keywords Phishing attack · Social engineering · Website · Machine learning · Hyperlink

1 Introduction

Phishing is an online identity theft, which can deceive Internet users into revealing their secret information and credentials, e.g., login id, password, credit card number, etc. Phishing is one of the major computer security threats faced by the cyber-world and could lead to financial losses for both industries and individuals [1]. Among the various cybersecurity attacks, phishing paid special attention because of its adverse effect on the economy [2–4]. According to APWG report, 122,0523 phishing attacks were found worldwide in 2016, and it is observed as 65% of growth over 2015 [5]. The per month attack growth also increased by 5753% over 12 years from 2004 to 2016 (1609 phishing attacks per month in 2004 and average of 92,564 attacks in 2106). Quarter 2 of 2016 represented an all-time high number of phishing attacks, which were 466,065 [5]. The motive of phishing attack is not only gaining the credentials; now it has become the number 1 delivery method for other types of malicious software like ransomware [6]. In August 2016, the financial loss due to the phishing scam is more than 17.36$ million in US only, followed by Japan and UK with the loss of 8.38 and 7.21 million dollar respectively [6].

Todays, cyber experts, and phisher are in a rat race condition. The cyber experts continue to improve anti-phishing solutions with the help of researchers and developers. The developer invents various anti-phishing tools that alert users to malicious emails and websites. (e.g., Calling-ID Toolbar, Netcraft Cloudmark Anti-Fraud Toolbar, etc.). A Recent study examines that only 3 out of 14 tools identified a phishing website hosted locally and it is a critical concern on the trust of the conventional tools [7]. Moreover, these tools are exposed in the public domain. Raising human awareness is not a sufficient mitigation method and deploying complementary technical solutions is a crucial requirement [8,9]. In the previous few years, researchers and developers build various phishing detection solutions. However, the phishing problem still available, and the development of efficient anti-phishing approach become a challenging task. Moreover, most of the anti-phishing solutions produce high false positive rate and not capable of dealing with zero hour attack. Blacklist based detection approaches have the quick access time; however, it cannot identify the zero-hour attack. Moreover, other solutions like heuristic and visual comparison produce high false predictions.

✉ B. B. Gupta
gupta.brij@gmail.com

[1] National Institute of Technology Kurukshetra, Kurukshetra, India

Therefore, it is essential to design an approach that can efficiently classify phishing webpages. The recent development of phishing detection employed numerous machine learning based approaches. These approaches train a classification algorithm with some features that can distinguish a phishing webpage from the legitimate one [10]. The efficiency of the detection approaches depends on training data, selection of good feature set, and classification algorithm used to train these features [11].

The existing machine learning based approaches extract features from various sources like URL, page source, search engine, and third party services like website traffic, DNS, whois record, etc. The extraction of third party features is a complicated and time-consuming process [12]. Integration of features from different sources is also a difficult process. Therefore, they are complicated, slow, and does not produce results in the real-time. To cope up this problem, this paper presents an efficient solution, which extracts the features from client side only. Identifying the outstanding features is one of the preconditions for the design of good phishing detection approach. Therefore, we have examined the various attributes of the phishing and legitimate websites in depth and identified various efficient client side features in order to detect the phishing websites. These nineteen features are obtained from the URL and source code of the website. Therefore, it makes our approach fast and reliable. We have evaluated proposed features on various machine learning algorithms using 4059 phishing and legitimate websites dataset. Evaluation results show that the proposed approach accurately filters the phishing sites as it has 99.39% of true positive rate and very less 1.25% of false positive rate. The main advantages of the proposed approach compared to existing phishing detection solutions are (1) it is fast, reliable and provide real-time phishing detection, (2) it can detect the phishing webpages hosted on the compromised domain, (3) it can detect the webpages written in any textual language, (4) it does not require any dedicated resources for phishing detection (5) it is platform independent, and (6) it is available as a client side desktop application.

The remainder of this paper is structured as follows. Section 2 describes the related work. Section 3 presents the overview of our proposed approach. Section 4 describes the proposed feature set. Section 5 shows the training dataset and performance metrics. We present the implementation and evaluation details in Sect. 6. Section 7 discuss the advantages of our proposed approach. Finally, Sect. 8 concludes the paper and present future work.

2 Related work

This section presents the overview of phishing detection approaches proposed in the literature. Phishing detection approaches split into two classes; user education based and software based. Software-based approaches are further classified into blacklist, visual similarity, search engine, and machine learning based solutions.

2.1 User education

User education approach aims to improve the capacity of Internet users in the detection of phishing attacks [13]. Internet users can be educated to distinguish the characteristics of phishing and legitimate emails and websites. In this, Sheng et al. [14] developed an interactive educational game "Anti-Phishing Phill", that teach users that how to identify the phishing websites. After spending 15 min on the game, users were better able to identify phishing websites compared to the other users who did not play the game. The main motto behind this game design is to provide conceptual knowledge to computer users behind the phishing attacks. This conceptual knowledge may help the users in avoiding phishing attacks.

2.2 Phishing blacklist

A blacklist contains the list of malicious domains, URLs, and IP addresses [15]. Sheng et al. [16] showed that a fake domain added in blacklist after the substantial amount of time and approximate 50–80% of fake domains added after performing the attack. The blacklist needs to be the regular update from their source because thousands of fake websites launch every day.

2.3 Visual similarity based techniques

These techniques [17] utilize various features to compute the similarity between websites like page source code, images, textual content, text formatting, HTML tags, CSS, website logo, etc. Most of the visual similarity based approaches compare the new website with previously visited or stored websites. Therefore, these techniques cannot detect the new phishing websites and produce high false negative rate. Some of the techniques take the snapshot of websites to compare which require high computation time, therefore does not fit in time constraint environment.

2.4 Machine learning based techniques

These methods [10,18–21] train a classification algorithm with some features that can distinguish a genuine website from the phishing one. In this, a website is declared as phishing, if the design of the websites matches with the predefined feature set. The performance of these solutions depends on features set, training data and classification algorithm. These features are extracted from various sources like URL, page

source, website traffic, search engine, DNS, etc. In this, some of the features are difficult to access, slow, third party dependent, and time consumable. Therefore, some of the machine learning solutions require a high computations to obtain and compute the features from various sources.

2.5 Search engine based techniques

The search Engine (SE) based techniques extract identity features (e.g., title, copyright, logo, domain name, etc.) from the webpage and make use of the search engine to check the legitimacy of webpage [22–24]. The FPR of these methods is high because newly constructed genuine sites do not appear in the top search results. Previous search based techniques believe that legitimate site appears in the top results of search engine. Although, only popular sites appear in the top search results. Moreover, these techniques do not provide desired results when webpages are in a language other than English because the search engine like Google does not give precise results for the non-English search query.

3 Proposed phishing detection approach

3.1 Design objectives

We designed our anti-phishing approach to satisfy following requirements:

- *High detection accuracy (D1)* Misclassification of genuine website as phishing (false positive) must be minimum and correct classification of phishing websites (true positive) must be high to provide high detection accuracy.
- *Language independent detection (D2)* The detection approach must not be reliant on a language specific content of the webpage.
- *Real time protection (D3)* The phishing detection method must provide its prediction before revealing the credential of the user on the phishing website.
- *Target independent detection (D4)* Phishing detection should not be dependent on a particular brand/sector, a phishing detection approach should be capable of detecting any kind of phishing website regardless of the newly created website (zero-hour attack).
- *Nondisclosure of privacy (D5)* The detection approach must not share the user's data (e.g., browser's history) to any third party.

3.2 Design concepts

We adapted following design concepts to fulfilling the above mentions requirements:

- *Client side implementation* The proposed approach is implemented at client side on user's system. Therefore, it provides better user's privacy (D5).
- *Feature set selection* The proposed features are extracted from the URL and source code of webpage (no third party features). Therefore, extraction and computing the features are easy and fast, and it provides a real time phishing prediction (approximately 2–6 s) (D3). Moreover, the most of the features are not affected by the textual language of the webpage (D2) and can detect any kind of phishing website (D4). Moreover, we proposed some new features that increase the detection accuracy of our method.
- *Sensitivity analysis of features* We conducted a sensitivity analysis on the feature set to ensure the higher detection accuracy (D1). Sensitivity analysis predicts the most powerful features in the detection of the phishing websites.

3.3 System architecture

Figure 1 presents the system architecture of the proposed approach. Our approach extracts and analyses various features of suspicious websites for successful detection of wide-ranging phishing attack. The selection of outstanding feature set is the major contribution of this paper. We proposed six new features to improve the detecting accuracy of phishing webpages. Our proposed features identify the relation between the page content and the URL of the webpage. We used pattern matching algorithms to match the domain name of page resource elements with the domain name of the queried webpage. Our features are based on URL and content of the webpage. The content obtained from the page source and document object model (DOM) of the webpage. A web crawler is used to gather the website features automatically. In particular, features 1, 2, 3, 5, 6, 17, 18, 19 are taken from other approaches [10,18–21]; features 7, 8, 11, 14, 15, 16 are novel and proposed by us. Moreover, the features 4, 9, 10, 12, 13 are proposed by other approaches but we modified them for better results. The features of our approach are classified into five categories as shown in Table 1. Section 4 of this paper give the detailed explanation of the proposed features. After extraction of features, we apply heuristics to generate the feature vector and creates a unique feature vector for every website to generate the labelled dataset. The feature vector is the numerical representation of feature for the statistical procedures in machine learning algorithm. In this, F = {F1, F2, F3, …, F19} is defined as the feature vectors corresponding to each feature. Each feature produces the value in the form of 1 and 0, where 1 indicates for phishing and 0 indicate for legitimate. In the training stage, a random forest (RF) classifier is trained using the feature vector taken from every

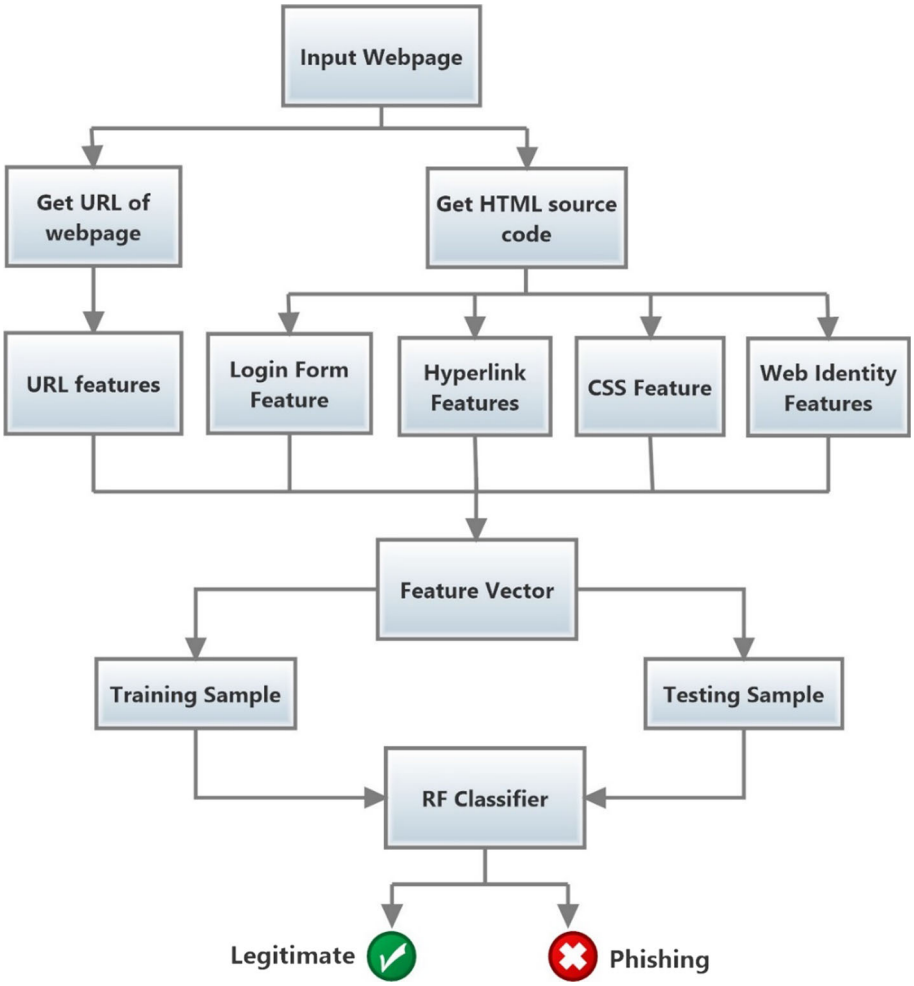

Fig. 1 System architecture of proposed approach

Table 1 Feature used in the proposed approach

S. No	Category	Features name	Total features
1	URL forgery	Number of dots, presence of special symbol, URL length, suspicious words in URL, position of top-level domain, http count, brand name in URL, Data URI	8
2	Fake login form	Fake login form identification	1
3	Hyperlink information	Number of hyperlinks, no hyperlink feature, foreign hyperlinks, empty hyperlinks, erroneous hyperlinks, hyperlinks redirection	6
4	Copied CSS	Suspicious CSS identification	1
5	Fake web identity	Copyright, identity keywords, favicon	3

entry in the training dataset. In the testing phase, the classifier determines whether a given website is a phishing site or not. A binary classifier classifies the websites into two possible categories namely phishing and legitimate. When a user requested for a new website, the trained classifier identify the legitimacy of given website from the generated feature vectors.

4 Features extraction

Given the limitation of search engine and third party dependent approaches presented in the literature, we utilize the client-side specific features in our approach. We have selected eight URL-based features (F1–F8), one login form feature (F9), six hyperlink specific features (F10–F15), one CSS feature (F16), and three web identity features

Fig. 2 URL structure

(F17–F20). We discuss all these features in the following subsections.

4.1 URL based features

A webpage is addressed by Uniform Resource Locator (URL). Figure 2 presents the structure of URL. The structure divides URL into five parts starting from protocol, sub-domain, base domain, top-level domain, and path segment. An attacker has the control over the full URL, and it can fix any value of subdomain, base domain, and path segment. This subsection presents how a cybercriminal trap users using URL obscure technique.

F1—Number of dots in URL This features checks the number of dot (.) in the URL. In general, the legitimate website does not contain more than three dots in the URL although phishing URL may contain more than three dots. The phishing URL contains many sub-domains in URL to confuse users. These subdomains are separated by the dot symbol. Consider an example https://support.appleid.itune.com-txutwo3wfh.store; it is a phishing site, but some users may believe that they are visiting the official Apple website. Therefore, if any URL has more than three dot symbol, it is considered as phishing.

$$F1 = \begin{cases} 1, & if\ dots\ in\ URL\ \geq\ 4 \\ 0, & Otherwise \end{cases} \quad (1)$$

F2—Presence of special symbol in URL This feature determines whether the URL address contains the special symbol at sign ("@") and dash symbol (-). The presence of "@" symbol in the URL ignores everything written before it, and the path after "@" consider as the real domain for retrieving the website. The dash (-) symbol is used in fake URL to looks like original URL. Attackers add some prefix or suffix keywords in brand name with dash symbol, so users believe that they are accessing the right website. e.g. www.paypal-india.com. It is the fake site, but a user may think that it is the official Indian site of PayPal.

$$F2 = \begin{cases} 1, & if\ URL\ contain\ @\ or\ -\ symbol \\ 0, & Otherwise \end{cases} \quad (2)$$

F3—Length of URL Phisher generally uses the long URL in the address bar to hide the brand or organization name. Usually, legitimate websites are short, significant, and easy-to-remember. On the other hand, phishing websites are normally longer, and may not contain any meaningful domain. Moreover, Phisher also hides the redirected information in long URL. In our experiment, we found that if URL length is greater than 74, then the website is more likely to be phishing.

$$F3 = \begin{cases} 1, & if\ URL\ length\ \geq\ 74 \\ 0, & Otherwise \end{cases} \quad (3)$$

F4—Suspicious words in URL This feature examines the presence of suspicious words in URL. Phisher adds suspicious keywords in URL to gain the trust on it. We identified nine keywords frequently present in the phishing URLs namely security, login, signin, bank, account, update, include, webs and online. If any of these keywords are found in URL, then this feature make the URL as phishing.

$$F4 = \begin{cases} 1, & if\ URL\ contain\ any\ suspicious\ word \\ 0, & Otherwise \end{cases} \quad (4)$$

F5—Position of the top-level domain This feature examines the two things regarding the top-level domains (TLD) in the URL. The first thing is to check the position of the TLD in the base domain part. The second point, it also verifies the occurrence of more than one TLD in the URL. In the legitimate site, the top-level domain appears one time in URL, and it is not present in base domain part.

Example 1 http://support.paypal.com.prodigitalmedia.org/signin/?country.x=US\&loc, in the given phishing URL, top-level domain .com (TLD) appear in the base domain part.

Example 2 http://romeiroseromarias.com.br/verify/0/asb.co.nz/, in the given phishing URL, two top-level domains appear (.com.br and .co.nz).

$$F5 = \begin{cases} 1, & if\ top\ level\ domain\ name\ present\ in\ base\ domain \\ 1, & if\ occurrence\ of\ top\ level\ domain\ in\ URL\ >\ 1 \\ 0, & otherwise \end{cases} \quad (5)$$

F6—http count in URL This feature counts the appearance of 'http' protocol in the URL. In phishing URL, http protocol may appear more than one time, however, in the legitimate site, 'http' appears only one time.

$$F6 = \begin{cases} 1, & if\ http\ count\ in\ URL\ >\ 1 \\ 0, & Otherwise \end{cases} \quad (6)$$

F7—Brand name in URL Most of the phishing websites have the brand name of the targeted website somewhere in the URL. According to the current report of APWG, the 45.97% of phishing websites contain brand name of the targeted site

in URL [25]. In this feature, if a brand name is present and its position is not at the right place, then site marked as phishing. We have selected top 500 phishing targets including banks, payment gateways, etc. The top name found in the phishing URLs are PayPal, Amazon, Apple, Yahoo, Dropbox, Google, AOL, USAA, etc.

Example http://forlittledrops.org/asd/Paypalaccount/, in the given phishing URL, "PayPal" found in the path segment.

$$F7 = \begin{cases} 1, & \textit{if any top brand name present at incorrect position in URL} \\ 0, & \textit{Otherwise} \end{cases} \quad (7)$$

F8—Data URI These days data URI (uniform resource identifier) based attack seems to be most common phishing attacks [26]. Data URI scheme provides a facility to add data in-line in web pages as if they were external resources. This scheme allows fetching different elements such as HTML, images, and javascript in a single HTTP request rather than multiple HTTP requests. The syntax of data URI is given below:

data : [<media type>][; base64], <data>
With the data URI method, it is possible to show media contents in a web browser without hosting the actual data on the internet. Traditional anti-phishing techniques fail to detect it because the phishing web pages not hosted anywhere on the internet. In this attack, users do not require to communicate with a server to get phished.

$$F8 = \begin{cases} 1, & \textit{if} \text{ Data URI present} \\ 0, & \textit{Otherwise} \end{cases} \quad (8)$$

4.2 Login form based feature

F9—Fake login form The fake websites always include the login form because it is the only way to obtain the user's personal data. In the legitimate website, the action field of login form usually contains a link that has the same base domain as appear in the address bar of the browser. However, as per our observation, form action field of the phishing websites includes the URL having the different base domain, null links (URL may be in footer section), or a simple PHP file. The action attribute of phishing website includes a PHP file, which named as mail.php, login.php, index.php, etc. PHP file contains a script that saves the inputted data (e.g., user id or password) in a text file at hacker's computer. The algorithm to detect fake login form is presented in Fig. 3.

4.3 Hyperlink specific features

F10—Number of webpages A legitimate website usually contains many web pages while a phishing website has very limited pages. Moreover, sometimes a phishing site only consist of one or two web pages (usually attackers create only login page). This feature calculates the number of pages in a website by visiting hyperlinks in the source code. In our approach, we extracted the hyperlinks from the "src" attribute of img, script, frame, input, link tags and anchor attribute of the href tag.

Algorithm1: Fake Login form Detector Algorithm
Input: DOM tree of Suspicious URL
Output: Existence of Fake login form, F9 ϵ {0, 1},
Start
1. If the value of action field is blank, # or javascript:void(0)) then set F9 = 1
2: If the value of action field is in the form of "filename.php" then set F9 = 1
3: If action field contain foreign base domain then set F9 =1 else set F9 = 0
End

Fig. 3 Algorithm for detection of fake login form

$$F10 = \textit{total hyperlinks present in website} \quad (9)$$

F11—No hyperlink feature This feature checks whether any hyperlinks present on the website or not. Sometimes attackers use the hyperlink hidden techniques [27] to bypass the anti-phishing solutions. Moreover, attackers also use server site script to cover up the page source content. From our study, we analyse that a legitimate website contains at least one hyperlink. Moreover, if a website does not include the hyperlinks, it depicts the phishing attack.

$$F11 = \begin{cases} 1, & \textit{if number of hyperlinks are zero} \\ 0, & \textit{Otherwise} \end{cases} \quad (10)$$

F12—Foreign hyperlinks The Foreign hyperlink contains domain name different from the website domain name. Cybercriminals usually copy the HTML coding from their targeted official website to construct the phishing website, and it may have numerous foreign hyperlinks that point to their targeted site [28]. In a legitimate website, most of the hyperlinks point to the browsed domain name. On the other hand, phishing sites have the many hyperlinks that point to the foreign domain. In this features, we calculate the ratio of the external hyperlink to the total hyperlink present in the website. The feature results in 1 if the ratio is greater than 0.5 otherwise, the result is 0.

$$F12 = \begin{cases} 1, & if \ \frac{Foreign\ Hyperlinks}{Total\ Hyperlinks} > 0.5 \ and \ Total\ Hyperlinks\ in\ Webpage > 0 \\ 0, & Otherwise \end{cases} \quad (11)$$

F13—Empty hyperlinks Empty or null hyperlink returns on the same page when a user clicks on it. It increases the chance of user falling for the phishing scam since if a hyperlink is active, the user may end up reaching the original website if it is clicked. Thus, attacker prevents any chance of user's redirection to the original website by removing hyperlinks. Moreover, Phisher also exploits the vulnerability of web browser with the help of empty links. <a href = "#">, <a href = "#content"> and <a href = "JavaScript::void(0)"> HTML coding are used to create empty hyperlinks. This feature calculates the ratio of the empty hyperlinks to the total hyperlinks present on the website. The feature results in 1 if the ratio is greater than 0.34 otherwise, the result is 0.

$$F13 = \begin{cases} 1, & if \ \frac{Empty\ Hyperlinks}{Total\ Hyperlinks} > 0.34 \ and \ Total\ Hyperlinks\ in\ Webpage > 0 \\ 0, & Otherwise \end{cases} \quad (12)$$

F14—Error in hyperlinks This feature checks the error in hyperlinks. Error "404 not found" occurred when a user requested for an URL and server cannot locate the URL. The attacker also adds some hyperlinks in the fake page which not exists. We consider the 403 and 404 response code of hyperlinks. In this feature, we calculate the ratio of hyperlinks occurring error.

$$F14 = \begin{cases} 1, & if \ \frac{Number\ of\ error\ in\ Hyperlinks}{Total\ Hyperlinks} > 0.3 \ and \ Total\ Hyperlinks\ in\ Webpage > 0 \\ 0, & Otherwise \end{cases} \quad (13)$$

F15—Hyperlink redirection In this feature, the system checks the number of hyperlinks redirected to some other place out of the total hyperlinks available in the website. In the phishing attack, sometimes URL redirection confuse user about which website they are surfing. Proposed approach consider response code 301 and 302 for URL redirection. This feature results in 1 if the ratio of redirection hyperlinks is greater than 0.3, else results is 0.

$$F15 = \begin{cases} 1, & if \ \frac{Number\ of\ hyperlinks\ which\ are\ redirecting}{Total\ Hyperlinks} > 0.3 \ and \ Total\ Hyperlinks > 0 \\ 0, & Otherwise \end{cases} \quad (14)$$

4.4 CSS based feature

F16—Copied CSS Cascading Style Sheets (CSS) is a language used for setting the visual appearance of a website. An attacker always tries to mimic the same visual design of the phishing website as the legitimate website. CSS of any website either includes with external CSS file or within the HTML tags itself. Phishing website usually contains the external CSS file, which includes the link of the targeted legitimate site. However, numerous genuine websites use more than one external CSS file or include internal CSS style.

$$F16 = \begin{cases} 1, & if\ CSS\ file\ is\ external\ and\ contain\ foreign\ domain\ name \\ 0, & Otherwise \end{cases} \quad (15)$$

4.5 Web identity based features

The phishing website is the mimicked fake copy of popular brand or organisation, and it may have many identity features, which are copied from the targeted page (e.g., favicon, copyright information, etc.), and claiming a false identity.

Algorithm 2: To find fake identity of Website
Input: the DOM tree of a website
Output: F18 ϵ {0, 1}, 1- Phishing, 0- legitimate
Start
1. Extract the identity keywords from title and meta tag
2. Extract the top keywords using tf-idf algorithm from website
3. Construct the identity keywords set from step 1 and step 2
4. If one of the identity keyword matched with the domain name then set F18 = 0
5. else set F18 = 1
End

Fig. 4 Algorithm to find fake identity of the website

F17—Copyright features The identity of a website can be extracted using copyright information given in the text form. Copyright field of a website contains the name of the organization. This feature extracts the keywords from the copyright field, tokenized them, and matches with the suspicious domain name. The symbol and keywords used to locate the copyright information are the @ symbol, © symbol, & copy, copyright and all right reserved.

$$F17 = \begin{cases} 0, & \textit{if copyright keyword matched with base domain} \\ 1, & \textit{Otherwise} \end{cases} \tag{16}$$

F18—Identity Keywords Some specific keywords present in the website by which a developer can know the information about the exact identity of the website. We make a set of identity keywords, which include title, meta and frequent appeared keywords. Approach apply TF-IDF algorithm [22] to extract the most frequently appeared keywords. These extracted keywords, matched with the domain name of the suspicious site. If the site is legitimate, then one of identity keyword should be the part of the domain name. However, phishing websites include the identity keyword in the path segment of URL to fool users. e.g. http://www.shoppiingg.com/www.amazon.com/cgi-bin/index.htm. The algorithm to find the fake identity of a website is explained in Fig. 4.

F19—Favicon Favicon is a unique image icon associated with the particular website. An attacker may use the same favicon of the targeted website to fool innocent users. Favicon is an .ico file linked to an URL, which is available in link tag of the DOM tree. If the favicon shown in the address bar is different from the present website, then it is considered as phishing attempt. Therefore, if favicon contains the foreign domain, the feature results 1, otherwise it results as 0.

Table 2 Datasets used for training and testing

#	Dataset	Number of instances	Category
1	Phishtank [29]	1528	Phishing
2	Openphish [30]	613	Phishing
3	Alexa [31]	1600	Legitimate
5	Payment gateway [32]	66	Legitimate
6	Top banking website [33]	252	Legitimate

$$F19 = \begin{cases} 1, & \textit{if foreign domain found in favicon link} \\ 0, & \textit{Otherwise} \end{cases} \tag{17}$$

5 Training dataset and performance metric

The proposed approach build a binary classifier based on the features described in Sect. 4, which classify phishing and legitimate websites correctly. This section describes the training and testing dataset, and performance matrix used in our approach.

5.1 Training dataset

Our training dataset consists of 2141 phishing and 1918 legitimate websites. Table 2 presents the number of instances and the sources of phishing and legitimate datasets. We have collected Phishing dataset from two sources namely Phishtank [29] and Openphish [30]. These phishing datasets consist of verified URLs. The phishing websites are short lived. Therefore, we crawled when phishing websites are active. The legitimate dataset is taken from various sources as shown in Table 2. The legitimate dataset Alexa is a most reliable source websites, and it ranks the website based on page views and unique site users. The popular sites got the high rank and unpopular sites situated at the low rank. We added some high ranked and some low ranked websites in our dataset. Moreover, we have added the payment gateways websites in the dataset because these are the perfect target of cyber-criminals. Moreover, our dataset comprises of numerous languages websites (e.g., English, Russian, Spanish, Portuguese, Hindi, Chinese, etc.) to test the language independent performance of our method. Every feature vector has the one entry in the dataset for defined nineteen features.

```
<link rel="shortcut icon" href="https://www.facebook.com/rsrc.php/yl/r/H3nktOa7ZMg.ico" />
<link rel="shortcut icon" href="//in.bmscdn.com/webin/common/favicon.ico" type="image/x-icon" />
<link type="image/png" href="/css/img/favicon.png" rel="shortcut icon">
```

Example of HTML coding for favicon

Table 3 Performance measures used in our approach

Measure	Formula	Description
TPR	$TPR = \frac{N_{P\to P}}{N_P} \times 100$	Rate of phishing websites classified as phishing out of total phishing websites
FPR	$\text{FPR} = \frac{N_{L\to P}}{N_L} \times 100$	Rate of legitimate websites classified as phishing out of total legitimate websites
FNR	$\text{FNR} = \frac{N_{P\to L}}{N_P} \times 100$	Rate of phishing websites classified as legitimate out of total phishing websites
TNR	$\text{TNR} = \frac{N_{L\to L}}{N_L} \times 100$	Rate of legitimate websites classified as legitimate out of total legitimate websites
Accuracy (A)	$\text{Accuracy} = \frac{N_{L\to L}+N_{P\to P}}{N_L+N_P} \times 100$	The rate of phishing and legitimate websites which are identified correctly with respect to all the websites

The feature vector is having identical values removed from the dataset.

5.2 Performance metric

We have calculated the true positive rate (TPR), false positive rate (FPR), true negative rate (TNR), false negative rate (FNR) and accuracy to evaluate the performance of proposed anti-phishing approach. These are the standard metrics to judge any anti-phishing approach. N_L and N_P denote the total number of legitimate and phishing websites respectively. $N_{L\to L}$ are the legitimate websites classified as legitimate, $N_{L\to P}$ are the legitimate websites misclassified as phishing. $N_{P\to P}$ are the phishing websites classified as phishing, and $N_{P\to L}$ are the phishing websites misclassified as legitimate. Table 3 presents the measures used for classification of phishing and legitimate websites.

6 Implementation and evaluation

6.1 Implementation details

In the process of phishing website identification, we first identified the relevant and useful features, construct the dataset by extracting features from legitimate and phishing websites. The labelled dataset is used to train the random forest classifier. A laptop machine having core Pentium i5 processor with 2.4 GHz clock speed and 4 GB RAM is used to implement the proposed anti-phishing solution. Our proposed approach is implemented using the Python programming language. Python offers a vast support of its libraries, and it has a reasonable compile time. We have created the separate function for each feature. Different libraries are required for the extraction of features from the webpage. These libraries can be installed individually using either the pip installer for python or downloading and extracting them from the official websites. Following libraries that are used during execution of the code are -

BeautifulSoup This library is used for pulling data from HTML and XML files.

urllib2 This library is used to get response object from the URL, which extracts all the resources from the webpage.

re This library is used to perform a regular expression match of the desired string to another.

Time This library is used to capture time of a particular instance.

6.2 Complexity of the proposed approach

Feature extraction from the source code of the webpage helps in reducing the processing time as well as response time, hence making the approach more reliable and efficient. The computational complexity of the proposed approach depends on the extraction and computing the proposed features. The URL based features are easy to calculate. To compute feature *F1, F2, F3, F5, F6, F8,* we implemented single pattern matching algorithm (i.e. Knuth–Morris–Pratt algorithm) which required $O(n)$ time and space complexity, where n is the length of URL. URL based feature F4 and F7 are calculated using multiple pattern matching where we have implemented Karp–Rabin algorithm. Its best and average case running time is $O(n + m)$, where m is the combined length of all pattern. To compute hyperlink specific feature *F10, F11, F12, F13, F14, F15, F16, F18*, we need to obtain all hyperlinks from the webpage. A regular expression, which can include and identify all the ways in which hyperlinks can be present on the webpage. Every text in the page source that matches the given regular expression is identified as a hyperlink, and it is calculated in term of linear time complexity of $O(t)$, where t is source code length of the webpage. The login form based feature ($F9$) required regex matching pattern, which is calculated in $O(t)$ time. Copyright feature $F17$ needed a string matching algorithm and computed in $O(t)$ time. Feature $F18$ extracted identity keywords using TF-IDF algorithm. The TF-IDF is based on the frequency of each term and indexing a document of p tokens is $O(p)$. The learning model of our approach is the random forest. The time complexity of building a complete unpruned decision tree is $O(x * y\ log(y))$, where y is the number of records and x is the number of features. In our approach the value of x (i.e., number of features = 19) is constant, so the time complexity of the random forest is $O(y\ log(y))$. In summary, our crawler calculates the proposed features in $O(t)$ time, and learning algorithm

Table 4 Performance of proposed approach on various classifiers

Algorithm	TPR (%)	FPR (%)	Accuracy (%)
Random forest	99.39	1.25	99.09
Support vecor machine	98.23	6.15	96.16
Neural networks	98.93	2.92	98.05
Logistic regression	98.41	1.93	98.25
Naive Bayes	98.46	3.39	97.59

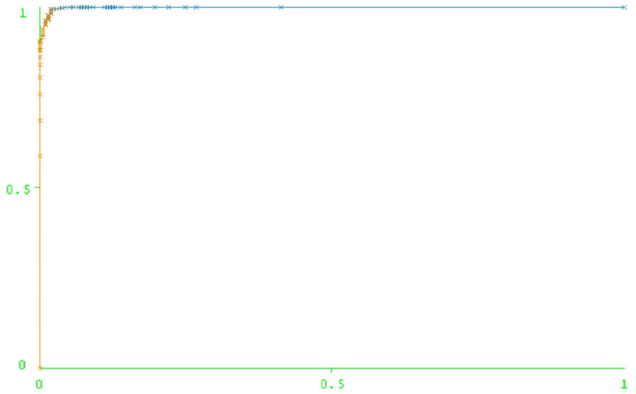

Fig. 5 ROC curve on random forest classifier

required *O(ylog(y))* time. Moreover, the proposed method is not dependent on any third party services, and it does not need to wait for the results return by these services.

6.3 Results on popular classification algorithms

Table 4 presents the performance of our approach on popular and widely accepted classifiers in term of TPR, FPR, and accuracy. WEKA software is used to judge the performance of proposed technique on various machine learning classifiers. We have evaluated our dataset with 10-fold cross-validation, which uses 90% of data for training purpose, and remaining 10% data for testing purpose. It is noticed that random forest outperformed SVM, neural networks, logistic regression and naïve Bayes. Random forest performs best regarding highest TPR, and accuracy. We have also explored the area under ROC (Receiver Operating Characteristic) curve to find a better metric of precision. In our experiment, the area under the ROC curve for phishing website is 99. 85 for the random forest as shown in Fig. 5, and it shows that our approach has the high accuracy in classification of correct websites.

6.4 Evaluation of features

In this experiment, we evaluated the performance of our approach. Random forest classification algorithm is used to

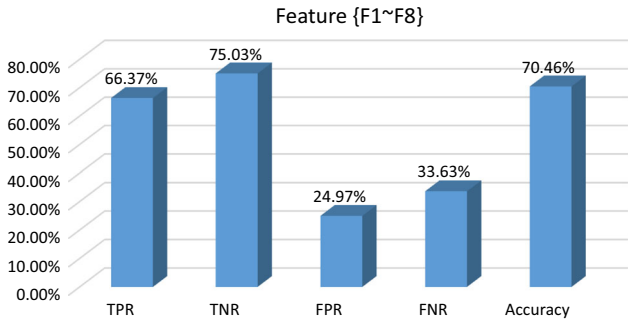

Fig. 6 Results of URL based features

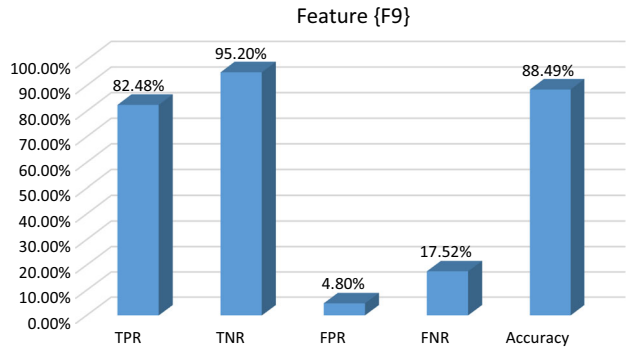

Fig. 7 Performance of login form feature

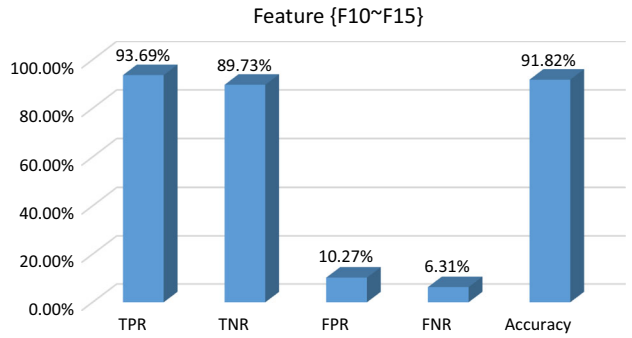

Fig. 8 Results of hyperlink based features

classify the websites. Moreover, we also evaluated the efficiency of each category of the proposed feature set. Figure 6 shows the classification results of URL-based features (feature 1 to feature 8). As seen in the figure, URL based features can correctly filter 75.03% of legitimate websites and 66.37% of phishing websites. Figure 7 presents the performance of fake login form detector. As seen in the figure, our login form detection algorithm can correctly classify high amount of legitimate websites and provide 88.49% accuracy. This is because, attackers change the link of login form handler to send the user's detail to their desired source, and our fake login form algorithm successfully detects it. Figure 8 presents the results of hyperlink based feature, and it represents that these features are most significant in the classification of

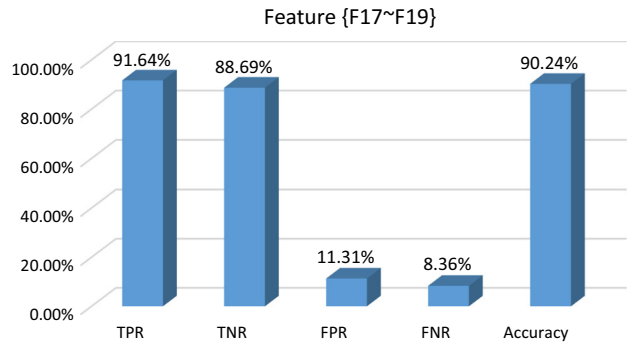

Fig. 9 Results of identity based features

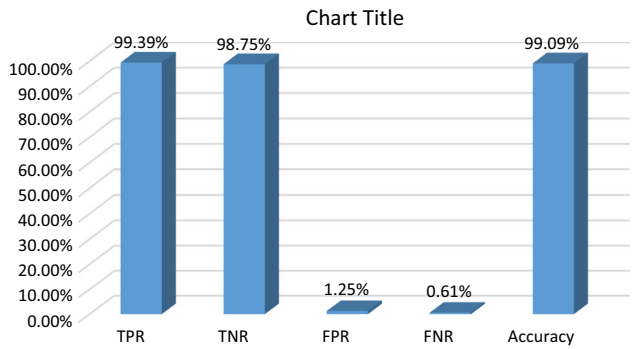

Fig. 10 Overall results of proposed approach

phishing websites accurately. The only hyperlink specific features can detect the 93.69% of phishing websites, and produces 91.82% of overall detection accuracy. As seen in Fig. 9, the identity based features deliver the high accuracy because the phishing websites always claim the wrong identity to trap the Internet user. The results demonstrate that the given feature successfully determine the correct identity of the website. Figure 10 presents the results of proposed approach by combining all kind of features. It is noticed that URL, hyperlink, identity, login form and CSS features are useful in phishing detection. However, a single kind of feature is not sufficient to detect all types of phishing websites and does not produce high accuracy. Therefore, we integrated all features to improve the detection accuracy of the proposed approach. If any approach uses only URL based feature, it yields high false negative rate, which wrongly judged the phishing websites. Our approach results high true positive rate (i.e., more than 99% of phishing sites correctly identified), and low false positive rate (i.e., less than 1.3% of legitimate websites misclassified as phishing). Table 5 presents the confusion matrix of the proposed approach. This matrix shows the number of correct and false predictions. The results on various combination are stated in Table 6 in numeric form.

Table 5 Confusion matrix

	Classified as legitimate	Classified as phishing
Legitimate websites	1894	24
Phishing websites	13	2128

Table 6 Performance of proposed approach on different combination of features

Features	TPR (%)	FPR (%)	Accuracy (%)
F_{URL}	66.37	24.97	70.46
$F_{HYPERLINK}$	93.69	10.27	91.82
$F_{IDENTITY}$	91.64	11.31	90.24
$F_{CSS} + F_{LOGINFORM}$	85.61	6.26	89.46
$F_{URL} + F_{HYPERLINK} + F_{IDENTITY} + F_{CSS} + F_{LOGINFORM}$	99.39	1.25	99.09

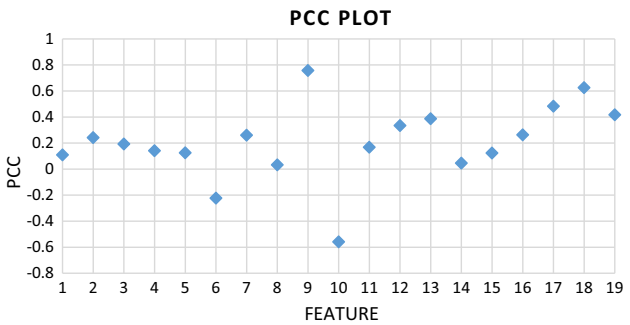

Fig. 11 PCC values of proposed feature set

6.5 Importance of proposed features

The proposed feature set is carefully projected to ensure the correct classification of legitimate and phishing websites. We experimentally determine the importance of proposed features using the Pearson product-moment Correlation Coefficient (PCC). PCC measures of the linear correlation between two variables by producing a value between + 1 and − 1. The higher absolute value indicates more dominant feature in classification result. For example − 0.71 PCC value has greater significant compared to + 0.34. We have calculated the PCC values of each feature. If the feature is relevant in correct classification, it produces the non-zero value. Figure 11 presents the plot of the PCC of each of the 19 features of the proposed approach with the label. From the figure, we analyse that PCC values for all features are non-zero, represent that every feature in proposed approach is important.

6.6 Runtime analysis

The response time is the time duration between inputting URL to producing output. When user input URL the approach

Table 7 Comparison of proposed approach with other standard approaches

Approach	TPR (%)	FPR (%)	ACC (%)	SEI	LI	TSI
Montaze et al. [18]	88	12	88	Yes	Yes	No
Xiang et al. [19]	92	0.4	95.8	No	No	No
Gowtham et al. [10]	98.24	1.71	98.25	No	Yes	No
Zhang et al. [22]	97	6	95	No	No	No
Tan et al. [23]	99.68	7.48	96.10	No	No	No
Chiew et al. [24]	99.8	13	93.4	No	Yes	Yes
El-Alfy et al. [20]	97.24	3.88	96.74	No	No	No
Zhang et al. [21]	98.64	0.53	99.04	Yes	Yes	No
Proposed approach	99.39	1.25	99.09	Yes	Yes	Yes

tries to fetch all defined features from the webpage URL and textual content as discussed in Sect. 4. Then, based on the extracted feature value, the trained classifier classifies the current URL and shows the result in the form of phishing or legitimate. We selected some random URLs from our dataset to check runtime analysis of our approach. The total response time of our approach in extraction and computing the feature vector, and producing the result is 2350 ± 3500 ms. This response time is relatively low and acceptable in real time environment. The classifier is not dependent on any third party services, and it does not need to wait for the results return by these services. Hence, our approach is fast as compared to other solutions.

6.7 Comparison with existing anti-phishing approaches

In this experiment, we have compared the proposed method with the benchmarked anti-phishing approaches. Table 7 present comparison that is based on TPR, FPR, accuracy (ACC), language independent solution (LI), Search engine independent solution (SEI), and third party services independent (TSI). As seen in the table, our work gives highest detection accuracy among the approaches discussed in the literature. The work of Tan et al. [23] and Chiew et al. [24] give the TPR higher than our approach. However, these two approaches produce very high FPR as compared to our approach. There is a trade-off between TRR and FPR so a good anti-phishing system should provide balanced TPR and FPR. Most of the previous approaches have used the search engine in feature set [10,19,20,22–24]. However, there are several drawbacks to search engine based feature. First, the new genuine sites do not appear in top search results, and this feature leads to the wrong prediction. Second, the search engines do not produce the accurate outcomes in non-English query search [34]. Therefore, our approach detects the websites missed by the search engine and produce low FPR.

7 Advantages of our approach

7.1 Language independency

The language barrier has been a bottleneck in most of the existing approaches. English is used as the textual language for only about 52.1% [35] of the websites. Therefore, language independence becomes a critical issue for any anti-phishing scheme. In our approach, most of the features are language independent (F1–F16, F19). Therefore, our system yields accurate results to a large extent. Most of the previous approaches have used the search engine and textual content in feature set [10,19–23]. The search engines do not produce the correct outcomes in non-English query search [34]. Therefore, our approach detects the websites missed by the search engine and produce low FPR. Furthermore, our testing dataset comprises various language websites, and experiment results on the dataset show the high detection accuracy of our approach.

7.2 Low response time

Low response time is a necessary and critical requirement for a phishing detection system that acts as another reason for choosing the client side features. Using proposed features for the detection of phishing webpages provides an average response time of around 2–6 s, which is quite low as compared to the existing alternatives such as other machine learning and visual similarity techniques, which gives a response time of around 10–13 s. Accessing the source code and producing result requires a negligible amount of time.

7.3 Third party independency

Several approaches use third-party dependent feature in the classification [10,18–23]. We have not chosen these features (such as DNS, blacklist/ whitelist, whois record, certifying authority, etc.) for the following reasons.

- Blacklist and whitelist contain a certain number URLs and does not cover all the websites.
- Some of the approaches verify the domain age from whois lookup. From the Phishtank dataset [29], we analyse that more than 30% of phishing webpages hosted on the compromised domain. If a website hosted on the compromised domain, domain age feature give the age of compromised domain and it leads approach in the wrong prediction.
- Certifying authority does not certify to each legitimate websites, and wrongly classify the new legitimate websites.
- DNS database may also be poisoned.

- Third party dependent features create additional network delay that can cause the high prediction time.

7.4 Compromised domain detection

Nowadays, cybercriminals host the phishing webpage on publicly available websites by exploiting vulnerabilities using various phishing tools. Phishing webpage on the compromised domain is a large scale deployment, and it provides numerous advantages to cybercriminals. A hacker does not require a web hosting server to deploy the phishing webpage. Our approach is based on source code, and it is not included features which predict false in the case of the compromised domain. Some of the features like the age of domain, certifying authority, WHOIS lookup, etc. provide incorrect information in case of the compromised website and produce false results. The search base techniques compare the domain name from the top 'T' search results to check the legitimacy. In compromised domain attack, the domain name is genuine, and most of the time it appears in top 'T' search results and these methods fail to detect the compromised domain.

Visual similarity techniques compare the visual appearance of the suspicious webpage with it's corresponding authenticate webpage are stored in a local database. These techniques only detect the compromised webpage only if its corresponding legitimate webpage present in the local database. Maintaining the large database that contains every legitimate webpage is a tough task. Though, our approach can identify the compromised domain up to a large extent because it does not depend on the local database for comparisons.

7.5 Client side application

The webpage is labelled the phishing webpage as phishing, legitimate based on the page source of the webpage, and it does not require any other resource. Most of the anti-phishing techniques resource such as OCR, DNS, SE, It makes our approach platform independent. Our approach uses best features, delivers competitive detection rate with minimum resources. This makes our approach portable and platform independent (i.e. performance is not affected if changes in the facilities supplied by third parties or protocol)

8 Conclusion and future work

This paper presented a novel approach for filtering phishing websites at client side where URL, hyperlink, CSS, login form, and identity features are used. The main contributions of this paper is the identification of various new client-side specific features that are previously not studied. Furthermore, we also created a new heuristic for each feature. We have constructed a dataset collected from various sources and included the variety of websites to validate the proposed solution. Our experimental results on dataset showed that the solution is very efficient as it has 99.39% true positive rate and only 1.25 % false positive rate. The proposed approach also has good accuracy as compared to other existing anti-phishing solutions.

The feature set of our phishing detection approach entirely depends on the URL and source code of the website, which can detect the webpages written in HTML code only. Therefore, the identification of non-HTML websites is the aim of our future scope. Nowadays, Mobile devices are more popular and seem to be a perfect target for malicious attacks like mobile phishing. Therefore, detecting the phishing websites in the mobile environment is a challenge for further research and development.

Acknowledgements This research work is being supported by Sir Visvesvaraya Young Faculty Research Fellowship Grant from Ministry of Electronics & Information Technology (MeitY), Government of India.

References

1. Jain, A. K., & Gupta, B. B. (2017). Detection of phishing attacks in financial and e-banking websites using link and visual similarity relation. *International Journal of Information and Computer Security, Inderscience*, 2017 (Forthcoming Articles).
2. Gupta, S., & Gupta, B. B. (2017). Detection, avoidance, and attack pattern mechanisms in modern web application vulnerabilities: Present and future challenges. *International Journal of Cloud Applications and Computing*, *7*(3), 1–43.
3. Almomani, A., et al. (2013). A survey of phishing email filtering techniques. *IEEE Communications Surveys & Tutorials*, *15.4*, 2070–2090.
4. Gupta, B. B., et al. (2017). Fighting against phishing attacks: State of the art and future challenges. *Neural Computing and Applications*, *28*(12), 3629–3654.
5. APWG Q4 2016 Report available at: http://docs.apwg.org/reports/apwg_trends_report_q4_2016.pdf. Last accessed on September 22, 2017.
6. Razorthorn phishing report, Available at : http://www.razorthorn.co.uk/wp-content/uploads/2017/01/Phishing-Stats-2016.pdf. Last accessed on September 22, 2017.
7. Purkait, S. (2015). Examining the effectiveness of phishing filters against DNS based phishing attacks. *Information & Computer Security*, *23*(3), 333–346.
8. Huang, Z., Liu, S., Mao, X., Chen, K., & Li, J. (2017). Insight of the protection for data security under selective opening attacks. *Information Sciences, Volumes*, *412–413*, 223–241.
9. Li, J., Chen, X., Huang, X., Tang, S., Xiang, Y., Hassan, M. M., et al. (2015). Secure distributed deduplication systems with improved reliability. *IEEE Transactions on Computers*, *64*(12), 3569–3579.
10. Gowtham, R., & Krishnamurthi, I. (2014). A comprehensive and efficacious architecture for detecting phishing webpages. *Computers & Security*, *40*, 23–37.
11. Aboudi, N. E., & Benhlima, L. (2017). Parallel and distributed population based feature selection framework for health monitoring.

International Journal of Cloud Applications and Computing, *7*(1), 57–71.
12. Sahoo, D., Liu, C., & Hoi, S. C. H. (2017). Malicious URL detection using machine learning: A survey. arXiv:1701.07179.
13. Arachchilage, N. A. G., Love, S., & Beznosov, K. (2016). Phishing threat avoidance behaviour: An empirical investigation. *Computers in Human Behavior*, *60*, 185–197.
14. Sheng, S., Magnien, B., Kumaraguru, P., Acquisti, A., Cranor, L. F., Hong, J. & Nunge, E. (2007). 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,* (pp. 88–99).
15. Jain, A. K., & Gupta, B. B. (2016). A novel approach to protect against phishing attacks at client side using auto-updated white-list. *EURASIP Journal of Information Security*, *2016*, 1–11.
16. Sheng, S., Wardman, B., Warner, G., Cranor, L. F., Hong, J., & Zhang, C. (2009). An empirical analysis of phishing blacklists. In *Proceedings of the 6th Conference on Email and Anti-Spam (CEAS'09)*.
17. Jain, A. K., & Gupta, B. B. (2017). Phishing detection: Analysis of visual similarity based approaches. *Security and Communication Networks*, 2017, Article ID 5421046, 20 pages, https://doi.org/10.1155/2017/5421046.
18. Montazer, G. A., & ArabYarmohammadi, S. (2015). Detection of phishing attacks in Iranian e-banking using a fuzzy-rough hybrid system. *Applied Soft Computing*, *35*, 482–492.
19. Xiang, G., Hong, J., Rose, C. P., & Cranor, L. (2011). Cantina+: A feature-rich machine learning framework for detecting phishing web sites. *ACM Transactions on Information and System Security (TISSEC)*, *14*(2), 21.
20. El-Alfy, E. S. M. (2017). Detection of phishing websites based on probabilistic neural networks and K-medoids clustering. *The Computer Journal.* https://doi.org/10.1093/comjnl/bxx035.
21. Zhang, W., Jiang, Q., Chen, L., & Li, C. (2017). Two-stage ELM for phishing Web pages detection using hybrid features. *World Wide Web*, *20*(4), 797–813.
22. Zhang, Y., Hong, J. I., & Cranor, L. F. (2007). Cantina: A content-based approach to detecting phishing web sites. In *Proceedings of the 16th international conference on world wide web*, (pp. 639–648).
23. Tan, C. L., Chiew, K. L., Wong, K., & Sze, S. N. (2016). PhishWHO: Phishing webpage detection via identity keywords extraction and target domain name finder. *Decision Support Systems*, *88*, 18–27.
24. Chiew, K. L., Chang, E. H., & Tiong, W. K. (2015). Utilisation of website logo for phishing detection. *Computers & Security*, *54*, 16–26.
25. APWG 2014 H2 Report Available at : https://docs.apwg.org/reports/apwg_trends_report_q3_2014.pdf. Last accessed on September 22, 2017.
26. Dataurization of URLs for a more effective phishing campaign. Available at: https://thehackerblog.com/dataurization-of-urls-for-a-more-effective-phishing-campaign/index.html. Last accessed on September 10, 2017.
27. Geng, G. G., Yang, X. T., Wang, W., & Meng, C. J. (2014). A Taxonomy of hyperlink hiding techniques. In *Asia-Pacific web conference*, (pp. 165–176).
28. Jain, A. K., & Gupta, B. B. (2017). Two-level authentication approach to protect from phishing attacks in real time. *Journal of Ambient Intelligence and Humanized Computing*. https://doi.org/10.1007/s12652-017-0616-z.
29. Verified Phishing URL, Available at : https://www.phishtank.com. Last accessed on September 22, 2017.
30. Phishing dataset available at : https://www.openphish.com/. Last accessed on September 27, 2017.
31. Alexa Most Popular sites, Available at : http://www.alexa.com/topsites. Last accessed on September 22, 2017.
32. List of online payment gateways. available at: http://research.omicsgroup.org/index.php/List_of_online_payment_service_providers. Last accessed on September 27, 2017.
33. Top banking websites in the world. Available at: https://www.similarweb.com/top-websites/category/finance/banking. Last accessed on September 27, 2017.
34. Chu, P., Komlodi, A., & Rózsa, G. (2015). Online search in english as a non-native language. *Proceedings of the Association for Information Science and Technology*, *52*(1), 1–9.
35. Percentages of websites using various content languages. Available at https://w3techs.com/technologies/overview/content_language/all. Last accessed on September 22, 2017.

Ankit Kumar Jain is presently working as Assistant Professor in National Institute of Technology, Kurukshetra, India. He received Master of technology from Indian Institute of Information Technology Allahabad (IIIT) India. Currently, he is pursuing Ph.D. in cyber security from National Institute of Technology, Kurukshetra. His general research interest is in the area of Information and Cyber security, Phishing Website Detection, Web security, Mobile Security, Online Social Network and Machine Learning. He has published many papers in reputed journals and conferences.

B. B. Gupta received Ph.D. degree from Indian Institute of Technology Roorkee, India in the area of Information and Cyber Security. In 2009, he was selected for Canadian Commonwealth Scholarship awarded by Government of Canada. He published more than 100 research papers (including 02 books and 14 book chapters) in International Journals and Conferences of high repute including IEEE, Elsevier, ACM, Springer, Wiley, Taylor & Francis, Inderscience, etc. He has visited several countries, i.e. Canada, Japan, Malaysia, China, Hong-Kong, etc to present his research work. His biography was selected and published in the 30th Edition of Marquis Who's Who in the World, 2012. Dr. Gupta also received Young Faculty research fellowship award from Ministry of Electronics and Information Technology, government of India in 2017. He is also working as principal investigator of various R&D projects. He is serving as associate editor of IEEE Access and Executive editor of IJITCA, Inderscience, respectively. He is also serving as reviewer for Journals of IEEE, Springer, Wiley, Taylor & Francis, etc. He is also serving as guest editor of various reputed Journals. He was also visiting researcher with Yamaguchi University, Japan in January 2015. At present, Dr. Gupta is working as Assistant Professor in the Department of Computer Engineering, National Institute of Technology Kurukshetra India. His research interest includes Information security, Cyber Security, Cloud Computing, Web security, Intrusion detection and Phishing.