Journal of Ambient Intelligence and Humanized Computing
https://doi.org/10.1007/s12652-018-0798-z

ORIGINAL RESEARCH

A machine learning based approach for phishing detection using hyperlinks information

Ankit Kumar Jain[1] · B. B. Gupta[1]

Received: 11 December 2017 / Accepted: 14 April 2018

Abstract

This paper presents a novel approach that can detect phishing attack by analysing the hyperlinks found in the HTML source code of the website. The proposed approach incorporates various new outstanding hyperlink specific features to detect phishing attack. The proposed approach has divided the hyperlink specific features into 12 different categories and used these features to train the machine learning algorithms. We have evaluated the performance of our proposed phishing detection approach on various classification algorithms using the phishing and non-phishing websites dataset. The proposed approach is an entirely client-side solution, and does not require any services from the third party. Moreover, the proposed approach is language independent and it can detect the website written in any textual language. Compared to other methods, the proposed approach has relatively high accuracy in detection of phishing websites as it achieved more than 98.4% accuracy on logistic regression classifier.

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

1 Introduction

1.1 Context

Today, phishing is one of the most serious Internet security threats. In this attack, the user enters his/her sensitive credential such as credit card details, password, etc. to the fake website which looks like a genuine one (Jain and Gupta 2017a). The online payment services, e-commerce, and social networks are the most affected sectors by this attack.

A phishing attack is performed by taking advantage of the visual resemblance between the fake and the authentic webpages (Jain and Gupta 2017b). The attacker creates a webpage that looks exactly similar to the legitimate webpage. The link of phishing webpage is then send to thousands of Internet users through emails and other means of communication. Usually, the fake email content shows some sense of fear, urgency or offer some price money and asks the user to take urgent action. E.g., the fake email will impel user to update their PIN to avoid debit/credit card suspension. When the user unknowingly updates the confidential credentials, the cyber criminals acquire user's details (Bhuiyan et al. 2016; Fan et al. 2016; Li et al. 2018). Phishing attack performed not only for gaining information; now it has become the number 1 delivery method for spreading other types of malicious software like ransomware. 90% of all active cyber-attacks start with a phishing emails (Phishingpro Report 2016). Phishing attack encompasses over a half of all cyber fraud that influences the Internet users. According to APWG report, 291,096 unique phishing websites were detected between January to June 2017 (APWG H1 2017 Report 2017). The per month attack growth has 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 2016). Figure 1 presents the growth of phishing attack from 2005 to 2016.

1.2 Problem definition

Recent developments in phishing detection have led to the growth of various new machine learning based techniques. In the machine learning based techniques, a classification algorithm is trained using some features, which can differentiate a phishing website from the legitimate one (Jain and Gupta 2016a). These features are extracted from various

✉ B. B. Gupta
gupta.brij@gmail.com

[1] National Institute of Technology, Kurukshetra, India

Published online: 26 April 2018

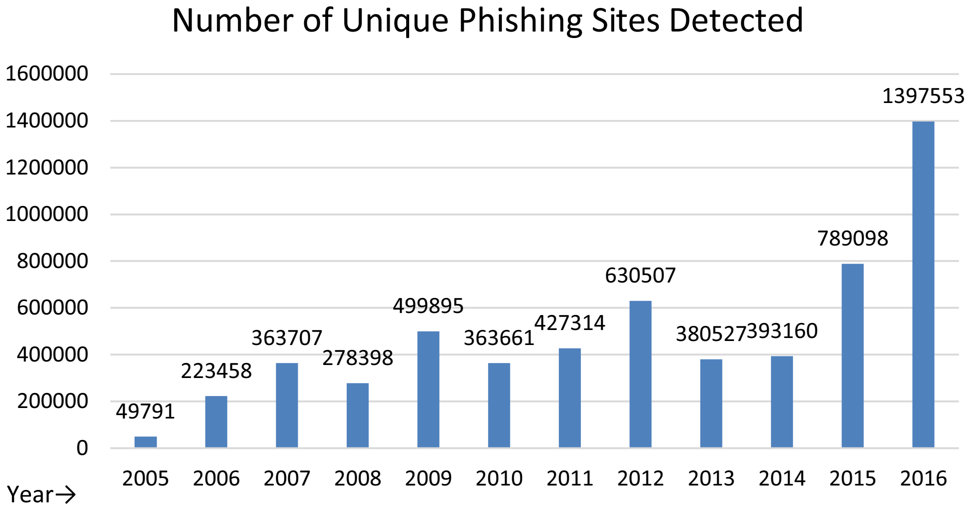


Fig. 1 Growth of phishing attack

sources like URL, search engine page source, website traffic, search engine, DNS, etc. The 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. Phishing websites are short-lived, and thousands of fake websites are generated every day. Therefore, there is requirement of real-time, fast and intelligent phishing detection solution.

1.3 Proposed solution

To solve above said problem, this paper presents a machine learning based novel anti-phishing approach that extracts the features from client side only. This paper presents an approach that can detect phishing websites using the hyperlink information present in the source code of the website. Proposed approach extract the hyperlinks from the page source and analyse them to detect whether the given website is phishing or not. We have divided the hyperlink features into 12 different categories namely total hyperlinks, no hyperlink, internal hyperlinks, external hyperlinks, internal error, external error, internal redirect, external redirect, null hyperlink, login form link, external/internal CSS, and external/internal favicon.

1.4 Contributions

The followings are the major contributions of our paper:

- Proposed approach extracts the outstanding features from the web browser only and does not depend on third party services (e.g. search engine, third party DNS, Certification Authority, etc). Therefore, it can be implemented at the client side and provide better privacy.
- Proposed approach can identify “zero-hour” phishing attack with high accuracy.
- Proposed approach can detect the phishing websites written in any textual language.
- We have also conducted a sensitivity analysis to predict the most powerful features in the detection of the phishing websites.

1.5 Experimental results

We have evaluated our proposed phishing detection approach on various classification algorithms and used the dataset of 2544 phishing and non-phishing websites. Experimental results show that logistic regression performs best in the detection of phishing websites. The proposed approach has the relatively high accuracy in detection of phishing websites as it achieved more than 98.39% true positive rate and only 1.52% false positive rate. Moreover, the accuracy, precision, and f1 score of our approach are 98.42, 98.80, and 98.59%, respectively. We have also explored the area under receiver operating characteristic (ROC) curve to find a better metric of precision. In our experiment, the area under the ROC curve for phishing website is 99.6, and it shows that our approach has high accuracy in classification of correct websites.

1.6 Outlines

The remainder of this paper is organized as follows. Section 2 presents the related work. Section 3 describes our proposed approach in detail. Section 4 presents the extractions of various features to train the machine learning algorithms. Section 5 presents the implementation detail, evaluation metrics, and experimental results. Finally, Sect. 6 concludes the paper and presents future work.

2 Related work

In this section, we present an overview of various anti-phishing solutions proposed in the literature. Phishing detection approaches are divided into two categories. First, based on user education, and another relies on the software. In the user education-based approaches, Internet users are educated to understand the characteristics of phishing attacks, which eventually leads them to appropriately identifying phishing and legitimate websites and emails (Kumaraguru et al. 2007). Software-based approaches are further classified into machine learning, blacklist, and visual similarity based approaches. Machine learning based approach trains a classification algorithm with some features and a website is declared as phishing, if the design of the websites matches with the predefined feature set. Visual similarity based approaches compare the visual appearance of the suspicious website and its corresponding legitimate website (Jain and Gupta 2017a). Blacklist matches the suspicious domain with some predefined phishing domains which are blacklisted. The negative aspect of the blacklist and visual similarity based schemes is that they usually do not cover newly launched (i.e. zero hour attack) phishing websites. Most of the phishing URLs in the blacklist are updated only after 12 h of phishing attack (Sheng et al. 2009). Therefore, machine learning based approaches are more effective in dealing with phishing attacks. Some of the machine learning based approaches given in the literature are explained below.

Pan and Ding (2006) proposed an anti-phishing method, which inspects the anomalies in the website. The approach extracts the anomalies from the various sources like URL, page title, cookies, login form, DNS records, SSL certificates, etc. The approach used SVM and achieved 88% true positive rate and 29% false positive rate. However, the proposed scheme used a dataset of only 379 websites Zhang et al. (2007) proposed a content specific approach CANTINA that can detect the phishing webpage by analysing text content and using TF-IDF algorithm. Top five keywords with highest TF-IDF are submitted into the search engine to extract the relevant domains. CANTINA also uses some heuristic like the special symbol in URL "@" (at sign), "–" (dash) symbol, dot count, domain age, etc. However, the accuracy of the scheme depends on TF-IDF algorithm and language used on the website. CANTINA achieved 6% of false positive rate, which is considered very high.(Abu-Nimeh et al. 2007) compared six machine learning algorithms for phishing e-mail detection namely Logical regression, Bayesian additive regression trees, SVM, RF, Neural network, and Regression trees. The result shows that there are no standard machine learning algorithms which can efficiently detect phishing attack. Garera et al. (2007) proposed a technique based on phishing URLs. The given approach discussed four different kinds of obfuscation techniques of phishing URLs. The approach uses logistic regression as a classifier. However, this technique cannot identify tiny URL based phishing websites. Mohammad et al. (2014) proposed an intelligent phishing detection system using the self-structuring neural network. Authors have collected 17 features from URL, source code and the third party to train the system using the neural network. Back propagation algorithm is used to adjust the weights of the network. Nevertheless, the design of network was a little bit complex. However, the training and testing set accuracy were 94.07 and 92.18, respectively on 1000 epochs. Aburrous et al. (2010) have used 27 features to construct a model based on fuzzy-logic for detection of phishing attack in banking websites. The authors used the features from the URL, page content (e.g. spelling error), SSL certificates, etc., to identify the phishing attack. This approach focused only on e-banking websites and did not discuss the detection results on another type of websites. Whittaker et al. (2010) published research on a large-scale classification of phishing websites, which uses the features from URL, page hosting, and page content. The TPR and FPR of the approach is 90 and 0.1%, respectively. Xiang et al. (2011) proposed CANTINA+, which takes 15 features from URL, HTML DOM (Document object model), third party services, search engine, and trained these features using support vector machine (SVM). Although, the performance of the scheme is affected by third party services like WHOIS lookup and search results. He et al. (2011) have used 12 features from the legitimate and phishing websites and achieved 97% true positive rate and 4% false positive rate. These features are taken from the meta tags, webpage content, URL, hyperlinks, TF-IDF, etc. Zhang et al. (2017) extract hybrid features from the URL, text content, and web and uses extreme learning machine (ELM) technique. The first phase of this technique built a textual content classifier to predict the label of textual content using ELM. In this, OCR software is used to extract text from images. The second phase combine text and another hybrid feature-based classifier. El-Alfy (2017) proposed an approach, which builds probabilistic neural networks (PNNs). The benefits of the PNN are fast training time, insensitivity to outliers and optimal generalisation. However, PNN may require high space and time with enormous increase of data. Therefore, the authors use K-medoids clustering with PNN to reduce the training instances. Montazera and ArabYarmohammadi (2015) proposed an anti-phishing method for the e-banking system of Iran. Authors identified 28 features utilized by the attackers to deceive the Irani banking websites. The detection accuracy is 88% on Iranian banking system. The approach is particular designed to identify the Iranian banking websites only while our approach can filter all kinds of phishing and legitimate websites.

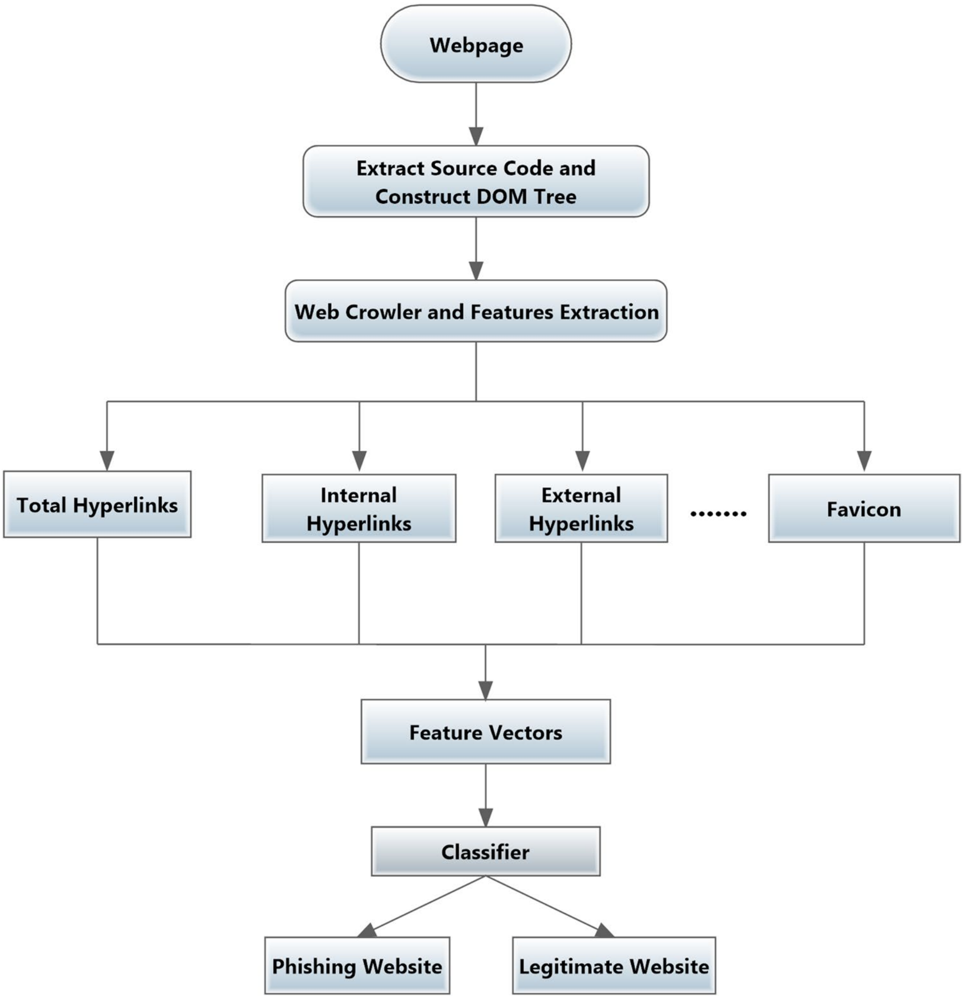
Usually, machine learning based techniques compare the features of the suspicious website with the predefined feature set (Wang et al. 2018; Lin et al. 2018). Therefore, the accuracy of the scheme depends on feature set and how accurately a defender chooses the features (Maio et al. 2017, 2018).

3 Proposed approach

Figure 2 presents the system architecture of the proposed approach. The selection of outstanding feature set is the major contribution of this paper. We have proposed six new features to improve the detecting accuracy of phishing webpages. Our proposed features identify the relation between the webpage content and the URL of the webpage. Our features are based on hyperlinks of the webpage. A website can be transformed into a Document Object Model (DOM) tree, and it is used to extract the hyperlink features as shown in Fig. 3. In our approach, we have gathered the website hyperlink features automatically using a web crawler as shown in Fig. 4. In hyperlink extraction process, the relative links are replaced by their hierarchically known absolute links. Our proposed approach takes the decision based on 12 features namely total hyperlink, no hyperlinks, internal hyperlinks, external hyperlinks, null hyperlinks, internal error, external error, internal redirect, external redirect, login form link, external/internal CSS and external/internal favicon.

In particular, features 2, 6, 7, 8, 9, 10 are novel and proposed by us. Features 1, 3, 4, 5, 11, 12 are taken from other approaches (Mohammad et al. 2014; Whittaker et al. 2010; Xiang et al. 2011; He et al. 2011). However, we fine-tuned these adopted features by performing various experiment to get the better results. After extraction of these features, a feature vector is created corresponding to each website. We have constructed the training and testing dataset by extraction of defined 12 features from the phishing and non-phishing websites. The training phase generates a binary classifier by applying the feature vectors of phishing and legitimate websites dataset. In the testing phase, the classifier determines whether a new site is a phishing site or not. A classifier takes the decision based on the learning from the labelled dataset. A binary classifier classify the websites

Fig. 2 System architecture



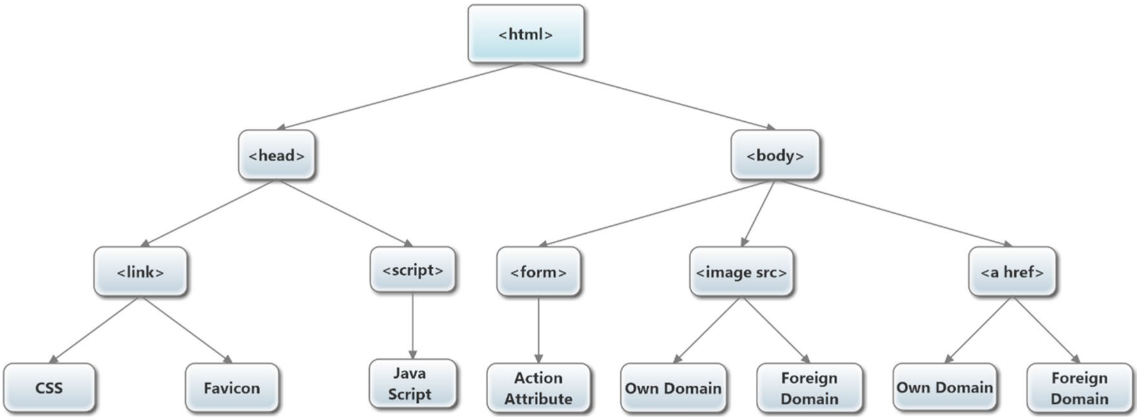


Fig. 3 HTML DOM tree

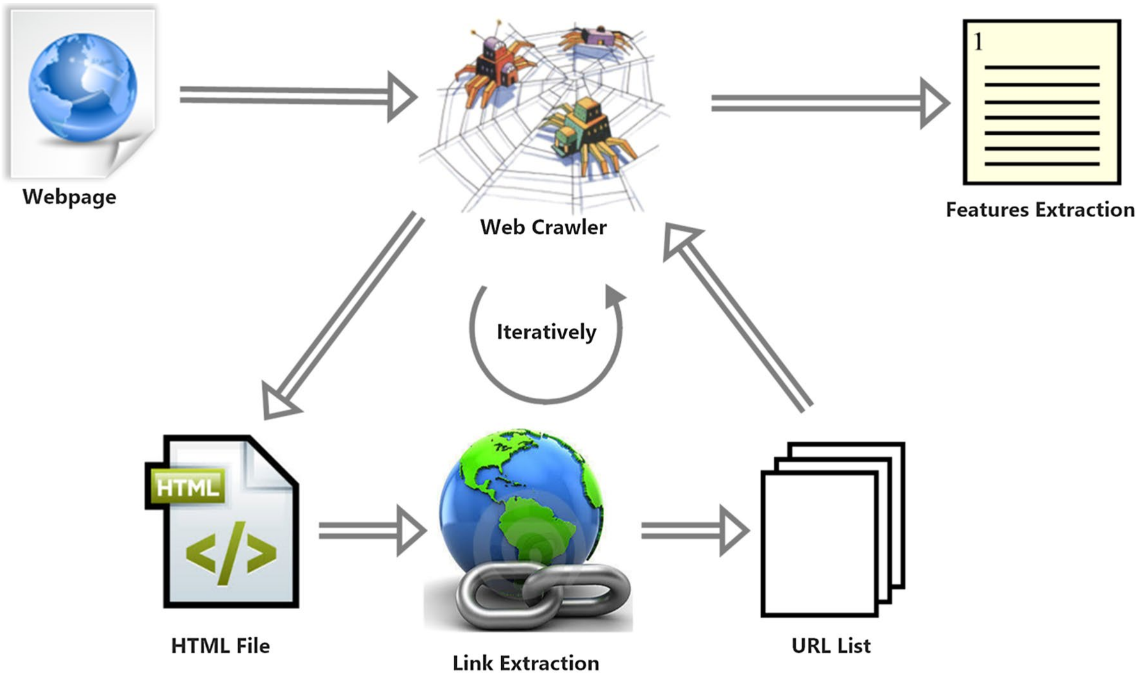


Fig. 4 Web crawler to extract features

into two possible categories namely phishing and legitimate. When a user requests for a new website, the crawler generates the feature values and the binary classifier that correctly identifies the given website.

4 Features extraction

The accuracy of a phishing detection scheme depends on the feature set which distinguish the phishing and legitimate website. Based on the given limitation of individual and third party dependent approaches in the Sect. 2, we have adopted the hyperlink specific features in the proposed approach. These features are extracted from the client side and not dependent on any third party services. In this, F = {F1, F2,…, F12} is defined as the feature vector corresponding to each feature. Some features produce the value in the form of 1 and 0, where 1 indicates for phishing and 0 indicate for legitimate. We will discuss all these features in the following subsections.

4.1 Total and no hyperlink feature (F1 and F2)

Phishing websites are small as compared to legitimate websites. A legitimate website usually contains many webpages. However, a phishing website consists of very limited webpages, sometimes only one or two. Moreover, sometimes the phishing website does not provide any hyperlink because the attackers use the hyperlink hidden techniques (Geng et al. 2014). Also, attacker also uses server-side scripting and frameset to hide the source code of webpage (Jain and Gupta 2016b). From our experiments, we analyse that if a website is genuine, we can extract at least one hyperlink from the source code. Therefore, if the approach does not extract any link from the source code, the website is considered as a phishing website (feature 2). Total hyperlinks are calculated by adding href, link, and src tags. Taking the no hyperlink as a different feature increases the true positive rate of the proposed approach.

$$F1 = Total\ hyperlink\ present\ in\ a\ website \tag{1}$$

$$F2 = \begin{cases} 0 & if\ F1 > 0 \\ 1 & if\ F1 = 0 \end{cases}. \tag{2}$$

4.2 Internal and external hyperlinks (F3 and F4)

The internal and external hyperlink means hyperlink contains the same and different base domain respectively. The phishing website usually copies the source code from its targeted official website, and it may have many hyperlinks that point to the targeted website. In the legitimate website, most of the hyperlinks contain same base domain while in phishing website many hyperlinks may contain the domain of the corresponding legitimate website. In our experiment, we found that out of 1428 phishing websites, 593 websites include direct hyperlinks to their official website. To set the internal hyperlink feature, we calculate the ratio of internal hyperlinks to the total links present in a website (Eq. 3) and if the ratio is less than 0.5 then set as 1 else 0 as given in Eq. 4. Furthermore, to establish the external hyperlink feature, we calculate the ratio of external hyperlinks to the total available links (Eq. 5) and if the ratio is greater than 0.5 then set as 1 else 0 as represented in Eq. 6.

$$Ratio_{internal} = \begin{cases} \frac{H_{Internal}}{H_{total}}\ if\ H_{total} > 0 \\ 0\ if\ H_{total} = 0 \end{cases} \tag{3}$$

$$F3 = \begin{cases} 0 & Ratio_{Internal} \geq 0.5 \\ 1 & Ratio_{Internal} < 0.5 \end{cases} \tag{4}$$

$$Ratio_{External} = \begin{cases} \frac{H_{External}}{H_{total}} & if\ H_{total} > 0 \\ 0 & if\ H_{total} = 0 \end{cases} \tag{5}$$

$$F4 = \begin{cases} 0 & Ratio_{External} \leq 0.5 \\ 1 & Ratio_{External} > 0.5 \end{cases}, \tag{6}$$

where $H_{internal}$, $H_{External}$, and H_{total} are the number of internal, external and total hyperlinks in a website. $Ratio_{internal}$ and $Ratio_{External}$ are the ratios of internal and external hyperlinks to total available hyperlinks.

4.3 Null hyperlink (F5)

In the null hyperlink, the href attribute of anchor tag does not contain any URL. When the user clicks on the null link, it returns on the same page again. A legitimate website consists of many webpages, therefore to behave like the legitimate website, phisher places no values in hyperlinks, and the links appear active on the website. Phisher also exploits the vulnerability of web browser with the help of empty links (Jain and Gupta 2016b). The HTML coding used for designing null hyperlinks are < a href="#">, <a href="#content">, <a href="JavaScript ::void(0)">. To set the null hyperlink feature, we calculate the ratio of null hyperlinks to the total number of links present in a website and if the ratio is greater than 0.34 then set as 1 else 0. Following equations are used to calculate null hyperlink feature.

$$Ratio_{Null} = \begin{cases} \frac{H_{Null}}{H_{total}} & if\ H_{total} > 0 \\ 0 & if\ H_{total} = 0 \end{cases} \tag{7}$$

$$F5 = \begin{cases} 0 & Ratio_{Null} \leq 0.34 \\ 1 & Ratio_{Null} > 0.34 \end{cases}, \tag{8}$$

where H_{Null} and H_{total} are the numbers of null and total hyperlinks in a website.$Ratio_{Null}$ is the ratio of null hyperlinks to total hyperlinks present in the website.

4.4 Internal/external CSS (F6)

Cascading Style Sheets (CSS) is a language used for depicting the formatting of a document and setting the visual appearance of a website written in the HTML, XHTML, and XML. An attacker always tries to mimic legitimate website and keep the same design of the phishing website as that of targeted website to attract potential victim. Formally, a CSS contains a list of rules, which can associate a group of selectors, properties, and values to a set of declarations. CSS of any website is either included with external CSS file or within the HTML tags itself. External CSS files are associated with some HTML website by using the tag <link>. To extract external CSS file, we try to find a tag with other values such as <link… rel = 'stylesheet''… href = 'URL of CSS file'…>. However, during the experiment, we found that in the case of the phishing website, it uses only one CSS file or internal style and this external CSS file contain the link of targeted legitimate website. Whereas, several legitimate websites use more than one CSS file or internal style. We develop an algorithm to find the suspicious CSS in a website as shown in Fig. 5.

4.5 Internal and external redirection (F7 and F8)

Redirection indicates whether a website redirects to some other place. When a browser tries to open an URL, which has been redirected, a webpage with a different URL opens. Sometimes URL redirection confuses users about which website they are surfing. Moreover, redirection may also take the user to a website which is bogus. In a phishing website, there may be some links that redirect to the corresponding legitimate domain and sometimes the fake website can also be redirected to legitimate one after filling the login form. In this paper, we consider only response code 301 and 302 for URL redirection. We select both, internal and external URL redirection in our feature set. In this feature, we calculate the ratio of hyperlinks which are redirecting. Internal Redirection (F7) is calculated by dividing total internally redirected hyperlinks to the total internal hyperlinks. External Redirection (F8) is calculated by dividing the total external redirected hyperlinks to the total external hyperlinks.

$$F7 = \begin{cases} \frac{H_{i\text{-}redirect}}{H_{Internal}} & if\ H_{Internal} > 0 \\ 0 & if\ H_{Internal} = 0 \end{cases} \tag{9}$$

$$F8 = \begin{cases} \frac{H_{e\text{-}redirect}}{H_{External}} & if\ H_{External} > 0 \\ 0 & if\ H_{External} = 0 \end{cases}, \tag{10}$$

where $H_{i\text{-}redirect}$, $H_{e\text{-}redirect}$, $H_{Internal}$ and $H_{External}$ are the number of internal redirect, external redirect, total internal and total external hyperlinks present in the website.

4.6 Internal and external error (F9 and F10)

In this heuristic, we check the errors in hyperlinks of the website. Error "404 not found" occurs when a user request for an URL and server is not able to determine the requested URL. Phisher also adds some hyperlinks in the fake page which do not exists. "404 not found" error is generated when a user attempts to access dead or broken link. We consider the 403 and 404 response code of hyperlinks. The proposed approach uses web crawler to fetch the response code of each hyperlink. Internal error (F9) is calculated by dividing the total internal error hyperlinks to the total internal hyperlinks. External error (F10) is calculated by dividing the total external error hyperlinks to the total external hyperlinks.

$$F9 = \begin{cases} \frac{H_{i\text{-}error}}{H_{Internal}} & if\ H_{Internal} > 0 \\ 0 & if\ H_{Internal} = 0 \end{cases}, \tag{11}$$

$$F10 = \begin{cases} \frac{H_{e\text{-}error}}{H_{External}} & if\ H_{External} > 0 \\ 0 & if\ H_{External} = 0 \end{cases}, \tag{12}$$

where $H_{i\text{-}error}$, $H_{e\text{-}error}$, $H_{Internal}$ and $H_{External}$ are the number of internal error, external error, total internal and total external hyperlinks in a website.

4.7 Login form link (F11)

Phishing websites usually contain login form to steal credentials of the Internet users. The personal information of the user is transferred to the attacker after filling the form on a fake website. The login form of the phishing websites appears in the same manner as in the legitimate website. In this feature, we check the authenticity of login forms. In the legitimate website, the action field typically contains the URL of the current website. However, Attackers either use the different domain (other than visited domain), null (hyperlink in footer section) or a PHP file in the form action field of phishing websites (Jain and Gupta 2017c). PHP file contains a script which saves the input data (e.g. user id or password) in a text file saved on the attacker's computer. The PHP file usually named as index.php, login.php, etc. We construct an algorithm to check the authenticity of the login form as shown in Fig. 6. The input of algorithm is the URL of the suspicious website and output results as {0,1}, 0 for legitimate and 1 for phishing. If hyperlink present in the action field is relative, then system replaces it by the absolute link.

Fig. 5 Algorithm to detect suspicious CSS

```
Algorithm to detect suspicious CSS
Input: URL of suspicious website
Output: F6 ϵ {0, 1}, 0- Legitimate, 1- Phishing
Start
Step1: Extract all the CSS file of the website
Step 2: If the CSS is internal then set F6 = 0
Step 3: If the CSS is external and base domain is equal to current domain then set F6 = 0
else set F6 = 1
End
```

Fig. 6 Algorithm to find suspicious login form

```
Algorithm to find suspicious login form
Input:  URL of suspicious website
Output: F11 ϵ {0, 1}, 0- Legitimate, 1- Phishing
Start
Step1: Extract the action field value of each form
Step 2: If the value of action field is blank, # or, javascript:void(0)) then set F11 = 1
Step 3: If the value of action field is in the form of "filename.php" then set F11 =1
Step 4: If action field contain foreign domain then set  F11 =1 otherwise set F11 = 0
End
```

```
a.   action= " "
b.   action= "#"
c.   action= "javascript:void(0)"
d.   action= "filename.php"// e.g. filename is the name of php file
```

4.8 Internal/external favicon (F12)

Favicon is an image icon associated with the particular website. An attacker may copy the favicon of targeted website. Favicon is an .ico file linked to an URL, and found in link tag of the DOM tree. If the favicon shown in the address bar is other than the current website, it is considered as a phishing attempt. This feature contains the two values, 0 (legitimate) and 1(phishing). If the favicon belongs to the same domain, then make this features 0 else 1. Following HTML coding is used in designing of favicon.

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

4.9 Unused features

We have mentioned the usefulness and importance of the proposed features used in our approach. However, there are several other features, which are used by various existing approaches and are not appropriate for the proposed approach due to the following reasons:

1. *Search engine based features* Various approaches have used search engine based features (Zhang et al. 2007; He et al. 2011; Varshney et al. 2016). These approaches verify the authenticity of the webpage by searching the URL, domain name, title keywords, most frequent word, website logo, etc. in the popular search engine (Google, Yahoo, Bing, etc). In Zhang et al. (2007) and He et al.(2011) the presented approaches are based on the TF-IDF algorithm. In this, if the algorithm extracts the wrong keywords, then the results are defective. Moreover, the rank provided to a website determines its position in the list of the searched links. Newly published websites and alienated blogs which have no connection to the mainstream websites are pushed back in the search results. Furthermore, different search engines allow the search string to be specified in the desired way so that it may give the particular result user is looking for. e.g. Google has special query pattern to search exact phrases, exclude a word, search a specific domain, search specifying a location, etc. If the search string that user enters, matches a special case, then the search results could be irrelevant and in some cases, the search engine may fail to produce results for such queries.

2. *Third Party dependent features* We have not chosen features which are dependent on third party services such as DNS, blacklist/whitelist, WHOIS record, certifying authority, search engine, etc. Third party dependent features make our approach dependent on the third party and create additional network delay which can result in high prediction time. Moreover, DNS database may also be poisoned.

3. *URL based features* Various approaches used URL features (Whittaker et al. 2010; Xiang et al. 2011; He et al. 2011; Zhang et al. 2017) (e.g. number of dots, Presence of special "@", "#", "–" symbol, URL length, Suspicious words in URL, Position of Top-Level Domain, http count,

Brand name in URL, IP address, etc.). Nowadays phisher are changing their way to perform attacks, and these techniques cannot detect tiny URL, and Data URI based phishing websites which are considered as popular one.

5 System design, implementation and results

This section presents the construction of the dataset, evaluation measures, implementation details, and results outcomes of proposed anti-phishing approach. The detection of phishing websites is a binary classification problem where various features are used to train the classifier. Moreover, this trained classifier is used to classify the new website as phishing and legitimate category.

5.1 Training dataset

We have collected proposed features from 2544 different phishing and legitimate websites. Table 1 presents the number of instances and the sources of phishing and legitimate datasets. The life of phishing websites is very short. Therefore, we crawled when they are alive. We have used a wide range of websites in our dataset like blogs, social media networking, payment gateways, banking, etc. Table 2 presents a sample of phishing and legitimate websites datasets. The Alexa dataset includes 500 high ranked website (Rank 1–500) and 500 low ranked websites (Rank 999,500–1,000,000). Some features contain the feature values like "Legitimate" and "Phishing" in this case; we replaced these values with numerical value 0 and 1, respectively. The feature vector is having identical values removed from the dataset. Our solution is the language independent. Therefore, we have also considered the website of different languages to test our approach.

5.2 Evolution metrics

We use true positive rate, false positive rate, true negative rate, false negative rate, f1 score, accuracy, precision, and recall to evaluate the performance of the proposed approach. Table 3 shows the results of true and false possible classification. The performance of our approach is evaluated in the following manner:

True positive rate (TPR): measures the rate of phishing websites classified as phishing out of entire phishing websites.

False positive rate (FPR): measures the rate of legitimate websites classified as phishing out of total legitimate websites.

False negative rate (FNR): measures the rate of phishing websites classified as legitimate out of total phishing websites.

True negative rate (TNR): measures the rate of legitimate websites classified as legitimate out of total legitimate websites.

Accuracy (A): it measures the overall rate of correct prediction.

Precision: it measures the rate of instances correctly detected as phishing with respect to all instances detected as phishing.

f1 Score: It is the harmonic mean of Precision and Recall.

5.3 Implementation tool

The experiments were conducted on Pentium i5 computer with 2.4 GHz processor. The proposed approach is implemented in Java platform standard edition 7. Jsoup (Jsoup HTML parser 2018) is used to extract hyperlinks from website and Guava library (Guava libraries, Google Inc 2018) is used to obtain the base domains of the hyperlinks. We have used WEKA to judge the performance of our proposed approach on various classifiers. Weka is Java open source code which means "Waikato Environment for Knowledge Analysis". Numerous data mining and machine learning algorithms are implemented in WEKA. It contains the rich collection of modelling, clustering, classification, regression and data pre-processing techniques. The experiments on various classification algorithm namely SMO, Naive Bayes, Random forest, Support Vector Machine (SVM), Adaboost, Neural Networks, C4.5, and Logistic Regression have been performed.

5.4 Training with classifier

We have used the logistic regression (LR) as a binary classifier because it gives the better accuracy as compared to other classifiers. Logistic regression is a classification

Table 1 Training and testing dataset

S. no.	Database	Number of instances	Phishing/legitimate
1	Phishtank dataset (2018)	1428	Phishing
2	Alexa top websites (2018)	1000	Legitimate
3	Stuffgate Free Online Website Analyzer (2018)	50	Legitimate
4	List of online payment service providers (2018)	66	Legitimate

Table 2 Sample datasets

S. no.	Total hyperlink	No hyperlink	Internal hyperlink	External hyperlink	Null links	Internal/ external CSS	Internal redirection	External redirection	Internal error	External error	Login form link	Internal/ external favicon
Legitimate websites												
1	537	0	484	52	1	1	47	14	2	1	0	0
2	54	0	38	14	2	0	3	5	0	0	0	0
3	19	0	19	0	0	0	0	1	0	0	0	0
4	635	0	618	12	5	0	20	6	1	0	1	0
Phishing websites												
1	27	0	15	11	0	1	1	8	0	0	0	1
2	13	0	2	0	11	0	0	0	0	0	1	0
3	38	0	28	9	0	1	0	2	27	0	1	1
4	115	0	2	100	0	1	0	11	0	84	0	0

Table 3 Confusion matrix

	True results	
	1 (phishing)	0 (legitimate)
Prediction		
1 (phishing)	True positive rate	False positive rate
0 (legitimate)	False negative rate	True negative rate

Table 4 Coefficient and odd ratio of feature set

Feature	Coefficients	Odd ratio
Total hyperlinks	−0.017236	0.982911
No hyperlink	23.231230	1.23E+10
Internal hyperlink	2.327730	10.254638
External hyperlink	1.914151	6.781177
Null hyperlink	20.263021	6.31E+08
CSS	2.515211	12.369218
Internal redirect	0.149838	1.161646
External redirect	0.826801	2.285995
Internal error	2.089872	8.083884
External error	0.222953	1.249762
Login form	5.445638	231.745272
Favicon	2.910151	18.359569
Intercept	−3.714364	0.024371

technique used to predict a binary dependent variable with the set of independent variables. Logistic regression estimates the occurring probability of the dependent variable. In our approach, the dependent variable is used to decide whether a website is phishing, and the independent variables are the proposed feature set which were explained in Sect. 4. A labelled training dataset is used to train the logistic regression classifier. Our labelled data set consists of 2544 websites in which 1428 are phishing, and 1116 are legitimate, as described in Sect. 5.1. Phishing websites are defined under the positive (true, 1) class, and legitimate websites are described under the negative (false, 0) class.

Table 4 presents the coefficient and odd ratio corresponding to each feature. The odd ratio is the ratio of the odd of an event in the positive class (phishing) to the odd of it happening in the negative class (non-phishing). Odd ratio 1 means a feature is equally useful in identification of both categories (phishing and non-phishing). If the value of the odd ratio is greater than 1, then the related feature is more valuable in recognizing the positive class. Higher odd ratio means most helpful feature in determining the phishing websites. From Table 4 we can analyse that the feature 2, 3, 4, 5, 6, 9, 11 and 12 have the very high odd ratio, and identify as the most useful features in our proposed feature set. However, these eight features are not sufficient to detect all kind of phishing websites. Therefore, we have also used other features to improve

Table 5 Results of our proposed approach

Total dataset	True positive rate/recall	False positive rate	True negative rate	False negative rate	Accuracy	Precision	f1 Score
2544	98.39%	1.52%	98.48%	1.61%	98.42%	98.80%	98.59%

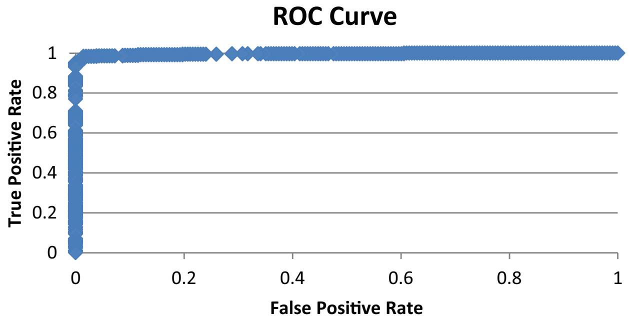


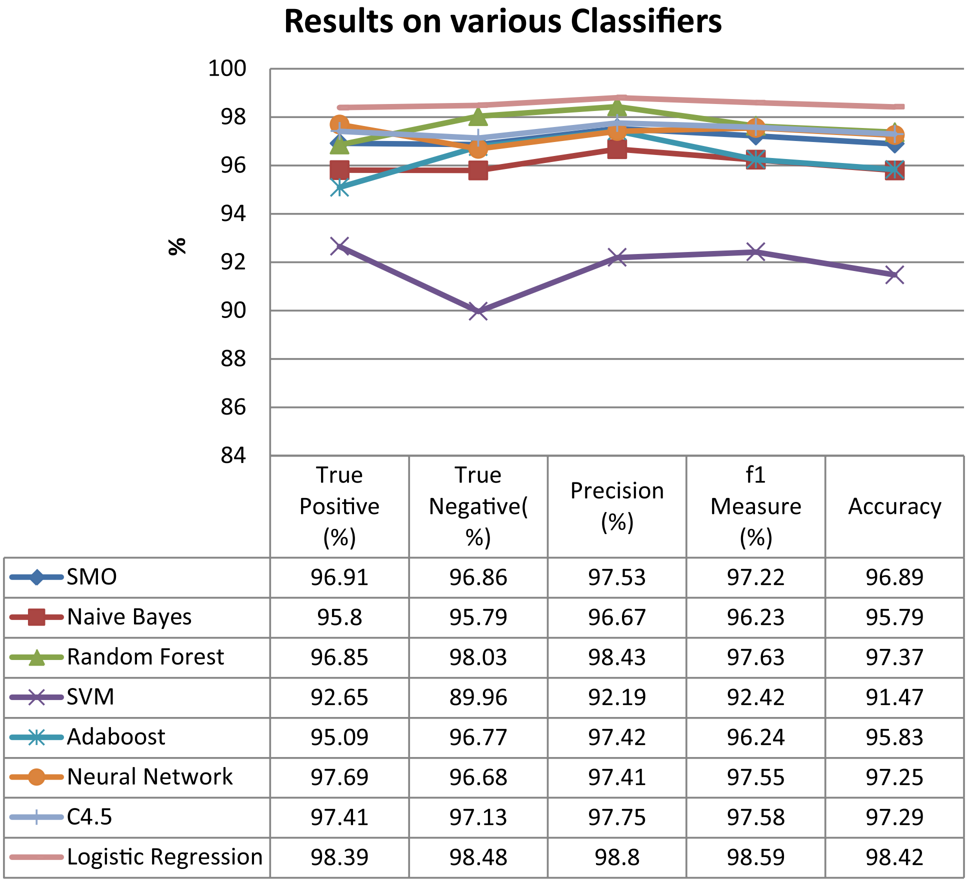
Fig. 7 ROC curve of logistic regression classifier

the accuracy of the proposed approach. We have evaluated our dataset with tenfold cross validation. It uses 90% of data for training purpose, and 10% data for testing purpose. The TPR of the approach is 98.39%, and FPR is 1.52%. In other words, 98.39% of phishing websites are caught by our approach, and 1.61% (false negative) will be missed. The accuracy, precision, and f1 score of our approach are 98.42, 98.80, and 98.59%, respectively as presented in Table 5. 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 99.6 as shown in Fig. 7, and it shows that our approach has high accuracy in classification of correct websites. Results of our approach on different classifiers are presented in Fig. 8. The probability of a website is phishing in logistic regression shown by the following equation.

$$p = \frac{e^{b_0+b_1x_1+b_2x_2+\ldots+b_nx_n}}{1+e^{b_0+b_1x_1+b_2x_2+\ldots+b_nx_n}} = \frac{1}{1+e^{-(b_0+b_1x_1+b_2x_2+\ldots+b_nx_n)}} \tag{13}$$

In the Eq. 13, 'p' is the probability of occurring the event. x_1, x_2,…x_n are the values corresponding to each

Fig. 8 Evaluation results of our approach on the various classifiers



	True Positive (%)	True Negative(%)	Precision (%)	f1 Measure (%)	Accuracy
SMO	96.91	96.86	97.53	97.22	96.89
Naive Bayes	95.8	95.79	96.67	96.23	95.79
Random Forest	96.85	98.03	98.43	97.63	97.37
SVM	92.65	89.96	92.19	92.42	91.47
Adaboost	95.09	96.77	97.42	96.24	95.83
Neural Network	97.69	96.68	97.41	97.55	97.25
C4.5	97.41	97.13	97.75	97.58	97.29
Logistic Regression	98.39	98.48	98.8	98.59	98.42

feature and b_0, b_1,… . b_n are the coefficient corresponding to each feature. In our experiment, we set the classification cut-off at 0.5, since at 0.5 system get the maximum accuracy. If the score of the website is less than 0.5, then website is more likely to be a genuine website, and if it is greater than 0.5, then the website considers as a phishing website.

In this paper, our primary objective is to design an approach which has high TPR and TNR and, low FPR and FNR. If classification cut-off increases, then the FPR decreases but at the same time TPR also decreases. Furthermore, if we reduce the classification cut-off then TPR increases but FPR increases as well. A good phishing detection approach requires both high TPR and low FNR.

5.5 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. We need to obtain all hyperlinks from the webpage to compute features. 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(n), where n is source code length of the webpage. A single pattern matching algorithm (i.e. Knuth–Morris–Pratt algorithm) used to match the domain name of hyperlinks with the URL of webpage. Moreover, the proposed method is not dependent on any third party services, and hence it does not need to wait for the results return by these services.

5.6 Comparison with other machine learning based phishing detection method

This experiment compares our proposed method with the existing machine learning based approaches given in the literature. The comparison is based on TPR, FPR, accuracy, third party independent, language independent solution, zero hour detection, and search engine independent solution. Table 6 presents the result comparison of our approach with other previous phishing detection methods. The search engine based techniques believe that legitimate site appears in the top results of search engine. Although only popular sites appear in the top search results. Therefore, we have not considered search engine based feature. Moreover, most of the previous methods have used the dataset of famous sites while we have also considered the low ranked websites. Our approach gives FPR of 1.52% for the legitimate websites. Only the work of Garera et al. (2007), Whittaker et al. (2010), Xiang et al. (2011) gives a FPR lower than our approach but their TPR and overall detection accuracy is very low as compared to our approach. The TPR of Garera et al. (2007) is 88%, i.e. this scheme fails to detect 12% of phishing websites, which is very high. Another important issue of comparison is the language used in the website. Only 52.1% of the website are used English language (Usage of content languages for websites 2017). Many approaches (Garera et al. 2007; Aburrous et al. 2010) are dependent on the textual language of the website. The proposed approach used the hyperlink specific features because it is very efficient and language independent. Some of the approaches (Aburrous et al. 2010; Montazera and ArabYarmohammadi 2015) cannot detect the zero hour attack because these approaches are designed to detect special kind of phishing website. On the other hand, our approach can detect all kind of phishing websites. Moreover, most of the approaches use

Table 6 Comparison between various anti-phishing approaches based on results obtained

Approach	TPR (%)	FPR (%)	Accuracy (%)	Search engine independent	Language independent	Zero hour detection	Third party independent
(Pan and Ding 2006)	88	29	84	Yes	No	Yes	No
(Zhang et al. 2007)	97	6	95	No	No	Yes	No
(Garera et al. 2007)	88	0.7	97.3	Yes	Yes	Yes	No
(Aburrous et al. 2010)	86.38	13.6	88.4	Yes	Yes	No	No
(Whittaker et al. 2010)	91.85	0.0001	95.92	Yes	No	Yes	No
(Xiang et al. 2011)	92	0.4	95.8	No	No	Yes	No
(He et al. 2011)	97	4	96.5	No	No	Yes	No
(Zhang et al. 2017)	97	2	97.50	Yes	Yes	Yes	No
(El-Alfy 2017)	97.89	4.59	96.74	No	Yes	Yes	No
(Montazera and ArabYarmohammadi 2015)	88	12	88	Yes	No	No	No
Our method	98.39	1.52	98.42	Yes	Yes	Yes	Yes

the third party features, e.g. WHOIS lookup, DNS, certifying authority, etc. and the accuracy also depends on the result returned by the third party and it is also time consuming process. Therefore, we have not considered the third party dependent features in our proposed approach.

6 Discussion

With the rapid growth of e-commerce, e-banking, and social networking, the phishing attack is also growing day by day. This results in enormous amount financial losses to industries and Internet users. Therefore, there is need of effective solution to detect phishing attack which has high accuracy and less response time. We proposed a novel anti-phishing approach, which includes various unique hyperlink specific features that have never been considered. We implemented these hyperlink specific features on different machine learning algorithms, and find that logistic regression achieved the best performance. There are certain limitations of our proposed approach. The feature set of our phishing detection approach completely depends on the source code of the website. We believe that attacker use the source code from targeted legitimate website to construct the phishing website and they modify the login form handler to steal user's credential. If a cybercriminal may alter all the page resource references (i.e. images, CSS, Favicon, JavaScript, etc.), then our approach predicts false result too. Also, if the attacker uses embedded objects (images, JavaScript, Flash, ActiveX, etc.) instead of DOM to hide the HTML coding from the phishing detection approaches, then our technique may incorrectly classify the phishing websites.

7 Conclusion and future work

In this paper, we have recognized various new features for identifying phishing websites. These features are based on hyperlink information given in source code of the website. We have used these features to train logistic regression classifier, which achieved high accuracy in detection of phishing and legitimate websites. One of the major contributions of this paper is the selection of hyperlink specific features which are extracted from client side and these features do not depend on any third party services. Moreover, these features are sufficient enough to detect a website written in any language. The experimental results showed that proposed method is very efficient in classification of phishing websites as it has 98.39% true positive rate and 98.42% overall accuracy. The accuracy of our approach may be improved by adding certain more features. Our proposed phishing detection approach completely depends on the source code of the website. Adding certain more features may increase the classification accuracy. However, extracting other features from the third party will increase the running time complexity of the scheme. In future work our aim to design a system which can also detect non-HTML websites with high accuracy. 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.

References

Abu-Nimeh S, Nappa D, Wang X, Nair S (2007). A comparison of machine learning techniques for phishing detection. In: Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit, Pittsburgh, pp 60–69

Aburrous M, Hossain MA, Thabatah F, Dahal K (2010) Intelligent phishing detection system for e-banking using fuzzy data mining. Expert Syst Appl 37(12):7913–7921

Alexa top websites (2018) http://www.alexa.com/topsites. Retrieved 22 Aug 2017

APWG H1 2017 Report (2017) http://docs.apwg.org/reports/apwg_trends_report_h1_2017.pdf. Retrieved 25 March 2018

Bhuiyan MZA, Wu J, Wang G, Cao J (2016) Sensing and decision making in cyber-physical systems: the case of structural event monitoring. IEEE Trans Ind Inform 12(6):2103–2114

El-Alfy E-SM (2017) Detection of phishing websites based on probabilistic neural networks and K-Medoids clustering. Comput J. https://doi.org/10.1093/comjnl/bxx035

Fan L, Lei X, Yang N, Duong TQ, Karagiannidis GK (2016) Secure multiple amplify-and forward relaying with cochannel interference. IEEE J Select Topics Signal Process 10(8):1494–1505

Garera S, Provos N, Chew M, Rubin AD (2007) A framework for detection and measurement of phishing attacks. In: Proceedings of the 2007 ACM workshop on recurring malcode, Alexandria, pp 1–8

Geng G-G, Yang X-T, Wang W, Meng C-J (2014) A taxonomy of hyperlink hiding techniques. In: Asia-Pacific web conference, vol 8709, Lecture Notes in Computer Science. Springer, Suzhou, pp 165–176

Guava libraries, Google Inc. (2018) https://github.com/google/guava . Retrieved 18 Jan 2018

He M, Horng SJ, Fan P, Khan MK, Run RS, Lai JL, Sutanto A (2011) An efficient phishing webpage detector. Expert Syst Appl 38(10):12018–12027

Jain AK, Gupta BB (2016a) Comparative analysis of features based machine learning approaches for phishing detection. In: Proceedings of 3rd international conference on computing for sustainable global development (INDIACom). IEEE, New Delhi, pp 2125–2130

Jain AK, Gupta BB (2016b) A novel approach to protect against phishing attacks at client side using auto-updated white-list. EURASIP J Inf Secur 2016(9)

Jain AK, Gupta BB (2017a) Phishing detection: analysis of visual similarity based approaches. Secur Commun Netw. https://doi.org/10.1155/2017/5421046

Jain AK, Gupta BB (2017b) Two-level authentication approach to protect from phishing attacks in real time. J Ambient Intell Humaniz Comput, 1–14

Jain AK, Gupta BB (2017c). Towards detection of phishing websites on client-side using machine learning based approach. Telecommun Syst, 1–14. https://doi.org/10.1007/s11235-017-0414-0

Jsoup HTML parser (2018) https://jsoup.org/apidocs/org/jsoup/parser/Parser.html. Retrieved 20 Jan 2018

Kumaraguru P, Rhee Y, Acquisti A, Cranor LF, Hong J, Nunge E (2007) Protecting people from phishing: the design and evaluation of an embedded training email system. In: Proceedings of SIGCHI conference on human factors in computing systems, San Jose

Li J, Sun L, Yan Q, Li Z, Srisa-an W, Ye H (2018) Significant permission identification for machine learning based android malware detection. IEEE Trans Ind Inform

Lin Q, Li J, Huang Z, Chen W, Shen J (2018) A short linearly homomorphic proxy signaturescheme. IEEE Access

List of online payment service providers (2018) http://research.omicsgroup.org/index.php/List_of_online_payment_service_providers. Retrieved 25 March 2018

Maio CD, Fenza G, Gallo M, Loia V, Parente M (2017) Time-aware adaptive tweets ranking through deep learning. Future Gener Comput Syst. https://doi.org/10.1016/j.future.2017.07.039

Maio CD, Fenza G, Gallo M, Loia V, Parente M (2018) Social media marketing through time-aware collaborative filtering. Concurr Comput Pract Exp 30(1)

Mohammad RM, Thabtah F, McCluskey L (2014) Predicting phishing websites based on self-structuring neural network. Neural Comput Appl 25(2):443–458

Montazera GA, ArabYarmohammadi S (2015) Detection of phishing attacks in Iranian e-banking using a fuzzy–rough hybrid system. Appl Soft Comput 35:482–492

Pan Y, Ding X (2006) Anomaly based web phishing page detection. In: Proceedings of 22nd annual computer security applications conference, Miami Beach, pp 381–392

Phishingpro Report (2016) http://www.razorthorn.co.uk/wp-content/uploads/2017/01/Phishing-Stats-2016.pdf. Retrieved 14 Oct 2017

Phishtank dataset (2018) http://www.phishtank.com. Retrieved 22 Aug 2017

Sheng S, Wardman B, Warner G, Cranor LF, Hong J, Zhang C (2009) An empirical analysis of phishing blacklists. In: Proceedings of the sixth conference on email and anti-spam, Mountain View

Stuffgate Free Online Website Analyzer (2018) http://www.stuffgate.com/. Retrieved 21 Jan 2018

Usage of content languages for websites (2017) https://w3techs.com/technologies/overview/content_language/all. Retrieved 22 Aug 2017

Varshney G, Misra M, Atrey PK (2016) A phish detector using lightweight search features. Comput Secur 62:213–228

Wang YG, Zhu G, Shi YQ (2018) Transportation spherical watermarking. IEEE Trans Image Process 27(4):2063–2077

Whittaker C, Ryner B, Nazif M (2010) Large-scale automatic classification of phishing pages. In: Proceedings of the network and distributed system security symposium, San Diego, pp 1–14

Xiang G, Hong J, Rose CP, Cranor L (2011) CANTINA+: a feature-rich machine learning framework for detecting phishing web sites. ACM Trans Inf Syst Secur 14(2)

Zhang Y, Hong JI, Cranor LF (2007) CANTINA: a content-based approach to detecting phishing websites. In: Proceedings of 16th international world wide web conference (WWW2007), Banff, pp 639–648

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

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.