Expert Systems With Applications 53 (2016) 231–242

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

New rule-based phishing detection method

Mahmood Moghimi, Ali Yazdian Varjani*

Faculty of Electrical and Computer Engineering, Tarbiat Modares University, Jalal Ale Ahmad Highway, Tehran 14115-111, Iran

ARTICLE INFO

Keywords:
Phishing
Internet banking
Classification
SVM
Sensitivity analysis
Browser extension
Rule-based

ABSTRACT

In this paper, we present a new rule-based method to detect phishing attacks in internet banking. Our rule-based method used two novel feature sets, which have been proposed to determine the webpage identity. Our proposed feature sets include four features to evaluate the page resources identity, and four features to identify the access protocol of page resource elements. We used approximate string matching algorithms to determine the relationship between the content and the URL of a page in our first proposed feature set. Our proposed features are independent from third-party services such as search engines result and/or web browser history. We employed support vector machine (SVM) algorithm to classify webpages. Our experiments indicate that the proposed model can detect phishing pages in internet banking with accuracy of 99.14% true positive and only 0.86% false negative alarm. Output of sensitivity analysis demonstrates the significant impact of our proposed features over traditional features. We extracted the hidden knowledge from the proposed SVM model by adopting a related method. We embedded the extracted rules into a browser extension named PhishDetector to make our proposed method more functional and easy to use. Evaluating of the implemented browser extension indicates that it can detect phishing attacks in internet banking with high accuracy and reliability. PhishDetector can detect zero-day phishing attacks too.

1. Introduction

Over the past few years, following the growth of communication networks, internet as the biggest has been widespread popular. Using anonymity provided by the internet, hustlers set out to deceive people with false offers and make themselves look legitimate in this medium (Arun et al., 2012). With increased terminals for access to information, internet banking creates the need for using reliable methods in order to control and use confidential and vital information. Today, financial crimes are transformed from direct attacks into indirect attacks. In other words, instead of bank robbery, criminals try to target bank's clients with a specific trick (Vrîncianu & Popa, 2010). Attacks on computer security are classified in three types: physical attacks, synthetic attacks, and semantic attacks (He et al., 2011). Phishing is one of the types of semantic attacks. In these types of attacks, vulnerabilities in the users are targeted; for example, the way users interpret computer messages (He et al., 2011), because most of the users read information sources without verifying them, and respond their demands.

Generally speaking, phishing is a kind of electronic identity theft in which a combination of social engineering and fake website creating methods is used to deceive user to disclose his/her confidential and invaluable details (Aburrous, Hossain, & Dahal, 2010). Most phishing attacks start with an electronic letter which claiming that has issued by a reputable company. This email encourages the user to click on the address that is provides in its content. This address directs the user to an illegal webpage, which is designed similar to a valid website, e.g. the site of a bank or a financial institution. According to report of Anti-Phishing Work Group (APWG) which is a non-profit organization working to provide anti-phishing education to enhance the public understanding of security, more than 66% of phishing attacks target financial institutions and online payment systems (Cassidy, 2013) (Fig. 1). In addition, most of the phishing attacks made through hacked web servers. According to this report, in September 2012, the United States of America's phishing was the host of more than 73% of websites (Cassidy, 2013).

To date, different methods have been proposed in order to detect phishing attacks. According to APWG, generally the defense mechanisms against phishing attacks are divided into three groups: identification methods, prevention methods, and modification methods (Abu-Nimeh, Nappa, Wang, & Nair, 2007). To identify and prevent from these types of attacks, different approaches are

* Corresponding author. Tel.: +98 2182883398
E-mail addresses: m.moghimi@modares.ac.ir (M. Moghimi), yazdian@modares.ac.ir (A.Y. Varjani).

http://dx.doi.org/10.1016/j.eswa.2016.01.028
0957-4174/

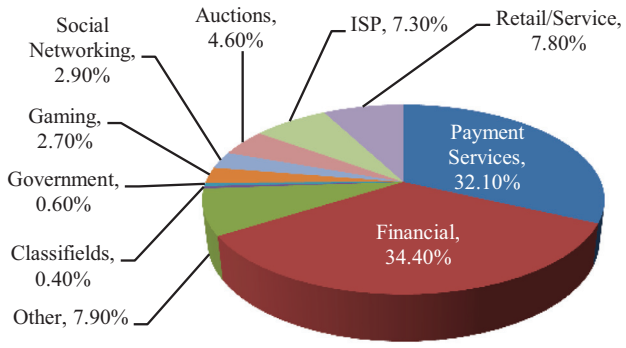


Fig. 1. Different industries affected by phishing in 2012 (Cassidy, 2013).

used whether in an email or in a website. We categorize these approaches into four groups:

- Black list/White list,
- URL evaluation,
- Visual assessment/Content evaluation,
- Hybrid approaches

Numerous tools are designed using white list or black list approach. Most of these tools authorize or reject a website through one or more references. Weakness of this approach is the lack of scalability and the ability to update the references quickly (Abu-Nimeh et al., 2007). Since creating websites often is cheap and easy, and their average life span may be a few hours, this approach in itself has low efficiency in prevention from these types of attacks (Aburrous et al., 2010). On the contrary, evaluating of apparent characteristic of webpage or its web address, is one of the best and simplest ways for phishing attacks detection, on which many studies have been conducted (Abu-Nimeh et al., 2007; Aburrous et al., 2010; Alkhozae & Batarfi, 2011; Arade, Bhaskar, & Kamat, 2011; ALmomani et al., 2012; Huang, Qian, & Wang, 2012; Rami, Thabtah, & McCluskey, 2014). In this approach, authors attempt to identify phishing attacks through analyzing website's information such as site logo, webpage visual similarity, security issues, and other web page characteristics (Alkhozae & Batarfi, 2011).

In this paper, we use this general approach to present a new rule-based method to detect phishing attacks on internet banking websites. Our method is based on novel feature sets, which determine the relationship between the content and the URL of a page. We used approximate string matching algorithms to compare address of each page resource element with the webpage URL. We also take consideration on how these resource elements load in a webpage; do they load from secure address or not? Our proposed feature sets include four features to evaluate the page resource identity, and four features to identify the access protocol of page resource elements. We used Support Vector Machine (SVM) algorithm to classify and detect phishing pages. We employ a related method to extract the hidden knowledge from our classification model. Finally, we implement our rule-based method in form of a browser extension called PhishDetector, which has a high accuracy in detection such attacks in our experimental results. The proposed method is independent of third-party services such as search engines or web browser history. Our main research contributions are summarized as follows:

- We proposed two novel feature sets to increase the accuracy of webpage classification.
- Proposed features are extracted from webpage content and they do not have any dependency to search engines, browser history and/or black/white lists.
- Since the proposed features extracted from page content, they are language-independent too.
- To make future development easy, we proposed a rule-based system by extracting the hidden knowledge from our classification model.
- We provide an easy to use chrome extension from our proposed rule-based method to detect phishing attacks on internet-banking websites.
- Our presented rule-based method can detect zero-day phishing attacks with high accuracy.

The rest of the paper is organized as follows: The next section reviews related works. Section 3 describes the proposed method and proposed features in detail and the way of the features vector made. Section 4 describes the data preparation for evaluation. In section 5, the method evaluated by real data includes phishing and legitimate websites. Limitation of our study discussed in section 6, while conclusion and future works discussed in section 7.

2. Related works

In 2010, Aburrous et al., provided an intelligent system to detect phishing pages in e-banking (Aburrous, Hossain, Thabatah, & Dahal, 2010). Their model is based on fuzzy logic combined with data mining algorithms to characterize the e-banking phishing website factors and to investigate its techniques by classifying the phishing types (Aburrous et al., 2010). Their approach was to apply fuzzy logic and data mining algorithms to assess e-banking phishing website risk on independent 27 characteristics and factors of a forged website. They got 86.38% classification accuracy with 10-fold cross-validation, which is relatively considered low. They also did not mention how they extract the 27 features for a webpage.

In the study carried out by He et al., they provided a system based on page content, HTTP transaction, and search engine results, which could identify phishing pages with an accuracy of 97% (He et al., 2011). In that study, the method was mainly based on CANTINA (Xiang, Hong, Rose, & Cranor, 2011), anomaly based web phishing page detection and with several additions and modifications (He et al., 2011). SVM algorithm also was used in order to classifying the output. The key feature used in the above study and other studies similar to Zhang et al. (Xiang et al., 2011), is the direct effect of the search engines ranking in page classification. If a webpage has high ranking in search engines, it will be considered as a legitimate website. In such cases, the attacker may use search engine poisoning techniques in order to increase the ranking of a forged website and index it legitimately (Gaurav, Mishra, & Jain, 2012).

By Arade et al., 2011, the authors used a new algorithm for approximate string matching in order to compare the webpage address and the addresses in the database of the proposed system. In the aforementioned algorithm, two strings are divided into a series of two-letter strings and then they will be compared with each other. If the similarity is more than 60%, the webpage will be considered as a legitimate otherwise, the webpage content will be checked and the aforementioned algorithm will run for all webpage links. The problem presented in this study is the probability of occurring false positive incidence. Thus, legitimate webpages may consider as phishing. Same feature used by Xiang et al., 2011, which may cause to produce false alarm. This lack of similarity existing between page resources address and page address (URL), has been used in related studies as an index to distinguish a valid site from a fake one (He et al., 2011; Xiang et al., 2011). In our study, we used this concept for all individual page resource elements (Hyperlinks, Cascade Style Sheets, Images, and Script reference addresses) not only page hyperlinks. He et al., used a same approach in their study (He et al., 2011). They defined a feature named "*ID foreign requests*". They compare the domain with the URL identity and *term identity set*. Their approach is completely

different with ours. We use approximate string matching algorithms instead on direct comparing. We discussed it in more detail in Section 3.2.

By Shahriar & Zulkernine, 2012, the authors presented a model to detect phishing webpages by testing the trustworthiness of suspected pages. They proposed a Finite State Machine (FSM) to evaluate webpage behavior by tracing the webpage form submission and the corresponding responses. Their approach has the ability of detecting advanced XSS-based attacks.

Ajlouni et al., adopted the 27 features from Aburrous et al., and employing MCAR (an AC classification algorithm) presented a phishing detection method (Ajlouni, Hadi, & Alwedyan, 2013). Their method can classify webpages with more than 98.5% accuracy, but they did not mention in their study how many rules were generated by employing the MCAR algorithm.

Barraclough et al., proposed a new rule-based model, which used five different inputs and a utilized Neuro-Fuzzy (a Fuzzy Logic and a Neural Network) scheme, to detect phishing websites (Barraclough, Hossain, Tahir, Sexton, & Aslam, 2013). Five inputs in their model were tables where features were stored which included: Legitimate site rules, User-behavior profile, PhishTank, User-specific sites and Pop-Ups from Emails. The proposed model can detect 98.5% of phishing attacks. One drawback with this study is the number of input features (288 features). This makes the implementation phase too complex.

Ramesh et al., proposed an approach to phishing webpages by take the webpage under scrutiny and identify all the direct and indirect links associated with the page (Ramesh, Krishnamurthi, & Kumara, 2014). Direct links extracted from the page content but the indirectly associated links of the page are extracted from the search engine result. They also used a third-party DNS lookup to map the domains of suspicious webpage and phishing target to corresponding IP. Their method could detect phishing webpages with 99.62% accuracy but it has external dependency (search engine result and 3rd-party DNS lookup) which makes the prediction time of their system depends upon the speed of the search engine and DNS lookup time. As mentioned in their paper, it takes 20,806 ms ± 27,272 ms to identify a webpage. The downside of this method is that it cannot detect phishing webpages hosted on the compromised domains as well as it cannot extract links or keywords from the suspicious webpages.

Abdelhamid et al., and Rami et al., had a same research in using rule-based classifications in detecting phishing webpages (Abdelhamid, Ayesh, & Thabtah, 2014; Rami et al., 2014). Rami et al., proposed a model (set of conventional features) and summarizes the prediction error-rate produced by set of Associative Classification (AC) algorithms (Rami et al., 2014). The results showed that C4.5 outperforms all algorithms with 5.76% average error-rate. Abdelhamid et al., used a developed AC algorithm called Multi-label Classifier based Associative Classification (MCAC) to extract their rules from training data (Abdelhamid et al., 2014). A limitation to inducing hidden knowledge (rules) by using AC methods is the large number of rules because all AC algorithms evaluate every single correlation between the attribute values and each class value during the training phase. They achieved most possible 97.5% accuracy of classification, which is relatively considered low in phishing detection systems.

Zhang et al., proposed a new model with five novel features and Sequential Minimal Optimization (SMO) algorithm for classification to detect Chinese phishing websites (Zhang, Yan, Jiang, & Kim, 2014). The proposed features have never been included or examined in any prior models of phishing detection. They proved that three features of the proposed five features have the most impact of phishing website detection. A limitation of this model is that the proposed features are just using to detect Chinese phishing webpages not all languages.

A brief summary of related works in comparison with our work shows in Table 1. We compare related works in four characteristics. These include independence of the methods on search engine and 3rd-party services (such as DNS-lookup, website traffic analysis, etc.), detection of new phishing websites that have not been identified yet (zero-day phishing detection), and detection of phishing websites in any language.

3. Proposed rule-based method

Our presented method is based on two novel feature sets. The overall system architecture of proposed method is shown in Fig. 2.

We proposed two feature sets to improve the performance of detecting phishing attacks and preventing data loss in internet

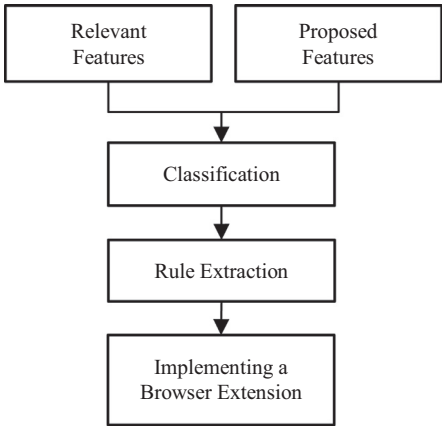


Fig. 2. Proposed model to detect phishing attacks.

Table 1
Comparison of related works with our work.

Work	Method	Search engines independence	3rd-party services* independence	Zero-day phishing detection	Language independence
(Aburrous et al., 2010)	Fuzzy logic	Yes	No	No	Yes
(Xiang et al., 2011)	CANTINA+	No	No	Yes	No
(He et al., 2011)	Page Classification	No	Yes	Yes	Yes
(Shahriar & Zulkernine, 2012)	Finite State Machine	No	Yes	Yes	Yes
(Ajlouni et al., 2013)	Rule-based (MCAR)	Yes	No	No	Yes
(Barraclough et al., 2013)	Neuro-fuzzy	No	No	Yes	Yes
(Ramesh et al., 2014)	New approach	No	No	Yes	Yes
(Rami et al., 2014)	Rule-based	Yes	No	Yes	Yes
(Abdelhamid et al., 2014)	Rule-based (MCAC)	Yes	No	Yes	Yes
(Zhang et al., 2014)	New method	Yes	No	Yes	No
Our work	**Rule-based**	**Yes**	**Yes**	**Yes**	**Yes**

* Other services such as DNS-lookup, website traffic analysis, and etc.

banking webpages. Our proposed feature sets, determine the relationship between the content and the URL of a page. These feature sets include four features to evaluate the page resource identity, and four features to identify the access protocol of page resource elements. We extract these two feature sets through examining the content and the page document object model (DOM). We used approximate string matching algorithms to compare address of each page resource element with the webpage URL (saved as our first proposed feature set). We also take consideration on how these resource elements load in a webpage; do they load from secure address or not? (We saved it as our second proposed feature set). We described more details about these feature sets and the way that we extract them in section 3.2. We adopted a subset of relevant features from related works along with our proposed feature sets to create webpage features vector. We employed support vector machine (SVM) algorithm to train our model and classify phishing webpages. The approach used in proposed method is to determine the relationship between page content and page address, which based on the aforementioned feature sets. We used a related method in order to extract the hidden knowledge from the proposed model. Then we embedded the extracted rules into a browser extension, which can detect phishing attacks in internet banking with high accuracy and reliability.

In this section at first, we describe about relevant features that we have used them in our feature vector. In section 3.2, we described about our proposed feature sets.

3.1. Relevant features

Based on related studies (Aburrous et al., 2010; He et al., 2011; Lakshmi V & MS, 2012), we analyzed the usage frequency of various features, which have been used to detect phishing attacks. During the frequency analysis, we collected 50+ different features for each webpage. Finally, we adopted a subset of first five features with some value extraction modification, in our feature vector: IP address, SSL certificate, Number of dots in URL, Web address length, and Blacklist keywords. Attackers often employ these features to create a fake webpage similar to legitimate one.

In this section, we describe how these features of our features vector is extract from a webpage.

3.1.1. IP address

One way to hide the webpage address is using an Internet Protocol (IP) address instead of a Domain Name System (DNS) address. Therefore, in this way the attacker prevents users to identify the actual page address. Another reason for using the IP address is that attackers often do not want to spend money to buy a domain for their fake webpages, and on the other hand, despite the possibility of tracking, they prefer to offer the host IP address to their users.

We define the first feature as F_1, which would be "1" if the webpage address contains IP address.

$$F_1 = \begin{cases} 1 & \textit{IP address} \\ 0 & \textit{no IP address} \end{cases} \quad (1)$$

3.1.2. SSL certificate

Using security protocols in internet banking sites is one of the most essential requirements of such websites (Vrîncianu & Popa, 2010). The Secure Hypertext Transfer Protocol (https) is a communication protocol that is using widely for secure communications in computer networks, especially internet. When confidential data are transmitting over the internet as a public path, their security is a matter of the utmost importance (Arun et al., 2012). Using a secure socket layer (SSL), digital certificates and digital signatures, provides a level of security for data exchange in internet environment. However, in the case of improperly implementation, information security will reduce (Arun et al., 2012). Not almost all of phishing sites use this protocol (Aburrous et al., 2010). However, attacker may apparently use this; by changing the browser's address or using noncredit certificates or self-signed certificates.

We define the existence of https protocol in webpage address as feature F_2, which will be "1" if the website uses https for data transfer.

$$F_2 = \begin{cases} 1 & \textit{Use SSL} \\ 0 & \textit{Do not use SSL} \end{cases} \quad (2)$$

3.1.3. Number of dots in URL

Other ways to mislead users to avoid detection of the primary webpage address is using high number of sub-domains in URL (Xiang et al., 2011). In this way, attacker prevents users from identifying the actual website by creating multiple sub-domains and inclusion of related bank names in the following domains. For example in a page with the following URL, at first glance for a novice user, it may be assumed that the address is related to PayPal website but actually, this address is pointed to com.mx website: http://paypal.co.uk.webdllscr.mpp.sipromedic.com.mx

In some previous studies that this feature used for classification, a specified number has been used as a way of detecting the suspicious addresses. For example, on average, in a URL address of a legitimate site there are maximally three dots. If the numbers of dots are more than this number, it will indicate that the webpage is phishing. Specifying a particular number as border detection may cause errors in classification result. Therefore, we define this feature as F_3 and contain the reverse of the number of dots in page URL.

$$F_3 = N_{dot} \quad (3)$$

3.1.4. Web address length

The length of an address is another characteristic by which the phishing addresses can often identified through it. Oftentimes, attacker use long URLs in order to hide the true address of a phishing webpage (Xiang et al., 2011). The length of an address cannot be the only representative of the phishing addresses. An address is consists of several parts. Consider a standard address as:

<protocol>://<host>/<path>/<file>?<query>

The length of an address can be explained through its different parts (Huang et al., 2012). Here, in order to increase the accuracy in detection of phishing webpages, we divide the length of an address into three parts including *host, path* and *query*. It helps us to determine a precise specification of a phishing webpage in internet banking.

We define three features for a webpage address to determine length of each part. Therefore, we define F_4 for length of the "host", F_5 for the length of the "path" and F_6 for the length of "file" and "query" parts.

3.1.5. Blacklist keywords

Most internet banking phishing webpages use some certain words in their page address that apparently look like as a legitimate (Xiang et al., 2011). These keywords usually are the names of famous banks or a combination of their websites address. An attacker uses these keywords into a phishing URL, in a way that at first sight mislead a novice user as a real internet-banking website.

By (Xiang et al., 2011), authors, used eight different keywords to evaluate a webpage address. The corresponding keywords should have specific characteristics. Using wrong keywords may increase system false positive errors in identifying correct webpages. In this regard, we have analyzed some collected phishing address to identify and extract such words. Among the extracted words, only some of them were selected as keywords, which firstly were not in an address of the related legitimate site and secondly be more

frequent in their part of a phishing address. In order to control the absence of any corresponding word in an original address, we checked URL of legitimate sites linked to phishing addresses by using search engines.

Finally, we selected 26 words for host, 23 words for path and 14 words for query part as our blacklist keywords list. Therefore, we define F_7 feature to determine the rate of keywords appears in host part of a webpage address. Similarly, F_8 as the rate of related keywords appears in path part and F_9 as the rate of related keywords in query part of a URL.

$$F_n = \begin{cases} -1 & Word_{count} = 0 \\ \dfrac{Keyword_{count}}{Word_{count}} & Word_{count} > 0 \end{cases} \quad n = 7,\ 8,\ 9 \tag{4}$$

3.2. Proposed feature sets

In this paper, we propose two feature sets, which include page resource identity and access protocol of page resource elements. In order to extract the desired feature sets from a webpage, we used the Document Object Model (DOM). In this section, we briefly summarize the methodology used to extract the proposed features from the page content.

3.2.1. Page resource identity feature set

In phishing attacks, the attacker tries to invite user to visit a fake webpage by different ways. In fact, he/she tries to create a false sense of confidence. One of these attraction ways is to register some addresses whose URL, at a glance, is similar to the address of the real website, so the novice user may not distinguish the difference. For example, the page address of http://www.Bank0fAmerjca.com is very similar to http://www.BankOfAmerica.com but it is a phishing site of the "Bank Of America" (At first URL, zero is replaced as "O").

In such cases, direct comparison of two addresses may produce wrong result. For this reason, we use approximate matching methods. Approximate string matching is a way for finding approximate patterns similar to a pattern in a textual string (Levenshtein, 1966). Normally, the output of approximate matching algorithms between two words represents the highest number of wrong characters in one of them, which needs to be similar to another one. To find these differences, these algorithms pay attention to such operations such as insertion, removal and replacement of characters.

There are many different approximate string matching algorithms. In this paper, we use "Levenshtein Distance" (Levenshtein, 1966) in order to evaluate the webpage resource identity. Levenshtein distance is a mathematical algorithm that is using to measure the difference between two sequences (Levenshtein, 1966). The Levenshtein distance between two words is the minimum number of single-character edits (insertion, deletion, and substitution) required to change one word into the other (Levenshtein, 1966). The higher result indicates higher difference between two words. For example for two words "bank" and "bank" the result of Levenshtein distance will be "0", which indicated that the mentioned words are equal. However, for "Box" and "Fox" it will be "1", because the letter "B" needs to change to "F". In general case, the Levenshtein distance between two strings a and b with the length i and j, respectively are defined as:

$$LD_{a,b}(i,j)\begin{cases} \max(i,j), & \min(i,j) = 0 \\ \min\begin{cases} LD_{a,b}(i-1,\ j)+1 \\ LD_{a,b}(i,\ j-1)+1, \\ LD_{a,b}(i-1,\ j-1)+[a_i \neq b_j] \end{cases} & else \end{cases} \tag{5}$$

Levenshtein distance algorithm has been used in wide range of studies such as spell checking, voice recognition and DNA analysis. In 2006, Fu proposed a new method to detect phishing attacks through employing two approaches; Visual Assessment & Semantic Assessment Approach (Fu, 2006). In semantic approach, he used Levenshtein distance algorithm to calculate the dissimilarity for the pair of Unicode strings in a Unicode attack detection system (Fu, 2006).

Normally, in legitimate sites the resource elements in a webpage (including images, CSS and Script references), are called from a related web address. On the contrary, a phishing webpage, which is created by an attacker, calls its required resource elements from the legitimate sites (He et al., 2011). This lack of similarity existing between resource address and page address in a phishing webpage is a good index to distinguish a valid site from a faked one. For this purpose, we extract all reference address of webpage resource elements, and then we compare each resource element address with the webpage URL separately. For hyperlinks, we extract the "*href*" attribute, and for images, CSS references, and script references, we extract the "*src*" attribute as well. The result of evaluating address of all related page resource elements shows the extent of nature of the page. We employ the normalization method used by (Fu, 2006) to normalize the calculated distance for each resource element in a webpage. We define the normalized distance as Eq. (6):

$$LD_{normalized} = \frac{LD(Domain_{link},\ Domain_{url})}{\max\left(|domain_{link}|,\ |domain_{url}|\right)} \tag{6}$$

where $LD_{normalized}$ is calculated by the proportion of Levenshtein distance between domain of webpage address and the domain of each resource in the webpage, by the maximum length of the mentioned domains. The value of $LD_{normalized}$ is between "1" and "0", which the larger value indicates more suspicious for the aforementioned resource. For example, assume that a webpage with URL of http://paypal.secure.login.web.com.cc/pages/web/login.html has one hyperlink with "*href*" attribute of http://www.paypal.com/pages/signup. By comparing domain of the webpage URL, and domain of the image reference address, we have:

$$LD(Domain_{link},\ Domain_{url}) = LD(\text{“}com\text{”},\ \text{“}paypal\text{”}) = 6$$

It means that six insert/updated/delete operations needed to transform "*com*" to "*paypal*". Therefore, we calculate the $LD_{normalized}$ according to Eq. 6 as follow:

$$LD_{normalized} = \frac{LD(\text{“}com\text{”},\ \text{“}paypal\text{”})}{\max(|com|,\ |paypal|)} = \frac{6}{6} = 1$$

The result reflects that the mention resource element is suspicious. Similarly, the normalized distance will calculate for the other resource types (including images, CSS and Script files). The average of normalized distance for each resource element type indicates the amount of relation between the webpage URL and the desired resource element. We define it as Eq. (7).

$$R_{resource} = \sum_{1}^{n} LD_{normalized} \Rightarrow Resource_{PhishingRate} = \frac{R_{resource}}{n} \tag{7}$$

where $R_{resource}$ is the summation of whole normalized distances for each resource element in a webpage and $Resource_{PhishingRate}$ is the average of it for the desired resource. Hence, we define four features F_{10}, F_{11}, F_{12} & F_{13} to determine the extent of suspicious of webpage resource elements: page links, JavaScript files, CSS files and Images respectively.

$$F_{10} = \frac{\sum_1^n Link\_LD_{normalized}}{n} \tag{8}$$

$$F_{11} = \frac{\sum_1^n Script\_LD_{normalized}}{n} \tag{9}$$

$$F_{12} = \frac{\sum_1^n Style\_LD_{normalized}}{n} \quad (10)$$

$$F_{13} = \frac{\sum_1^n Image\_LD_{normalized}}{n} \quad (11)$$

3.2.2. Page resource access protocol feature set

As described in Section 3.1.2, the security of an internet-banking site is one of the most important steps that need to be met in designing these sites (Vrîncianu & Popa, 2010). Internet banking websites receive their resource elements from their own or a related secure domain. If a webpage in an internet-banking website uses *https* protocol to display the page content, all the resources including images, style files and JavaScript files, should be loaded with the same webpage protocols. In contrast, phishing pages do not use *https* protocol, so the resource elements are not loaded via a secure protocol and therefore we can distinguish between a phishing and a legitimate webpage.

We define F_{14}, F_{15}, F_{16} and F_{17} as features to show the rate of secure access to page links, JavaScript files, style sheet files and images respectively.

$$F_{14} = \begin{cases} -1, & N_{links} = 0 \\ \frac{N_{secure\ links}}{N_{links}}, & N_{links} > 0 \end{cases} \quad (12)$$

$$F_{15} = \begin{cases} -1, & N_{scripts} = 0 \\ \frac{N_{secure\ scripts}}{N_{scripts}}, & N_{scripts} > 0 \end{cases} \quad (13)$$

$$F_{16} = \begin{cases} -1, & N_{styles} = 0 \\ \frac{N_{secure\ styles}}{N_{styles}}, & N_{styles} > 0 \end{cases} \quad (14)$$

$$F_{17} = \begin{cases} -1, & N_{images} = 0 \\ \frac{N_{secure\ images}}{N_{images}}, & N_{images} > 0 \end{cases} \quad (15)$$

where N is the number of each resource element used in a webpage. For example, N_{images} shows the number of images that used in the webpage. The variable $N_{secure\,images}$ indicates the number of images that loaded through a secure protocol (https). Similarly, for other resource elements, these two values represent the number of resource elements and the resource elements that use https protocol, respectively. The value of F_{14}, F_{15}, F_{16} & F_{17} is between "1" and "0", which the larger value indicates more secure access to the desired resource.

3.3. Classification

After preparing the features vector, we have to classify the webpage to identify it as a phishing or a legitimate. One of the most important issues in machine learning is using a classification method to train the system. Support Vector Machines (Vapnik, 1990) is a classification method, which are widely been used in recent years. SVM is a binary classifier that classifies two classes using a linear boundary. Vapnik the early 1990 s introduced SVM (Vapnik, 1990). SVM method, due to its high accuracy in anticipation and modeling, as well as having ability to work with high-dimensional data, is widely used in various branches of sciences such as pattern recognition, text classification and face detection (Vapnik, 1990).

In this paper, we use SVM to train our model and classify phishing pages in internet banking. This method has a good efficiency in related studies (He et al., 2011; Lakshmi V & MS, 2012; Huang et al., 2012; Abdelhamid et al., 2014). Learning a good SVM model from training data requires careful selection and optimization of the SVM training parameters (Barakat & Bradley, 2010). To select a best kernel for our SVM classification, we test the model using polynomial and RBF kernels. Results showed that the polynomial kernel with degree of three has the lowest error with high accuracy. Therefore, in this paper we employ polynomial kernel with degree three to classify webpages. We optimize the classifier performance by testing the model using different input parameters. In order to find the best parameters for the SVM model, we employ grid-search that is a standard method of exploring through two-dimensional space (Ben-Hur & Weston, 2008).

3.4. Rule extraction

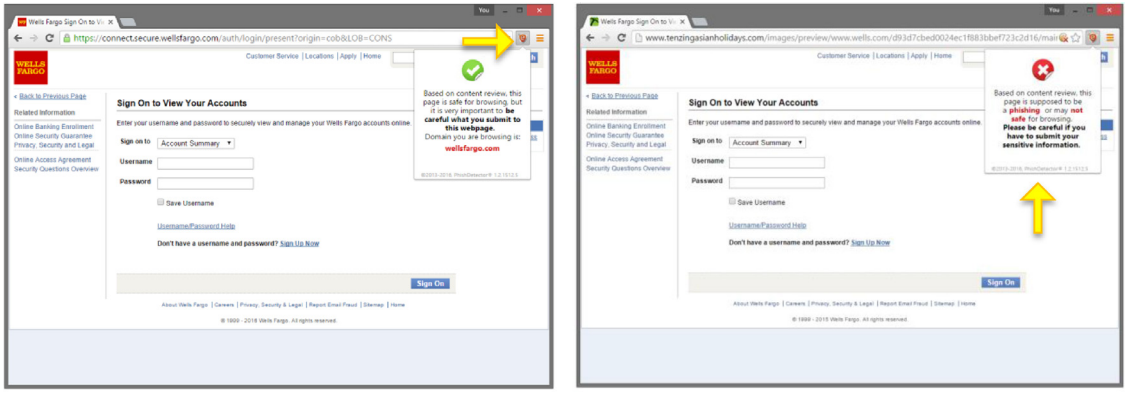
Despite the excellent performance of SVM algorithm in classification of phishing webpages, same as artificial neural network's models, the proposed model is a black box that humans cannot understand (He, Hu, Harrison, Tai, & Pan, 2006). Therefore, we need a way to emulate the logic that exists in this black box models and extract it in the form of rules to increase its comprehensibility. Then we can use the extracted rules to develop our rule-based system.

In this paper, we employed the method presented by (He et al., 2006) to extract our knowledgebase from the proposed SVM's model. This approach combines SVM with decision tree into a new algorithm called SVM_DT (He et al., 2006). In this approach the support vector machine is considered as a black box, and then rules are extracted based on the descriptive relationship exist between input and output of the model (Barakat & Bradley, 2010). This approach tries to generate rules by creating a synthetic dataset based on what SVM's model learned. The process of generating rules is done based on decision trees algorithms such as C4.5 that is trained on synthetic data sets (Barakat & Bradley, 2010). Since, SVM_DT algorithm can generate high quality rules with a better comprehensibility than SVM (He et al., 2006), we employed it to extract our knowledge. We extracted our rules from presented SVM model based on SVM_DT algorithm by following steps (He et al., 2006):

1. For dataset S, we divide it into N subsets with similar sizes (k) and similar distribution of classes.
2. Perform the tests for the N runs, each with a different subset as the test set (Te_svm^i, i=1,…,N) and with the union of the other N–1 subsets as the training set (Tr_svm^i, i=1,…,N).
3. Save the result of prediction for each run into (P_svm^i, i=1,…,N).
4. Select cases that are correctly predicted by SVMs into new data set (S_svm^i, i=1,…,N).
5. Create the artificial datasets: test set (Te_dt^i, i=1,…,N) from (Te_svm^i, i=1,…,N) and the union of the other N–1 subsets (S_svm^i) as the new training set (Tr_dt^i, i=1,…,N) to train decision trees and induce the rule sets.
6. Then we train and test the decision tree for each N subset by using datasets created in step 5.

After train and test the decision trees we have N different rules set which is made in step 6. It is necessary to select the best set of rules, which at least produce best and most effective conditions. After analyzing the rules that extracted from the produced trees, we found that most of rules produce same result with same conditions. Therefore, we can aggregate common rules into one rule. Given to the importance of how selecting rules, aggregating or removing duplicate, we selected rules in several steps.

- We removed the rules, which are often common together in each category.
- We compare rules in each set with each other based on used features, and then we ignore the rules that cover same range with common features.



(A) Legitimate Detection **(B) Phishing Detection**

Fig. 3. System output (PhishDetector extension); (A) Legitimate detection, (B) Phishing detection.

- We ignore the rules that have confidence level of 50% or below too.
- Then we implement the remaining rules in the form of a decision tree and then we combine rules that overlap and/or produce same result with each other.

Finally, we achieved 10 rules to implement the rule-based system to detect phishing attacks.

3.5. Implementation

We have embedded the proposed rule-based system in form of an extension called PhishDetector for Google Chrome browser. We developed our extension as a content script in order to interact with the webpage content. Content scripts are JavaScript files that run in the context of web pages (Google, 2014). It provides us to access the standard webpage DOM. Therefore, we can analyze webpage DOM and extract corresponding features as fast as possible. Content scripts execute in a special environment called an isolated world (Google, 2014). They have access to the DOM of the page they are injected into, but not to any JavaScript variables or functions created by the page (Google, 2014).

At runtime when the user calls our extension by clicking on its icon, our system tries to extract all features as described in previous section. These features extracted from the current webpage URL and content. Then, based on the extracted values and the rules contained in the knowledgebase of our system, the webpage is classified. The result will be shown to the user in the browser output through a popup window, which contains more detail about the inference result (Fig. 3). In image A of Fig. 3, the system detects the current page as a legitimate webpage but it warns the user to enter his/her information with caution if necessary. For more details, it shows the domain of the webpage too. When the system detects a page as a phishing webpage (image B of Fig. 3), it shows a popup page that recommend to close the page immediately and do not browse its content.

4. Experiment setup

In this section, we provide detail information of the setup needed to carry out the experiment. This includes how we will prepare our datasets and the metrics we use to evaluate our method.

4.1. Dataset

To classify webpages based on features vector, it is necessary to train the system on real data. Our dataset obtained from two different sources: Legitimate webpages collected from Yahoo directory service (*http://dir.yahoo.com*) and phishing webpages collected from PhishTank (*http://www.phishtank.com*). Our phishing webpage collection focused on internet banking web sites.

4.1.1. Data collection

We start our data gathering from mid-December 2012 to mid-February 2013. On average, we collected about 600 phishing site addresses daily. Out of the 600 sites, 300 sites were dedicated to internet banking or online payment. After analyzing the aforementioned sites, we found that more than 50% of the associated addresses has been expired or deactivated. Nonetheless, during the period, we extracted information of 3066 phishing webpages. We also selected 686 legitimate internet-banking webpages from the yahoo directory service after analyses and inspections. We extracted and recorded more than 50 features from collected webpages.

4.1.2. Preprocessing

Since phishing sites are active for a short period, we faced some difficulties during the process of data collection. For instance, the majority of phishing sites were infected with computer malwares, which cause some problems in our computer while data extracting. Most of phishing sites included in the PhishTank have not been classified properly, which led to decrease the speed of the data collection process. In order to prevent any further errors in our datasets, we use preprocessing step to prepare dataset for classification. In this step, we perform some operations such as removing irrelevant data, eliminating the data with error and loss, and removing redundant data:

- Because of the impact of duplicate data in machine learning (Witten, Frank, & Hall, 2011), we sorted our extracted features based on their webpage URL and then we excluded duplicate data of a repetitive domain.
- To scale all features value, we mapped all data range into –1 to +1.

4.1.3. Sampling

After preprocessing, the data is ready for sampling. There are many methods for data sampling, including Random Sampling, Stratified Sampling, Sampling without Replacement, and Sampling with Replacement (Witten et al., 2011). In this paper, we used stratified sampling to generate the dataset for training and testing. We assume 80% for train dataset and 20% for test dataset. At the end of preprocessing and sampling, our cleaned training dataset consists 1158 unique e-banking phishing pages and 549 legitimate webpages (Table 2).

Table 2
Collected Dataset after preprocessing.

		Phishing	Legitimate	Total
Raw instances		3066	1271	4337
Removal of missing/unrelated data		2189	935	3124
Duplicates removal		1448	686	2134
Stratified sampling	Dataset 1	1158	549	1707
	Dataset 2	290	137	427

Table 3
Confusion matrix for phishing detection.

		Predicted classes	
		Phishing	Legitimate
Actual classes	Phishing	True positive (TP)	False negative (FN)
	Legitimate	False positive (FP)	True negative (TN)

We also generate another dataset (*Dataset3*) which includes 103 and 73 non-duplicate unique phishing and legitimate webpage respectively. We use this dataset to evaluate our extension in next section.

4.2. Feature set

In this paper, we used 17 features to build the model. In order to identify the importance of the proposed features, we prepare three feature set as follow:

- Feature Set One (*FS1*): Contains the first nine features.
- Feature Set Two (*FS2*): Contains only the proposed features.
- Feature Set Three (*FS3*): All features include features from literature and proposed features, which results in total of 17 features.

4.3. Evaluation metrics

In phishing classification methods, the goal is to classify a phishing page correctly. By assuming a phishing page as positive and a legitimate page as negative, the confusion matrix shown in Table 3.

According to the confusion matrix, we have:

- True Positive (TP): The phishing pages that classified correctly as phishing.
- False Negative (FN): The phishing pages that classified incorrectly as legitimate.
- True Negative (TN): The legitimate pages that classified correctly as legitimate.
- False Positive (FP): The legitimate pages that classified incorrectly as phishing.

In our experiment, we adopted Kappa statistic, Sensitivity and *F*-Score as the main evaluation metrics of the model.

Kappa statistic: Kappa is one of the characteristics of evaluation a classification model. Cohen introduced it in 1960 (Witten et al., 2011). It shows the amount of agreement between judges (Manning, Raghavan, & Schütze, 2009). Kappa calculated as:

$$Kappa = \frac{P(A) - P(E)}{1 - P(E)} \tag{16}$$

where $P(A)$ is the proportion of the times the judges agreed, and $P(E)$ is the proportion of the times they would be expected to agree by chance (Manning et al., 2009).

Sensitivity: It refers to the proportion of positive samples, which correctly identifies as positive (Witten et al., 2011). Sensitivity defined as:

$$Sensitivity\ (Recall) = \frac{TP}{TP + FN} \tag{17}$$

F-Measure: This indicator measures the accuracy of classification. F-Score calculated based on Precision and Recall of a classification (Witten et al., 2011).

$$F - Score = \left(1 + \beta^2\right) \frac{Precision.\ Recall}{\left(\beta^2 . Precision\right) + \ Recall} \tag{18}$$

$$Precision = \frac{FP}{FP + TN} \tag{19}$$

$$F_1 - Measure = 2\ \frac{Precision.\ Recall}{Precision + \ Recall} \tag{20}$$

5. Results and discussion

In order to demonstrate the overall performance of our proposed method, we performed experiments on three aspects: model evaluation, proposed features evaluation and, evaluating extracted rules.

5.1. Model evaluation

We used 10-fold cross-validation to train and test the model. We create the model based on three feature sets separately. Fig. 4 shows the classification result.

In assessment the efficiency of phishing detection methods two characteristics are often considered; TP and FN. When the first feature set (*FS1*) is using to classify pages, the values of the aforementioned characteristics (TP, FN) are 94.39% and 5.61%, respectively. However, by using the new feature set (*FS2*) the values of them are 98.62% and 1.38%, respectively. The relative improvement of TP of the classification, which is ascribe to the increase in the precision of detecting phishing sites using the proposed features. It is though worth mentioning that the detection precision of the model in detecting internet banking pages decreased by 2.5% when using the second feature set (*FS2*), and the overall precision of the model was increased using the second feature set compared to the first too.

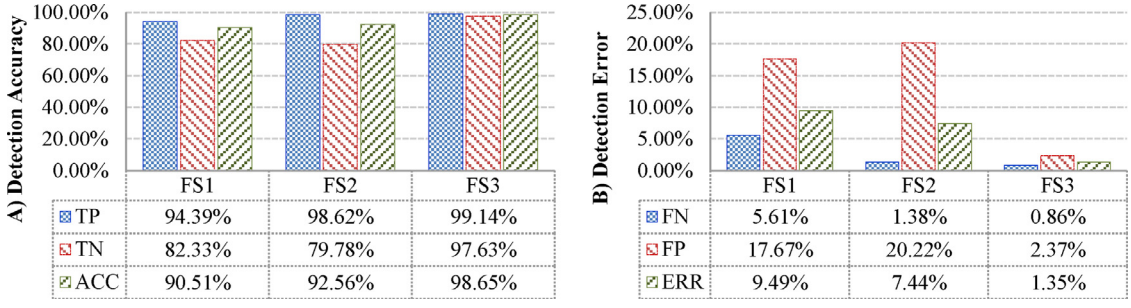


Fig. 4. Result of classification on each feature set, (A) Detection accuracy, (B) Detection error.

Table 4
The result of classification with SVM.

Classification output	Value
Correctly classified instances	98.65%
Incorrectly classified instances	1.35%
Kappa statistic	0.969
Sensitivity	0.9914
F-Score	0.9901
Root mean squared error (RMSE)	0.102

When we use the third feature set (*FS3*), the accuracy of detecting phishing webpages or the TP characteristic increased to 99.14%. The value of FN, which plays an important role in detecting phishing webpages, also decreased to 0.86%. The reduction indicates that the model demonstrates higher precision when we use 17 features to create the model and detect phishing pages in internet banking.

After page classification using the feature vector, which contains 17 features (*FS3*), we found that the proposed model is capable of detecting phishing pages in internet banking with accuracy of 98.65% and error rate of 1.35%. This error level is because of high amounts of errors associated with the FP of the model in detecting legitimate pages as phishing. The obtained results from classification of webpages using the SVM method and 17 features included in *FS3* listed in Table 4. As we see in Table 4, the Kappa value is 0.969, which according to the Rule of Thumb (Manning et al., 2009), indicates that the model has an acceptable output.

Another tool that is widely used for evaluating data mining schemes is Receiver Operating Characteristics or briefly ROC curves (Witten et al., 2011). A receiver operating characteristics (ROC) graph is a technique for visualizing, organizing and selecting classifiers based on their performance (Fawcett, 2006). An ROC graph depicts relative tradeoffs between benefits (true positives) and costs (false positives) (Fawcett, 2006). Fig. 5, shows the ROC graph for the proposed model. The area under curve (AUC) of a classifier is equivalent to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance (Fawcett, 2006). The value of AUC for the proposed model is 0.998.

5.2. Evaluating of proposed feature sets

One of the main objectives of this study was to evaluate the impact of proposed features to detect phishing pages. To determine the effect of each corresponding feature on output of the classification, we employed sensitivity analysis. In sensitivity analysis, the variability of outputs changes is measured through the variability of inputs changes (Pannell, 1997). In this paper, we used one-at-a-time method to evaluate the effect of each feature of the feature vector (Pannell, 1997). In this method, for changing the entries of each category, model output statistics is measured (Pannell, 1997). Finally, according to the sensitivity of classification model, the effectiveness of each desired feature is measured. We used accuracy, error, kappa, sensitivity and F-Score to measure each feature importance in our sensitivity analysis. Table 5 details the model statistics in eliminating of each feature.

Table 5
The result of sensitivity analysis.

Eliminated feature	Accuracy	Error	Kappa	Sensitivity	F1-measure
F1	98.65%	1.35%	0.969	0.9914	0.9901
F2	96.07%	3.93%	0.909	0.9801	0.9713
F3	98.54%	1.46%	0.966	0.9905	0.9892
F4	98.65%	1.35%	0.969	0.9914	0.9901
F5	98.65%	1.35%	0.969	0.9922	0.9901
F6	98.65%	1.35%	0.969	0.9905	0.9901
F7	98.07%	1.93%	0.956	0.9836	0.9857
F8	98.59%	1.41%	0.968	0.9914	0.9897
F9	98.77%	1.23%	0.972	0.9922	0.9909
F10	98.54%	1.46%	0.966	0.9896	0.9892
F11	98.59%	1.41%	0.968	0.9914	0.9897
F12	98.77%	1.23%	0.972	0.9931	0.9910
F13	98.54%	1.46%	0.966	0.9905	0.9892
F14	98.54%	1.46%	0.966	0.9896	0.9892
F15	98.30%	1.70%	0.961	0.9905	0.9875
F16	98.77%	1.23%	0.972	0.9931	0.9910
F17	97.83%	2.17%	0.951	0.9871	0.9841

Eliminating of the first feature (*F1*) does not change either the sensitivity or the F-Score of the model. Hence, this feature does not influence on new model. However, eliminating the second feature causes the sensitivity led to 0.011 decreases and the value of kappa has reduced to 0.909 too. Therefore, eliminating of this feature leads to a reduction in the efficiency of the model. Similarly, except for the 12th and 16th features from the proposed feature sets, eliminating of every feature leads to a decline in the efficiency of the model. However, when the ninth, 12th, and 16th features are omitted, the model sensitivity and *F*-Score are increased, which reflects the negative impact of these features on classifications

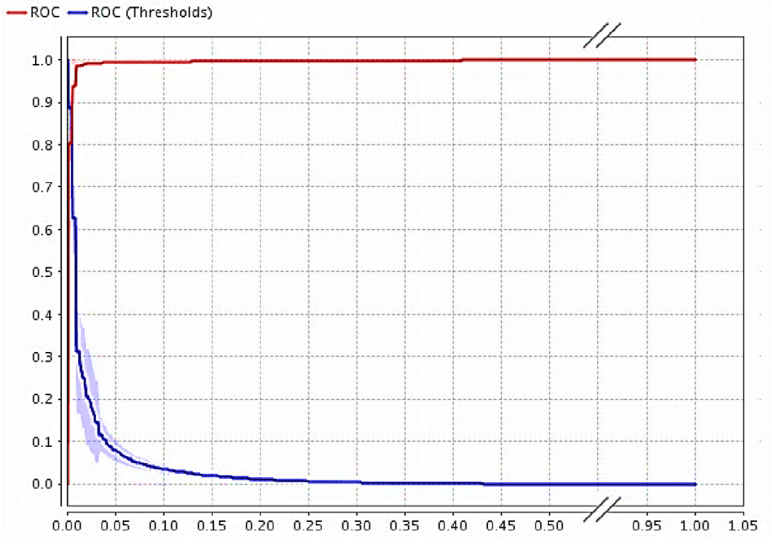


Fig. 5. ROC curve of SVM classification.

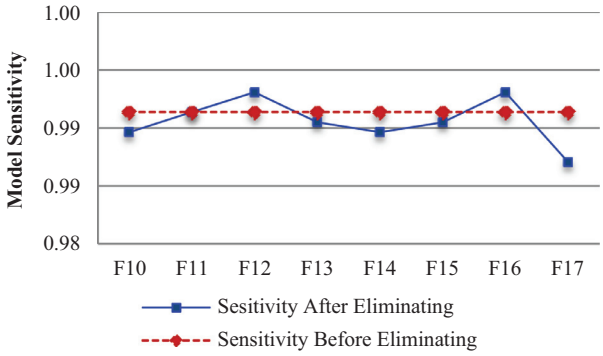


Fig. 6. Sensitivity of proposed features.

of phishing pages. Moreover, elimination of the aforementioned features leads to 98.77% increase in the model accuracy. The value of kappa is also increased by 0.972. This growth indicates that the reliability of the output of the classification process is developed as well. Output of sensitivity analysis of 17 features used in classification of phishing pages, demonstrates the positive effect of six features (*F10, F11, F13, F14, F15, F17*) of the eight proposed features, because by eliminating each of them from features vector, decreases the efficiency and accuracy of the model (Fig. 6).

5.3. Extracted rules evaluation

As described in Section 3.4, we extracted 10 rules by employing the method proposed by (He et al., 2006). In order to evaluate performance of our extracted rules, we compare them with the rules that could be extracted directly from our training dataset. To extract rules from training dataset we employ the C4.5 rule induction algorithm. After rule extraction, we compared them with our test dataset. Table 6 gives the comparison result of the rules extracted through above method with our rules. As shown in Table 6, relatively small numbers of rules, with both high accuracy and sensitivity have been extracted using the related method as described in section 3.4.

5.4. Implemented extension performance analysis

Table 7 shows the result of evaluating our extension (PhishDetector) with collected phishing and legitimate internet banking webpages in *Dataset3*. As we see, the proposed rule-based extension can identify phishing pages with 100% true positive in internet banking, but in identification of the legitimate webpages has 2.74% error rate.

In order to identify the rules that produce this error rate, we categorized sample instances according to rules, which used to predict the webpages (Fig. 7). According to Fig. 7, rules number 6 and 10 caused the misdiagnosis (detect two legitimate pages as phishing). As we see in phishing section in Fig. 7, these rules (Rule #10 & Rule #6) have detected most phishing pages. These rules are defined as follow:

Rule 6: *if F02-UseSSL = 0 and F05-DirLen > 0.667 and F13-ImagesLD > 0.37 then Phishing*

Rule 10: *if F02-UseSSL = 0 and F05-DirLen ≤ 0.667 then Phishing*

Table 7
Confusion matrix of extracted rules evaluation.

		Predicted classes	
		Legitimate	Phishing
Actual classes	Phishing	100%	0%
	Legitimate	2.74%	97.26%

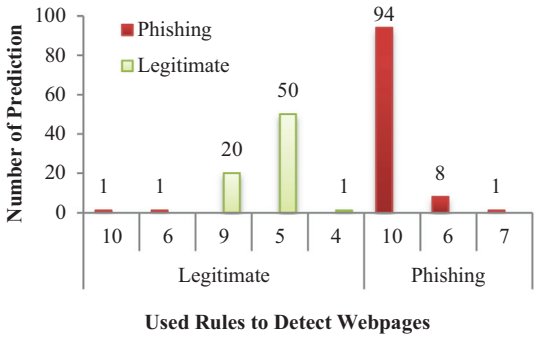


Fig. 7. Effective rules to detect webpages on test dataset.

Rule No. 6 states that "If a site has not a security certificate and the rate of directory length is greater the 0.667 and index of suspicious images of the page is greater than 0.37, then the webpage is Phishing." Feature number 13 (index of suspicious images of the page) shows the percentage of calling page images from a different domain, which is always high for phishing pages. However, this situation may also occur for some of legitimate sites; such a manner that pictures or style files are located on another URL which is different from the webpage URL. To address this problem, we can employ an "identity list" while the features vector is creating for each page. In data collection stage, this "identity list" will contain those addresses that webpage calls required information from. Then at the time of creating features vector, we will calculate the value of the page resource identity according to this list. If address of page resource element being evaluated, is as the same address in this list, then it considered as related address and the value for the first proposed feature set is close to zero. If we create the features vector in this way, the problem of incorrectly detection of legitimate webpages as phishing (false positive alarm) will be reduced.

The other rule that caused to produce false positive alarm is rule number 10. This rule is one of the essential rules in our proposed rule-based method, which can identify more than 80% of phishing webpages at the time of training the model.

Overall, our experiments demonstrate that the proposed rule-based extension can detect phishing webpages with 98.86% accuracy. Our proposed model with the lowest rules number produced higher true positive rate in compare to related rule-based methods (Aburrous et al., 2010; Ajlouni et al., 2013; Barraclough et al., 2013; Rami et al., 2014; Abdelhamid et al., 2014). Aburrous et al., in their study achieved 86.38% accuracy with a large number of rules set (306 rules). Methods of Barraclough et al., and Rami et al., produced 98.5% and 95.25% accuracy by using 15 and 17 rules respectively, whereas, we got more accuracy with our extension with

Table 6
Performance comparison of selected rules and the rules extracted directly from training dataset.

Rule set type	Number of rules	Acc.	Err.	Sensitivity	F1-measure
Rules extracted from proposed SVM model	10	98.59%	1.41%	1	0.9898
Rules extracted directly from training dataset	22	97.89%	2.11%	0.9966	0.9847

only 10 rules. We collected the results of other methods from respective papers and the test dataset is different for each method.

5.5. Runtime analysis

For the runtime performance evaluation of implemented extension, we use a desktop computer with a 2.0 GHz Core2Due processor and 2GB RAM. We evaluate the runtime performance with some phishing and legitimate webpages, which selected from our dataset randomly. In worst-case scenario, we got 50 ms and near to 12.7 ms on average delay until features extracted from a webpage. Total execution time of our extension on various webpages shows that it can extract features and classify the webpage based on features vector at 1,500 ms $\pm$ 4,800 ms, which is relatively acceptable.

6. Limitations

There are some limitations in our study. Firstly, our proposed feature sets are entirely depending on the webpage content. We assumed that the most attackers do not redesign a phishing webpage, and they use the legitimate page content only. They copy legitimate webpage content to another webpage (phishing) and then create a handler to send user information to their desired database. In reality, it might not be like this. Attacker may spend more time to redesign the phishing webpage and may change all reference addresses of page resource elements to make the phishing page less suspicious.

Another possible problem could be the limitation of our method in detecting phishing webpages with combination of image. We extract our proposed feature sets through tracing the webpage DOM. If an attacker use a flash media or an image of legitimate webpage instead of DOM in a phishing page, our method may not correctly detect and classify the webpage.

7. Conclusion and future works

Given wide range of the researches carried out as well as different ways provided to detect phishing attacks, the existing techniques are still not able to precisely identify these attacks and in many cases provide no accurate results. In this paper, we proposed two feature sets to improve the performance of detecting phishing attacks and preventing data loss in internet banking webpages. Our proposed feature sets, determine the relationship between the content and the URL of a page. This relation is measured by use of approximate string matching algorithm. We adopted a subset of relevant features from related works along with our proposed feature sets to create webpage features vector. Our proposed feature sets include four features to evaluate the page resource identity, and four features to identify the access protocol of page resource elements. These two feature sets will extract through examining the content and the page document object model (DOM). The approach used in proposed method is to determine the relationship between page content and page address, which based on the aforementioned feature sets. These features are independent from 3rd-party services such as search engines, web browser history and/or black/white lists. Our features vector consists of 17 features, which include a subset of relevant features (9 features), and the proposed features (8 features). To classify webpages by using our features vector, we employed support vector machine (SVM) algorithm.

The main contribution of this study is to provide some novel features to increase the performance of phishing detection methods. Our experiments indicate that the presented model by using the proposed feature sets along with some relevant features can detect phishing pages in internet banking with accuracy of 99.14% true positive and only 0.86% false negative alarm. Output of sensitivity analysis of proposed features in classification, demonstrates the positive effect of six features out of the eight proposed features. We used a related method in order to extract the hidden knowledge from the proposed model. We examined the extracted rules with the rules that extracted directly from training data. The results prove that our method have more accuracy than when we using the extracted rules from our model. We found the existing detection systems too slow with high false negative alarm to use with novice users. Therefore, we embed the extracted rules into a browser extension named PhishDetector to make our proposed method more functional and easy to use. Evaluating of the implemented browser extension indicates that it can detect phishing attacks in internet banking with high accuracy and reliability. With compare to related works, our rule-based method can detect phishing pages with zero false negative alarm with only 10 rules. Our presented rule-based method can detect zero-day phishing attacks too.

With the growing awareness of technology in all aspect of our life, hustlers always try to use new methods to target their victims too. It is necessary to improve methods and techniques to detect these frauds, and prevent more financial losses. This study reveals the importance of our proposed features. We believe that our model can detect phishing webpages if a good combination of relevant features is used. Our proposed feature sets are entirely depending on webpage content. In future studies we plan to include some new features in our model to decrease the dependency to the webpage content. As we said in limitations section, our method could not detect phishing webpage with non-HTML code. Therefor we also need to propose new methods to identify all type of webpages with high accuracy. Another important topic in this area requiring further investigation is how to detect phishing attacks in mobile devices. Nowadays, the popularity of smart phones has made them a common target of phishing attacks. Mobile users prefer to read email messages as soon as they arrive. Therefore, we need to propose new features and present new systems to detect such attack in mobile devices.

Acknowledgment

The authors would like to thank the reviewers for their detailed and useful comments.

References

Abdelhamid, N., Ayesh, A., & Thabtah, F. (2014). Phishing detection based associative classification data mining. *Expert Systems with Applications, 41*(13), 5948–5959. doi:10.1016/j.eswa.2014.03.019.

Abu-Nimeh, S., Nappa, D., Wang, X., & Nair, S. (2007). A comparison of machine learning techniques for phishing detection. *Proceedings of the Anti-Phishing Working Group eCrime Researchers Summit* (pp. 60–69). New York, NY: ACM. doi:10.1145/1299015.1299021.

Aburrous, M., Hossain, M. A., & Dahal, K. (2010). Experimental case studies for investigating e-banking phishing techniques and attack strategies. *Cognitive Computation, 2*(3), 242–253. doi:10.1007/s12559-010-9042-7.

Aburrous, M., Hossain, M., Thabatah, F., & Dahal, K. (2010). Intelligent detection system for e-banking phishing websites using fuzzy data mining. *Expert Systems with Applications, 37*(12), 7913–7921. doi:10.1016/j.eswa.2010.04.044.

Ajlouni, M., Hadi, W., & Alwedyan, J. (2013). Detecting phishing websites using associative classification. *European Journal of Business and Management, 5*(15), 36–40.

Alkhozae, M. G., & Batarfi, O. A. (2011). Phishing websites detection based on phishing characteristics in the webpage source code. *International Journal of Information and Communication Technology Research, 1*(9), 238–291.

ALmomani, A., Wan, T.-C., Altaher, A., Manasrah, A., ALmomani, E., Anbar, M., & Ramadass, S. (2012). Evolving fuzzy neural network for phishing emails detection. *Journal of Computer Science, 7*(8), 1099–1107. doi:10.3844/jcssp.2012.1099.1107.

Arade, M. S., Bhaskar, P., & Kamat, R. (2011). Antiphishing model with url & image based webpage matching. *International Journal of Computer Science and Technology (IJCST), 2*(2), 282–287.

Arun, S., Anandan, D., Selvaprabhu, T., Sivakumar, B., Revathi, P., & Shine, T. (2012, May). Detecting phishing attacks in purchasing process through proactive approach. *Advanced Computing: An International Journal (ACIJ), 3*(3), 81–93.

Barakat, N., & Bradley, A. (2010). Rule extraction from support vector machines: a review. *Neurocomputing, 74*(1-3), 178–190. doi:10.1016/j.neucom.2010.02.016.

Barraclough, P., Hossain, M., Tahir, M., Sexton, G., & Aslam, N. (2013). Intelligent phishing detection and protection scheme for online transactions. *Expert Systems with Applications, 40*(11), 4697–4706. doi:10.1016/j.eswa.2013.02.009.

Ben-Hur, A., & Weston, J. (2008). A user's guide to support vector machines. In O. Carugo, & F. Eisenhaber (Eds.), *Data Mining Techniques for the Life Sciences* (pp. 223–239). NJ, USA: NEC Labs America: Princeton. doi:10.1007/978-1-60327-241-4_13.

Cassidy, P. (2013). *Phishing Activity Trends Report, 3rd Quarter 2012*. Anti-Phishing Working Group (APWG), Lexington, MA, USA. APWG.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters, 27*(8), 861–874. doi:10.1016/j.patrec.2005.10.010.

Fu, A. (2006). *Web identity security: advanced phishing attacks and counter measures*. Doctor of Philosophy Thesis, University of Hong Kong, Hong Kong.

Gaurav, Mishra, M., & Jain, A. (2012). Anti-Phishing Techniques: A Review. *International Journal of Engineering Research and Applications (IJERA), 2*(2), 350–355.

Google, 2014 *Content Scripts*. Retrieved from Google Chrome Developers: https://developer.chrome.com/extensions/content_scripts.

He, J., Hu, H.-J., Harrison, R., Tai, P., & Pan, Y. (2006). Transmembrane segments prediction and understanding using support vector machine and decision tree. *Expert Systems with Applications, 30*(1), 64–72. doi:10.1016/j.eswa.2005.09.045.

He, M., Horng, S.-J., Fan, P., Khan, M., Run, R.-S., Lai, J.-L., & Sutanto, A. (2011). An efficient phishing webpage detector. *Expert Systems with Applications, 38*(10), 12018–12027. doi:10.1016/j.eswa.2011.01.046.

Huang, H., Qian, L., & Wang, Y. (2012). A SVM-based technique to detect phishing urls. *Information Technology Journal, 11*(7), 921–925. doi:10.3923/itj.2012.921.925.

Lakshmi, V, S., & Vijaya, M. S. (2012). Efficient prediction of phishing websites using supervised learning algorithms. *Procedia Engineering, 30*, 798–805. doi:10.1016/j.proeng.2012.01.930.

Levenshtein, V. (1966, February). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady, 10*(8), 707–710.

Manning, C., Raghavan, P., & Schütze, H. (2009). *Introduction to Information Retrieval*. Cambridge, England: Cambridge University Press.

Pannell, D. (1997). Sensitivity analysis of normative economic models: theoretical framework and practical strategies. *Agriculture Economics, 16*, 139–152.

Ramesh, G., Krishnamurthi, I., & Kumara, K. (2014). An efficacious method for detecting phishing webpages through target domain identification. *Decision Support Systems, 61*, 12–22. doi:10.1016/j.dss.2014.01.002.

Rami, M., Thabtah, F. A., & McCluskey, T. (2014). Intelligent rule-based phishing websites classification. *Information Security, IET, 8*(3), 153–160. doi:10.1049/iet-ifs.2013.0202.

Shahriar, H., & Zulkernine, M. (2012). Trustworthiness testing of phishing websites: a behavior model-based approach. *Future Generation Computer Systems, 28*(8), 1258–1271. doi:10.1016/j.future.2011.02.001.

Vapnik, V. (1995). *The nature of statistical learning theory*. New York: Springer.

Vrîncianu, M., & Popa, L. (2010). Considerations regarding the security and protection of e-banking services consumers' interests. *Amfiteatru Economic, 12*(28), 388–403.

Witten, I., Frank, E., & Hall, M. (2011). *Data Mining, Practical Machine Learning Tools and Techniques* (3rd Edition ed.). Burlington: Morgan Kaufmann Publishers, Elsevier Inc.

Xiang, G., Hong, J., Rose, C., & 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), 1–21. doi:10.1145/2019599.2019606.

Zhang, D., Yan, Z., Jiang, H., & Kim, T. (2014). A domain-feature enhanced classification model for the detection of Chinese phishing e-Business websites. *Information & Management, 51*(7), 845–853. doi:10.1016/j.im.2014.08.003.