World Wide Web (2017) 20:797–813
DOI 10.1007/s11280-016-0418-9

Two-stage ELM for phishing Web pages detection using hybrid features

Wei Zhang[1,2] · Qingshan Jiang[1,2] · Lifei Chen[3] ·
Chengming Li[1]

Received: 13 December 2015 / Revised: 7 July 2016 /
Accepted: 31 July 2016 / Published online: 29 September 2016

Abstract Increasing high volume phishing attacks are being encountered every day due to attackers' high financial returns. Recently, there has been significant interest in applying machine learning for phishing Web pages detection. Different from literatures, this paper introduces predicted labels of textual contents to be part of the features and proposes a novel framework for phishing Web pages detection using hybrid features consisting of URL-based, Web-based, rule-based and textual content-based features. We achieve this framework by developing an efficient two-stage extreme learning machine (ELM). The first stage is to construct classification models on textual contents of Web pages using ELM. In particular, we take Optical Character Recognition (OCR) as an assistant tool to extract textual contents from image format Web pages in this stage. In the second stage, a classification model on hybrid features is developed by using a linear combination model-based ensemble ELMs (LC-ELMs), with the weights calculated by the generalized inverse. Experimental results indicate the proposed framework is promising for detecting phishing Web pages.

Keywords Phishing Web page detection · Two-stage ELM · LC-ELMs · Linear combination model · Hybrid features

✉ Qingshan Jiang
qs.jiang@siat.ac.cn

Wei Zhang
wei.zhang1@siat.ac.cn

Lifei Chen
Clfei@fjnu.edu.cn

Chengming Li
cm.li@siat.ac.cn

[1] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China

[2] Shenzhen College of Advanced Technology, University of Chinese Academy of Sciences, Shenzhen, China

[3] Fujian Normal University, Fuzhou, China

1 Introduction

Phishing is an online scam for financial returns. Fraudsters forge the identity of well-known Websites and send emails, short message service (SMS) or instant messenger with an attached link to a phishing Website. Victims believe they have reached a trusted Website, and provide their debit or credit card numbers, PIN codes or other private information. Then, fraudsters steal the bank funds of victims, or send messages to victims' friends to obtain illegal economic interests. Phishing not only seriously causes pecuniary loss to Internet users, but also brings reputation loss to the phishing target Website, which has been one of the most serious problems in the Internet [24].

In order to detect constantly emerging phishing Web pages,various methods have been proposed, such as user education [3, 20], blacklist-based methods [10, 31] and heuristic-based methods [2, 34]. User education trains Internet users to identify phishing Web pages correctly by themselves, which is helpful for users only if they actually read the material [21]. In addition, users must be trained incessantly for identifying newly arrived phishing Web pages. However, users may not be trained continuously [19], so these methods are not in effect for some users. Blacklist-based methods check whether the URL of the current Web page matches any record in the blacklist. Therefore, they cannot identify the phishing URLs which are not in the blacklist in advance. Heuristic-based methods are considered effective in academia [21], which can identify phishing Web pages not seen before. Machine learning is the most extensive used tool in heuristic-based methods, and most of them take phishing Web page detection as a classification problem. The following two strategies can be implemented to construct an efficient classification model with high accuracy. One is using a classification algorithm with high efficiency. The other one is choosing a set of effective features.

Extreme Learning Machine (ELM) is one of the most popular machine learning algorithms for classification problem due to its advantage of high accuracy and fast learning speed. Therefore, this research takes it as the base classifier for phishing detection. As ensemble-based ELM tends to perform better than single ELM [5] in classification, this paper utilizes it for hybrid features classification, where a linear combination model is used to combine the base classifier for it considers the performance of each base classifier [22]. Generalized inverse is adopted in this study to calculate the weights of the linear combination model [5, 22].

Features of Web pages are essential for machine learning. Features have a major influence on phishing detection accuracy and time consumption. Most of literatures focus on URL-based, Web-based, HTML-based or their combinations [21, 25]. Some other researchers pay attention to textual content classification for phishing detection [36, 37]. Different types of features may provide information complementary to each other for classification. This research therefore proposes hybrid features for phishing detection for superior accuracy. Our hybrid features contain nine basic features, a rule-based feature that represents hyperlinks relationship, and two textual content features which are labels predicted by using ELM.

This paper focuses on improving the accuracy of phishing detection by proposing an efficient two-stage ELM framework using hybrid features. In the first stage, we take ELM as the classifier to predict labels of textual contents. The second stage utilizes a linear combination model-based ensemble ELMs (LC-ELMs) to construct detection module on hybrid features. The main contributions in this paper are summarized as follows.

1) A novel framework is proposed by making use of two-stage ELM to detect Phishing Web pages.
2) This paper proposes using labels of textual contents via classification models and hyperlinks relationship by using $if - then$ rules, combining with features from URL, HTML source code and the Web to represent Web pages.
3) LC-ELMs is proposed for hybrid features classification. Generalized inverse is utilized to solve the weights of the linear combination model.
4) We evaluate the importance of each feature by three different criteria including intra-class and inter-class distance based criterion (J_2 criterion), Information Gain (IG) and accuracy, and investigate the effectiveness of proposed framework in English phishing Web pages detection and Chinese phishing Web pages detection.

The remainder of this paper is structured as follows. Section 2 gives related work of phishing Web pages detection and reviews ELM algorithm for learning the Neural Network. Section 3 details how to represent the Web pages based on hybrid features. Section 4 describes the two-stage ELM framework for phishing detection and proposes LC-ELMs for Web pages classification. The performance of the proposed framework is tested for phishing Web pages detection in Section 5. Finally, Section 6 concludes the paper.

2 Related work

In this section, a sampling of related work on phishing Web pages detection is described, followed by ELM algorithm. As ELM is a machine learning algorithm, this section first discusses some typical machine learning methods for phishing Web pages detection. Then, we review ELM algorithm for classification.

2.1 Machine learning for phishing detection

Booming phishing instances have drawn growing attention, and multiple approaches have been developed to deal with phishing Web pages. A good anti-phishing approach must meet the challenges including realtime, ability to identify new URLs and effectiveness [17]. Machine learning is heuristic to detect phishing Web page, which has ability to detect zero-day phishing. Machine learning has been proved effective to detect phishing Web pages. This section will review some typical a pproaches that utilize machine learning methods for phishing Web pages detection.

Generally speaking, machine learning applied in phishing detection involves selecting appropriate features to represent the Web page and subsequently employing an operational machine learning method to distinguish phishing Web pages and legitimate ones. A number of such methods including Bayesian classifier [11, 36], Support Vector Machine (SVM) [11, 12], association rules [2, 9], Logistic Regression (LR) [9] and artificial Neural Networks [4, 29], have been used for phishing detection. They employ URL-based features [12, 29] such as length of URL and suspicious characters in the URL, or domain name features [4, 29] such as domain history and Website traffic, or make use of content-based features [36, 37] such as the page title, hyperlinks and meta description.

Zhang et al. [36] took Naive Bayes (NB) as a textual content feature classifier and Earth Movers Distance (EMD) method as an image classifier, and employed Bayesian theory to combine the results of the two classifiers. They used this model to determine if the given Web page was similar to eight protected pages including eBay, PayPal, RapidShare, HSBC, Yahoo, Alliance-Leicester, Optus and Steam. They generated very high accuracy. However, if the phishing Web page does not imitate any of the protected pages, this approach may fail to identify it. He et al. [12] took the search engine result as a feature, and combined content-based feature and URL-based features to detect phishing Web pages. They employed SVM as the classifier and generated 97 % true positive rate (TPR) and 4 % false positive rate (FPR). Gu et al. [11] first constructed classification model on URL-based features by using NB. If the Web's legality was suspicious, they extracted the HTML-based features and constructed classification model by SVM. Their approach generated 96.90 % TPR and 1.25 % FPR. Neda et al. [2] proposed an associate classification method to discover correlations among sixteen features and generate new rules, which introduced new types of useful features to enhance the classification accuracy.

Some researchers compared the effectiveness of different machine learning methods in phishing Web page detection. Xiang et al. [34] compared six machine learning methods for training the phishing detector, including SVM, LR, Bayesian Network (BN), J48 Decision Tree, Random Forest (RF) and Adaboost, and produced the result that BN performed better than other classifiers. They achieved an over 92 % TPR on unique testing phishing Web pages. Feroz et al. [9] compared LR, J48 Decision Tree, NB, BN and BFTree, and obtained the result that LR performed best among all the classifiers with 97.98 % accuracy.

Neural Network classifier has also been proposed in phishing detection due to it good classification performance. Mohammad et al. [29] made use of self-structuring Neural Network to construct detection model based on 17 features from URL and HTML source code, and they considered that Neural Network was a suitable classifier for phishing detection. Barraclough et al. [4] utilized Neuro-Fuzzy with five inputs including legitimate site rules, User-behavior profile, PhishTank, User-specific sites and Pop-Ups from E-mails based on 288 features to identify phishing Web pages, which mainly targeted at online transaction. The heuristic-based approaches have the ability to identify new phishing Web pages. However, most machine learning methods may mistake a legitimate Web page as a phishing one and produce a high FPR.

Zhuang et al. [37] focused on textual content-based feature classification. They extracted seven different types of features including tag 'Title', 'Keyword', 'H1-H6', 'Description', 'Link text', 'Alt' and 'String' to represent the Web page. Then, seven classifiers were built based on these different features. They classified 'Title', 'Keyword', 'H1-H6' by using associative rule, 'Description', 'Link text', 'Alt' by NB classifier, and 'String' by SVM. Finally, they combined the predicted results of each classifier for phishing detection. They obtained over 96 % TPR in all the experiments. Moreover, topic models are popular for text classification [32, 35]. Ramanathan et al. [32] utilized Latent Dirichlet Allocation to obtain distribution probabilities of topics and made use of Adaboost to build the classifier for phishing detection.

However, literatures that utilize content-based features sometimes encounter image format Web pages. They cannot extract the features from HTML source code, such as 'Brand' of the page, text of tag 'Title' and 'Keyword'. Dunlop et al. [8] utilized Optical Character Recognition (OCR) technique to convert an image into text, and then utilized search engine to retrieve the text. However, Search engine techniques may lead to high FPR for new Websites that have low search engine ranking.

Given the state of current researches, we also treat phishing Web pages detection as a classification problem and utilize some basic features which have been used as suspicious features in previous researches [2, 11, 12, 27, 29, 34], textual content-based features and rule-based feature. Textual content-based features and rule-based feature represent hidden information of textual content of Web pages and hidden relationship of hyperlinks respectively. This research also takes Neural Network as the classifier because of its ability to solve nonlinearity classification problems.

However, most of Neural Network learning algorithms have the disadvantages of slow training speed or local minimum. Meanwhile, phishing detection requires the classification method with high speed and accuracy. In order to utilize Neural Network in phishing detection, this paper learns Neural Network via ELM algorithm. Because of randomly assigned input weights and biases, single ELM may generate unstable results. In this paper, we train the Neural Network using ELM En times with different input weights and biases, get En classifiers and then combine the results of classifiers by the linear combination model.

2.2 ELM for classification

Neural Network has been used as the classifier to solve Web pages classification problem and proved to be an effective method for phishing Web pages detection [4, 29]. ELM is a learning algorithm for single hidden layer feed forward Neural Networks (SLFNs), which is based on least square to train the networks [13, 14]. ELM has been widely used in classification applications [7, 15, 16] due to its good performance in dealing with imprecision and nonlinearity efficiently. The ELM randomly assigns for input weights and biases, and analytically determines the output weights of the SLFNs [14].

The training data set is denoted as $\{(x_i, t_i)|x_i \in \mathbb{R}^n, t_i \in \mathbb{R}^m, i = 1, 2, , \ldots, N\}$. Denote K as the number of hidden nodes, randomly choose the input weights and biases, and calculate the output matrix of hidden neurons [13],

$$H = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \cdots & g(w_K \cdot x_1 + b_K) \\ \vdots & \ddots & \vdots \\ g(w_1 \cdot x_N + b_1) & \cdots & g(w_K \cdot x_N + b_K) \end{bmatrix} \tag{1}$$

where $w_j = [w_{j_1}, w_{j_2}, \ldots, w_{j_n}](j = 1, 2, \ldots, K)$ is input weight, b_j denotes the bias of the j-th hidden neuron, N denotes total number of input samples and $g(\bullet)$ is the activation function. The output of the Neural Network respecting to x_i is denoted as o_i [13],

$$o_i = \sum_{j=1}^{K} \beta_j g(w_j \cdot x_i + b_j) \quad i = 1, 2, \ldots, N \tag{2}$$

where $\beta_j = [\beta_{j_1}, \beta_{j_2}, \ldots, \beta_{j_m}]$ is the j-th weight connecting the j-th hidden neuron and output neurons. There exists β that makes the SLFNs appropriate N samples with zero error, which means that $\sum_{i=1}^{N} \|o_i - t_i\| = 0$ [13].

$$\sum_{j=1}^{K} \beta_j g(w_j \cdot x_i + b_j) = t_i \quad i = 1, 2, \ldots, N \tag{3}$$

Rewrite (3) as a linear system [13],

$$H\beta = T \tag{4}$$

The ELM trains the SLFNs based on minimum norm least-square, which is equivalent to finding a least-square $\hat{\beta}$ of the linear system.

$$\|H\hat{\beta} - T\| = min_{\beta}\|H\beta - T\| \tag{5}$$

For the general cases, the number of hidden nodes is much less than the number of samples, and the smallest norm least-square solution of (4) is,

$$\hat{\beta} = H^{\dagger}T \tag{6}$$

where $H^{\dagger}$ denotes the Moore-Penrose generalized inverse operation of H. $\hat{\beta}$ is the unique minimum norm least-square solution of the (4) [13]. Then, the decision function is,

$$f(x) = sign(H(x)\hat{\beta}) \tag{7}$$

Input weights and biases are randomly assigned, single ELM may result in misclassification for certain samples [6]. Optimal Pruning Extreme Learning Machine (OPELM) is an improved ELM method and shows robustness for classification [28]. Ensemble-based ELMs [6, 23, 26] are proposed for higher accuracy. Majority voting strategy is used to fuse the results of ELM classifiers in most of the experiments, which gives no consideration to performance of each classifier. Therefore, linear combination model [5, 22] is studied by some researchers to combine the ensemble ELMs. Cao et al. [5] constructed weighted voting-based ELM for classification based on the linear combination model. Similarly, Laencina et al. [22] utilized linear combination model to combine the ELMs for regression prediction, which exhibited superior results. In this paper, LC-ELMs is proposed to construct the classification model on hybrid features, which considers the performance of each classifier.

3 Hybrid features for phishing detection

Feature representation is essential, which provides data matrix for training and detecting phishing Web pages. This research represents the Web pages by using nine basic features, eight of which have been used as suspicious features in previous researches, one of which is a new feature that is statistical from the first data set in this research, two textual content-based features, and a rule-based feature which represents the relationship between the internal and external links. Phishing Web pages used for feature generation are obtained from Phishtank http://www.phishtank.com/, and legitimate ones are from Statscrop http://www.statscrop.com/Websites/top/. The features used in this research are explained as follows, which serve as the input data of the second stage of ELM.

3.1 Basic features

Basic features are directly analyzed from URL or HTML, or obtained by submitting the URL to third-party tools. Features 1-8 are adopted from literatures.

1) IP address [2, 11, 12, 27, 29, 34]: almost all the researches based on URL features figure out that if the domain name of a URL is instead by IP address or hexadecimal format, it is a suspicious URL.

2) @ in the URL [2, 11, 12, 29, 34]: @ symbol is used to hide the suspicious part of the URL. Usually, the part of the URL before @ symbol is a credited URL which is familiar to the users, and @ symbol redirects Internet users to visite a phishing URL after it.
3) Sub-domain [2, 11, 12, 29, 34]: famous domain names are often added to the phishing URL as a sub-domain, which is used to confuse Internet users and make they believe they visite a legitimate Website.
4) Length of URL [2, 29]: a long URL is often utilized which contains famous domain name in order to confuse Internet users. For instance, Internet users are familiar with 'paypal'. When they see there is a 'paypal' in the long URL, they think there's hardly a doubt that it is a legitimate Web page, and login to it. Therefore, they are deceived.
5) Website traffic [2, 29]: Website traffic is a criterion to measure the importance of a Website. Legitimate Websites usually have a high Website traffic ranking in the Alexa database, while phishing Websites may have a low Website traffic ranking due to their short life [2].
6) DNS record [2, 27, 29]: DNS record records domain information of a Website. If DNS record is missing or empty, the research takes this feature as suspicious.
7) Age of domain [2, 29, 34]: the intention of phishing is to acquire private information as fast as possible, so most of the phishing Web pages have a relatively younger age than legitimate ones.
8) Login form [34]: almost all of the phishing attacks aim at stealing private information of the Internet users by means of using a fake login form in the HTML [19]. Therefore, this research takes 'login form' as a feature. In order to prevent mistaking the common search form as the login form, we search for the word 'search' in the same scope. If there is a 'form' but no 'search' in the same scope, we consider this feature as a suspicious one.
9) Number of URLs: the feature counts the number of URLs linking to the current Website. Shown as Figure 1, most of the phishing Web pages have less than 10 URLs linking to them, while legitimate cases tend to have more. Therefore, this research also takes it as a feature to identify the type of Web pages. This feature is obtained by submitting the URL to 'Who.is' database.

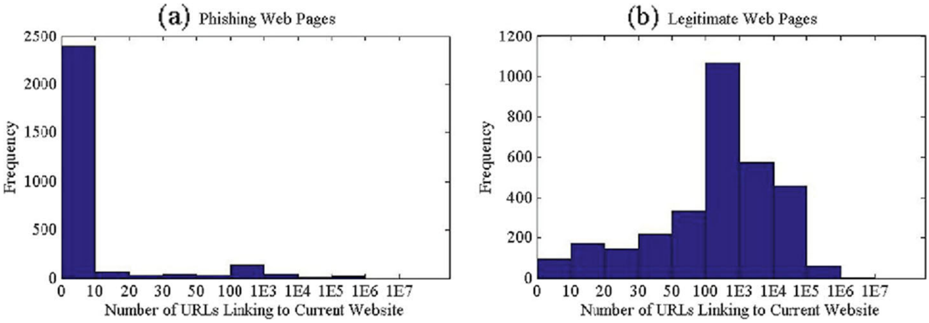

Figure 1 Frequency of URLs Linking to the Current Website

3.2 Textual content-based features

These features represent if the textual contents are suspicious, which are the predicted labels obtained from textual classification models by using ELM. We extract text of tag ‘Title’ from the HTML source code to construct the ‘Title’ textual classification model. Then, we extract tag ‘Head’ and ‘Body’ to construct the ‘String’ textual classification model. The steps of ‘Title’ textual content generation are described as follows.

1) Extract the text of tag ‘Title’ from the HTML source code by utilizing Jsoup.
2) Remove stop words from the text.
3) Utilize information extraction approach Term Frequency-Inverse Document Frequency (TF-IDF) [33] to obtain the score of each term in the text.
4) Select N terms with highest TF-IDF value in phishing data set, and take them as keywords of tag ‘Title’. If the i-th keyword is contained in the tag ‘Title’ of the j-th Web page, set $flag_{ij}$ to ‘1’. Otherwise, set $flag_{ij}$ to ‘0’. Then, we can obtain the data matrix for classification.
5) Train the data matrix by using ELM, and get the type of tag ‘Title’ textual content feature. If the tag ‘Title’ textual content feature is suspicious, set $TitleFeature$ to ‘1’. Otherwise, set $TitleFeature$ to ‘0’.

Likewise, the value of $StringFeature$ which represents textual content of tag ‘Head’ and ‘Body’ of the HTML source code can be obtained. This research also implements OCR to extract text from image format Web pages. However, we cannot determine the extracted text is from ‘Title’ or ‘String’. Therefore, we search for keywords of ‘Title’ and ‘String’ separately in the same extracted text, and obtain two data matrixes.

3.3 Rule-based feature

Legitimate Web pages have a tendency to point at the Web pages with the same domain name. Most of the phishing Web pages are partial replicas [1], and link to other Web pages with different domain name. Shown as Figure 2a, the number of internal links is much more than external ones in majority legitimate Web pages, which is opposite in phishing samples, shown as Figure 2b.

Then, we can define $if - then$ rules to determine whether the relationship of internal and external links is suspicious. Denote this feature as ‘$linkRelation$’.

Rule1: If $N_{IL} >= 0$ and $N_{EL} >= N_{IL}$, then $linkRelation$ is suspicious
Rule2: If $N_{IL} > 0$ and $N_{EL} < N_{IL}$, then $linkRelation$ is legitimate

where N_{IL} is the number of internal links and N_{EL} is the number of external links. If the $linkRelation$ is suspicious, $linkRelation = 1$. Otherwise, $linkRelation = 0$.

4 Two-stage ELM for phishing detection

4.1 Framework description

The overview of our framework is illustrated in Figure 3. This framework is based on two-stage ELM for phishing Web page detection. In the first stage, ELM is used to construct textual content classification model, and predicts the labels of textual contents.

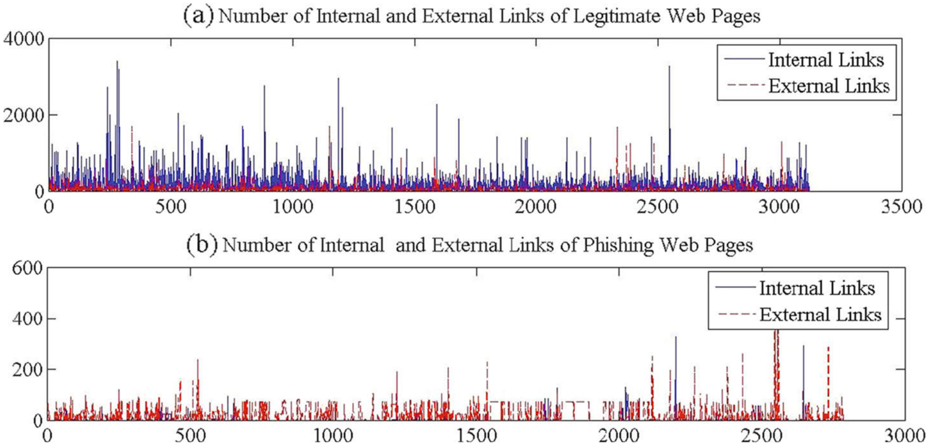

Figure 2 Number of Internal and External Links in Web Pages

In the second stage, ELM is taken as base classifier, and a linear combination model is utilized as the combiner to combine each classifier. Details of the detection framework are described as follows.

1) Feature extractor. The features will be extracted in this component. URL is the original data, based on which, all the basic features are extracted and HTML source code will be crawled. Then, textual contents and hyperlinks of the Web page are extracted from HTML source code by utilizing Jsoup.
2) ELM classifier. This component utilizes ELM to predict the labels of textual contents.
3) LC-ELMs for hybrid features classification. Ensemble-based ELM classifiers are employed for training. In this component, a linear combination model is utilized as the combiner to combine the result of each base classifier.
4) Detection module. This module is used to identify if the given Web page is phishing.

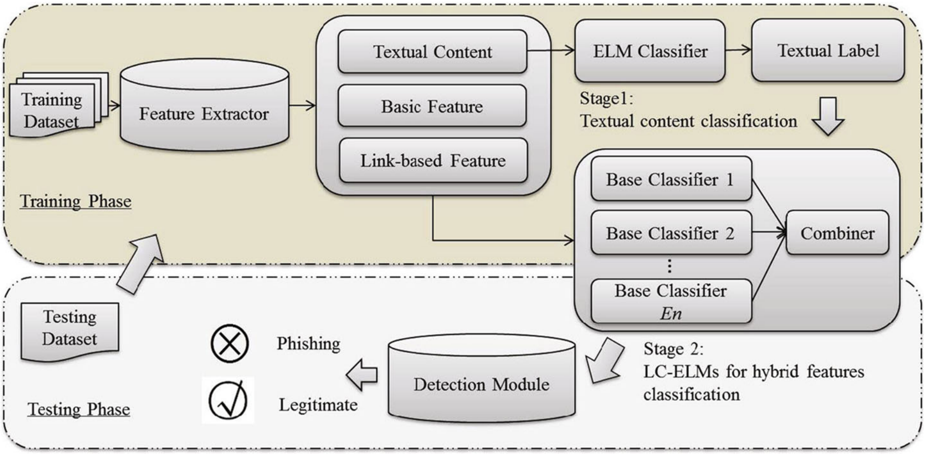

Figure 3 Phishing Web Pages Detection System

4.2 LC-ELMs for hybrid features classification

ELM algorithm has been shown in Section 2.2. It is obvious that there is no iteration in ELM, which overcomes slow speed in the training step. Huang et al. [14] have discussed that ELM algorithm can obtain the minimum training error and $\hat{\beta}$ is unique, which avoid the local optimal solution [7]. However, due to the fact that the input weights and biases are randomly assigned, single ELM may result in misclassification for certain samples [6]. Therefore, this research makes use of an ensemble-based ELM method to construct a classification model for hybrid features, which could generate stable and high accuracy results.

In the training steps, we train the ELM En times, and get En classification results. For the i-th sample x_i, the linear combination model is described as,

$$F(x_i) = \sum_{j=1}^{En} f_j(x_i) v_j \tag{8}$$

where $i = 1, 2, \ldots, N$, $f_j(x_i)$ is the predicted label by using the j-th ELM for the i-th sample, and v_j is the weight of the j-th ELM. Let $\sum_{i=1}^{N} \|t_i - F(x_i)\| = 0$. This is actually to determine the solutions $v = [v_1, v_2, \ldots, v_{En}]^{\mathrm{T}}$ of linear equations. According to the (4), (5) and (6), we can obtain,

$$\hat{v} = \mathcal{F}(x)^{\dagger} T \tag{9}$$

where $\mathcal{F}(x) = \begin{bmatrix} f_1(x_1) & \cdots & f_{En}(x_1) \\ \vdots & \ddots & \vdots \\ f_1(x_N) & \cdots & f_{En}(x_N) \end{bmatrix}$, $\mathcal{F}(x)^{\dagger}$ is the generalized inverse of $\mathcal{F}(x)$ and $\hat{v}$ is the unique minimum norm least-square solution of the equation. Then, we calculate $F(x_i)$ by (8), and $sign(F(x_i))$ is the final decision of the i-th Web page. The LC-ELMs algorithm is shown as below.

Algorithm 1 LC-ELMs

```
Input: training data, $D = \{(x_1, t_1), (x_2, t_2), \ldots, (x_N, t_N)\}$; ensemble size, $En$.
Output: base classifiers $f_j$ $(j = 1, 2, \ldots, En)$ and their weights $\hat{v}$
for $j = 1$ to $En$ do
    train base classifier $\xi_j = f_j(D)$
end
for $i = 1$ to $N$ do
    for $j = 1$ to $En$ do
        $f_j(x_i) = \xi_j(x_i)$
    end
end
Calculate weights $\hat{\boldsymbol{v}} = \mathcal{F}(\boldsymbol{x})^{\dagger}\boldsymbol{T}$;
```

OPELM is robust in some fields and generates relatively high accuracy. Majority voting and Adaboost method are popular in the ensemble classification field. This paper takes OPELM, Majority Voting ELMs (MV-ELMs) and Adaboost ELM as comparisons.

5 Experiments and results

This section first details data sets, and then measures the importance of individual feature by three different criteria. At last, we utilize two data sets to evaluate the effectiveness of the proposed framework.

5.1 Data sets

The details of data sets in the experiments are shown in Table 1.

In data set 1, phishing samples are English Websites that are collected from Phishtank http://www.phishtank.com/, which is a free community Website and reports newly discovered phishing Web pages. Legitimate samples are downloaded from Statscrop http://www.statscrop.com/Websites/top/, which provides top 1,000,000 sites on the Web, ordered by Alexa Traffic Rank. We randomly select 5000 Websites from Statscrop, choose the English Websites, and get 3121 legitimate samples. Data set 1 is used to evaluate the effectiveness of each feature and the performance of the proposed framework.

Data set 2 is collected for measuring the performance of the proposed framework for Chinese Web pages detection. Phishing samples are obtained by search engine, or from phishing SMSs and emails. The legitimate samples are also collected from Statscrop.

The ELM classifier and LC-ELMs on all the data sets are carried out in MATLAB 2012B environment running in Intel Pentium G850 Core2 CPU clocked at 2.89 GHz with 2GB RAM. The feature extractor is carried out in JAVA, and Jsoup is used to parse the HTML source code and extract textual content and hyperlinks of the Web pages. OCR is utilized to extract textual content from image format Web pages, which is downloaded from http://code.google.com/p/tesseract-ocr/. The code used for ELM is from http://www.ntu.edu.sg/home/egbhuang/. The sigmoidal function $g(w, x, b) = 1/(1+e^{-(wx+b)})$ is taken as the activation function in all the simulations. The input weights and biases are randomly generated from the range $[-1, 1]$.

5.2 Evaluation of individual feature

In order to verify the effect of our proposed features, we evaluate each feature by utilizing different criteria including J_2 criterion, IG and accuracy. The criteria are described as follows.

***J_2* criterion** This criterion measures class distance, which represents the ability of features for classification. Lager value of J_2 has more capability of distinguishing.

$$J_2 = \frac{S_b}{S_w} \tag{10}$$

Table 1 Details of Data Sets in Experiments

Data set	Data sources	Data size	Validation method
1	Phishtank http://www.phishtank.com/	2784 phishing samples	5-fold
	Statscrop http://www.statscrop.com/Websites/top/	3121 legitimate samples	cross-validation
2	SMS, Email,	500 phishing samples	5-fold
	Search Engine, Statscrop http://www.statscrop.com/Websites/top/	500 legitimate samples	cross-validation

where $S_w = \sum_{i=1}^{2} P_i E\{(x_i - \bar{m}_i)(x_i - \bar{m}_i)^T | x_i \in class_i\}$ is intra-class scatter, P_i is the probability of i-th class and $\bar{m}_i$ is the mean of the i-th class. $S_b = \sum_{i=1}^{2} P_i(\bar{m}_i - \bar{m})(\bar{m}_i - \bar{m})^T$ is inter-class scatter, and $\bar{m}$ is the mean of all the samples.

IG This index measures information of a feature contributing to the classification. This index is usually used for feature selection, and the feature with lager value will be selected. This paper utilizes this index to measure the importance of each feature.

$$IG(X) = -\sum_{i=1}^{2} p(c_i)log(p(c_i)) + \sum_{j=1}^{2} p(x_j)(\sum_{i=1}^{2} p(c_i|x_j)log(p(c_i|x_j))) \tag{11}$$

where $p(c_i)$ is the probability of the class c_i, $p(x_j)$ is the probability of feature x_j, and $p(c_i|x_j)$ is the probability of c_i with regard to x_j.

Accuracy This index measures correct classification ratio by using single feature.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

where TP is the number of correctly classified phishing samples, FN is the number of phishing ones that are misclassified as legitimate ones, FP is the number of legitimate ones that are misclassified as phishing ones, and TN is the number of correctly classified legitimate ones. Table 2 shows the performance of each feature.

Shown as Table 2, the new proposed features TitleFeature, StringFeature and linkRelation have high value in terms of all the criteria. OCR has a certain effect for improving the performance of textual content-based features.

We also compare our proposed hybrid features with original textual contents and hyperlinks, which are not processed by using ELM classification and $if - then$ rule separately.

Table 2 Performance of Individual Feature

Feature	J_2	IG	Acc(%)
IP address	0.06	0.08	55.27
@ in the URL	0.01	0.01	53.24
Sub-domain	0.47	0.05	67.67
Length of URL	0.28	0.03	65.96
Website traffic	1.21	0.10	77.78
DNS record	5.21	0.20	92.01
Age of domain	3.00	0.15	88.72
Login form	0.22	0.02	65.89
Number of URLs	4.76	0.18	91.82
TitleFeature	0.87	0.078	74.70
TitleFeature(OCR)	0.90	0.080	75.02
StringFeature	2.37	0.131	86.71
StringFeature(OCR)	2.46	0.134	86.98
linkRelation	2.29	0.14	85.44

We implement 100 simulations to test each combination by using single ELM, and list average value of accuracy, shown as Figure 4.

Figure 4 shows that basic features combining with our proposed features manage to perform better than basic features with original textual contents and hyperlinks in all the combinations, which demonstrates the effectiveness of our proposed hybrid features. Our hybrid features perform best in terms of accuracy among all the combinations.

5.3 Results and analysis

In order to evaluate the performance of LC-ELMs, we compare it with ELM, OPELM, MV-ELMs, and Adaboost. Optimal number of hidden nodes is the only parameter to be determined for single ELM. We set the number of hidden nodes to be 5 to 20, and construct 100 simulations for each number using 5-fold cross-validation to determine this parameter, and get the result that 10 is the optimal number. Number of hidden nodes and ensemble size are the parameters in MV-ELMs, Adaboost ELMs and LC-ELMs. We set the numbers of hidden nodes to be 10 in each base ELM classifier. We set ensemble size to be 5, 10, 15, 20, 25, 30, and implement 100 simulations for each of the given ensemble size. Testing accuracies for each ensemble size are shown in Figure 5.

Figure 5 presents that accuracy of LC-ELMs is higher than MV-ELMs in all the experiments with the same ensemble size. Adaboost ELMs with small ensemble size generates higher accuracy than LC-ELMs. However, when the ensemble size is larger than 20, LC-ELMs obtain higher accuracy. It is evident that when ensemble size is set to be 25, results of this experiment are the best. Therefore, this research takes 25 as the optimal ensemble size for LC-ELMs, and applies it for Chinese phishing Web pages detection.

Table 3 summaries the comparison results in terms of accuracy, standard deviation (SD) of accuracy, FPR and training time. The values are averaged across 100 simulations. We also list results of Back-Propagation (BP) Neural Network, SVM, NB and *k*-Nearest Neighbors (*k*-NN) as comparisons.

Table 3 presents that single ELM generates lower accuracy than BP Neural Network and SVM, but it costs 0.014s, which is much less than SVM and BP Neural Network. NB encounters lowest accuracy and highest FPR, and it costs more training time than single

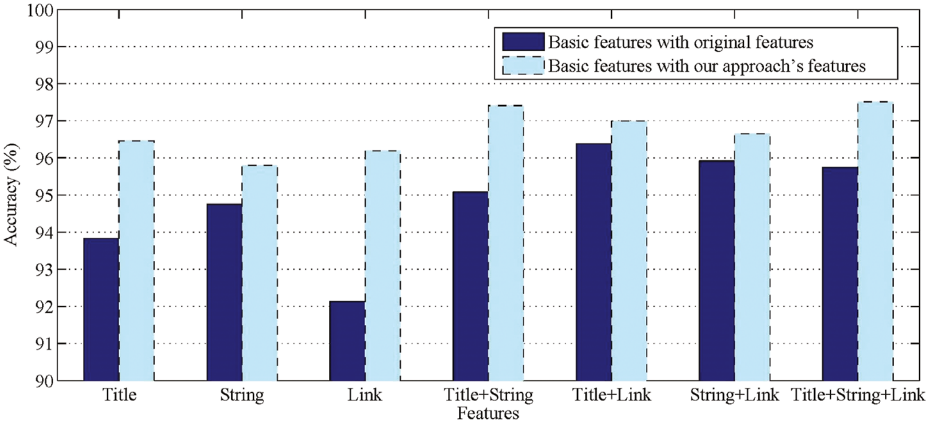

Figure 4 Accuracy from Different Feature Combinations

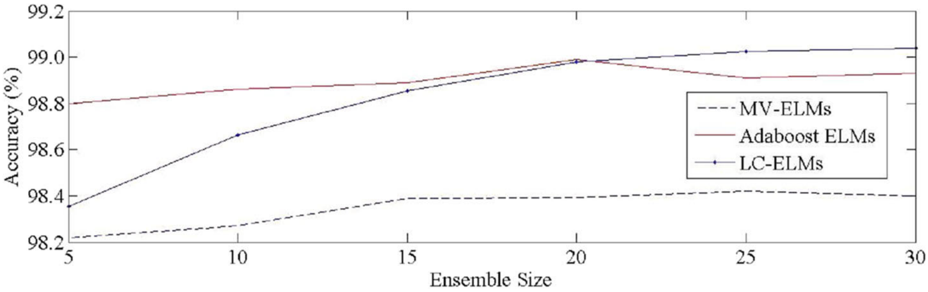

Figure 5 Accuracy from Different Ensemble Size

ELM. K-NN encounters 5.13 % FPR, which is some high for phishing detection. LC-ELMs generates higher accuracy and lower FPR than BP Neural Network and SVM, and the training time is also lower than BP Neural Network.

OPELM generates higher accuracy and lower FPR than ELM, but takes more time. Accuracy of MV-ELMs, Adaboost ELMs and LC-ELMs in all the cases is always higher than single ELM. Meanwhile, FPR and SD values are lower than single ELM's. It is demonstrated that ensemble-based ELMs achieve better performance than a single ELM. Adaboost ELMs cost more training time than LC-ELMs. Especially, FPR values are reduced to some extent via LC-ELMs. LC-ELMs also generates lower SD than MV-ELMs and Adaboost ELMs, which indicates LC-ELMs performs best among all the ensemble-based ELMs methods.

We compare our framework with the other two approaches in detecting English phishing Web pages. The first one [37] utilized textual contents to detect phishing, but did not use URL-based features. The other approach [29] made use of URL-based, Web-based and HTML-based features, but did not use textual content of Web pages. We simulate their experiments using our data set 1, and Figure 6 gives a graphical illustration of the performance of different approaches. Comparison indicates that our framework performs superior than the other two approaches with higher accuracy, lower FPR and lower FNR.

Table 3 Results of Different Classifiers

Classifier	Accuracy(%)	SD	FPR(%)	Training time(s)
ELM	97.51	0.97	1.43	0.014
BP Neural Network	98.38	0.06	0.98	2.66
SVM	98.73	–	1.12	0.39
NB	94.51	–	9.07	0.074
k-NN	97.02	–	5.13	0.32
OPELM	98.35	0.38	0.94	3.91
Adaboost ELMs	98.93	0.86	1.08	1.21
MV-ELMs	98.42	0.24	0.68	1.01
LC-ELMs	99.04	0.16	0.53	1.16

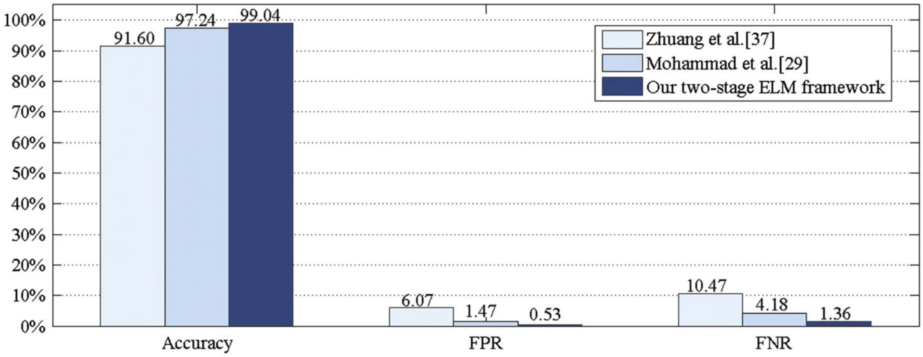

Figure 6 Comparison of the Approaches' Performances in Detecting English Phishing Web Pages

5.4 Applying the framework for chinese web pages detection

This experiment is constructed to measure the performance of our framework in Chinese phishing Web pages detection. Different from English Web pages, word segmentation is needed to deal with textual content before the step of 'remove stop word'. This research utilizes ICTCLAS (Institute of Computing Technology, Chinese Lexical Analysis System) [18] to segment words. Table 4 presents the performance of the proposed framework and the other two approaches in detecting Chinese phishing Web page using data set 2.

The comparison results show that our approach performs best in terms of accuracy, FPR and FNR. Zhuang et al. [37]'s approach cannot deal with the phishing Web pages which are image format or the HTML source code is obfuscated, and results in high FNR. If the phishing page is a sub page of a legitimate Website, such as http://www.zgdjzyw.com/qqdrsign=010a6, Mohammad et al. [29]'s approach may fail to detect it. Our framework can detect some phishing Web pages with image format by using URL-based and Web-based features. That is why our approach can generate higher accuracy than Zhuang et al. [37]' approach. Through textual content classification, our approach can identify some Web pages which are the sub pages of legitimate Websites. Therefore, our approach generates a little higher accuracy than Mohammad et al. [29]'s.

Besides, the time consumption on training the two-stage ELM is 1.21 seconds, which is less than Zhuang et al. [37]'s and Mohammad et al. [29]'s training time. The average time of detecting a Web page is 1.89 seconds, which is less than the tolerable waiting time for Web users (2 seconds) [30, 36]. As a consequence, our two-stage ELM is efficient in detecting phishing Web pages.

Table 4 Results of Chinese Phishing Web Pages Detection

Approach	Accuracy(%)	FPR(%)	FNR(%)
Zhuang et al. [37]	90.50	3.00	16.00
Mohammad et al. [29]	96.50	3.00	4.00
Our two-stage ELM framework	97.50	2.00	3.00

6 Conclusions and future work

Although a variety of approaches have been used to mitigate phishing attacks, they are continuing to be a serious problem for the Internet security and become more and more complicated. This paper presents a novel framework by using two-stage ELM to detect phishing Web pages. In this framework, hybrid features consist of basic features, rule-based feature and labels of textual content predicted by ELM. LC-ELMs is proposed to construct the detection model. Experimental results show that the LC-ELMs performs best in respect of accuracy, FPR and training time. Furthermore, we also compare the proposed framework with another two approaches for English and Chinese phishing detection, and obtain the result that the proposed framework demonstrates superior performance for both English and Chinese phishing detection. Future development of our framework intends to apply incremental method to update the detection model, and topic models such as [35] for higher accuracy of textual content classification.

Acknowledgements This research work is supported by Special Fund on Guangdong Province Chinese Academy of Sciences Comprehensive Strategic Cooperation (NO.2013B091300019), Shenzhen Fundamental Research Foundation (NO.CXZZ20150813155917544, JCYJ20150630114942277), and Guangdong National Natural Science Foundation of China (NO.U1401258).

References

1. Abbasi, A., Chen, H.: A comparison of tools for detecting fake Websites. Computer **42**(10), 78–86 (2009)
2. Abdelhamid, N., Ayesh, A., Thabtah, F.: Phishing detection based associative classification data mining. Expert Syst. Appl. **41**(13), 5948–5959 (2014)
3. Arachchilage, N.A.G., Love, S.: A game design framework for avoiding phishing attacks. Comput. Hum. Behav. **29**(3), 706–714 (2013)
4. Barraclough, P.A., Hossain, M.A., Tahir, M.A., Sexton, G., Aslam, N.: Intelligent phishing detection and protection scheme for online transactions. Expert Syst. Appl. **40**(11), 4697–4706 (2013)
5. Cao, J.J., Kwong, S., Wang, R., Li, K.: A weighted voting method using minimum square error based on Extreme Learning Machine. In: Proceedings of International Conference on Machine Learning and Cybernetics, 1, 411–414 (2012)
6. Cao, J., Lin, Z., Huang, G.B., Liu, N.: Voting based extreme learning machine. Inf. Sci. **185**(1), 66–77 (2012)
7. Ding, S., Zhao, H., Zhang, Y., Xu, X., Nie, R.: Extreme learning machine: algorithm, theory and application. Artif. Intell. Rev. **44**(1), 103–115 (2013)
8. Dunlop, M., Groat, S., Shelly, D.: GoldPhish: Using Images for Content-Based Phishing Analysis. In: Proceedings of International Conference on Internet Monitoring and Protection, 123-128, IEEE (2010)
9. Feroz, M.N., Mengel, S.: Examination of data, rule generation and detection of phishing URLs using online logistic regression. In: Proceddings of 2014 IEEE International Conference on Big Data, IEEE, 241-250 (2014)
10. Google Safe Browsing, https://developers.google.com/safe-browsing/?hl=zh-CN
11. Gu, X., Wang, H., Ni, T.: An Efficient Approach to Detecting Phishing Web. J. Comput. Inf. Syst. **9**(14), 5553–5560 (2013)
12. He, M., Horng, S.J., Fan, P., Khan, M.K., Run, R.S., Lai, J.L., et al.: An efficient phishing Webpage detector. Expert Syst. Appl. **38**(10), 12018–12027 (2011)
13. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: a new learning scheme of feedforward Neural Networks. In: Proceedings of IEEE International Joint Confrence on Neural Networks, 2, 985-990, IEEE (2004)
14. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: theory and applications. Neurocomputing **70**(1-3), 489–501 (2006)
15. Huang, G.B., Ding, X.J., Zhou, H.M.: Optimization method based extreme learning machine for classification. Neurocomputing **74**(1-3), 155–163 (2010)
16. Huang, G.B., Zhou, H.M., Ding, X.J., Zhang, R.: Extreme learning machine for regression and multiclass classification. IEEE Trans. Syst. Man Cybern. B Cybern. **42**(2), 513–529 (2012)

17. Huang, D., Xu, K., Pei, J.: Malicious URL detection by dynamically mining patterns without pre-defined elements. World Wide Web **17**(6), 1375-1394 (2014)
18. ICTCLAS, http://ictclas.nlpir.org/
19. Iraqi, Y., Jones, A., Khonji, M.: Phishing detection: a literature survey. IEEE Commun. Surv. Tutorials **15**(4), 2091–2121 (2013)
20. Kumaraguru, P., Sheng, S., Acquisti, A., Cranor, L.F., Hong, J.: Lessons from a real world evaluation of anti-phishing training. In: Proceedings of eCrime Researchers Summit, 1-12, IEEE (2008)
21. Kumaraguru, P., Sheng, S., Acquisti, A., Cranor, L.F., Hong, J.: Teaching johnny not to fall for phish. ACM Trans. Internet Technol. **10**(2), 890–895 (2010)
22. Laencina, P.J.G.: Improving predictions using linear combination of multiple extreme learning machines. Inf. Technol. Control **42**(1), 86–93 (2013)
23. Lan, Y., Soh, Y.C., Huang, G.B.: Ensemble of online sequential extreme learning machine. Neurocomputing **72**, 3391–3395 (2009)
24. Li, S., Schmitz, R.: A novel anti-phishing framework based on honeypots. In: Proceedings of eCrime Researchers Summit, 1-13, IEEE (2009)
25. Li, Y., Chu, S., Xiao, R.: A pharming attack hybrid detection model based on IP addresses and Web content. Optik-Inter. J. Light and Electron Optics **126**, 234–239 (2015)
26. Liu, N., Wang, H.: Ensemble based extreme learning machine. IEEE Signal Process Lett. **7**(8), 754–757 (2010)
27. Ma, J., Saul, L.K., Savage, S., Voelker, G.M.: Beyond blacklists: learning to detect malicious Web sites from suspicious URLs. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1245-1254, ACM (2009)
28. Miche, Y., Sorjamaa, A., Bas, P., Jutten, C., Lendasse, A.: OP-ELM: optimally pruned extreme learning machine. IEEE Trans. Neural Netw. **21**(1), 158–62 (2010)
29. Mohammad, R.M., Thabtah, F., Mccluskey, L.: Predicting phishing Websites based on self-structuring Neural Network. Neural Comput. & Applic. **25**(2), 443–458 (2014)
30. Nah, F.H.: A study on tolerable waiting time: how long are Web users willing to wait? Behav. Inform. Technol. **23**(3), 153–163 (2003)
31. Netcraft, http://www.netcraft.com/anti-phishing
32. Ramanathan, V., Wechsler, H.: Phishing Website detection using Latent Dirichlet Allocation and AdaBoost. In: Proceedings of IEEE International Conference on Intelligence and Security Informatics, 102–107 (2012)
33. Salton, G., McGill, M.: Introduction to modern information retrieval. McGraw-Hill (1983)
34. Xiang, G., Hong, J., Rose, C.P., Cranor, L.: CANTINA+: a feature-rich machine learning framework for detecting phishing Web sites. ACM Trans. Inf. Syst. Secur. **14**(2), 1–28 (2011)
35. Yao, W., He, J., Wang, H., Zhang, Y., Cao, J.: Collaborative topic ranking: leveraging item meta-data for sparsity reduction. In: Proceedings of AAAI, 374-380 (2015)
36. Zhang, H., Liu, G., Chow, T.W.S., Liu, W.: Textual and visual content-based anti-phishing: a bayesian approach. IEEE Trans. Neural Netw. **22**(10), 1532–1546 (2011)
37. Zhuang, W.W., Jiang, Q.S.: Intelligent anti-phishing framework using multiple classifiers combination. J. Comput. Inf. Syst. **8**(17), 7267–7281 (2012)