Application of Feature Engineering for Phishing Detection

Wei ZHANG\textsuperscript{(a)}, Member, Huan REN\textsuperscript{(b)}, and Qingshan JIANG\textsuperscript{(c)}, Nonmembers

SUMMARY  Phishing attacks target financial returns by luring Internet users to expose their sensitive information. Phishing originates from e-mail fraud, and recently it is also spread by social networks and short message service (SMS), which makes phishing become more widespread. Phishing attacks have drawn great attention due to their high volume and causing heavy losses, and many methods have been developed to fight against them. However, most of researches suffered low detection accuracy or high false positive (FP) rate, and phishing attacks are facing the Internet users continuously. In this paper, we are concerned about feature engineering for improving the classification performance on phishing web pages detection. We propose a novel anti-phishing framework that employs feature engineering including feature selection and feature extraction. First, we perform feature selection based on genetic algorithm (GA) to divide features into critical features and non-critical features. Then, the non-critical features are projected to a new feature by implementing feature extraction based on a two-stage projection pursuit (PP) algorithm. Finally, we take the critical features and the new feature as input data to construct the detection model. Our anti-phishing framework does not simply eliminate the non-critical features, but considers utilizing their projection in the process of classification, which is different from literatures. Experimental results show that the proposed framework is effective in detecting phishing web pages.

key words: phishing detection, feature engineering, feature selection, feature extraction, two-stage projection pursuit

1. Introduction

As Internet becomes ubiquitous and there is a bulge in the number of Internet users, especially the rapid development of e-commerce, Internet security is particularly important. In recent decades, Internet crimes are indeed pervasive, which threatens Internet security. Phishing detection is one of the topics in Internet security for it torments Internet users continuously. Phishing is an identity theft by using a fake web page which is well-designed and spread by E-mail, Short Message Service (SMS) or social networks. Estimated by Cyren\cite{1}, phising web pages increased by a whopping 233\% in the year 2014. Clearly, there is a need for an efficient and automatic method to identify the phishing web pages.

Phishing web pages disguise as their targets – legitimate web pages by imitating their page content and layout, or Uniform Resource Locator (URL). High degree of visual similarity between phishing web pages and their targets confuses Internet users because of their carelessness or no knowledge about phishing. Internet users are lured to provide their private information such as user name, password, or credit card numbers and PIN codes. The leak of private information could lead to the loss of money or another identity theft.

As the proliferation of phishing leads to enormous security problems, a variety of approaches have been developed to resist phishing attacks. Blacklist\cite{2},\cite{3} is usually used in browsers. Blacklist maintains a list of phishing URLs and checks whether the URL of the current web page matches any record in the blacklist before Internet users open the web page. However, there is always a delay between phishing URLs reporting and the list updating\cite{4}, and the blacklist cannot identify the phishing URLs which are not in the blacklist. Therefore, blacklist approaches are considered vulnerable\cite{4},\cite{5} to detect new phishing attacks. Heuristic-based approaches are proposed to avoid the disadvantage of blacklist. Heuristic-based approaches employ features of web page, and score the web page\cite{6} or utilize machine learning methods\cite{7},\cite{8} to identify the phishing web pages. There are some other approaches such as User-education to protect human from phishing attacks. However, high similarity with legitimate ones makes phishing web pages difficult to be distinguished, and phishing continues to be a primal intimidation of Internet security.

1.1 Motivation

Phishing is a continuing problem in Internet security, which cannot be permanently solved. Heuristic-based approach is considered to be effective for detecting phishing web pages because of its ability to identify new ones. Currently, mainstream heuristic-based approaches usually utilize page content features, domain features or URL features combining with machine learning algorithm to detect phishing web pages. Feature is crucial in an anti-phishing system, which is an important element for detection results. However, only a few researches performed feature assessment and feature selection techniques. In previous literatures, most of the heuristic-based approaches chose the features by researchers’ careful consideration according to their experience. This motivates our new anti-phishing framework based on feature engineering for high detection accuracy and low false positive (FP) rate, which implements automat-
rical feature selection and feature extraction.

1.2 Contribution

This research employs genetic algorithm (GA) to perform feature selection, and proposes a two-stage projection pursuit (PP) algorithm to generate a new feature. We evaluate this feature engineering by different classifiers including Support Vector Machine (SVM), Logistic Regression (LR), k-Nearest Neighbor (kNN), Neural Networks (NN) and Naive Bayes (NB). In addition, we propose if-then rule to reduce the FP rate. Our main contributions in this research are summarized as follows.

- This research constructs an effective framework to detect phishing web pages based on feature engineering.
- This research proposes a two-stage PP algorithm to transform non-critical features to a new feature for phishing web pages detection.
- This research investigates effectiveness of feature engineering combining with different classifiers for detecting phishing web pages.
- This research generates if-then rule to reduce the FP rate in phishing detection.

The remainder of this paper is organized as follows. Section 2 gives the related work of phishing web pages detection. Section 3 describes features used in this paper, reviews theoretical background about GA for feature selection, and proposes a two-stage PP algorithm for feature extraction. Section 4 describes system architecture. Section 5 presents experimental results of the proposed framework for phishing web page detection. Finally, Sect. 6 concludes the paper and gives future work.

2. Related Work

2.1 Phishing Web Page Detection

The emerging of phishing attacks has produced negative consequences to Internet security, which is a high recognition of industry and academe. A lot of researches have been carried out in recent years for detecting phishing web pages. In the first subsection, we review some typical approaches to detect phishing web pages including blacklist, heuristic-based approaches and user-education. In the second subsection, we present feature evaluation and selection methods in phishing detection.

2.1.1 Blacklist

Blacklist maintains a list of phishing URLs, which needs to be updated frequently. Google Safe Browsing API [9] makes use of client applications to check whether a URL is a phishing one. If the given URL is in the blacklist, it is considered as a phishing one. Otherwise, it is a legitimate one. Google published a new version (V3) of the Safe Browsing API in 2014, which is based on a list of hashed URLs. This version can protect user privacy. The API users exchange data with the server using hashed URLs, so the server never knows the actual URLs queried by the clients [9]. Besides, API users maintain a local cache of the hashed URLs in lists, so they need not query the server every time when they want to check a URL. However, the API users need to update their local cache of the hashed URLs periodically and download the new lists of hashed URLs. Prakash et al. [10] proposed to discover new phishing URLs by enumerating simple combinations of known phishing URLs. Then, they tested whether the new URLs generated were indeed phishing by employing DNS queries and content matching techniques. Their approach generated low false negative (FN) rate (less than 3%), but suffered 5% FP rate.

2.1.2 Heuristic-Based Approaches

Heuristic-based approaches detect phishing web pages based on features such as URL features, page content features, domain features and web-based features. Zhang et al. [6] extracted the textual page content and utilized term frequency-inverse document frequency (TF-IDF) to generate a lexical signature. Then, they submitted this lexical signature to Google search engine, and checked if the searching results contained the domain name of the given web page. If the domain name of the given web page did not match any of the top \( N \) searching results, the web page was considered as a phishing one. Xiang et al. [8] developed Zhang’s approach, and took the page content as one of the features. They utilized HTML source code, search engines and third party services with machine learning techniques to detect phishing web pages. In order to reduce the FP rate and achieve runtime speedup, they designed two filters. One filter was a near-duplicate detector to catch highly similar phishing. The other was a login form filter, which directly defined web page as legitimate if the web page had no login form. Their approach achieved over 92% true positive (TP) rate on unique testing. Zhang et al. [11] extracted and employed five URL features, eight domain features and two page content features to detect phishing web pages. They compared sequential minimal optimization (SMO) algorithm, LR, NB and RF. Their experimental results suggested that SMO achieved the best performance with 94.27% TP rate for Chinese e-Business phishing detection.

Visual content is also proposed to be feature for phishing detection. Zhang et al. [12] proposed to employ textual and visual contents to detect phishing web pages. They constructed a text classifier based on SVM to classify the textual content, and an image classifier making use of the earth mover’s distance to classify the visual content. Finally, they utilized a Bayesian approach to combine the results. Zhou et al. [13] compared the snapshot image pair of the suspected web page and the protected web page to detect phishing. They first took a snapshot of the given web page and logo image of the protected web page. Then, they utilized Speeded Up Robust Features (SURF) detector to detect...
user-education approaches can train and enhance the ability of users to identify the phishing URLs. However, user-education should be carried on continuously [14], which is hard for the users.

The above mentioned researches have some disadvantages for phishing detection. Blacklist could not identify the phishing URL which is not in the blacklist. Therefore, blacklist is no closer to solving zero-day phishing problem. In user-education approach, it is impossible to be trained continuously for Internet users. Therefore, this approach is very limited for detecting phishing web pages. Heuristic-based approaches can deal with zero-day phishing problem, but most of the heuristic-based approaches encountered high FP rate, low TP rate or high time cost.

2.2 Feature Engineering for Phishing Web Pages Detection

Most of the existing researches seem to choose the features depending upon experience. Currently, a few researches evaluate features and focus on feature engineering. Statistical methods are commonly used for evaluating the importance of features. Whitaker et al. [19] calculated the percentage of feature attended in phishing web pages and feature attended in all the web pages. They used all the listed features to construct detection model, but did not study how to select features by the percentage value. Mohammam et al. [20] calculated each feature frequency in the phishing data set to reflect the feature importance, and all the features were associated with a weight corresponding to the percentage of features attended in the data set. They suggested that ‘request URL’, ‘age of domain’ and ‘HTTPS and SSL’ are the most significant features, while ‘Disabling Right Click’ and ‘@ in the URL’ are the lowest significant features. Actually, ‘@ in the URL’ never attends in legitimate web pages, and some literatures took it as significant feature for phishing detection [8], [11], [21].

Basnet et al. [22] investigated correlation-based feature selection (CFS) and wrapper feature selection (WFS) techniques for phishing web pages detection. CFS evaluates feature subsets by searching space of all possible feature subsets according to a given correlation-based heuristic function. WFS uses machine learning to evaluate feature subsets and selects the feature subsets providing the better estimate accuracy. They concluded that feature selection can improve classification results, and wrapper-based technique performed better than correlation-based technique by combining with different classifiers. Our research is also motivated by their research and adopts GA from their research to perform feature selection.

3. Theoretical Backgrounds

In this section, we first describe features commonly used in

2.2 Feature Engineering for Phishing Web Pages Detection

Most of the existing researches seem to choose the features depending upon experience. Currently, a few researches evaluate features and focus on feature engineering. Statistical methods are commonly used for evaluating the importance of features. Whitaker et al. [19] calculated the percentage of feature attended in phishing web pages and feature attended in all the web pages. They used all the listed features to construct detection model, but did not study how to select features by the percentage value. Mohammam et al. [20] calculated each feature frequency in the phishing data set to reflect the feature importance, and all the features were associated with a weight corresponding to the percentage of features attended in the data set. They suggested that ‘request URL’, ‘age of domain’ and ‘HTTPS and SSL’ are the most significant features, while ‘Disabling Right Click’ and ‘@ in the URL’ are the lowest significant features. Actually, ‘@ in the URL’ never attends in legitimate web pages, and some literatures took it as significant feature for phishing detection [8], [11], [21].

Basnet et al. [22] investigated correlation-based feature selection (CFS) and wrapper feature selection (WFS) techniques for phishing web pages detection. CFS evaluates feature subsets by searching space of all possible feature subsets according to a given correlation-based heuristic function. WFS uses machine learning to evaluate feature subsets and selects the feature subsets providing the better estimate accuracy. They concluded that feature selection can improve classification results, and wrapper-based technique performed better than correlation-based technique by combining with different classifiers. Our research is also motivated by their research and adopts GA from their research to perform feature selection.

3. Theoretical Backgrounds

In this section, we first describe features commonly used in

2.2 Feature Engineering for Phishing Web Pages Detection

Most of the existing researches seem to choose the features depending upon experience. Currently, a few researches evaluate features and focus on feature engineering. Statistical methods are commonly used for evaluating the importance of features. Whitaker et al. [19] calculated the percentage of feature attended in phishing web pages and feature attended in all the web pages. They used all the listed features to construct detection model, but did not study how to select features by the percentage value. Mohammam et al. [20] calculated each feature frequency in the phishing data set to reflect the feature importance, and all the features were associated with a weight corresponding to the percentage of features attended in the data set. They suggested that ‘request URL’, ‘age of domain’ and ‘HTTPS and SSL’ are the most significant features, while ‘Disabling Right Click’ and ‘@ in the URL’ are the lowest significant features. Actually, ‘@ in the URL’ never attends in legitimate web pages, and some literatures took it as significant feature for phishing detection [8], [11], [21].

Basnet et al. [22] investigated correlation-based feature selection (CFS) and wrapper feature selection (WFS) techniques for phishing web pages detection. CFS evaluates feature subsets by searching space of all possible feature subsets according to a given correlation-based heuristic function. WFS uses machine learning to evaluate feature subsets and selects the feature subsets providing the better estimate accuracy. They concluded that feature selection can improve classification results, and wrapper-based technique performed better than correlation-based technique by combining with different classifiers. Our research is also motivated by their research and adopts GA from their research to perform feature selection.

3. Theoretical Backgrounds

In this section, we first describe features commonly used in

2.2 Feature Engineering for Phishing Web Pages Detection

Most of the existing researches seem to choose the features depending upon experience. Currently, a few researches evaluate features and focus on feature engineering. Statistical methods are commonly used for evaluating the importance of features. Whitaker et al. [19] calculated the percentage of feature attended in phishing web pages and feature attended in all the web pages. They used all the listed features to construct detection model, but did not study how to select features by the percentage value. Mohammam et al. [20] calculated each feature frequency in the phishing data set to reflect the feature importance, and all the features were associated with a weight corresponding to the percentage of features attended in the data set. They suggested that ‘request URL’, ‘age of domain’ and ‘HTTPS and SSL’ are the most significant features, while ‘Disabling Right Click’ and ‘@ in the URL’ are the lowest significant features. Actually, ‘@ in the URL’ never attends in legitimate web pages, and some literatures took it as significant feature for phishing detection [8], [11], [21].

Basnet et al. [22] investigated correlation-based feature selection (CFS) and wrapper feature selection (WFS) techniques for phishing web pages detection. CFS evaluates feature subsets by searching space of all possible feature subsets according to a given correlation-based heuristic function. WFS uses machine learning to evaluate feature subsets and selects the feature subsets providing the better estimate accuracy. They concluded that feature selection can improve classification results, and wrapper-based technique performed better than correlation-based technique by combining with different classifiers. Our research is also motivated by their research and adopts GA from their research to perform feature selection.

3. Theoretical Backgrounds

In this section, we first describe features commonly used in
literatures. Then, we present theories about feature engineering and propose a two-stage PP for feature extraction.

3.1 Feature Vector

Several features have been proposed combining with machine learning for phishing web page detection. In this research, we list 16 frequently used features in literatures [8], [21], [23]–[25] and one new feature, and make use of them to perform feature engineering.

- IP address ($f_1$): almost all the literatures [8], [21], [23]–[25] using URL features to detect phishing have mentioned this feature. If the domain name of the URL is an IP address, such as http://66.199.185.139/805d6d477117c59fad677e76eee2e20b/, it is suspicious.

- Length of URL ($f_2$): phishers use long URL to hide suspicious part of the URL. Mohammad et al. [20] obtained the following rule by statistical method on 2500 phishing samples. If the length of URL is less than 54, they consider it as legitimate. If the length of URL is greater than 74, they take it as phisy. Otherwise, they consider it as suspicious. Other literatures [21], [24], [25] followed and used this rule to define the value of this feature. However, this rule was highly affected by the samples. In this paper, we directly employ the length of URL to be the feature.

- @ in URL ($f_3$): symbol @ redirects the user to the link after it, which is usually a phishing one. Once there is a symbol @ in the given URL, it is definitely a phishing web page.

- Prefix or suffix separated by ‘-’ ($f_4$): prefix or suffix is usually added to famous domain name by ‘-‘ to confuse Internet users.

- Number of sub domains/dots in domain name ($f_5$): phishing URLs often insert famous domain name as sub domain. Therefore, if the number of sub domain exceeds a certain value, the given web page is suspicious.

- Abnormal URL ($f_6$): this feature supposes the website identity is a part of URL, and checks the identity in the WHOIS database [26]. If the website identity dose not match its record shown in the WHOIS database, the website is considered as suspicious.

- Misuse of HTTPs ($f_7$): HTTPs protocol is used for communication security of sensitive information, which reflects it is certainly a legitimate one [21]. However, phishers employ a fake HTTPs to confuse the Internet users. If the HTTPs exists but certificate issuer is not within the trusted issuer list, this feature is considered as suspicious. If the HTTPs does not exist, it is taken as phisy. Otherwise, it is legitimate [21], [25].

- Request URL ($f_8$): phishing web pages are often duplicates of legitimate ones. Their text, images or videos are often downloaded from other server. Therefore, we take the ratio of internal links (linking with the sever of current domain) and external links (linking to other server with different domain) as the value of this feature.

- URL of Anchor ($f_9$): an Anchor is an element defined by tag ‘a’ [24], and we also take the radio of internal links and external links of Anchor as the value of this feature.

- Redirect page ($f_{10}$): phishers utilize this technique to redirect Internet users to the phishing pages. We count the number of redirected times to be the value of this feature.

- Domain history ($f_{11}$): phishing aims at obtaining sensitive information quickly. Therefore, Phishing web pages usually have relatively short history. We submit domain name to the WHOIS database and search the history of the domain name. We consider the URL as a suspicious one, if its domain history is less than six months. Otherwise, it is considered as a legitimate one, which refers to literature [24].

- DNS record ($f_{12}$): DNS record records the information about website that is still alive at the moment. If the DNS record is empty or not found in the WHOIS database [26], this feature is taken as suspicious.

- Website traffic ($f_{13}$): this feature measures the importance of website. If the website is visited frequently, it has relatively high website traffic. For being visited regularly, legitimate websites usually have high website traffic. As phishing websites have a short history, they usually have low ranking or no website traffic. This feature is also obtained from WHOIS database [26]. The URL is considered as legitimate if it is ranked in top 1000000. If the URL is ranked more than 1000000 or has no traffic rank, it is taken as phishing.

- Hiding suspicious link ($f_{14}$): a fake URL is displayed in the status bar of browser to the Internet users by using JavaScript [24]. This feature is obtained by checking HTML source code. If the ‘onmouseover’ makes changes to the status bar, it is suspicious.

- Using popup window ($f_{15}$): legitimate websites hardly ask users to submit information through a popup window, while a part of phishing websites steal sensitive information of Internet users by means of using popup window.

- Login form ($f_{16}$): almost all the phishing web pages acquire sensitive information by using login forms, which are generally displayed in the same way as that used on the their phishing targets [23].

- Number of URLs linking to the current website ($f_{17}$): this feature is a direct measurement to evaluate the importance of a website. The legitimate websites usually have more URLs linking to them, while phishing websites have fewer. This feature is also obtained from WHOIS database [26]. We utilize the original numerical value to be the feature value.
3.2 Feature Engineering

Feature engineering is performed before proceeding classification step. In this research, features are automatically classified to critical features and non-critical features by WFS method, and then non-critical features are transformed to a new feature by PP. WFS is a method which evaluates features by using a machine learning algorithm [22]. We take GA as the optimization algorithm integrating with different classifiers to perform WFS for distinguishing critical features and non-critical features.

For a given feature set \( F = \{f_1, f_2, \ldots, f_N\} \), we intend to divide the features into two subsets \( F_1 \) and \( F_2 \), where \( F_1 \) is the critical feature set, \( F_2 \) is non-critical feature set, \( F_1 \cup F_2 = F \) and \( F_1 \cap F_2 = \emptyset \). GA utilizes binary vector to define the subset feature.

\[
S = [s_1, s_2, \ldots, s_N]
\]

where \( s_i = 1 \) represents the feature is critical and \( s_i = 0 \) denotes it is a non-critical feature. In this research, we take classification accuracy as the fitness. The steps for performing feature selection are simply summarized as follows [27], [28].

1. Create initial population to generate \( N_{ch} \) chromosomes \( \{S_1, S_2, \ldots, S_{N_{ch}}\} \).
2. Construct classification model, and calculate fitness of each chromosome \( S_i \).
3. If satisfy the stopping criterion, designate solution, else goto step 4.
4. Select chromosomes with the 30% highest fitness to create new generations with mutation, goto step 2.

3.3 Two-Stage PP for Non-Critical Feature Extraction

PP maps the features into a lower dimension, which involves directions in the feature space, a criterion to measure the ‘usefulness’ of each direction as a projection axis, and an optimization technique which is used to vary the direction to maximize the criterion [29]. In this research, we do not drop the non-critical features, but utilize a two-stage PP to transform them into a new feature space.

In the first stage, we randomly select a set of initial projection directions, and take the following radio as the criterion of PP [30] to select the projection direction with the highest criterion value.

\[
W_{opt} = \arg \max_W \frac{|W^T S_B W|}{|W^T S_W W|}
\]

where \( S_W \) is within-class scatter matrix and \( S_B \) is between-class scatter matrix. \( W \) is projection matrix, and the column vectors of \( W \) are the eigenvectors of \( S_W^{-1} S_B \).

\[
S_W = \sum_{i=1}^{C} \sum_{x \in X_i} (x - m_i)(x - m_i)^T
\]

\[
S_B = \sum_{i=1}^{C} N_i (m_i - m)(m_i - m)^T
\]

where \( C \) is the number of classes, \( N_i \) is the number of samples of class \( X_i \), \( m_i \) is the mean of class \( X_i \), and \( m \) is the mean of all the classes.

We repeat the first stage \( Ea \) times, and get a set of projection directions, which are taken as the initial values of the second stage.

In the second stage, we take the projection directions from the first stage to calculate the criterions. We utilize the projection feature combining with critical features to construct the classification model, and calculate the accuracy of classification as the criterion to determine the projection direction. We utilize GA as the optimization technique to seek the projection direction according to the maximizing criterion.

4. System Architecture

Our anti-phishing system consists of five major components: a feature generation module, a feature engineering module, a training module, a detection module and a FP filter module, as illustrated in Fig. 1. All of the components run in Intel Core2 CPU clocked at 2.89 GHz with 1.8GB RAM.

The functionality of the feature generation module is to generate the features from the given web pages, which is implemented in JAVA. Feature engineering module first divides the features into critical features and non-critical features, and then projects the non-critical features to a new feature. Training module utilizes machine learning method to train a classification model. Detection module is corresponding to training module, which is used to identify if the given web page is phishing or not. The feature engineering module, training model and detection module are implemented in MATLAB 2012B.

FP filter module is proposed to reduce FP rate. High FP rate is a barrier for applying machine learning in phishing detection. According to the literature [14], 3% FP rate is considered some high for phishing detection. In order to reduce FP rate, we seek to find rules to filter the legitimate ones that are mislabeled as phishing based on calculating confusion matrix of mislabeled legitimate samples and original phishing samples. The rules satisfy the criterion: none
From the comparisons shown in Figs. 2–7, we observe LR, kNN and NB combining with feature engineering perform better than without feature engineering in all the experiments in terms of accuracy. LR combining with feature engineering generates above 99% accuracy in all the experiments, which outperforms other combinations. NB combining with feature engineering has a notable improvement in all the experiments. However, SVM and NN with fea-

| Table 2 | Projection directions of two-stage PP. |
|---------|--------------------------------------|
| Classifier No. | Projection directions |
| SVM | E1 0.72, 0.44, 0.35, 0.20, 0.27, 0.07, 0.19, 0.06 |
| | E2 0.72, 0.34, 0.30, 0.14, 0.21, 0.44, 0.09, 0.12 |
| | E3 0.71, 0.31, 0.27, 0.15, 0.18, 0.38, 0.30, 0.09 |
| | E4 0.72, 0.14, 0.32, 0.14, 0.37, 0.24, 0.37, 0.04 |
| | E5 0.71, 0.45, 0.26, 0.25, 0.10, 0.32, 0.16, 0.09 |
| | E6 0.74, 0.35, 0.28, 0.13, 0.19, 0.40, 0.08, 0.15 |
| LR | E1 0.71, 0.20, 0.07, 0.31, 0.11, 0.33, 0.42, 0.21 |
| | E2 0.71, 0.31, 0.11, 0.08, 0.18, 0.48, 0.18, 0.26 |
| | E3 0.71, 0.27, 0.31, 0.20, 0.05, 0.22, 0.45, 0.11 |
| | E4 0.72, 0.51, 0.10, 0.20, 0.14, 0.31, 0.21, 0.10 |
| | E5 0.71, 0.11, 0.44, 0.26, 0.19, 0.38, 0.06, 0.17 |
| | E6 0.74, 0.55, 0.09, 0.21, 0.13, 0.49, 0.10, 0.13 |
| NN | E1 0.72, 0.35, 0.16, 0.28, 0.31, 0.38, 0.08, 0.07 |
| | E2 0.72, 0.31, 0.23, 0.40, 0.13, 0.34, 0.16, 0.12 |
| | E3 0.71, 0.42, 0.29, 0.25, 0.11, 0.09, 0.34, 0.14 |
| | E4 0.72, 0.30, 0.27, 0.31, 0.16, 0.31, 0.29, 0.11 |
| | E5 0.72, 0.48, 0.31, 0.11, 0.13, 0.33, 0.06, 0.12 |
| | E6 0.74, 0.49, 0.22, 0.12, 0.07, 0.26, 0.16, 0.19 |
| NB | E1 0.72, 0.22, 0.23, 0.48, 0.26, 0.11, 0.26, 0.08 |
| | E2 0.72, 0.19, 0.32, 0.37, 0.24, 0.34, 0.15, 0.11 |
| | E3 0.71, 0.29, 0.37, 0.03, 0.10, 0.27, 0.15, 0.20 |
| | E4 0.72, 0.19, 0.08, 0.16, 0.28, 0.55, 0.07, 0.14 |
| | E5 0.72, 0.40, 0.33, 0.26, 0.07, 0.18, 0.32, 0.03 |
| | E6 0.74, 0.44, 0.20, 0.21, 0.11, 0.26, 0.16, 0.13 |
| | E7 0.74, 0.20, 0.30, 0.16, 0.40, 0.16, 0.30, 0.11 |

Fig. 2 Accuracy result of experiment E1.

Fig. 3 Accuracy result of experiment E2.
ture engineering combinations perform unstably. In some experiments, these combinations generate lower accuracy than without feature engineering. This may be related to the characteristics of the SVM and NN. SVM and NN are both based on feature projection. They map input data to high-dimensional feature space and implement classification in the new space, which might explain why their combinations have less effect in this research.

In the experiments E1–E5, the training data sizes are the same. We give the average training time and detecting time of them. The training time of feature engineering combining with SVM, LR, kNN, NN and NB is 10388.93s, 11571.67s, 11279.42s, 11945.95s and 10614.62s, respectively, and the corresponding detecting time is 0.34s, 0.29s, 3.14s, 0.26s and 4.18s. In the last experiment E6, the training time of feature engineering combining with SVM, LR, kNN, NN and NB is 5892.26s, 6301.11s, 5975.40s, 6464.32s and 5910.91s, respectively, and the corresponding detecting time is 1.98s, 0.32s, 5.07s, 0.29s and 10.34s. The feature selection by utilizing GA and searching for the projection directions of non-critical features using GA cost most of the training time, which needs to be improved. The average detecting time of each sample is $6.76 \times 10^{-5}$s by using LR classifier, so it is suitable for actual application.

5.3 Performance of FP Filter

We utilize the critical features to construct FP filter. Then, we obtain one rule for generating FP filter: if $f_8 > 1$ and $f_{17} >= 10$, then the given web page is legitimate. This rule is employed to filter the web pages that are classified as phishing. If features of web pages satisfy the rule, it is reclassified as legitimate. Table 3 illuminates FP rate of each combination.

The same as accuracy, LR, kNN and NB combining with feature engineering respectively have lower FP rate than without feature engineering. The FP filter in all the combinations has effectiveness to reduce the FP rate.

6. Conclusion and Future Work

This paper describes our research effort on phishing web pages detection by proposing a feature engineering method. We investigate the features that have been used in some literatures, and perform feature selection by using GA. The features are classified into two sets, critical features and non-critical features. A two-stage PP algorithm is proposed to perform the feature extraction on non-critical features instead of eliminating them. We construct five different traditional machine learning algorithms to evaluate the performance of the proposed feature engineering, and obtain the result that feature engineering can improve the accuracy of phishing web page detection and feature engineering combining with LR performs best. The FP filter is also proved effective to reduce the FP rate from experiments.

We plan to extend our anti-phishing framework from the following aspects as the future work: (1) Extract new

| Classifiers | Features | E1 | E2 | E3 | E4 | E5 | E6 |
|-------------|----------|----|----|----|----|----|----|
| SVM         | original | 2.19 | 1.57 | 2.05 | 1.64 | 1.77 | 1.47 |
|             | FE       | 1.77 | 1.95 | 1.45 | 1.32 | 1.59 | 1.56 |
|             | FP filter| 1.32↓ | 1.95↓ | 1.33↓ | 1.31↓ | 1.43↓ | 1.41↓ |
| LR          | original | 3.09 | 3.27 | 3.18 | 4.14 | 2.43 | 2.56 |
|             | FE       | 1.05 | 1.05 | 1.14 | 1.91 | 1.29 | 1.38 |
|             | FP filter| 1.05↓ | 1.05↓ | 1.14↓ | 1.71↓ | 1.29↓ | 1.30↓ |
| kNN         | original | 3.82 | 3.27 | 2.65 | 3.41 | 3.50 | 3.24 |
|             | FE       | 1.41 | 1.23 | 2.09 | 1.68 | 1.91 | 1.44 |
|             | FP filter| 1.35↓ | 1.23↓ | 1.78↓ | 1.57↓ | 1.91↓ | 1.44↓ |
| NN          | original | 1.23 | 1.14 | 1.24 | 1.36 | 1.23 | 1.80 |
|             | FE       | 1.14 | 3.09 | 3.41 | 1.24 | 1.68 | 1.42 |
|             | FP filter| 1.14↓ | 2.10↓ | 2.47↓ | 1.24↓ | 1.68↓ | 1.38↓ |
| NB          | original | 5.53 | 6.08 | 5.76 | 5.12 | 6.31 | 5.51 |
|             | FE       | 3.77 | 3.50 | 3.55 | 3.86 | 3.54 | 4.27 |
|             | FP filter| 1.91↓ | 2.37↓ | 2.89↓ | 2.56↓ | 2.99↓ | 2.21↓ |
features from new types of phishing web pages, and implement incremental learning for detection; (2) Improve the effectiveness of our FP filter to reduce the FP rate.

Acknowledgements

This research work is supported by Special Funds on Guangdong Province Chinese Academy of Sciences Comprehensive Strategic Cooperation, Key Technologies and Industrialization NO. 2013B091300019. We would like to thank phishtank for supplying us with the phishing samples.

References

[1] Cyren, “2015 Cyber Threats Yearbook,” http://pages.cyren.com/SecurityYearbook_2015.html, accessed July 1, 2015.
[2] Mozilla, https://support.mozilla.org/en-US/kb/how-does-phishing-and-malware-protection-work, accessed July 1, 2015.
[3] Neteraft, http://toolbar.neteraft.com/, accessed July 1, 2015.
[4] W.W. Zhuang, and Q.S. Jiang, “Intelligent anti-phishing framework using multiple classifiers combination,” Journal of Computational Information Systems, vol.8, no.17, pp.7267–7281, March 2012.
[5] J. Ma, L.K. Saul, S. Savage, and G.M. Voelker, “Beyond blacklists: Learning to detect malicious web sites from suspicious URLs,” Proc. 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.1245–1254, Paris, France, June 2009.
[6] Y. Zhang, J.J. Hong, and L.F. Cranor, “CANTINA: A content-based approach to detecting phishing web sites,” Proc. 16th International Conference on World Wide Web, pp.639–648, Banff, Canada, May 2007.
[7] H. Huang, L. Qian, and Y. Wang, “A SVM-based technique to detect phishing URLs,” Information Technology Journal, vol.11, no.7, pp.921–925, May 2012.
[8] G. Xiang, J. Hong, C.P. Rose, and L. Cranor, “CANTINA+: A feature-rich machine learning framework for detecting phishing websites,” ACM Trans. Information and System Security, vol.14, no.2, pp.1–28, Sept. 2011.
[9] Google Safe Browsing API, https://developers.google.com/safe-browsing?csw=1, accessed June 1, 2015.
[10] P. Prakash, M. Kumar, R.R. Kompella, and M. Gupta, “Phishnet: Predictive blacklisting to detect phishing attacks,” Proc. IEEE INFOCOM, pp.1–5, San Diego, CA, March 2010.
[11] D. Zhang, Z. Yan, H. Jiang, and T. Kim, “A domain-feature enhanced classification model for the detection of Chinese phishing e-business websites,” Information & Management, vol.51, no.7, pp.845–853, Aug. 2014.
[12] H. Zhang, G. Liu, T.W.S. Chow, and W. Liu, “Textual and visual content-based anti-phishing: A Bayesian approach,” IEEE Trans. Neural Netw., vol.22, no.10, pp.1532–1546, Oct. 2011.
[13] Y. Zhou, Y. Zhang, J. Xiao, Y. Wang, and W. Lin, “Visual similarity based anti-phishing with the combination of local and global features,” Proc. IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp.189–196, Beijing, China, Sept. 2014.
[14] M. Khonji, Y. Iraji, and A. Jones, “Phishing detection: A literature survey,” Communications Surveys & Tutorials, vol.15, no.4, pp.2091–2121, Dec. 2013.
[15] L. Wenyin G. Liu, B. Qiu, and X. Quan, “Antiphishing through phishing target discovery,” IEEE Internet Computing, vol.16, no.2, pp.52–61, March 2012.
[16] G. Ramesh, I. Krishnamurthi, and K.S.S. Kumar, “An efficacious method for detecting phishing webpages through target domain identification,” Decision Support Systems, vol.61, no.5, pp.12–22, May 2014.
[17] S. Sheng, B. Magnien, P. Kumaraguru, A. Acquisti, L.F. Cranor, J. Hong, and E. Nunge, “Anti-phishing phil: The design and evaluation of a game that teaches people not to fall for phishing,” Proc. 3rd Symposium on Usable Privacy and Security, pp.88–99, Pittsburgh Pennsylvania, USA, July 2007.
[18] N.A.G. Arachchilage, and S. Love, “A game design framework for avoiding phishing attacks,” Computers in Human Behavior, vol.29, no.3, pp.706–714, May 2013.
[19] C. Whittaker, B. Ryner, and M. Nazif, “Large-scale automatic classification of phishing pages,” Proc. Network and Distributed System Security Symposium (NDSS), San Diego, USA, Feb. 2010.
[20] R.M. Mohammad, F. Thabtah, and L. McCluskey, “An assessment of features related to phishing websites using an automated technique,” Proc. International Conference for Internet Technology and Secured Transactions, pp.492–497, London, UK, Dec. 2012.
[21] N. Abdelhamid, A. Ayesh, and F. Thabtah, “Phishing detection based associative classification data mining,” Expert Systems with Applications, vol.41, no.13, pp.5948–5959, Oct. 2014.
[22] R.B. Basnet, A.H. Sung, and Q. Liu, “Feature selection for improved phishing detection,” Advanced Research in Applied Artificial Intelligence, Lecture Notes in Computer Science, vol.7345, pp.252–261, Springer, Berlin, Heidelberg, June 2012.
[23] M. He, S.-J. Hong, P. Fan, M.K. Khan, R.-S. Run, J.-L. Lai, R.-J. Chen, and A. Sutanto, “An efficient phishing webpage detector,” Expert Systems with Applications, vol.38, no.10, pp.12018–12027, Sept. 2011.
[24] R.M. Mohammad, L. McCluskey, and F. Thabtah, “Intelligent rule-based phishing websites classification,” IET Information Security, vol.8, no.3, pp.153–160, May 2014.
[25] R.M. Mohammad, F. Thabtah, and L. McCluskey, “Predicting phishing websites based on self-structuring neural network,” Neural Computing and Applications, vol.25, no.2, pp.443–458, Aug. 2014.
[26] WHOIS, www.who.is
[27] J. Yang, and V. Honavar, “Feature subset selection using a genetic algorithm,” in Handbook of Feature Extraction, Construction and Selection, ed. H. Liu, and H. Motoda, pp.117–136, Springer USA, March 1998.
[28] H. Frohlich, O. Chapelle, and B. Scholkopf, “Feature selection for support vector machines by means of genetic algorithm,” Proc. 15th IEEE International Conference on Tools with Artificial Intelligence, pp.142–148, Sacramento, USA, Nov. 2003.
[29] K. Fukunaga, ed., Handbook of Introduction to Statistical Pattern Recognition, Academic Press, Boston, 1990.
[30] Y. Su, S. Shan, X. Chen, and W. Gao, “Classifiability-based discriminatory projection pursuit,” IEEE Trans. Neural Netw., vol.22, no.12, pp.2050–2061, Dec. 2011.
[31] Phishtank, http://www.phishtank.com
[32] Statscrop, http://www.statscrop.com/websites/top-sites
[33] DMOZ, http://www.dmoz.org

Wei Zhang received the M.Sc. degree in mechanics in 2009 from Peking University, China. She is currently a Ph.D. student in Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences. Her research interests include data mining, phishing detection and malware detection.
Huan Ren is currently a master student of University of Science and Technology of China. His research interests include data mining and malware detection on Android platform.

Qingshan Jiang received the Ph.D. degree in mathematics from Chiba University, Japan, in 1996 and the Ph.D. degree in computer science from the University of Sherbrooke, Canada, in 2002. He is a Full Professor at Shenzhen Key Laboratory for High Performance Data Mining, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, China. His current research interests include data mining, network security, pattern recognition, statistical analysis, and fuzzy logic.