Journal of Theoretical and Applied Information Technology
15th September 2016. Vol.91. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

FEATURE-BASED PHISHING DETECTION TECHNIQUE

[1]XIN MEI CHOO, [2]KANG LENG CHIEW, [3]DAYANG HANANI ABANG IBRAHIM, [4]NADIANATRA MUSA, [5]SAN NAH SZE, [6]WEI KING TIONG

Faculty of Computer Science and Information Technology,

University Malaysia Sarawak, 94300 Kota Samarahan, MALAYSIA

E-mail: [1]xmchoo@gmail.com, [2]klchiew@unimas.my, [3]hananii@unimas.my, [4]nadia@unimas.my, [5]snsze@unimas.my, [6]wktiong@unimas.my

ABSTRACT

Phishing is an Internet fraud to entice unsuspecting victims. The tactic of phishing is to impersonate the trusted entities by employing both social engineering and technical subterfuge. Moreover, phishing is a form of online identity theft that creates a fake copy of popular site. There are many types of anti-phishing techniques available. However, they are mostly still in the infancy stage which may give false alarm to the user. Therefore, this research aims to develop a feature-based phishing detection technique to overcome the limitation. The proposed method involves aggregating new features with several existing features to form a sensitive features set. Based on the features set, the proposed method will utilise support vector machine to perform the classification. The experimental results show convincing performance with 95.33 percent of accuracy.

Keywords: *URL Features, Website Features, Phishing Detection, Anti-Phishing, Feature Extraction, Classification*

1. INTRODUCTION

Phishing attacks usually target user confidential information such as username, password and financial ID. Phishers would use their sophisticated attack vector such as emailing, or pop up window notification to lure the victim to visit the phishing website which has legitimate-looking layout. This will allow the phishers to harvest the victim credentials and sell them in the black market (as depicted in Figure 1).

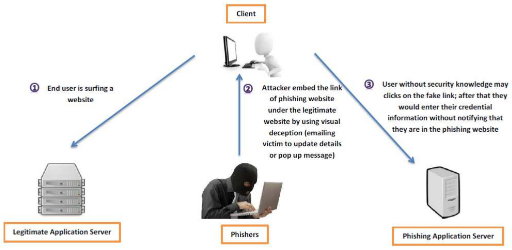

Client
① End user is surfing a website
② Attacker embed the link of phishing website under the legitimate website by using visual deception (emailing victim to update details or pop up message)
③ User without security knowledge may clicks on the fake link; after that they would enter their credential information without notifying that they are in the phishing website
Legitimate Application Server
Phishers
Phishing Application Server

Figure 1: Example of Phishing Attack Scenario

Anti-phishing refers to the method that is employed to prevent and defend phishing attacks. There are many techniques that offer protection at different domain. Some techniques work on emails, while others work on website attributes [2]. The proposed method works on the later. The contribution of this paper is twofold: first, it identifies and analyses attributes exhibited in phishing websites. Second, it proposes several new features and integrates to an existing method to enhance the overall detection performance. The works start by analysing and looking for abnormal attributes of a phishing website. The abnormal attributes that usually appear on phishing website include some uncommon symbols in the URL and some irregular HTML form and title elements. Therefore, extracting features from these attributes will enhance the phishing detection ability.

2. LITERATURE REVIEW

Many anti-phishing techniques focus on enabling the client to recognize and filter various types of phishing attacks [1, 4, 9]. In general, anti-phishing techniques can be classified into four categories, which are content filtering, blacklisting, symptom-based prevention and domain binding. Content filtering uses machine learning techniques, such as Bayesian Additive Regression Trees (BART) or Support Vector Machines (SVM) to filter phishing email that matches phishing attributes. Blacklist is a collection of known phishing websites that are published by a trust entity like Google and Microsoft. However, blacklist requires constant update in order to keep effective. Symptom-based prevention approach analyses the content of each webpage, then generate phishing alerts according to the type and number of symptoms detected. Domain binding is a browser-based technique where sensitive information such as username and password are combined with the domain. If this information is not bind, the technique will warn the users [2].

2.1 Blacklist and Whitelist

Blacklist and whitelist techniques are the most common and straightforward solutions. However, their effectiveness is determined by the completeness of the list. As a result, these techniques are not effective against new phishing websites [4]. Furthermore, majority of phishing websites are short-lived and the updated list is of less functional.

2.2 Page Analysis

Page analysis inspects the properties of a webpage based on the features, which are extracted from the HTML source code or derived from a URL. For page source, the number of HTML form tags might provide an indicator to detect phishing website. In addition, the number of input fields such as user ID and password are also crucial and suitable to be used as an indicator [3]. Phishers may trick users to provide their credentials through these input elements. Unsuspecting users are also susceptible to visually deceptive text, images mask underlying text, images mimicking windows and windows mask underlying windows [2]. Hence, these elements are important for the analysis.

2.3 URL Structure Analysis

The URL string can be broken down into multiple tokens that constitutes of binary features. Examples of features include length of the URL, number of dots, existence of IP address in the URL and URL with HTTPS and SSL [1]. In order to get a more comprehensive analysis, Alexa database and WHOIS database are usually used to check the URL domain name, domain registrar, name server and age of domain.

3. METHODOLOGY

Motivated by Rami et al. work [1], this paper proposes a feature-based technique to detect the phishing webpage. The paper will propose additional features and combine with some of the selected features from [1]. The features are shown in Table 1 where the proposed additional features are indicated with asterisk *.

Table 1: Proposed Features Set

URL-Based Feature	**Webpage Source Feature**	**Third Party Feature**
1. IP Address 2. Length of URL 3. Sum of '.' Symbol 4. 'HTTP' and 'SSL' 5. Claim Identity (Double URL) 6. '-' Symbol 7. '@' Symbol 8. '~' Symbol * 9. '#' Symbol * 10. Sum of '%' Symbol * 11. Sum of '=' Symbol * 12. Sum of '&' Symbol * 13. '?cmd=' Symbol * 14. 'paypal' Key Word * 15. Numeric and Alphabet *	16. Request URL 17. URL of Anchor 18. Server Form Handler 19. Redirect Page 20. *onMouseOver* to Hide the Link 21. Disable Right Click 22. Popup Window 23. Irregularity Form * 24. Absent of Title *	25. Traffic Rank 26. DNS Record 27. Domain Age

The syntax and structure of URL is a very crucial feature to detect the phishing website [7]. The adversary usually obfuscates the URL with @ symbol, request URL, anchor of URL, server from handler, uncertain keywords in URL, adding prefix and suffix in domain. Phishing website URLs also have characteristic like long URL, absent of HTTPS and SSL [1] and [6]. Therefore, using URL-based features to detect phishing website are useful and beneficial. Instead of waiting for victim to submit the form then harvest the credentials, phishers can get more active by using JavaScript to capture key-press events. Phishers could intercept each key that is pressed and send the information back to the server. Thus, all data will be transmitted to the phishing server by using JavaScript. Although one may claim that JavaScript can be disabled to protect the user. However, disabling JavaScript is not feasible because there are many websites that are heavily depended on JavaScript to fully functional. The second class of feature is webpage source feature. Some examples of these features include webpage redirection, disable the hyperlink with onMouseOver, disable right click function and use popup window [1]. The third class of feature involves utilising third party information. In this paper, WHOIS and Alexa lookup are used to check whether the claimed identity of a website matches its real identity in the database. Phishing websites usually are short lived and ranked low in search engine ranking. Thus, website traffic and age of domain information provided by the third party can help to determine the legitimacy of a website [5].

Following the notation used in [1], the listing below shows the rules for each feature:

3.1 Class 1: URL-Based Feature

1 If {IP address exists in URL → feature = True}
Else {feature = False}

2 If {URL length < τ_{1a} → feature = False}
Else If {τ_{1a} ≤ URL length ≤ τ_{1b} → feature = Suspicious}
Else {feature = True}

3 If {Dots in domain part < τ_2 → feature = False}
Else {feature = True}

4 If {HTTPS exist in URL → feature = False}
Else {feature = True}

5 If {Claim Identity or Double URL → feature = True}
Else {feature = False}

6 If {Domain part includes '-' symbol → feature = True}
Else {feature = False}

7 If {URL contains '@' symbol → feature = True}
Else {feature = False}

8 If {URL contains '~' symbol → feature = True}
Else {feature = False}

9 If {URL contains '#' symbol → feature = True}
Else {feature = False}

10 If {URL contains '%' symbol → feature = True}
Else {feature = False}

11 If {URL contains '=' symbol → feature = True}
Else {feature = False}

12 If {URL contains '&' symbol → feature = True}
Else {feature = False}

13 If {URL contains '?' symbol → feature = True}
Else {feature = False}

14 If {URL contains 'paypal' keyword → feature = True}
Else {feature = False}

15 If {URL contains alphanumeric → feature = True}
Else {feature = False}

3.2 Class 2: Webpage Source Feature

1 If {Request URL '<img>' < τ_{6a} → feature = False}
Else If {τ_{6a} ≤ Request URL '<img> ≤ τ_{6b} → feature = Suspicious}
Else {feature = True}

2 If {URL of Anchor '<a href>' < τ_{7a} → feature = False}
Else If {τ_{7a} ≤ URL of Anchor '<a href> ≤ τ_{7b} → feature = Suspicious}
Else {feature = True}

3 If {Server Form Handler '<form>' < τ_{8a} → feature = False}
Else If {τ_{8a} ≤ Server Form Handler '<form>' ≤ τ_{8b} → feature = Suspicious}
Else {feature = True}

4 If {Redirect page # < τ_{9a} → feature = False}
Else If {τ_{9a} ≤ Redirect page # ≤ τ_{9b} → feature = Suspicious}
Else {feature = True}

5 If {*OnMouseOver* < τ_{10} → feature = False}
Else {*OnMouseOver* > τ_{10} → feature=True}

6 If {Right Click Disable < τ_{11} → feature = False}
Else {Right Click Disable > τ_{11} → feature=True}

7 If {Popup < τ_{12} → feature = False}
Else {Popup > τ_{12} → feature=True}

8 If {Irregularity Form < τ_{13} → feature = False}
Else {Irregularity Form > τ_{13} → feature=True}

9 If {Title Element Exist → feature = False}
Else {Title Element Exist → feature=True}

3.3 Class 3: Third Party Information Feature (Alexa and WHOIS)

1 If {Website Rank < τ_{14a} → feature = False}
Else If {τ_{14a} ≤ Website Rank ≤ τ_{14b} → feature = Suspicious}
Else {feature = True}

2 If {There is no DNS record for the domain → feature = True
Else {feature = True}

3 If {Age of Domain < τ_{15a} → feature = False}
Else If {τ_{15a} ≤ Age of Domain ≤ τ_{15b} → feature = Suspicious}
Else {feature = True}

Based on the rules discussed, the proposed method will extract and form the feature set for a webpage. Figure 2 shows the template of feature set and Figure 3 shows some samples of the extracted feature sets. The label column will store the label for the webpage, namely, +1 for phishing webpage and -1 for legitimate webpage. There are a total of 27 rules used in this paper and each represents for one feature. Therefore, an extracted feature set for a webpage will consist of one label and 27 features.

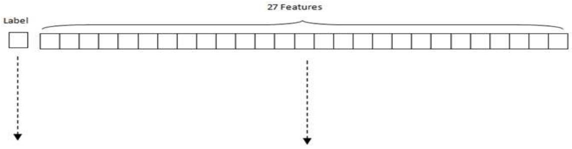
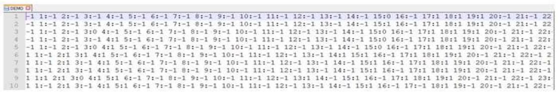


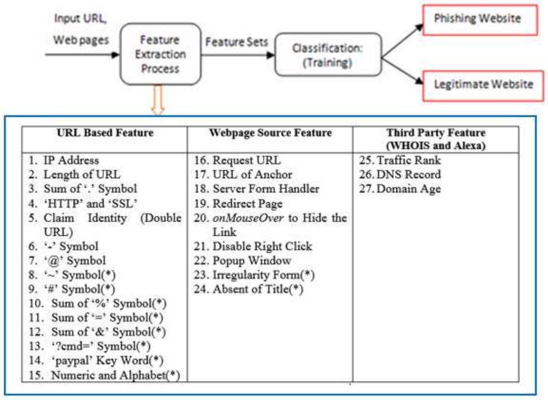
Figure 2: Template of Feature Set



```
-1 1:-1 2:-1 3:-1 4:-1 5:-1 6:-1 7:-1 8:-1 9:-1 10:-1 11:-1 12:-1 13:-1 14:-1 15:0 16:-1 17:1 18:1 19:1 20:-1 21:-1 22
-1 1:-1 2:-1 3:-1 4:-1 5:-1 6:-1 7:-1 8:-1 9:-1 10:-1 11:-1 12:-1 13:-1 14:-1 15:1 16:-1 17:1 18:1 19:1 20:-1 21:-1 22
-1 1:-1 2:-1 3:0 4:-1 5:-1 6:-1 7:-1 8:-1 9:-1 10:-1 11:-1 12:-1 13:-1 14:-1 15:0 16:-1 17:1 18:1 19:1 20:-1 21:-1 22:
-1 1:-1 2:-1 3:-1 4:1 5:-1 6:-1 7:-1 8:-1 9:-1 10:-1 11:-1 12:-1 13:-1 14:-1 15:0 16:-1 17:1 18:1 19:1 20:-1 21:-1 22:
-1 1:-1 2:-1 3:0 4:1 5:-1 6:-1 7:-1 8:-1 9:-1 10:-1 11:-1 12:-1 13:-1 14:-1 15:0 16:-1 17:1 18:1 19:1 20:-1 21:-1 22:-
1 1:-1 2:1 3:1 4:1 5:-1 6:-1 7:-1 8:-1 9:-1 10:-1 11:-1 12:-1 13:-1 14:1 15:1 16:-1 17:1 18:1 19:1 20:-1 21:-1 22:-1 2
1 1:-1 2:1 3:-1 4:1 5:-1 6:-1 7:-1 8:-1 9:-1 10:-1 11:-1 12:-1 13:-1 14:-1 15:1 16:-1 17:1 18:1 19:1 20:-1 21:-1 22:-1
1 1:-1 2:1 3:-1 4:1 5:-1 6:-1 7:-1 8:-1 9:-1 10:-1 11:-1 12:-1 13:-1 14:-1 15:1 16:-1 17:1 18:1 19:1 20:-1 21:-1 22:-1
1 1:1 2:1 3:0 4:1 5:1 6:-1 7:-1 8:-1 9:-1 10:-1 11:-1 12:-1 13:1 14:-1 15:1 16:-1 17:1 18:1 19:1 20:-1 21:-1 22:-1 23:
1 1:-1 2:1 3:-1 4:1 5:1 6:-1 7:-1 8:-1 9:-1 10:-1 11:-1 12:-1 13:-1 14:-1 15:1 16:-1 17:-1 18:1 19:-1 20:-1 21:-1 22:-
```

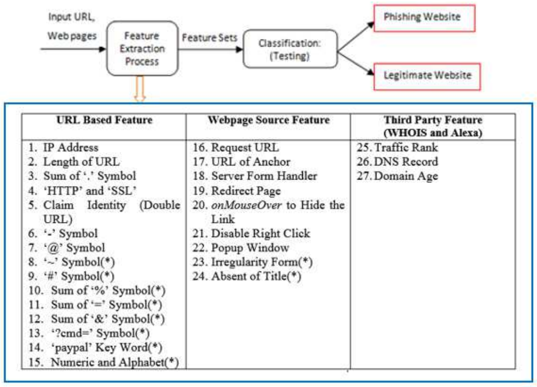
Figure 3: Sample of Some Extracted Feature Sets

The proposed method will feed the extracted feature sets to a classifier. In this paper, we use Support Vector Machine (SVM) as the classifier. It is noteworthy to mention that the use of SVM may be suboptimal; however, changing to an optimal classifier later is effortless. We use the SVM library implemented in [8] with the default setting (i.e., radial basis function is used as the kernel function, and the values for parameter γ and C is set to 1/n[1] and 1.0, respectively). The classification process involves two phrases; there are training and testing phrases. During the training phrase, SVM will produce a model based on the pattern of the extracted feature sets. With the model, SVM will predict the sample's label (phishing or legitimate) in the testing phrase. Figure 4 and 5 show the general flow of training and testing phrases, respectively.



URL Based Feature	Webpage Source Feature	Third Party Feature (WHOIS and Alexa)
1. IP Address 2. Length of URL 3. Sum of '.' Symbol 4. 'HTTP' and 'SSL' 5. Claim Identity (Double URL) 6. '-' Symbol 7. '@' Symbol 8. '~' Symbol(*) 9. '#' Symbol(*) 10. Sum of '%' Symbol(*) 11. Sum of '=' Symbol(*) 12. Sum of '&' Symbol(*) 13. '?cmd=' Symbol(*) 14. 'paypal' Key Word(*) 15. Numeric and Alphabet(*)	16. Request URL 17. URL of Anchor 18. Server Form Handler 19. Redirect Page 20. *onMouseOver* to Hide the Link 21. Disable Right Click 22. Popup Window 23. Irregularity Form(*) 24. Absent of Title(*)	25. Traffic Rank 26. DNS Record 27. Domain Age

Figure 4: General Flow of Training Phrase



URL Based Feature	Webpage Source Feature	Third Party Feature (WHOIS and Alexa)
1. IP Address 2. Length of URL 3. Sum of '.' Symbol 4. 'HTTP' and 'SSL' 5. Claim Identity (Double URL) 6. '-' Symbol 7. '@' Symbol 8. '~' Symbol(*) 9. '#' Symbol(*) 10. Sum of '%' Symbol(*) 11. Sum of '=' Symbol(*) 12. Sum of '&' Symbol(*) 13. '?cmd=' Symbol(*) 14. 'paypal' Key Word(*) 15. Numeric and Alphabet(*)	16. Request URL 17. URL of Anchor 18. Server Form Handler 19. Redirect Page 20. *onMouseOver* to Hide the Link 21. Disable Right Click 22. Popup Window 23. Irregularity Form(*) 24. Absent of Title(*)	25. Traffic Rank 26. DNS Record 27. Domain Age

Figure 5: General Flow of Testing Phrase

4. EXPERIMENTAL RESULTS

We have constructed a dataset from a total of 1000 webpages. The dataset contains 500 phishing and 500 legitimate webpages. Phishing webpages are downloaded from PhishTank[2] and legitimate webpages are from Alexa[3] with different categories (i.e., banking, social networking, news, e-commerce, forums and blogging). The experiment has dedicated 70 percent of the dataset for training phrase and the remaining 30 percent for testing phrase. Three experiments are designed to evaluate the effectiveness of the proposed method, as well as comparing our performance with the existing method proposed by Rami et al. [1]. The experiments also include comparison between the three classes of features and also the comparison within all the 27 features. Table 2 shows all the threshold values used in the experiments.

Table 2: Threshold Values Used in the Experiments

URL-based Feature	Webpage Source Feature	Third Party Feature (WHOIS and Alexa)
Length of URL $\tau_{1a} = 54$ $\tau_{1b} = 75$	Request URL $\tau_{6a} = 2$ $\tau_{6b} = 5$	Traffic Rank $\tau_{14a} = 1000$ $\tau_{14b} = 5000$
Sum of '.' Symbol $\tau_2 = 3$	URL of Anchor $\tau_{7a} = 2$ $\tau_{7b} = 5$	Domain Age $\tau_{15a} = 1$ $\tau_{15b} = 50$
Sum of '%' Symbol $\tau_3 = 2$	Server Form Handler $\tau_{8a} = 2$ $\tau_{8b} = 5$	
Sum of '=' Symbol $\tau_4 = 2$		
Sum of '&' Symbol $\tau_5 = 2$	Redirect Page $\tau_{9a} = 2$ $\tau_{9b} = 5$	
	onMouseOver to Hide the Link $\tau_{10} = 1$	
	Disable Right Click $\tau_{11} = 1$	
	Popup Window $\tau_{12} = 1$	
	Irregularity Form $\tau_{13} = 2$	

4.1 Performance Comparison between the Proposed Method and Rami et al. Method [1]

We implemented both the proposed method and Rami et al. method [1] using Matlab and evaluated on the same dataset. In order to get a fair comparison, we run our experiment using a few types of cross validations. Table 3 shows the detection results for the experiment (we abbreviate Rami et al. method [1] as Rami) and are computed using the following equation:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \qquad (1)$$

[1] n is the number of features used during the classification and in our case, it is 27

[2] http://www.phishtank.com

[3] http://www.alexa.com

where TP, TN, FP and FN denote true positive, true negative, false positive and false negative, respectively. The results show the proposed method has improved Rami’s method. The results also show consistency across different cross validations. The highest improvement result of four percent is obtained using cross validation with K=15. Whereas cross validation with K=5 and holdout method obtained same improvement percentage of 3.34 percent. The improvements have validated the efficiency of the newly proposed features.

Table 3: Performance Comparison between the Proposed Method and Rami et al. Method [1]

	Holdout Method	Cross Validation Method	
		K=5	K=15
Proposed Method	95.67%	94.67%	95.33%
Rami	92.33%	91.33%	91.33%
Differences	3.34%	3.34%	4.00%

4.2 Performance comparison between the Three Classes of Features

This section is devoted to gain some insight on the performance of each individual class and combination of them. We run six tests in this experiment. Test 1 to 3 involves a single class of features and Test 4 to 6 involves a combination of two classes of features. For example, Test 1 is using only URL-based feature. Whereas Test 4 is using the combination of URL-based feature and webpage source feature. Table 4 shows the experimental results for this comparison. The table clearly shows that URL-based feature is superior compared to the other two classes and achieved 93.33 percent. Due to the URL-based feature's effectiveness, its combination with the other classes of features also produced promising results. This is showed in Test 4 and 5. From the table, third party information feature has the lowest performance. However, its role is still important to detect some outliners. For example, there are some legitimate website which are not W3C compliance and this may cause a false alarm to the proposed method. Therefore, using the information extracted from WHOIS and Alexa will remedy the problem. This experiment is important because it shows the effectiveness level of each feature class as well as the combination among the classes. This evaluation is necessary and allows us to decide the final feature set.

Table 4: Performance Comparison between Different Classes of Features

Comparison Between 3 Classes	Holdout Method
Test 1: URL-based Feature	93.33%
Test 2: Webpage Source Feature	76.00%
Test 3: Third Party Information Feature	68.33%
Test 4: URL-Based Feature + Webpage Source Feature	96.67%
Test 5: URL-Based Feature + Third Party Feature	93.00%
Test 6: Webpage Source Feature + Third Party Feature	79.00%

4.3 Performance comparison between Individual Feature

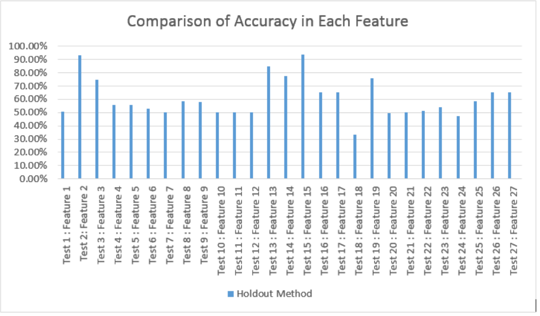


Figure 6: Performance Comparison between Individual Features

In addition to the evaluation of different classes of features, this section discusses performance comparison between individual features. The experiment involves 27 tests that use only one feature at a time to perform the detection. Figure 6 shows the results for all the tests. Notably, there are three most effective features and have achieved detection performance above 80 percent. They are feature 15 (alphanumeric), feature 2 (length of URL) and feature 13 (existence of ?cmd=). Feature 15 and 13 are the newly proposed features and feature 2 is from [1].

5. DISCUSSION

The accuracy of proposed technique is 95.33 percent, while the accuracy of the benchmarked method [1] is 91.33 percent. The four percent enhancement indicates that the proposed technique is superior than the benchmarked method. Hence, the addition of 10 newly added features are important to classify the websites. During the experiments when we tested with single class of features alone, URL-based feature is the most effective and achieved 93.33 percent of accuracy. This indicates that URL-based feature plays an important role in detecting phishing website. Mainly because it is technically easier for phishers to deceive user by constructing an URL with identity liked keyword (e.g., PayPal as the keyword to deceive PayPal user). Unsuspecting user tends to click on an URL when they see the URL contains keyword that they recognise. The experiment also shows that the combination between URL-based and webpage source features yield the highest accuracy of 96.67 percent. The improved accuracy can be attributed to the discriminative features extracted from the content of a webpage. It is noteworthy to mention that phishers will usually alter the webpage to behave maliciously in order to harvest the login credentials. Such attempts are detected in our webpage source features. Although the combination of these two classes of features shows higher accuracy than the combination of all the classes in the experiments, we still opt to combine all classes as the final features set. This is because the third party information feature can serve as a complementary feature to capture some outliner of phishing websites and reduce false positive for some new legitimate website.

Note that the feature extraction of proposed method is currently focused on the textual elements. Phishers may be able to bypass the detection by manipulating on graphical elements. For example, phishers may use an image as the canvas and overlap on the whole webpage. This image will contain the textual content of the webpage and looked like a normal webpage.

6. CONCLUSION

In this paper, a feature-based phishing detection technique is proposed. The aggregation between ten new features and 17 existing features from [1] has shown improvement in the detection performance. A series of experiments have been conducted to validate the efficiency of the newly added features. The experimental results show a promising outcome with as high as 95.33 percent of accuracy and manage to improve four percent compared to the original method by Rami et al. [1].

7. ACKNOWLEDGEMENT

The funding for this project is made possible through the research grant obtained from UNIMAS and the Ministry of Education, Malaysia under the RAGS Grant [RAGS/SG/05(4)/929/2012 (30)].

8. REFERENCES

[1] M. M. Rami, F. Thabtah and L. McCluskey, An Assessment of Features Related to Phishing Websites using An Automated Technique, *7th International Conference for Internet Technology and Secured Transaction*, 2012.

[2] M. M. Gaurav and J. Anuraj, Anti-Phishing Techniques: A Review, *International Journal of Engineering Research and Applications*, Vol. 2 (2), 2012, pp. 350-355.

[3] C. Ludl, S. McAllister, E. Kirda and C. Kruegel, On Effectiveness of Techniques to Detect Phishing Sites, *4th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, 2007, pp. 20-39.

[4] H. H. Jun and H. Kim, Phishing Detection with Popular Search Engines: Simple and Effective, *4th Canada-France MITACS Workshop on Foundations and Practice of Security*, 2011, pp. 194-207.

[5] S. Garera, N. Provos, M. Chew and A. D. Rubin, A Framework for Detection and Measurement of Phishing Attacks, *Proceedings of the 2007 ACM Workshop on Recurring Malcode*, 2007, pp. 1-8.

[6] R. Dhamija, J. D. Tygar and M. Hearst, Why Phishing Works, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2006, pp. 581-590.

[7] A. Le, A. Markopoupou and M. Faloutsos, PhishDef: URL Names Say It All, *IEEE International Conference on Computer Communications*, 2011, pp. 191-195.

[8] C-C Chang and C-J Lin, LIBSVM: A library for support vector machines, *ACM Transactions on Intelligent Systems and Technology, Vol. 2 (3)*, 2011, pp. 27:1-27:27.

[9] K. L. Chiew, E. H. Chang, S. N. Sze and W. K. Tiong, Utilisation of website logo for phishing detection, Computers & Security, Vol. 54, 2015, pp. 16-26.