# Towards Developing a Tool to Detect Phishing URLs: A Machine Learning Approach

Ram B. Basnet[1], Tenzin Doleck[2]
[1]Colorado Mesa University, rbasnet@coloradomesa.edu
[2]McGill University, tenzin.doleck@mcgill.ca

***Abstract*— Despite efforts to curb online fraud, there continues to be a significant proliferation of fraud in the online space. In the same vein, Phishing attacks are a significant and growing problem for users, and carrying out certain actions such as mouse hovering, clicking, etc., on malicious URLs may cause unwary users to unwittingly fall victim to identity theft and problems. In this paper, we propose a methodology that could be used towards developing an anti-phishingURL tool to thwart a phishing attack by either masking the potentially phishing URL or by alerting the user about the potential threat.**

***Keywords- machine learning, phishing, tools, phishing URLs***

## I. INTRODUCTION

Phishing URLs are URLs that lead users to a phishing web page and are usually distributed via phishing messages with links to the phishing site, Internet downloads, social networking sites, vulnerable web sites (such as blogs, forums), instant messaging (IM), etc. Blacklisting is the most common anti-phishing technique used by modern web browsers. However, study [1] shows that centralized, blacklist-based protection alone is not adequate to protect end users from new and emerging phishing webpages that appear in droves and quickly disappear. Furthermore, the study [1] highlights that heuristics based phishing techniques outperform centralized blacklisting techniques. Thus, methods that are discovery-oriented, dynamic, and semi-automated are needed to address the shortcomings of blacklisting. We present a heuristic-based methodology for automatically classifying URLs as being potentially phishing. This methodology could then be used towards developing an anti-phishingURL tool to thwart a phishing attack by either masking the potentially phishing URL or by alerting the user about the potential threat.

The work by Garera et al. [2] is related to our work; the study employed logistic regression over 18 hand-selected features to classify phishing URLs. Though similar in goal, our approach differs significantly in both methodology (considering new publicly available features based on URLs alone and comparing several machine learning algorithms) and scale (considering more features and samples). Ma et al. propose a method to classify malicious URLs using variable number of lexical and host-based properties of the URLs; using these features, they compare the accuracy of batch and online learning algorithms [3. 4]. Though we use some similar features and classification models, our approach is different in a number of ways: firstly, the scope of our work is limited to detecting phishing URLs as opposed to detecting wide range of malicious URLs; secondly, we have a fixed set of smaller number of features; thirdly, we do not use host-based properties of web pages such as WHOIS entries, connection speed, etc. Whittaker et al. [5] describe the scalable machine learning classifier that has been used in maintaining Google's phishing blacklist automatically. Their proprietary classifier system classifies web pages submitted by end users and URLs collected from Gmail's spam filters. Though some URL based features are similar, we propose several new features and evaluate our approach with publicly available machine learning algorithms and public data sets. Unlike their approach, we do not use any proprietary and page content-based features.

## II. METHOD

We propose a heuristic-based approach to classify phishing URLs by using the information available only on URLs. We treat the problem of detecting phishing URLs as a binary classification problem with phishing and benign URLs belonging to the positive and negative class respectively. We first run scripts to collect our phishing and benign URLs to create our data sets. Our next batch of scripts then extracts a number of features by employing various publicly available resources in order to classify the instances into their corresponding classes. We then apply various machine learning algorithms to build models from training data. Separate set of test data are then supplied to the models, and the predicted class of the data instance is compared to the actual to the class of the data to compute the accuracy of the classification models. Figure 1 provides the overview of graphical representation of phishing URL detection framework. This methodology could then be used towards developing an anti-phishingURL tool to thwart a phishing attack by either masking the potentially phishing URL, or by alerting the user about the potential threat.

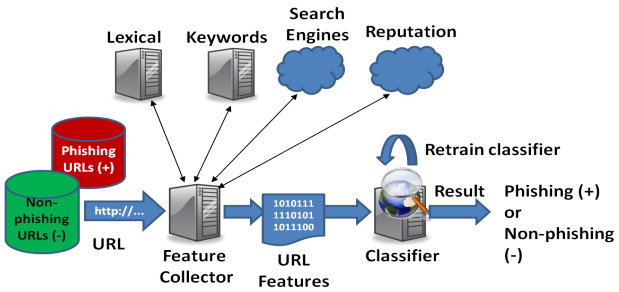


Fig. 1. Overview of framework

## III. DATA SET

For phishing URLs, we coded scripts to automatically download confirmed phishing websites' URLs from PhishTank

978-1-4799-6023-1/15 $31.00 © 2015 IEEE
DOI 10.1109/CICT.2015.63

[6]. We collected first set of 11,361 phishing URLs from June 1 to October 31 of (*"OldPhishTank" data set*). Phishing tactics used by scammers evolve over time; to track these evolving URL features, we collected second batch of 5,456 phishing URLs that were submitted for verification from Jan 1 to May 3, 2011 (*"NewPhishTank" data set*). In order to address URLs that were produced using shortening services such as bit.ly, goo.gl, etc., we developed a Python library [7] to utilize the web service API provided by longurl.org to automatically detect and expand shortened URLs. Non-phishing URLs were collected from two public data sources: Yahoo! directory and DMOZ Open Directory Project. We used Yahoo's server redirection service (http://random.yahoo.com/bin/ryl), which randomly selects a link from Yahoo directory and redirects browser to that page. To cover wider URL structures, we made a list of URLs of most commonly phished targets (using statistics of top targets from PhishTank). We then crawled those URLs, parsed the retrieved HTML contents, and harvested the hyperlinks therein (to use as non-phishing URLs). We use 22,213 legitimate URLs using these sources collected between Sep 15 to Oct 31, 2010 (*Yahoo data set*). We use 9,636 randomly chosen non-phishing URLs from DMOZ, a directory whose entries are vetted by editors (*DMOZ data set*). We then paired *"OldPhishTank"* and *"NewPhishTank"* data sets with non-phishing URLs from a benign source (either Yahoo or DMOZ). We refer to these data sets as the *OldPhishTank-Yahoo (OY)*, *OldPhishTank-DMOZ (OD)*, *NewPhishTank-Yahoo (NY)*, and *NewPhishTank-DMOZ (ND)*.

## IV. Feature Analysis

We developed our set of 138 features in detecting phishing URLs based on related works, drawing primarily from [2, 5, 8, 9, 10]. Some of these features are modified to fit our needs, while others are newly proposed. We group features that we gathered into 4 broad categories. We describe each feature category with their statistics from a randomly selected 80% of *OldPhishTank-Yahoo* training data set (we call it *"Random Set"*) in the following sub sections.

### A. Lexical based features

Lexical features, the textual properties of the URL itself, have been widely used in literature [3, 12, 18, 19, 33, 34] in detecting phishing attacks. We examine various obfuscation techniques phishers may employ and derive a number of phishing like features to use in our classifiers. We summarize the real-valued and binary features separately in Tables 1 and 2, respectively.

TABLE I. Lexical-Based Features and Their Statistics

| Feature Description | URL Type | Max | Min | Mean | Median |
|---|---|---|---|---|---|
| Length of Host | Phishing | 240 | 4 | 21.38 | 19 |
| | Non-phishing | 70 | 5 | 18.77 | 18 |
| Number of '.' in Host | Phishing | 30 | 0 | 2.13 | 2 |
| | Non-phishing | 5 | 1 | 2.14 | 2 |
| Number of '.' in Path | Phishing | 18 | 0 | 0.86 | 1 |
| | Non-phishing | 13 | 0 | 0.25 | 1 |
| Number of '.' in URL | Phishing | 30 | 0 | 3.00 | 3 |
| | Non-phishing | 15 | 1 | 2.38 | 2 |
| Length of Path | Phishing | 380 | 0 | 24.55 | 15 |
| | Non-phishing | 360 | 0 | 10.74 | 1 |
| Length of URL | Phishing | 999 | 13 | 66.09 | 18 |
| | Non-phishing | 383 | 15 | 41.22 | 33 |

TABLE II. Lexical-Based Binary Valued Features and Statistics

| Feature Description | % Phishing URLs | % Non-phishing URLs |
|---|---|---|
| '-' in Host | 2.02% | 9.03% |
| Digit [0-9] in Host | 30.06% | 3.11% |
| IP Based Host | 4.15% | 0.00% |
| Hex Based Host | 0.18% | 0.00% |
| '-' in Path | 15.82% | 6.64% |
| '/' in Path | 98.39% | 96.18% |
| '=' in Path | 4.58% | 0.16% |
| ';' in Path | 0.07% | 0.00% |
| ',' in Path | 0.15% | 0.28% |
| Has Parameter Part | 0.18% | 0.77% |
| Has Query Part | 0.07% | 0.01% |
| '=' in Query Part | 13.45% | 10.43% |
| Has Fragment Part | 0.18% | 0.77% |
| '@' in URL | 0.33% | 0.08% |
| 'Username' in URL | 0.33% | 0.08% |
| 'Password' in URL | 0.02% | 0.00% |
| Has Non-Standard Port | 0.01% | 0.00% |
| '_' in Path | 11.16% | 8.41% |

### B. Keyword based features

Many phishing URLs contain word tokens like login, verify, etc. to attract users' attention. Using the *"Random Set"*, we tokenize each phishing URL by splitting it using non-alphanumeric characters and apply Porter stemmer [11] to obtain 12,012 unique root tokens and their frequencies. We discard all tokens with: A). length < 3 (such as d, it), B). common URL parts (such as http, www) and webpage file extensions (such as html, asp), C). top target organizations such as paypal, ebay (since they are covered under reputation-based features), D). random characters (such as ykokejox, njghlfi). This selection resulted in 1,127 tokens. We then computed mutual information (MI) of each term in phishing class. MI measures how much information the presence or absence of a term contributes to making correct classification decision on a class [12]: $MI(U;C) = \frac{N_{11}}{N} log_2 \frac{NN_{11}}{N_{1.}N_{.1}} + \frac{N_{01}}{N} log_2 \frac{NN_{01}}{N_{0.}N_{.1}} + \frac{N_{10}}{N} log_2 \frac{NN_{10}}{N_{1.}N_{.0}} + \frac{N_{00}}{N} log_2 \frac{NN_{00}}{N_{0.}N_{.0}}$ Terms with high MI values indicate that they are more relevant to the class. Table 3 shows only top 10 terms based on MI along with the percentage of each term appearing in phishing and non-phishing URLs. By ordering the terms based on MI from high to low, we use these terms as binary features on data set OY. Using forward feature selection method, we train and test Naïve Bayes 1,127 times for each feature set size from 1 to 1,127 and record its error rate for each run. Figure 2 shows the error rate on feature size from 1 to 1,127, which decreases from ~29% to ~24% as feature size increases. After 100 features, change in error rate is statistically insignificant (< 0.1%); thus, top 101 terms based on their MI as keyword-based features were used.

TABLE III. TOP 10 ROOT TERMS (BASED ON MI) AND THEIR STATISTICS

| Root Term | MI | % Phishing URLs | % Non-phishing URLs |
|---|---|---|---|
| log | 0.1740 | 21.77% | 1.71% |
| pay | 0.1027 | 13.26% | 0.50% |
| web | 0.0778 | 14.90% | 1.62% |
| cmd | 0.6840 | 10.08% | 0.37% |
| account | 0.0559 | 7.86% | 0.34% |
| dispatch | 0.0390 | 5.69% | 0.01% |
| free | 0.0362 | 7.20% | 0.48% |
| run | 0.0331 | 4.89% | 0.16% |
| net | 0.0320 | 13.05% | 5.05% |
| confirm | 0.0292 | 3.42% | 0.00% |

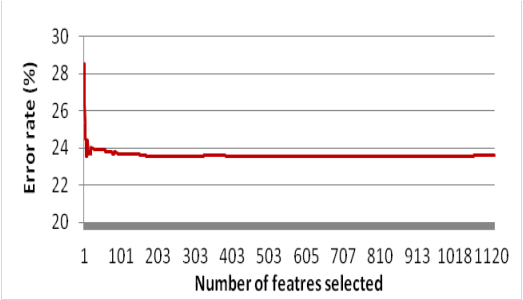


Fig. 2. Error rates on feature size

### C. Reputation based features

We downloaded 3 types of statistics: Top 10 Domains, Top 10 IPs, and Top 10 Popular Targets from PhishTank's statistics published from Oct '06 to Oct '10, to make use of historical data on top IPs and domains that host phishing websites. There were 311 unique domains, 354 unique IPs, and 43 unique targets during the aforementioned period. We also include statistics from StopBadware.org; StopBadware.org works with its network of partner organizations such as Google and individuals to counter viruses, spyware, etc. [13]. It produces top 50 IP address report from a number of reported URLs. If the IP address of a URL belongs to this top 50 report, we flag it as potentially phishing. We use Safe Browsing API [14] to check URLs against Google's constantly updated blacklists of suspected phishing and malware pages, and use 3 binary features for membership in those blacklists. Table 4 summarizes the distribution of reputation-based features in phishing and non-phishing URLs.

TABLE IV. REPUTATION BASED FEATURES AND THEIR STATISTICS

| Feature Description | % Phishing URLs | % Non-phishing URLs |
|---|---|---|
| PhishTank Top 10 Domain in URL | 20.98% | 4.87% |
| PhishTank Top 10 Target in URL | 32.65% | 14.21% |
| IP in PhishTank Top 10 IPs | 17.30% | 0.87% |
| IP in StopBadware Top 50 IPs | 2.31% | 1.37% |
| URL in Phishing Blacklist | 42.41% | 0.00% |
| URL in Malware Blacklist | 0.45% | 0.05% |
| URL in RegTest Blacklist | 0.16% | 0.00% |

### D. Search engine based features

We check if the URL exists in the search engines' index. We search for the whole URL and retrieve the top 10 results. If the results contain the URL, we consider it as a potentially legitimate URL, phishing otherwise. We also check if the domain part of the URL matches the domain part of any links in the results. If there is a match, we flag the URL as a potentially legitimate URL. Otherwise, we query the search engine again with just the domain part of a URL. If none of the returned links matches the queried URL, we flag the URL as potentially phishing. If both the URL and the domain do not exist in search engines' index, it is likely that the domain is a newly created one and the URL in question is likely to be phishing. We employ 3 major search engines (Google, Bing, and Yahoo); reason being that at least one of them may have indexed legitimate website. Search engine based features & their statistics are shown in Table 5.

TABLE V. SEARCH ENGINE BASED FEATURES AND THEIR STATISTICS

| Feature Description | % Phishing URLs | % Non-phishing URLs |
|---|---|---|
| URL NOT in Google Top Results | 98.71% | 4.85% |
| Domain NOT in Google Top Results | 98.27% | 2.64% |
| URL NOT in Bing Top Results | 96.95% | 34.63% |
| Domain NOT in Bing Top Results | 96.34% | 12.77% |
| URL NOT in Yahoo Top Results | 98.93% | 17.74% |
| Domain NOT in Yahoo Top Results | 98.71% | 13.95% |

## V. RESULTS

We evaluate several supervised batch-learning classifiers to empirically compare a number of classifiers and determine the one that yields the best performance in the problem of detecting phishing URLs. We evaluate the following 7 classifiers implemented in WEKA library [15] with their default parameter values: 1). Support Vector Machines (SVMs with rbf kernel) [16], 2). SVMs with linear kernel 3). Multilayer Perceptron (MLP) [17], 4). Random Forest (RF) [18], 5). Naïve Bayes (NB) [19], 6). Logistic Regression (LR) [20], 7). C4.5 [21] – which is implemented as J48 in WEKA. Using the features, we encode each individual URL into a feature vector with 138 dimensions. We scale the real-valued features, available mostly in lexical based features, to lie between 0 and 1; scaling equalizes the range of features in real-valued and binary features further emphasizing that we are treating each feature as equally informative and important. We use 10 times 10-fold cross-validation (unless otherwise stated) to evaluate the classifiers. Figure 3 compares the overall error rates of all classifiers on the four datasets. The differences in overall error rates on all the classifiers are not significant on each data set. Random Forest (RF) performs the best in all performance metrics followed by J48 on each of four data sets. Naïve Bayes (NB) consistently performs the worst followed by SVM-rbf on all data sets.

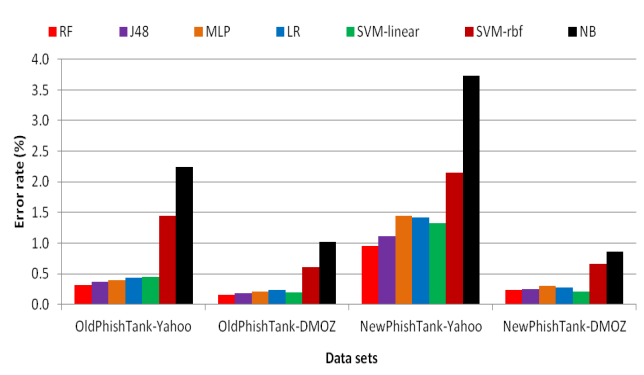


Fig. 3. Overall error rates of Classifiers using all features.

## VI. CONCLUSION

In this paper, we proposed new search engines, reputation, and statistically mined keyword based features for classifying phishing URLs. We empirically demonstrated that the proposed features are highly relevant to the automatic discovery and classification of phishing URLs. We evaluated our approach on real-world data sets with more than 16,000 phishing and 31,000 non-phishing URLs. Our experiments obtained an error rate of less than 0.3% while maintaining about 0.2% false positive and 0.5% false negative rates. Featured with high accuracy rate, we believe that our lightweight approach can be used to develop a tool for phishing URL detection.

## REFERENCES

[1] S. Sheng, B. Wardman, G. Warner, L.F. Cranor, J. Hong, C. Zhang, An empirical analysis of phishing blacklists, In: Proc. 6th Int. Conf. Email and Anti-Spam, CEAS'09, Mountain View, California, USA, 2009.

[2] S. Garera, N. Provos, M. Chew, A.D. Rubin, A framework for detection and measurement of phishing attacks. In: Proc. 5th ACM Workshop on Recurring Malcode, WORM'07, ACM, New York, NY, USA, 2007, pp. 1-8.

[3] J. Ma, L.K. Saul, S. Savage, G.M. Voelker, Beyond blacklists: Learning to detect malicious web sites from suspicious URLs, In: Proc. 15th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, Paris, France, 2009, pp. 1245-1254.

[4] J. Ma, L.K. Saul, S. Savage, G.M. Voelker, Identifying suspicious URLs: an application of large-scale online learning, In: Proc. 26th Annual Int. Conf. Machine Learning, ICML'09, Montreal, Quebec, Canada, 2009, pp. 681-688.

[5] C. Whittaker, B. Ryner, M. Nazif, Large-scale automatic classification of phishing pages, In: Proc. 17th Annual Network and Distributed System Security Symposium, NDSS'10, San Diego, CA, USA, 2010.

[6] PhishTank. Out of the net, into the tank, http://www.phishtank.com, accessed on June 18, 2010.

[7] R.B. Basnet, PyLongURL - Python library for longurl.org, software available at: http://code.google.com/p/pylongurl/, 2010.

[8] R.B. Basnet, S. Mukkamala, A.H. Sung, Detection of phishing attacks: a machine learning approach, In: Bhanu Prasad (Ed.), Studies in Fuzziness and Soft Computing, Springer, 2008, pp. 373-383.

[9] I. Fette, N. Sadeh, A. Tomasic, Learning to detect phishing emails, In: Proc. Int. Conf. World Wide Web, WWW'07, Banff, Alberta, Canada, 2007, pp. 649-656.

[10] Y. Zhang, J. Hong, L. Cranor, CANTINA: a content-based approach to detecting phishing web sites, In: Proc. 16th Int. Conf. World Wide Web, WWW'07, Banff, Alberta, Canada, 2007, pp. 639-648.

[11] Natural Language Toolkit (NLTK), http://www.nltk.org, accessed on July 15, 2011.

[12] H. Peng, F. Long, C. Ding, Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy, IEEE Transaction on Pattern Analysis and Machine Intelligence 27 (2005) 1226-1238.

[13] StopBadware, IP Address Report – Top 50 by number of reported URLs, http://stopbadware.org/reports/ip, accessed on June 12, 2010.

[14] Google Safe Browsing API - Google Code, http://code.google.com/apis/safebrowsing/, accessed on June 12, 2010.

[15] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The WEKA data mining software: an update, SIGKDD Explorations, 11 (2009) 10-18.

[16] V.N. Vapnik, The nature of statistical learning theory, Springer, 1995.

[17] K. Hornik, Multilayer feedforward networks are universal approximators, Neural Networks 2 (1989) 359-366.

[18] L. Breiman, Random forests, http://oz.berkeley.edu/users/breiman/randomforest2001.pdf, 2001, accessed on February 20, 2011.

[19] G.H. John, P. Langley, Estimating continuous distributions in Bayesian classifiers, In: Proc. 11th Conf. Uncertainty in Artificial Intelligence, San Mateo, CA, USA, 1995, pp. 338-345.

[20] M.T. Brannick, Logistic regression, http://luna.cas.usf.edu/~mbrannic/files/regression/Logistic.html, accessed on February 27, 2011.

[21] J. R. Quinlan, C4.5 programs for machine learning, Morgan Kaufmann Publishers, San Mateo, CA, USA, 1993.