International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 06, June - 2017

Real Time Phishing Detection on Genrated URL's

Mr. Sagar Hangarage
Department Of IT
MMCOE, Pune, Maharashtra, India

Ms. Shilpa Arsul
Department Of IT
MMCOE, Pune, Maharashtra, India

Mr. Rohan Mithapelli
Department Of IT
MMCOE, Pune, Maharashtra, India

Mr. Vishal Palse
Department Of IT
MMCOE, Pune, Maharashtra, India

Mr. Nikhil Dhawase
Assistant Professor Department of IT
MMCOE, Pune, Maharashtra, India

***Abstract*— Phishing URLs usually have few relationships between the part of the URL that must be registered (low-level domain) and the remaining part of the URL. Define the new concept of intra-URL relatedness and evaluate it using features extracted from words that compose a URL based on query data from Google and Yahoo search engines. These features are then used in machine-learning- based classification to detect phishing URL's from a real dataset. Phishing is currently one of the most lucrative cybercrime activities. Although accurately evaluating the financial loss caused by phishing is difficult, some surveys have been conducted, suggesting losses of several billion dollars every year. In this paper, we propose an automated real-time URL phishingness rating system to protect users against phishing content: PhishStorm. our approach evaluates the relatedness of words that compose a URL and highlights the differences between legitimate and phishing URL's. PhishStorm gives a generic solution for phishing URL detection relying on intra URL relatedness computation. This technique only needs access to search engine query data to operate. Hence the application range of PhishStorm is wide. It can operate at different network level to prevent phishing. This paper introduces PhishStorm, an efficient phishing URL detection system relying on URL lexical analysis. The approach is based on the intra-URL relatedness. This relatedness reflects the relationship among the words blended into a URL and particularly into the part of the URL that can be freely defined and the registered domain. In recent years, many techniques have been developed to cope with phishing and have focused on the real-time identification of this threat. One approach is to compare the content of presumed phishing.**

Keywords*— *Security management, machine learning, phishing detection, URL rating, word relatedness, search engine query data, STORM.

I. INTRODUCTION

PHISHING scams are typically fraudulent email messages appearing to come from legitimate enterprises. These messages usually direct you to a spoofed website or otherwise get you to divulge private information. More ever, Evaluating the financial loss due to phishing is difficult, surveys has been done, which suggest several billion losses every year. According to Anti-Phishing Working Group[1] , the number of commercial brands being attacked by phishing just hit a new record: 356 brands in October 2009. With major industry targets, such as, financial and payment services, phishing has caused billions of dollars loss annually. Javelin Strategy & Fraud published a report that identity theft led to a loss of $54 billion in 2009, due to Cybercrime. Various techniques are used to perform phishing attacks are Spear phishing, Session Hijacking, Email/Spam, Link Manipulation etc. to social engineering.

Luring Internet users by making them click on rogue links that seem trustworthy is an easy task because of widespread credulity and unawareness. To cope with this threat, the best strategy is to prevent connection to phishing Web sites by the identification of phishing URLs.

In this paper, we propose an Real-time URL phishingness rating system to protect users against phishing content: PhishStorm. Over here this underlying method targets identification of phishing URLs that are based on registered domains that are not related to their targeted brand. The intra-URL relatedness is the quantification of the relatedness among the words composing the different parts of a URL and more precisely between the registered domain and the rest of the URL. To Lure their victims, phishers blend many phishing keywords (Popular brands, Trusted names) into the remaining parts of the URL. Therefore, our approach evaluates the relatedness of words that compose a URL and highlights the differences between legitimate and phishing URLs. Most Internet users are not aware of the DNS hierarchy. Seeing words like paypal or ebay at any level of a URL will make them feel confident that the fake link actually leads to the official Web site of these brands.

These tools, coming from the natural language processing field, usually have no entries for domain names and most of the words that compose a URL. We leverage search engine query data from Google to compute this relatedness. The Google Safe Browsing service protect the Internet users from visiting phishing site[2].

We define the term of intra-URL relatedness. Efficient feature computation methods leveraging distributed streaming analytics techniques and space-efficient data

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 06, June - 2017

structure are used. These reduce the delay of detecting phishing URLs which we observed in our previous work [3] and permit phishing email as wider applications. Over here we extract 12 features from a single URL which are given as input to machine learning algorithms to do the process of identify phishing URLs. Our technique is assessed on ground truth data of 96018 URL sleading to a correct classification rate of 94.91%. Finally, a phishingness score is computed for every single URL based on Random Forest classifier.

To summarize the major contributions of this paper:

— We introduce the concept of intra-URL relatedness representing the relation between a registered domain and other part of the domain.

— We leverage search engine query data to establish relatedness between words and show that this is more suited to Internet vocabulary than existing methods.

— We propose new features based on intra-URL relatedness and build a machine learning based approach relying on these for distinguishing between phishing and non-phishing URLs.

This paper is an extension of [3] with the following additional contributions:

— We use distributed real-time computation technique (STORM) to infer intra-URL relatedness.

— We use data structures with the ability to store and manage data that consumes the least amount of space with little to no impact on performance (Bloom filter) to reduce the delay in intra URL relatedness features calculation. A detailed feature engineering process is performed. The rest of this paper is structured as follows: we start by presenting URL obfuscation techniques in Section II. Then we have intra-URL relatedness analysis in Section III. Further we define the architecture implementation of PhishStrom in Section IV. Details about the phishing URL detection process in Section V. We have conclusion and future work of Phishstrom in Section VI.

II. PHISHING URL OBFUSCATION

In the following Type explained below we present the most used URL obfuscation techniques [4], with examples given in Table I for the domain paypal.com:

• Type I: URL obfuscation with other domain: The mld.ps is a real domain name, usually registered by the phisher, while the original domain being phished is part of the path, the query or the upper level domain.

• Type II: URL obfuscation with keywords: Again the mld.ps is are al domain name, and the brand being phished and related words are part of the path, the query or upper level domain.

• Type III: Typosquatting domains or long domains: the mld.ps of the URL is the domain being phished but misspelled, with letters or words missing or added, or the domain is pronounced the same way as the original but written differently. The targeted brand can also be combined with other words to create an unregistered domain.

• Type IV: URL obfuscation with IP address: the URL's hostname is replaced by an IP address and the brand being phished is part of the path or the query.

• Type V: Obfuscation with URL shortner: A URL shortening service is used to hide the name of the real host. Such URLs are not meaningful and are mainly used in phishing attacks targeting services that use this kind of short URL, like Twitter.

Table I
Examples of Obfuscated URL's For Domain paypal.com

Obf. Type	Example
Type I	http://school497.ru/222/www.paypal.com/29370274276105805/ http://paypal.com.eu.compte.client.update.condst.com.br/
Type II	http://www.quadrodeofertas.com.br/www1.paypal-com/encripted/ssl218 http://sezopoztos.com/paypalitlogin/us/webscr.html?cmd=_login-run
Type III	http://cgi-3.paypal-secure.de/info2/verikredit.html http://paypal-shopping.co.il/
Type IV	http://69.72.130.98/janaseva/https.paypal.com/uk/onepagepaypal.htm ftp://212.13.144.72/SERVICE/PayPal.com/security/alert/paypal.com
Type V	http://tiny.cc/clientID00858JD8 http://goo.gl/HQx5g

We focus on the identification of the four first types of URL obfuscation techniques since our technique relies on natural language processing, which is clearly not suited to shortened URLs. The common feature of these obfuscated URLs is that the brand and some related terms are included in the path,the query and low level domain. These terms are related as these have relationships with the targeted brand and have no obvious relation with the *mld.ps* that is used for phishing.

III. INTRA-URL RELATEDNESS ANALYSIS

The examples of obfuscated phishing URL from Type I to IV highlight a global characteristic in URL obfuscation, namely that there is no relation between the *mld.ps* and the rest of the URL. To Analyse this, we split the URL in the two parts that are presumed to have no relationship: extract the *mld.ps* and separate it from the rest. As the *ps* may be composed of multiple level domain, we use Public Suffix List[2] to identify it and then retrieve the immediately preceding level domain as the *mld*. For the rest of the URL, a split according to non-alpha-numeric characters is first performed. From extracted parts composed of several words such as *paypalitlogin* in *http://sezopoztos.com/paypalitlogin/us/*. . . we use a dictionary based word splitter [5]. For instance, the three words *paypal*, *it* and *login* are extracted from *paypalitlogin* through this process. Based on this splitting two sets are composed: one, called RD_{url} (for Registered Domain), consists just of two elements: RD_{url} = {mld, mld.ps}. The other, REM_{url} (for REMaining part), is composed of all extracted words from the URL except *mld.ps*. Given *http:// sezopoztos.com/ paypalitlogin/ us/webscr. html?cmd=_login-run*, the following sets are extracted:

• RD_{url} = {sezopoztos, sezopoztos.com}
• REM_{url} = {paypal, it, login, us,web, src, html, cmd, login, run}

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 06, June - 2017

B. Word Relatedness Evaluation Tools Shortcomings

Automated techniques and measures have also been developed to evaluate word relatedness. Latent Semantic Analysis (LSA) proposed by Landauer and Dumais [6] or Pointwise Mutual Information (PMI), introduced by Church and Hanks [7], then used by Turney [8] based on statistical data from search engine results for the queried words, are examples of these techniques. The Normalized Google Distance (NGD [9]) computes the semantic similarity between two words by querying the Google search engine for these words and counts the number of Web pages where they appear together and individually. Disco [10] relies on mutual information evaluation between two words based on corpora.

Table II
NUMBER OF LABELS MATCHING AT LEAST ONE RELATED WORD FOR 4 TOOLS

Tool	#mld	%mld	#mld.ps	%mld.ps
WordNet	20	21.3%	0	0%
Disco	23	24.5%	0	0%
Yahoo Clues	87	92.6%	68	72.3%
Google Trends	92	97.9%	76	80.9%
Total	94	-	94	-

The testing set consisted of the RD_{url} extracted from a set of 94 URLs. These URLs come from PhishTank (described in Section V-A), i.e.,phishing URLs present in PhishTank blacklist are categorised according to the brand they target. As of March 2014, when we made our evaluation, 94 brands and associated URLs were present in this list. The result of the test for each tool is given in the two first rows of Table II. The numbers of *mld* and *mld.ps* for which the tested tools can give at least one related word are given in absolute value and percentage terms.

C. Search Engine Query Data

To perform the evaluation of intra-URL relatedness, we are using search engine query data. The reason for using search engine query data is that URL obfuscation is a social engineering lure. Usually Phishing URLs target a brand, so clever phishers blend within them the brand and words that Internet users associate with the brand, such as a provided service: *payment* for PayPal. Generally, most people use search engine to excess these services. To began with the search for these service, User start with typing some keyword that typically matches the brand or with the domain name and the service needed like *paypal payment* or *statebank.com on-line banking*. These word associations reflect the cognitive process of users searching for PayPal or statebank.

Google Trends shows the relative interest of Google users over time in a term. It depicts the geographic interest for this term and provides related terms according to users related searches. Google Trends provides the top ten related searches over time as well as the ten rising related searches namely those on which interest has increased recently. This allows us to gather up to twenty related terms for one given term.

Yahoo Clues provides the same kind of services as Google Trends. It offers an insight into the search flows, the terms requested just before (5 terms) and after (5 terms) a term. Like Google Trends it also provides a set of related searches

Having a URL *url* and the extracted sets RD_{url} and REM_{url}, Google Trends and Yahoo Clues are automatically requested for each element of the two sets. We define $Term_w$,
as the set of terms resulting from the requests of the word w in
both Google Trends (related & rising) and Yahoo Clues (related & requests). A subset of $Term_{paypal}$ is given in Table IV with $Term_{paypal}$ = {{paypal, account}, {paypal, login}, . . }.

We define four sets of words built from a URL *url*:
$REL_{rd}(url)$, $REL_{rem}(url)$, $AS_{rd}(url)$ and $AS_{rem}(url)$. $REL_{set}(url)$ consists of all the words *related* to the words of set, i.e., words included in terms that are results of requests for elements of set. Here set is either RD_{url} or REM_{url}. The formulas for these sets are given in (1) and (2).

$$REL_{rd}(url)=\{w \in t \mid t \in Term_{w'}, w' \in RD_{url}\} \quad (1)$$

$$REL_{rem}(url)=\{w \in t \mid t \in Term_{w'}, w' \in REM_{url}\}. \quad (2)$$

$AS_{set}(url)$ is the set of words that are *associated* with the words of *set*, i.e., the words that appear in a common single term. Assuming a term t composed of three words {w_1, w_2, w_3}, there is a symmetric *association* relationship between w_1 and w_2, w_1 and w_3, w_2 and w_3. The two sets $AS_{rd}(url)$ for RD_{url} and $AS_{rem}(url)$ for REM_{url} are defined in (3) and (4) respectively.

$$AS_{rd}(url)=\{w \in t \mid \exists w' \in RD_{url}, w' \in t, w' \not\models w\} \quad (3)$$

$$AS_{rem}(url)=\{w \in t \mid \exists w' \in REM_{url}, w' \in t, w' \not\models w\}. \quad (4)$$

These four sets are extracted to quantify the relationship between and inside each set RD_{url} and REM_{url}. Assume the URL *http:// sezopoztos.com/ paypalitlogin/ us/webscr.html?cmd=_login-run*, Fig. 1 presents the full process from word extraction to $AS_{rem}(url)$ and $REL_{rem}(url)$ composition based on a subset of $Term_{paypal}$. We have:

$REL_{rem}(url)$ = {amazon, paypal, fees, ebay, uk, login}

$AS_{rem}(url)$ = {amazon, fees, login}

Table III
EXAMPLE OF TERM RESULTS FROM GOOGLE TRENDS AND YAHOO CLUES FOR {*paypal*}

Google related	Google rising	Yahoo related	Yahoo requests
{paypal, account}	{amazon, paypal}	{bill, me, later}	{paypal, login}
{paypal, login}	{paypal, fees}	{netspend}	{paypal.com}
{paypal, credit, card}	{ebay, uk}	{suntrust}	{paypal, buyer, credit}
{paypal, email}	{paypal, login}	{regions}	{paypal, customer, service}

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 06, June - 2017

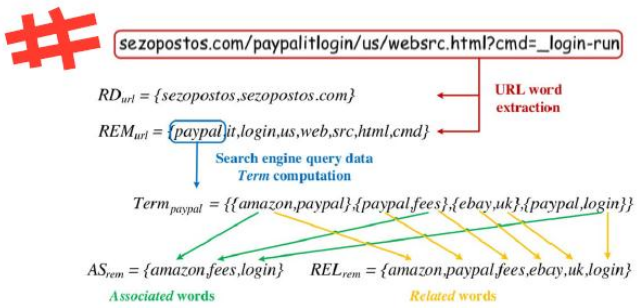


Fig. 1. Word extraction and word set composition for http://sezopoztos.con paypalitlogin/us/webscr.html?cmd=_login-run.

Fig. 1.

D. Feature Computation

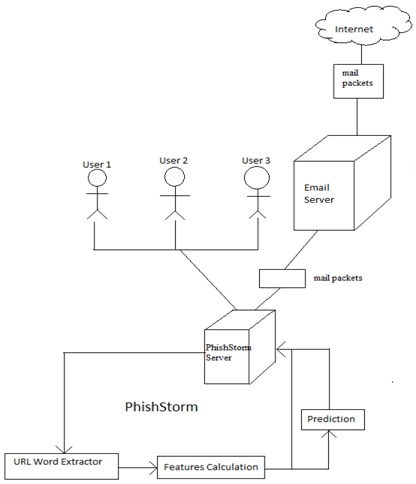
Based on the sets defined in the previous subsection, we introduce 12 features characterising intra-URL relatedness and URL popularity. The popularity criteria is based on the search count for components of a URL (registered domain, *mld*, etc.). These features are described in Table V. The features 1–6 define intra-URL relatedness by calculating the Jaccard index pairwise between the four sets defined in Section III-C (RELrd(url), RELrem(url), ASrd(url) and ASrem(url)). The Jaccard index is a long-established metric used to calculate similarity and diversity between two sets A
and B. The closer J(A,B) is to 1 the more similar are A and B. These six features quantify the relatedness between the two parts of the URL (*mld.ps* and the rest) through JRR, JRA, JAA and JAR, as these compute Jaccard indexes between sets extracted from different parts (RDurl and REMurl). These also measure the relatedness inside each part through JARrd and JARrem, as these features are calculated from sets extracted from the same part of a URL.

Features 7–12 reflect the popularity of a URL and its components with the number of words that compose it (cardrem) and the number of related and associated words found in search engine query data based on these words with ratioArem and ratioRrem. These two features are weighted by cardrem. Features mld.psres and mldres represent the popularity of the registered domain by giving Boolean values describing whether the mld.ps and mld match results while queried in Google Trends and Yahoo Clues. The final feature (*ranking*) is the ranking of the *mld.ps* according to the Alexa5 Web site ranking list. Alexa gives a ranking for the top 1 000 000 most visited Web sites; if a particular *mld.ps* is not in the list, the value10 000 000 is considered.

Features 10–12 can be considered as relying on the reputation of a domain and not on the intra-URL relatedness. Even if features 10 and 11 are new—*ranking* has been used already in state of the art work—we compare in Section VI classification results with and without these three features to assess the relevancy of intra-URL relatedness features.

IV. PHISHSTORM IMPLEMENTATION

PhishStorm gives a generic solution for phishing URL detection relying on intra-URL relatedness computation. This
technique only needs access to search engine query data to operate. PhishStorm has a wide range of applications. To prevent phishing PhishStorm can be operate on different network level. Even while surfing on a web it provide a personal protection for users if implemented locally as a browser add-on. PhishStorm provides phishingness score for URL and can act as a Web site reputation rating systems, displaying a Web site rating while using a search engine or typing a URL into a Web
browser. Centralized phishing protection is another option as
for instance at theWeb proxy level of a local company network, filtering HTTP packets sent from URLs identified as phishing. However, as the main vector of phishing attacks is spoofed emails embedding phishing URLs, we implement PhishStorm as a centralized phishing email detection tool which is positioned in front of the email server. Nowadays, spam filtering is performed centrally in many organization and PhishStorm can be added to such process to increase detection performance. Fig. 2 depicts the implementation of PhishStorm and the four steps of the phishing email detection process. While incoming emails from the Internet reach PhishStorm (1), potential embedded URLs are extracted therefrom. The system then proceeds to features computation thanks to search engine query data and predicts a phishingness score using machine learning techniques (2). A detection threshold is applied to every predicted score, determining if the email must be forwarded to the email server (3) and then to users (4) with its phishingness score or dropped. We give in this section a detailed description of the implementation of the features computation .



PhishStorm Architechture

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 06, June - 2017

V. PHISHING URL DETECTION

To automatically detect phishing URLs, we use supervised classification techniques. We build a feature vector matrix from the dataset. Each feature vector is composed of 12 elements, namely the 12 features described in Section III-D. The predicted variable is 0 for a legitimate URL and 1 for a phishing URL. This gives a matrix of 96 018 feature vectors representing the 96 018 URLs of the testing dataset.

A. URL Classification

Since there is a wide range of supervised classification algorithms, we assessed our dataset according to several classifiers using Weka [11]. Seven classifiers were tested covering treebased (Random Tree, Random Forest, C4.5, LMT) rule-based (PART, JRip) and function-based (SVM). The classification was made without parameters tuning through a ten-fold cross validation as a first step to select the most promising approach. Results for accuracy, true positives and true negatives are given in Fig. 3 for each classifier. To give additional information about confidence interval of classification results for these classifiers, Table IV shows the median, 5th percentile, 95th percentile and standard deviation (SD) values for the Accuracy of each classifier out of 100 runs. For sake of clarity we define for URLs:

• Phishing classified as phishing: true positives (TP) and

$$TP_{rate} = \frac{TP}{TP+FN}$$

• Legitimate classified as phishing: false positives (FP) and

$$FP_{rate} = \frac{FP}{TN+FP}$$

• Legitimate classified as legitimate: true negatives (TN) and

$$TN_{rate} = \frac{TN}{TN+FP}$$

• Phishing classified as legitimate: false negatives (FN) and

$$FN_{rate} = \frac{FN}{TP+FN}$$

and the accuracy:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

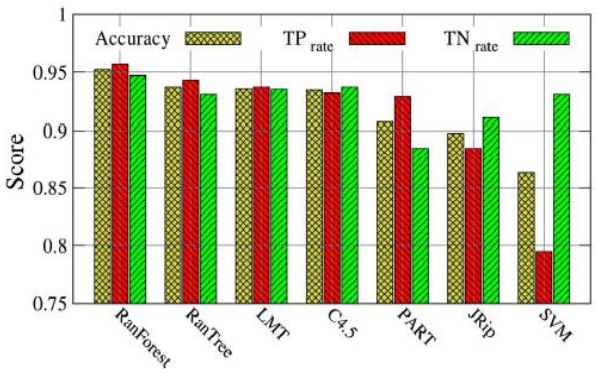


Fig. 3

Table IV

CONFIDENCE INTERVAL FOR CLASSIFICATION RESULTS

Classifier	5^{th} perc.	median	95^{th} perc.	SD
RanForest	94.68	95.22	95.41	0.23
RanTree	93.35	93.81	94.1	0.23
LMT	93.10	93.55	94.01	0.27
C4.5	93.01	93.45	93.87	0.26
PART	89.84	90.62	91.49	0.47
JRip	88.32	89.66	90.48	0.66
SVM	85.23	86.31	87.53	0.62

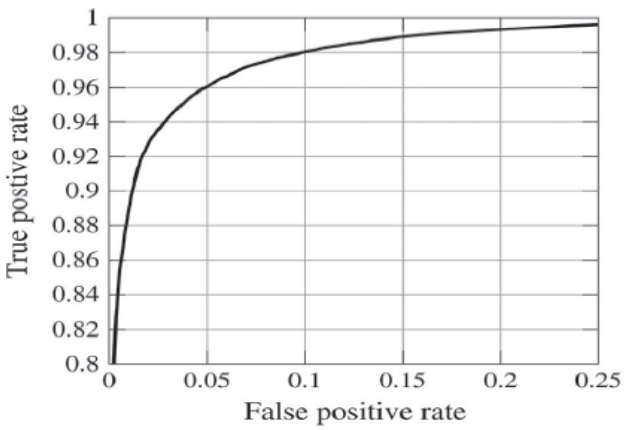
Among the tested classifiers, SVM yields the worst accuracy (86.31%) while being efficient in identifying legitimate URLs (93.1%). Rule-based classifiers have approximately the same performance (around 90%) with disproportionate true positives and true negatives. The best performers are tree-based classifiers, with Random Forest, correctly classifying 95.22% of URLs, being the best. In addition Table VIII shows that these top performer classifiers give accurate results having a standard deviation around 0.25% over 100 runs.

The detailed classification metrics for the Random Forest algorithm with a 0.76 discrimination threshold are given in Table V. The two first columns represent the rate of wellclassified and misclassified instances for each class: TP_{rate}, FP_{rate}, FN_{rate} and TN_{rate}. The Precision corresponds to the ratio of phishing URLs classified as phishing with respect to the total URLs classified as phishing such as

$$Precision = \frac{TP}{TP+FP}$$

The F − measure is defined with Recall = TP_{rate}.

$$F - measure = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}.$$



ROC curve for Random Forest classification.

Fig. 4

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 06, June - 2017

Table V

DETAILED CLASSIFICATION RESULTS FOR RANDOM FOREST (threshold = 0.76)

Class	Class. as phish.	Class. as leg.	*Precision*	*F-measure*	*Accuracy*
Phishing	91.27% (TP)	8.73% (FN)	98.44%	94.72%	94.91%
Legitimate	1.44% (FP)	98.56% (TN)			

To show the relevancy of intra relatedness features, we classified with different features the set of URLs. Using only features 1–9 for classification yields an accuracy of 93.48% whereas using reputation based features 10–12 yields 83.97%.

While having the best Information Gain as shown in Table VI, feature 12 (ranking) and other state of the art features are not sufficient to distinguish between phishing and non-phishing URLs alone. However we show that the proposed feature set yields good results in doing this task. In addition, combining the new proposed features with reputation based features can lead to an improvement in the classification accuracy making this work complementary to the state of the art.

Even though this technique, which gives a *hard decision* for URL class, is proved efficient, correctly classifying 94.91% of URLs with only 1.44% of legitimate URLs classified as malicious, we further leverage machine learning to build a reputation system.

Table VI

INFORMATION GAIN VALUES OF THE 12 FEATURES

Feature	IG	Feature	IG
$ranking$	0.388	mld_{res}	0.178
J_{ARrd}	0.286	J_{RA}	0.133
$card_{rem}$	0.273	J_{RR}	0.125
$ratio_{Arem}$	0.217	J_{AA}	0.123
J_{ARrem}	0.208	J_{AR}	0.12
$ratio_{Rrem}$	0.208	$mld.ps_{res}$	0.07

VI. CONCLUSION AND FUTURE WORK

This paper introduces PhishStorm, an efficient phishing URL detection system relying on URL lexical analysis. The approach is based on the intra-URL relatedness. This relatedness reflects the relationship among the words blended into a URL and particularly into the part of the URL that can be freely defined and the registered domain. We leverage search engine query data in order to extract 12 features from a URL characterizing its intra relatedness and its popularity. The proposed features were used in supervised classification on a ground truth dataset of 96 018 phishing and legitimate URLs. This experiment yielded a classification accuracy of 94.91% with a low false positive rate of 1.44%. This experiment was extended to introduce a URL rating system, PhishStorm, to dynamically compute a *risk score* for URLs. The *risk score* on the testing dataset is able to correctly identify 99.22% of the legitimate and phishing URLs for 83.97% of the dataset. We have extended an initial approach [3] towards real-time analytics by leveraging recent Big Data streaming architectures and patterns based on STORM and Bloom filters.

Future work will consist in releasing components of the tools as an add-on for a Web browser such as Mozilla Firefox. In addition, the technique proposed in [13], which is complementary to that introduced in this paper, will be merged to create a phishing detection system with a larger scope of action. We also plan to release the analytics related part in a larger Big Data security analytics stack, which is under current development in our lab.

VII. ACKNOWLEDGMENTS

This work was supported in part by CyberVault Security Solutions.

REFERENCE

[1] Anti Phising Working Group. Q4 2009 Report. http://antiphishing.org.
[2] Google Safe Browsing API. http://code.google.com/apis/safebrowsing/.
[3] S. Marchal, J. François, R. State, and T. Engel, “PhishScore: Hacking phishers’ minds,” in Proc. 10th Int. CNSM, 2014.
[4] S. Garera, N. Provos, M. Chew, and A. D. Rubin, “A framework for detection and measurement of phishing attacks,” in Proc. ACM Workshop Recurring Malcode, 2007.
[5] T. Segaran and J. Hammerbacher, Beautiful Data: The Stories Behind Elegant Data Solutions. Sebastopol, CA, USA: O’ Reilly Media, 2009.
[6] T. K. Landauer and S. T. Dumais, “A solution to Plato’s problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge,” Psychological Rev., vol. 104, no. 2, pp. 211–240, Apr. 1997.
[7] K. W. Church and P. Hanks, “Word association norms, mutual information, and lexicography,” Comput. Linguistics, vol. 16, no. 1, pp. 22–29, Mar. 1990.
[8] P. Turney, “Mining the web for synonyms: PMI-IR versus LSA on TOEFL,” in Proc. 12th Eur. Conf. Mach. Learn., 2001, pp. 491–502, Springer.
[9] R. L. Cilibrasi and P. M. Vitanyi, “The Google similarity distance,” Trans. Knowl. Data Eng., vol. 19, no. 3, pp. 370–383,Mar. 2007.
[10] P. Kolb, “Disco: A multilingual database of distributionally similar words,” in Proc. KONVENS, 2008, pp. 33–44.
[11] M. Hall et al., “The WEKA data mining software: An update,” ACM SIGKDD Explorations Newslett., vol. 11, no. 1, pp. 10–18, Jun. 2009.
[12] L. Breiman, “Random forests,” Mach. Learn., vol. 45, no. 1, pp. 5–32,Oct. 2001.
[13] S. Marchal, J. François, R. State, and T. Engel, “Proactive discovery of phishing related domain names,” in Research in Attacks, Intrusions, and Defenses. New York, NY, USA: Springer-Verlag, 2012, pp. 190–209.
[14] PhishDef: URL Names Say It All, Anh Le, Athina Markopoulou, Michalis Faloutsos, IEEE 2011.
[15] PhishNet: Predictive Blacklisting to Detect Phishing Attacks, Pawan Prakash, Manish Kumar, Ramana Rao Kompella, Minaxi Gupta, Purdue University, Indiana University, IEEE 2010.
[16] PhishAri: Automatic Realtime Phishing Detection on Twitter, Anupama Aggarwaly, Ashwin Rajadesingan_, Ponnurangam Kumaraguru, Indraprastha Institute of Information Technology, India, Arizona State University, USA
anupamaa@iiitd.ac.in, arajades@asu.edu, pk@iiitd.ac.in, IEEE