Wide Scope and Fast Websites Phishing Detection Using URLs Lexical Features

Ammar Yahya Daeef
School of Computer and Communication Engineering
Universiti Malaysia Perlis (UniMAP)
Middle Technical University, Baghdad, Iraq
Email: ammaryahyadaeef@gmail.com

R. Badlishah Ahmad
School of Computer and Communication Engineering
Universiti Malaysia Perlis (UniMAP)
Email: badli@unimap.edu.my

Yasmin Yacob
School of Computer and Communication Engineering
Universiti Malaysia Perlis (UniMAP)
Email: yasmin.yacob@unimap.edu.my

Ng Yen Phing
School of Computer and Communication Engineering
Universiti Malaysia Perlis (UniMAP)
Email: nyenphing@gmail.com

***Abstract*—Phishing is a considerable problem differs from the other security threats such as intrusions and Malware which are based on the technical security holes of the network systems. The weakness point of any network system is its Users. Phishing attacks are targeting these users depending on the trikes of social engineering. Despite there are several ways to carry out these attacks, unfortunately the current phishing detection techniques cover some attack vectors like email and fake websites. Therefore, building a specific limited scope detection system will not provide complete protection from the wide phishing attack vectors. This paper develops detection system with a wide protection scope using URL features only which is relying on the fact that users directly deal with URLs to surf the internet and provides a good approach to detect malicious URLs as proved by previous studies. Additionally, Anti-phishing solutions can be positioned at different levels of attack flow where most researchers are focusing on client side solutions which turn to add more processing overhead at the client side and lead to losing the trust and satisfaction of the users. Nowadays many organizations make centralized protection of spam filtering. This paper proposes a system which can be integrated into such process in order to increase the detection performance in a real time. The simulation results of the proposed system showed a phishing URLs detection accuracy with 93% and provided online process of a single URL in average time of 0.12 second.**

***Index Terms*—Phishing, Classifier, Machine learning, N-gram, Lexical features.**

I. Introduction

The internet today has become a vital communication tool, with many people using it to create an online environment to cope with the offline business management, or even to set up a wholly online business functionalities. In spite of the internet offers many benefits. There is also a negative side which users must be aware of. One of the risks is that when people are in the online environment, they may be vulnerable to online fraud by phishing. Phishing is an act whereby users are lured by an attacker to visit fake or malicious websites through using the attack channel in order to obtain the victim confidential information like passwords, credit card numbers, username, etc. [1]. Recently, Phishing attacks comprise over a half of all internet fraud cases that affecting the ordinary users [2]. Indeed, Google issues up to ten million warnings every day to the users who visiting a known websites for their phishing scams, and that list of dangerous sites is growing by around 10,000 every day. Different attack vectors are used to launch phishing attack such as search engines, fake websites, advisement, email, instant message, or phone call [3]. However, the simple and effective technique which is used widely by the phishers is the URL obfuscation [4]. URL obfuscation lures users by misleading them to forge websites via a URL or a genuine website familiar to the victim [5].

As the main data entry points are usually a masqueraded URL (or link), this fact gives a big motivation to propose a wide scope solution to detect such fake URLs in this work. Researchers and security experts have been proposed many techniques [6], [7], [8], [9], [10], [11], [12], [13], [14], [15] to detect phishing attacks. Implementation of a blacklist technique is widely used in search engines and browsers toolbars but it has many limitations such as miss a fresh phishing website and the slowly updating mechanism [16]. Furthermore, automated heuristics approaches do not have such bottleneck due to delay of updating since these methods detect phishing websites on the fly. Generally, heuristic techniques are used to detect phishing URLs, phishing emails, and phishing websites. Phishing websites and phishing emails are considered to be a specific to a special kinds of websites and emails, besides the lack discussion of the delayed producing by these methods. Moreover, phishing website detection relies on upon the downloading and intercepting of the full contents which in turn can provide high detection accuracies but it could present more security issues to the users. The most promising technique is the URL analysis due to less limitation in comparison with other methods, especially the technique which depends only on the lexical analysis because of the lexical features are extracted directly from the URLs without the needing of

978-1-5090-2160-4/16/$31.00 ©2016 IEEE

any external servers and in turn will not introduce any delay. Although prior researches had used URL lexical features, they lack the potential of using only lexical approach and test the robustness of such method using phishing URLs from different sources. This potential can provide wide scope and fast phishing detection system.

II. Related works

A large number of techniques are proposed for the purpose of confronting the threat of phishing attacks. Meanwhile, most of these techniques are depending on phishing features extraction. Some methods execute the page in order to extract the features and due to the time and resource consuming of this method. It is not suitable to support the desirable speed users which are normally looking for. As a result, detecting such malicious websites quickly with high accuracies is necessitated. In a view of achieving this requirement, Garera [17] has been proposed the extraction of the features from URLs only. Phishing URLs anatomy has been examined by McGrath and Gupta [18]. The results showed that phishing URLs normally presents different alphabet distributions and contained a target brand name. Additionally, short domain names and long URLs can be used as a strong indication of phishing. In this context, several methodologies are proposed relying on URLs lexical features only [19], [15], [20]. URL textual properties are so called lexical features such as host length, URL length, token length and dot numbers etc. A major benefit of this method can be summarized by the features extraction does not take a long time without the necessity of the websites downloading which in turn will prevent the threats and latency caused by website loading. Usually, extracted lexical features are used to train different machine learning techniques in view of implementing the phishing detection models.

Recent attempts of representing lexical features with a bag of words method have been presented by Kan and Thi [21]. The lexical features with a machine learning classifiers provide 95% accuracy. Furthermore, achieving such high accuracy is confirmed in [22]. As a result of these works, employing machine learning with URLs lexical features can lead to a high accuracy and lightweight phishing detection systems.

Some techniques use a real time analysis of URLs lexical which can be found in [23]. The authors present PhishStorm identified by a central automated classifier located before the email server. In this system, twelve features have been extracted through using the search engines followed by the machine learning algorithm in order to identify the phishing URLs. Despite, good accuracy of 94.91% with a low rate of 1.44% false positive, PhishStorm is considered to be a time consumed system due to the bottleneck of its search engines.

Lexical features can be used with supporting of other URLs features such as host information. Thomas [24] presents a such system with accuracy of 91% and processing time of 5.54 seconds per URL. The long processing time is a result of using host information features and in turn does not satisfy Internet users making this method not suitable for the online environment.

According to the related works, the features extracted from URLs and machine learning algorithms have proved its worth by providing high accuracy and the dependence on a few information as well. Moreover, as URLs have been used in several attack vectors, this method can cover a wide scope of the phishing attack. Despite all these advantages, lightweight real time phishing detection still subject to more research.

III. methodology

The methodology consists of three steps which are discussed in details as follows:

A. Datasets

In this paper, phishing datasets have been collected from the Phishtank and OpenPhish. We collected 46,5461 URLs from Phishtank and proposed a new methodology to divide these URLs into three datasets based on the listing year . In order to mimic the real life situation, 4647 URLs have been collected from OpenPhish that being launched recently. Keeping in mind that this dataset has been used for the first time in a pure phishing detection.

To cover the diversity of benign websites, we are randomly collected 10,275 URLs from dmoz.org and 10,275 URLs from webcrawler.com.

Table I shows the Phishtank dataset division and Table II depicts the paired phishing and legitimate datasets with the name of each generated dataset.

TABLE I
PHISHTANK DATASET DIVISION METHODOLOGY

Listing Year	Number of URLs	Name
2013	576	D2013
2014	25733	D2014
2015	23331	D2015

TABLE II
PHISHING AND LEGITIMATE DATASETS MERGING

Phish Dataset	Legitimate Dataset	Merged Dataset
D2013	dmoz (10275)	D13DM (10851)
D2013	Webcrawler (10275)	D13WC (10851)
D2014	dmoz (10275)	D14DM (36008)
D2014	Webcrawler (10275)	D14WC (36008)
D2015	dmoz (10275)	D15DM (33606)
D2015	Webcrawler (10275)	D15WC (33606)
OpenPhish	dmoz (10275)	ODM(14922)
OpenPhish	Webcrawler (10275)	OWC(14922)

B. Datasets Preprocessing and Phishing Features

The collected datasets are pre-processed in order to convert them into a suitable format for more processing. Phishtank and Openphish datasets include a number of basic columns such as phishing URL, target brand name, IP and country code etc. Regarding the processed dataset, all the unnecessary columns have been removed while the rest appropriate dataset

is identified by the label class 1 for phishing URLs and 0 for legitimate URLs.

In accordance to the relevant studies which have been presented in [19] and [25], each URL is separated into three parts a hostname, path and query. Therefore, each part is constructed into tokens where dots, slashes and question marks are used to separate the hostname tokens, path tokens and the query tokens respectively. The tokens of each part are further separated using the same delimiters which has been implemented in [26] which are '/','?', '.', '=','' and '_'.

Instead of a bag of word method, this paper proposes building URLs language model using all datasets (phishing and legitimate). The main function of any language model or LMs is assigning probabilities to a set of strings [27]. In this paper, we build a simplest and most effective model specified by N-gram that assigns low probabilities for uncommon strings and the reverse for common strings. An N-gram is a sequence of N strings: a 1-gram (or unigram) is one string like "a", a two string sequence like "aa" is 2-gram (or bigram) and the same for 3-gram (or trigram) and 4-gram (fourgram).

The N-gram is one of the best tools to build the language model as stated in [27] and it is difficult to beat [28]. In spite of building URLs language model using N-gram method is done in [21] and used for malicious websites detection in [25] this is the first time of a such method to be employed for detecting the phishing URLs only, in another word for pure phishing URLs detection only.

The strength of N-gram model approximates the history by using the last few strings only. This assumption is called a Markov method or model. In this way, looks only one past string is bigram, looks two past stings are trigram and thus to N-gram in which looks N-1 past strings. To explain the mathematical representation of N-gram building process, we will define the N-gram map of the phishing and legitimate datasets as a sequence of N sized strings e.g. $(w_1.....w_k)$ where k is the total number of N sized grams exists in the datasets and N takes the value from 1 to 4. We note that as the number of N increased to 5 or 6 there is no significant increase in accuracy and rise up the number of unseen grams in testing mode. In order to estimate these N-gram probabilities, Maximum Likelihood Estimation (MLE) is used by getting each gram number of occurrences from the datasets. In one hand, the unigram probability is computed using the frequency distribution of 1-gram divided by the sum of all unigrams occurrences as shown in (1) . On the other hand, to compute the bigram probability, we calculate the bigram count and divide it by the count of first half (the unigram) of that bigram. The same context is used for the trigram and fourgram. Equation (2) depicts the general formula of 2, 3, 4-gram probability computations.

$$P(w) = \frac{C(w)}{\sum_{i=1}^{k} C(w)} \tag{1}$$

Where k is the number of all unigram exists in datasets.

$$P(w_k|w_{k-N+1}^{k-1}) = \frac{c(w_{k-N+1}^{k-1} w_k)}{c(w_{k-N+1}^{k-1})} \tag{2}$$

In (2), N represents the grams size and k is the number of looks past strings.

C. Evaluation Metrics

The same performance metrics of our previous works [12], [29] have been implemented in this paper and that could be listed as follows:

- False Positive Rate (FPR): defined as the ratio of legitimate class that incorrectly classified as a phishing class to the total number of legitimate class instances.

$$FPR = \frac{N_{L\to P}}{N_{L\to L} + N_{L\to P}} \tag{3}$$

- False Negative Rate (FNR): defined as the ratio of phish class that incorrectly classified as a legitimate class to the total number of phish class instances.

$$FNR = \frac{N_{P\to L}}{N_{P\to P} + N_{P\to L}} \tag{4}$$

- True Positive Rate (TPR): defined as the frequency of correctly classified phishing instances which are computed by dividing the number of instances identified as phishing by the total number of phishing instances.

$$TPR = \frac{N_{P\to P}}{N_{P\to P} + N_{P\to L}} \tag{5}$$

- True Negative Rate (TNR): defined as the frequency of correctly classified legitimate instances which are computed by dividing the number of instances identified as legitimate by the total number of legitimate instances.

$$TNR = \frac{N_{L\to L}}{N_{L\to L} + N_{L\to P}} \tag{6}$$

- Accuracy: defined as the percentage of the correct classification over all attempts of classification.

$$Accuracy = \frac{N_{L\to L} + N_{P\to P}}{N_{L\to L} + N_{L\to P} + N_{P\to L} + N_{P\to P}} \tag{7}$$

IV. Experimental Results

The N-gram model is separately constructed into host, path and query using phishing and legitimate datasets. We developed Java codes to calculate the grams probabilities and stored as a MySQL database. In addition, Java code has been developed in order to extract the proposed N-gram features which are outlined in Table III through splitting URLs into host, path and query then extract all grams of each part. After that these grams are read in sequence and a MySQL is looked up to fetch each gram probability. If any gram probability is not matched then zero probability is assigned. Finally, in order to generate each feature value, grams probabilities of each part are added separately. In this paper, three widely used classifiers has been carried out in the field of phishing detection namely J48, Support Vector Machine (SVM), and Logistic Regression (LR). These classifiers are used with the default parameters as

implemented in Waikato Environment for Knowledge Analysis (WEKA) [30]. Additionally, 10-fold cross-validation is used to evaluate the classifiers performance and core-i7 2.57 GHz processor with 8 GB memory machine is utilized to run all the experiments.

TABLE III
FEATURES DESCRIPTION

Feature Name	Description
UniGram_Host	The probability summation of the host unigrams
BiGram_Host	The probability summation of the host bigrams
TriGram_Host	The probability summation of the host trigrams
FourGram_Host	The probability summation of the host fourgrams
UniGram_Path	The probability summation of the path unigrams
BiGram_Path	The probability summation of the path bigrams
TriGram_Path	The probability summation of the path trigrams
FourGram_Path	The probability summation of the path fourgrams
UniGram_Query	The probability summation of the query unigrams
BiGram_Query	The probability summation of the query bigrams
TriGram_Query	The probability summation of the query trigrams
FourGram_Query	The probability summation of the query fourgrams

The classifiers performance based on D13DM and D13WC is given in Table IV. From these results, accuracy demonstrates insignificant differences in a range between 95.3% to 96.9%. On one hand, the best classifier for detecting phishing samples is J48 which provides TPR in range of 48.3% to 50.7%. On the other hand, all classifiers perfectly detected the legitimate samples especially SVM which provides 100% TNR. However, we observed that classifiers performance based on these two datasets is the worst in TPR as compared with the other datasets results. This is due to insufficient number of phishing samples which are leading to a high FNR.

According to the rest datasets, the results in Table V, Table VI and Table VII show that the range of classifiers accuracies is 88% to 93% . Meanwhile, the rapid increasing of TPR reaches to around 97%. Generally, J48 provides the highest accuracies and the highest TPR followed by SVM and LR respectively.

TABLE IV
CLASSIFIERS RESULTS ON D13DM AND D13WC

Classifier	TPR	FPR	TNR	FNR	Accuracy	Error
D13DM						
J48	50.70%	0.40%	99.60%	49.30%	96.96%	3.03%
SVM	26%	0%	100%	74%	96.06%	3.93%
LR	40.10%	0.60%	99.40%	59.90%	96.26%	3.73%
D13WC						
J48	48.30%	0.40%	99.60%	51.70%	96.88%	3.11%
SVM	25.50%	0%	100%	74.50%	96%	3.99%
LR	25%	0.70%	99.30%	75%	95.35%	4.64%

In order to analyze clearly the overall error rates, bars in Fig.1 depict the error rates of all classifiers based on each dataset. The lowest error rate of 3% to 7.7% is achieved by J48 which is considered to be the best classifier followed by SVM and LR classifiers. J48 classifier is chosen for the next experiment because of the best overall results of classification.

A. Processing time and tradeoff between FNR and FPR

One of the primary goals of this work is to protect users from the phishing websites in a real time and based on a high accuracy of the classifier detection with a short processing time per URL as well. As stated above, J48 has the highest accuracies and provides the best trade off in FPR and FNR. Experimentally, the average time required to extract the features and convert them into classifier format in order to obtain the result from J48 takes around 0.12 second for a single URL. This superior processing time of this work is returned to the fact that the lexical features have been constructed only without using any other type of features which is required an external servers such as host information.

Using machine learning models allow to tune the false negatives and false positives by regulating the training data in the ratio of phishing to legitimate samples. False negatives can give the wrong sense of protection which is in the end leads to deliver the users data to the phishing websites. In case of false positive, users should be very careful when loading the website and personally ensure the safety and security of the website before giving any touchy information. A practical example of this process is to integrate the classifier with a firewall. The network administrator may want to make the level of false negatives as lowest as possible to reduce the viruses in the network, or may wish to make the false positives at the lowest level in order to minimize the notifications level of the blocked websites.

To show the effect of such trade off in FPR and FNR experimentally, D14WC as it contains sufficient number of phishing and non-phishing URLs is split into three different

TABLE V
CLASSIFIERS RESULTS ON D14DM AND D14WC

Classifier	TPR	FPR	TNR	FNR	Accuracy	Error
D14DM						
J48	97.30%	16.20%	99.70%	83.80%	93.45%	6.54%
SVM	95.80%	16.90%	83.10%	4.20%	92.18%	7.81%
LR	94.80%	18%	82%	5.20%	91.12%	8.87%
D14WC						
J48	96.80%	16.50%	83.50%	3.20%	93.01%	6.98%
SVM	95.80%	18.70%	81.30%	4.20%	91.65%	8.34%
LR	93.90%	23.30%	76.70%	6.10%	88.96%	11.03%

TABLE VI
CLASSIFIERS RESULTS ON D15DM AND D15WC

Classifier	TPR	FPR	TNR	FNR	Accuracy	Error
D15DM						
J48	96.60%	12.30%	87.70%	3.40%	93.91%	6.08%
SVM	95.40%	14.20%	85.80%	4.60%	92.43%	7.56%
LR	93.40%	15.20%	84.80%	6.60%	90.80%	9.19%
D15WC						
J48	96.30%	13.50%	86.50%	3.70%	93.32%	6.67%
SVM	95.40%	16.80%	83.20%	4.60%	91.69%	8.30%
LR	92.90%	21.30%	78.70%	7.10%	88.51%	11.48%

TABLE VII
CLASSIFIERS RESULTS ON ODM AND OWC

Classifier	TPR	FPR	TNR	FNR	Accuracy	Error
ODM						
J48	90.50%	5.40%	94.60%	9.50%	93.33%	6.66%
SVM	88.90%	6%	94%	11.10%	92.39%	7.60%
LR	80%	4%	96%	20%	91.01%	8.98%
OWC						
J48	87.80%	5.70%	94.30%	12.20%	92.25%	7.74%
SVM	88%	7.90%	92.10%	12%	90.78%	9.21%
LR	77.50%	4.40%	95.60%	22.50%	89.98%	10.01%

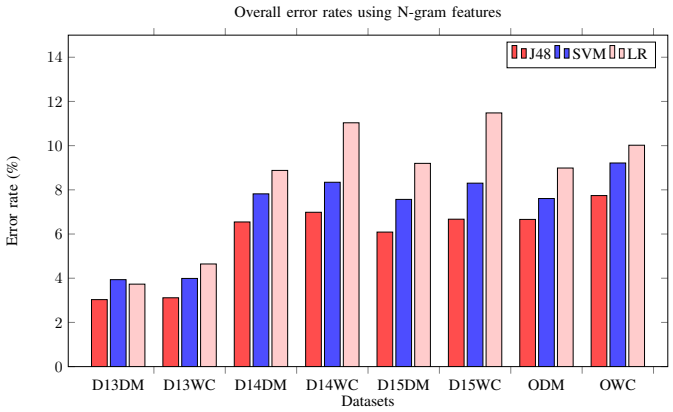

Fig. 1. Classifiers error rates comparison on all datasets

groups and then each group is giving a different ratio of phishing websites to non-phishing websites. In addition, 60% cross validation is used for training and the rest ratio is used for testing to make more samples in testing phase to better evaluate the trade off between FPR and FNR. Table VIII presents each group with the corresponding number of samples in each one.

Table IX illustrates the J48 trade off results where as phishing samples rise up, it leads to decrease the rate of false negative and false positive rates with respect to increasing the number of legitimate samples . Based on the achieved results, using balanced and sufficient training dataset is crucial to get better classification performance.

TABLE VIII
PHISHING TO LEGITIMATE RATIO IN EACH SET

Group	Phishing	legitimate	Training (60%)	Testing (40%)
Ratio 50:50	6850	6850	8220	5480
Ratio 75:25	10275	3425	8220	5480
Ratio 25:75	3425	10275	8220	5480

TABLE IX
TUNING CLASSIFICATION RESULTS

Ratio	Accuracy	FPR	FNR
50:50	89.92%	12.30%	7.80%
75:25	93.90%	17.70%	2.20%
25:75	92.24%	5.40%	14.70%

V. CONCLUSION

This paper presents a wide scope and fast phishing detection system. N-gram models are constructed using both phishing and legitimate URLs including the features which have been extracted from these models. In this paper, three classifiers are implemented through using WEKA and the best results are achieved by J48 classifier with accuracy of 93%. The average time required to identify a single URL is 0.12 seconds. Classifiers should be trained using balanced datasets in order to get higher performance. Despite of the promising performance and processing time of this technique, error rates still high and require additional efforts in order to reduce these error. The new direction of future work related to this interesting topic can be accomplished through combining N-gram strategy with the highest rank bag of word features that exists in literature. Therefore, this hybrid technique will lead to increasing the performance rapidly with a little rise up in the processing time.

REFERENCES

[1] A. Solanki and S. Dogiwal, "Implementation of an anti-phishing technique for secure login using usb (iatslu)," in *Computational Intelligence in Data Mining-Volume 1*. Springer, 2015, pp. 221–231.
[2] H. Orman, "The compleat story of phish," *IEEE Internet Computing*, no. 1, pp. 87–91, 2013.
[3] G. Liu, B. Qiu, and L. Wenyin, "Automatic detection of phishing target from phishing webpage," in *Pattern Recognition (ICPR), 2010 20th International Conference on*. IEEE, 2010, pp. 4153–4156.
[4] R. S. Rao and S. T. Ali, "Phishshield: A desktop application to detect phishing webpages through heuristic approach," *Procedia Computer Science*, vol. 54, pp. 147–156, 2015.
[5] M. Cova, C. Kruegel, and G. Vigna, "There is no free phish: An analysis of" free" and live phishing kits." *WOOT*, vol. 8, pp. 1–8, 2008.
[6] B. Braun, M. Johns, J. Koestler, and J. Posegga, "Phishsafe: leveraging modern javascript api's for transparent and robust protection," in *Proceedings of the 4th ACM conference on Data and application security and privacy*. ACM, 2014, pp. 61–72.
[7] Y. Cao, W. Han, and Y. Le, "Anti-phishing based on automated individual white-list," in *Proceedings of the 4th ACM workshop on Digital identity management*. ACM, 2008, pp. 51–60.
[8] S. Sheng, B. Wardman, G. Warner, L. F. Cranor, J. Hong, and C. Zhang, "An empirical analysis of phishing blacklists," 2009.
[9] Google safe browsing. [Online]. Available: http://code.google.com/p/google-safe-browsing/
[10] S. Gastellier-Prevost, G. G. Granadillo, and M. Laurent, "Decisive heuristics to differentiate legitimate from phishing sites," in *Network and Information Systems Security (SAR-SSI), 2011 Conference on*. IEEE, 2011, pp. 1–9.
[11] I. Fette, N. Sadeh, and A. Tomasic, "Learning to detect phishing emails," in *Proceedings of the 16th international conference on World Wide Web*. ACM, 2007, pp. 649–656.
[12] A. Y. DAEEF, R. B. AHMAD, Y. YACOB, N. YAAKOB, M. WARIP, and M. N. BIN, "Phishing email classifiers evaluation: Email body and header approach." *Journal of Theoretical & Applied Information Technology*, vol. 80, no. 2, 2015.
[13] K.-T. Chen, J.-Y. Chen, C.-R. Huang, and J.-Y. Chen, "Fighting phishing with discriminative keypoint features," *Internet Computing, IEEE*, vol. 13, no. 3, pp. 56–63, 2009.
[14] Y. Zhang, J. I. Hong, and L. F. Cranor, "Cantina: a content-based approach to detecting phishing web sites," in *Proceedings of the 16th international conference on World Wide Web*. ACM, 2007, pp. 639–648.
[15] M. Khonji, Y. Iraqi, and A. Jones, "Lexical url analysis for discriminating phishing and legitimate websites," in *Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference*. ACM, 2011, pp. 109–115.
[16] S. Sinha, M. Bailey, and F. Jahanian, "Shades of grey: On the effectiveness of reputation-based blacklists," in *Malicious and Unwanted Software, 2008. MALWARE 2008. 3rd International Conference on*. IEEE, 2008, pp. 57–64.
[17] S. Garera, N. Provos, M. Chew, and A. D. Rubin, "A framework for detection and measurement of phishing attacks," in *Proceedings of the 2007 ACM workshop on Recurring malcode*. ACM, 2007, pp. 1–8.
[18] D. K. McGrath and M. Gupta, "Behind phishing: An examination of phisher modi operandi." *LEET*, vol. 8, p. 4, 2008.

[19] A. Blum, B. Wardman, T. Solorio, and G. Warner, "Lexical feature based phishing url detection using online learning," in *Proceedings of the 3rd ACM Workshop on Artificial Intelligence and Security*. ACM, 2010, pp. 54–60.

[20] R. B. Basnet, A. H. Sung, and Q. Liu, "Learning to detect phishing urls," *IJRET: International Journal of Research in Engineering and Technology*, vol. 3, no. 6, pp. 11–24, 2014.

[21] M.-Y. Kan and H. O. N. Thi, "Fast webpage classification using url features," in *Proceedings of the 14th ACM international conference on Information and knowledge management*. ACM, 2005, pp. 325–326.

[22] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Learning to detect malicious urls," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 30, 2011.

[23] S. Marchal, J. François, R. State, and T. Engel, "Phishstorm: Detecting phishing with streaming analytics," *Network and Service Management, IEEE Transactions on*, vol. 11, no. 4, pp. 458–471, 2014.

[24] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song, "Design and evaluation of a real-time url spam filtering service," in *Security and Privacy (SP), 2011 IEEE Symposium on*. IEEE, 2011, pp. 447–462.

[25] M. Darling, G. Heileman, G. Gressel, A. Ashok, and P. Poornachandran, "A lexical approach for classifying malicious urls," in *High Performance Computing & Simulation (HPCS), 2015 International Conference on*. IEEE, 2015, pp. 195–202.

[26] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: learning to detect malicious web sites from suspicious urls," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 1245–1254.

[27] D. Jurafsky and J. H. Martin, *Speech and language processing*. Pearson, 2014.

[28] F. Jelinek, "Up from trigrams." Eurospeech, 1991.

[29] A. Y. Daeef, R. B. Ahmad, Y. Yacob, N. Yaakob, and K. N. F. K. Azir, "Multi stage phishing email classification," *Journal of Theoretical and Applied Information Technology*, vol. 83, no. 2, p. 206, 2016.

[30] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.