International Journal of Control Theory and Applications

ISSN : 0974–5572

Volume 9 • Number 42 • 2016

Websites Phishing Detection using URLs N-Grams as a Discriminating Features

Ammar Yahya Daeef [a] R. Badlishah Ahmad[b] and Yasmin Yacob[c]

[a]*School of Computer and Communication Engineering, Universiti Malaysia Perlis (UniMAP), Perlis, Malaysia*

E-mail: ammaryahyadaeef@gmail.com

[b]*Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin (UniSZA), Terengganu, Malaysia*

E-mail: badli@unimap.edu.my, badli@unisza.edu.my

[c]*Middle Technical University, Baghdad, Iraq*

E-mail: yasmin.yacob@unimap.edu.my

Abstract: Phishing is a challenging problem for users and security experts as well. Phishers use social engineering to acquire touchy information from a victim. Therefore, many works have been proposed to develop detection systems to prevent users from surfing phishing websites. Nowadays, users looking for fast Internet surfing so that it is important to have fast phishing detection systems to achieve users satisfied. Phishing classification using only the information exist in Uniform Resource Locator (URL) without visiting web content or using information from the external servers can provides such fast classification. In this context, this paper aims to robustly explore the effectiveness of using URLs N-grams as a discriminating features between phishing and legitimate URLs. The study analyzes URLs collected from different sources and according to this analysis, a statistical classifier is built and the performance is evaluated to measure the technique effectiveness.

Keywords: *Phishing, Lexical features, URLs N-grams, Statistical classifier.*

1. INTRODUCTION

The web has evolved widely in the life of people and since the beginning of Internet in the 1990s, a lot of new security issues and threats appear continuously which constitute a challenge to users and security experts as well. Phishing is a cutting edge threat that has an impact on commercial and banking sectors by means of the Internet which delivers huge misfortunes at the level of clients and organizations [1]. Phishing websites have high similitude to the honest ones trying to trap and bait users to enter these websites. In this sort of attack, phishers normally utilize technical and social designing traps together to begin their attacks. The attacks of social engineering are focusing on users not systems intended to get the data of users which are typically touchy and secret [2].

In spite of the broad field of phishing attack vectors, a typical purpose of numerous vectors is the utilization of link misleading victims to phishing websites. Utilization of obfuscated URL and domain names is widely used in phishing attacks [3]. Anti-Phishing Work Group (APWG) [4], reported that the number of phishing

websites increased by 250% in the period from the last three months of 2015 to the first quarter of 2016. The total number of discovered unique websites in the first quarter of 2016 is 289,371. Also, steadily rose per month was observed from October 2015 to March 2016 ranged from 48,114 to 123,555 respectively [5]. These statistics demonstrate the significance to distinguish URLs and domain names to battle phishing.

Most of the works in the field of phishing detection based on website content analysis or use external data from servers to classify URLs as legitimate or phishing class. This work focuses on feature extraction from URLs lexical itself because it needs less processing requirement compared with content or external features. Also, features extraction from URLs lexical can provide wide scope detection depending on the fact that users use URLs directly to search the Internet.

2. RELATED WORKS

A lot of techniques are proposed to detect phishing attacks, most of them based on extracting phishing features either from the website content or using external information. Extract features from website content is resource and time consuming and expose users to threats by downloading malicious content. Extracting features from external servers (*e.g*. website rank, DNS, Whois ….. etc.) adds more processing time to detect each URL which make such technique not applicable for real time applications.

As an alternative, some methods analysis URLs lexical properties as a discriminating features. Such features are the number of dots in URL, length of tokens and URL length etc. The features extracted by this method are not time consume and prevent downloading malicious code to the user machine. The anatomy of phishing URLs explored by McGrath and Gupta [18]. Their results state that phishing URLs normally contain the brand name of the target and present different distributions of the alphabet. Also, long URL and short domain name provide strong features of phishing. Take in account this, many works are proposed by utilizing only lexical features extracted from URLs [6], [7], [8].

Most of the works [9], [10] use a bag of word method to represent the lexical features for machine learning classifiers. However, representing lexical features using a bag of word produces high dimension vector which in turn increase the processing time to extract and prepare the features vectors and slow down the training and testing of machine learning classifiers. The authors of PhishStorm [11] present URLs lexical analyses in real time. This system is a central classifier placed in front of the email server to detect phishing URLs. PhishStorm uses 12 features extracted by aid of the search engines then these features are fed to machine learning classifier to make the decision. The accuracy achieved by this system is 94.91% combined with a low false positive rate of 1.44%. However, PhishStorm is time consume because of the search engines employed during features extraction process.

The results presented by Khonji [7] analyze the token distribution in both phishing and legitimate URLs. This study confirms that URLs provide additional information than just directing to a resource. Max accuracy achieved from this method is 97%. However, the robustness of this method is not evaluated by training and testing using completely different sources. Finally, some technique uses lexical features combined with different features such whois or DNS information. Such work is found in [12], this system provides 91% accuracy with 5.54 seconds processing time. This high processing time is a result of utilizing external servers to get the host information.

Using complex operations without fully evaluate simpler methods and check the productivity achieved from them is not a good practice. Therefore in this paper, we try to analyze URLs lexical features and construct statistical classifier to classify phishing and legitimate URLs. Additionally, we check the robustness of this method by out of sample test using different datasets for training and testing.

3. METHODOLOGY

We proposed to classify phishing URLs without segment them into tokens as presented in [7]. Alternatively, this work use similar technique proposed by Peng et al. [13] in which N-gram method is employed to classify text

documents. Usually, the value of N can be set to 1, 2, 3, or 4 where N-grams set for any text can be generated by moving N sized window of characters along that text with one character step at a time. In this paper, after the N-grams are extracted from URLs the number of occurrences of each N-gram is counted. Figure. 1 depicts the methodology phases of this technique.

To analyze the distribution of N-grams in both legitimate and phishing URLs, these URLs are treated one after one to calculate the percentage at which N-grams are reused in subsequent URLs. More clearly, the first URL N-grams are not seen before then the next URLs appeared to reuse N-grams already seen in previous URLs. Java script is written to automate the process described above.

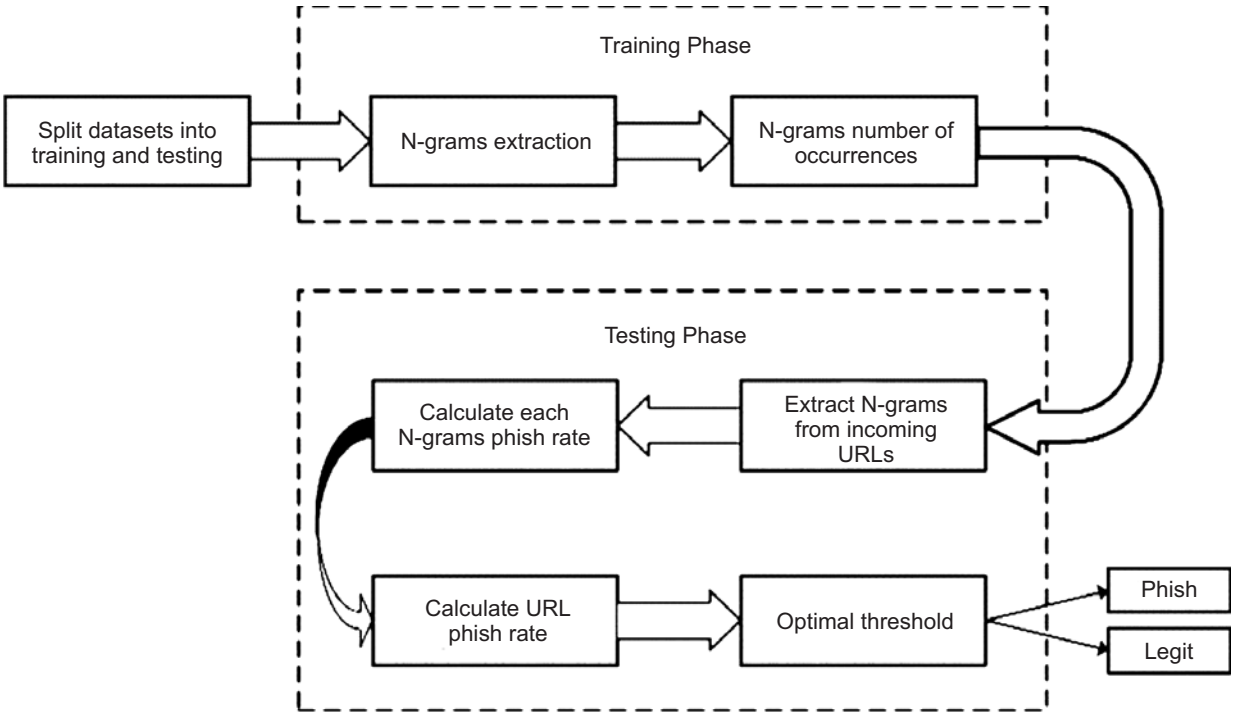


Figure 1: Research methodology phases

3.1. A Statistical Classifier

A binary classifier that constructed to classify each URL in the test phase as either phish or legit class. We use the same classifier proposed in [7], this classifier is built using a supervised learning phase by extracting the N-grams of characters from labeled URLs. After the completion of learning phase, the classifier is fed by unclassified URLs to predict the output class. To predict the output class, each incoming URL is broken into N-grams then try to find each N-gram frequency (the number of occurrences) from each class. After that for each N-gram, the phish rate is calculated using Eq. 1.

$$\text{Ngramphishrate}_i = \frac{\text{Count}_i \rightarrow \text{Phish}}{\text{Count}_i \rightarrow \text{Phish} + \text{Count}_i \rightarrow \text{Light}} \quad (1)$$

After calculating the phish rate of each N-gram in the input URL, The phish rate of that URL is calculated by adding the phish rate of all individual N-gram and divided by the number of N-grams exist in that URL as shown in Eq. 2.

$$\text{URLphishrate} = \frac{\sum_i^N \text{Ngramphishrate}_i}{N} \quad (2)$$

Where N is n is the number of N-grams in URL.

Each URL in the testing phase is classified as a phish if its phish rate value is more than a certain threshold. The classifier is tested using different values of threshold ranged between 0 and 1 with 0.001 increment for each test.

3.2. Datasets

The training data was drawn from four sources: Phishtank.org, Openphish.com, DMOZ.org, and Alexa.com. We collected 20000 phishing URLs from Phishtank and call it Tank dataset. For more closely following the evolving features of phishing URLs and to mimic the real-world scenario, we collected a second batch of 20000 confirmed phishing URLs that were submitted to OpenPhish and call it Open dataset.

To cover the diversity of legitimate websites, our legitimate URLs are gathered from two data sources provided publicly: DMOZ.org and Alexa.com. 20000 randomly chosen non-phishing URLs from DMOZ and we call it DMOZ data set. Also, 20000 randomly chosen non-phishing URLs are collected from Alexa and named this dataset as Alexa dataset. Additionally, in order to cover wider URL structures, we also made a list of URLs related to most commonly phished targets (using statistics of top targets from PhishTank and OpenPhish) to be part of DMOZ and Alexa datasets.

Finally, PhishTank and Openphish datasets are paired with non-phishing URLs from a benign source (either DMOZ or Alexa). We refer to these data sets as the Tank-DMOZ (TD), Tank-Alexa (TA), Open-DMOZ (OD), and Open-Alexa (OA).

3.3. Evaluation Metrics

There are several metrics to measure the quality of binary classification models. We present the most widely used ones that are briefly described in Table 1.

4. ANALYSIS AND EXPERIMENTAL RESULTS

Table 1
Classifier Performance Metrics

Evaluation Metric	*Definition*
False Positive Rate (FPR)	The ratio of legitimate URLs misclassified as phishing class divided by the total number of legitimate instances. $FPR = \frac{N_{L \to P}}{N_{L \to L} + N_{L \to P}}$
False Negative Rate (FNR)	The ratio of phishing URLs misclassified as legitimate class divided by the total number of phishing instances. $FNR = \frac{N_{P \to L}}{N_{P \to P} + N_{P \to L}}$
True Positive Rate (TPR)	The ratio of phishing URLs classified as phishing class divided by the total number of phishing instances. $TPR = \frac{N_{P \to P}}{N_{P \to P} + N_{P \to L}}$

Evaluation Metric	*Definition*
True Negative Rate (TNR)	The ratio of legitimate URLs classified as legitimate class divided by the total number of legitimate instances. $$TNR = \frac{N_{L \to L}}{N_{L \to L} + N_{L \to P}}$$
Accuracy	The ratio of correct classification over all attempts of classification. $$Accuracy = \frac{N_{L \to L} \quad N_{P \to P}}{N_{L \to L} + N_{L \to P} + N_{P \to P} + N_{P \to L}}$$

The analysis based on finding the percentage of reused and unique N-grams exist in each class beside explore the percentage at which these N-grams are overlap. The percentage at which N-grams are reused in each individual dataset is presented first. In this analysis, different N values are tried to see the effects of gram size change on the reused percentage. Figure 2 shows the results of different values of N in range of 1 to 6.

As the value of N increased, the N-grams reused percentage is decreased. For example when N is 2, the reused of 2-grams reached to 92% while decreased to 59% when N is 6. This observation is coupled with overlap results as shown in Table 2 and Table 3. According to the results, the N-gram overlap percentage is decreased as the value of N increased. So that, when N value is increased both percentage of reused and overlap are decreased. Generally, the more overlap between phishing sources besides the more overlap between legitimate datasets the more chance to get high accuracy in out of sample testing. Additionally, as the overlap percentage between phishing and legitimate sources is decreased, the overall accuracy will be increased. This is because most of the grams are not overlapped and can be better used as a discriminating features.

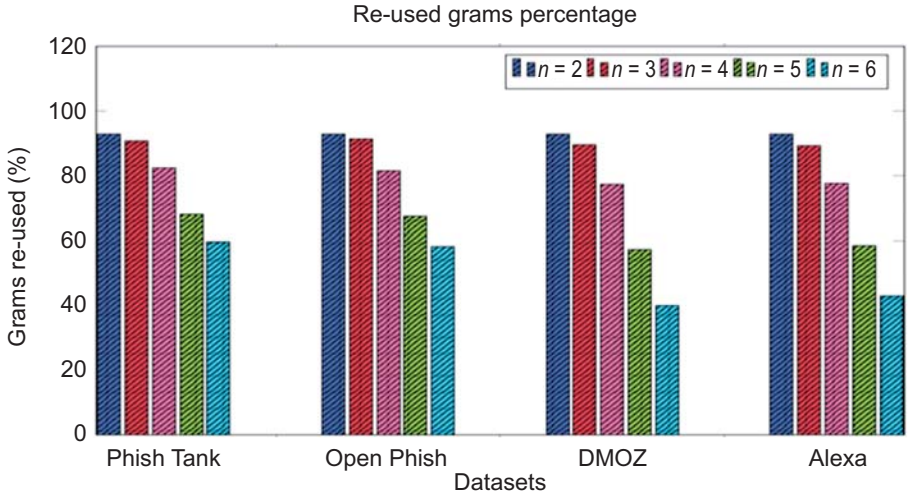


Figure 2: Percentage of reused N-grams in each dataset

Table 2
The percentage of overlapping 2-grams

	Tank	*Open*	*DMOZ*	*Alexa*
Tank	100%	70.85%	42.90 %	43.72 %
Open		100%	42.79%	43.55 %
DMOZ			100%	40.67%
Alexa				100%

Table 3
The percentage of overlapping 6-grams

	Tank	*Open*	*DMOZ*	*Alexa*
Tank	100%	37.85%	1.99 %	1.97 %
Open		100%	1.77%	1.74 %
DMOZ			100%	6.32%
Alexa				100%

The best value of N which provides the better trade off between percentages of reused and overlap when N is set to 4. Figure 3 depicts the percentage of reused 4-grams in each individual dataset.

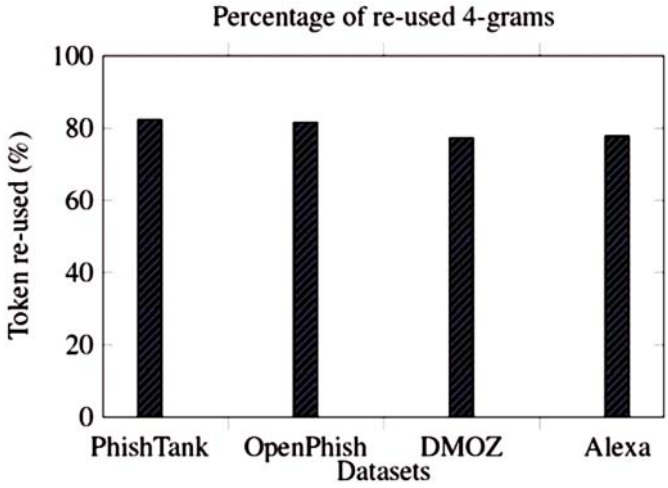
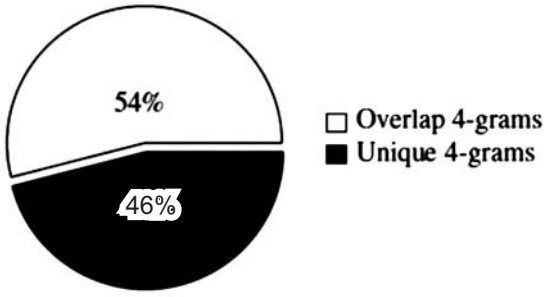
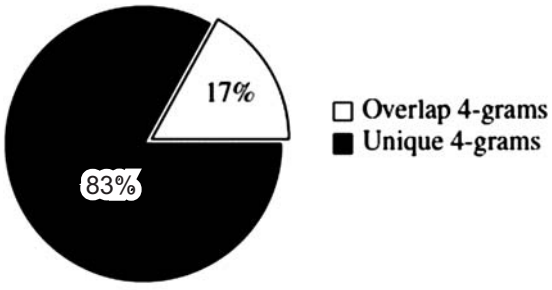


Figure 3: Percentage of reused 4-grams in each dataset



(a) Overlap between phishing sources



(b) Overlap between legitimate sources

Figure 4: Overlap percentage of phishing datasets (*a*) and legitimate datasets (*b*)

PhishTank dataset has reuse 4-grams percentage reached to 82.30% while the percentage in OpenPhish is 81.41%. 4-grams reuse in legitimate datasets shows less percentage with 77.35% and 77.75% for DMOZ and Alexa respectively. It is obvious that the percentage of reused phishing URL 4-grams is higher than legitimate 4-grams which in turn give evidence that the dictionary of phishing 4-grams is smaller than the dictionary of legitimate 4-grams. Such percentage is logical because phishers target famous brand frequently and mostly they reuse the same tricks to start the attack in contrast to the huge number of legitimate URLs exist nowadays. Although the dictionary of phishing 4-grams is less than legitimate one, 4-grams of legitimate URLs are still predictable as around 77% of the 4-grams are reappeared or reused.

From practical point of view as both classes have limited dictionaries of 4-grams, this can be exploited to build robust classification model using URLs 4-grams. To study the common characteristics of the datasets, the 4-gram overlap between different sources is explored. As shown in Figure. 4, the overlap between phishing sources is 54% which means that even with different sources of phishing URLs, these URLs share big percentage of 4-grams. This is very motivational point to create robust classifiers. On the other hand, low 4-grams overlap percentage is observed in legitimate datasets which reached to 17%. This is expected because of the wide variety exist in legitimate URLs.

As well as the analysis includes the overlap percentage of 4-grams between each phishing source and legitimate sources as depicted in Figure 5. In average, the percentage of tokens overlapping in relation to legitimate and phishing sources around 9%. As a result, the biggest percentage of 4-grams are not overlapped between phishing and legitimate sources. This observation is very important and promising to build a classification model using 4-grams only.

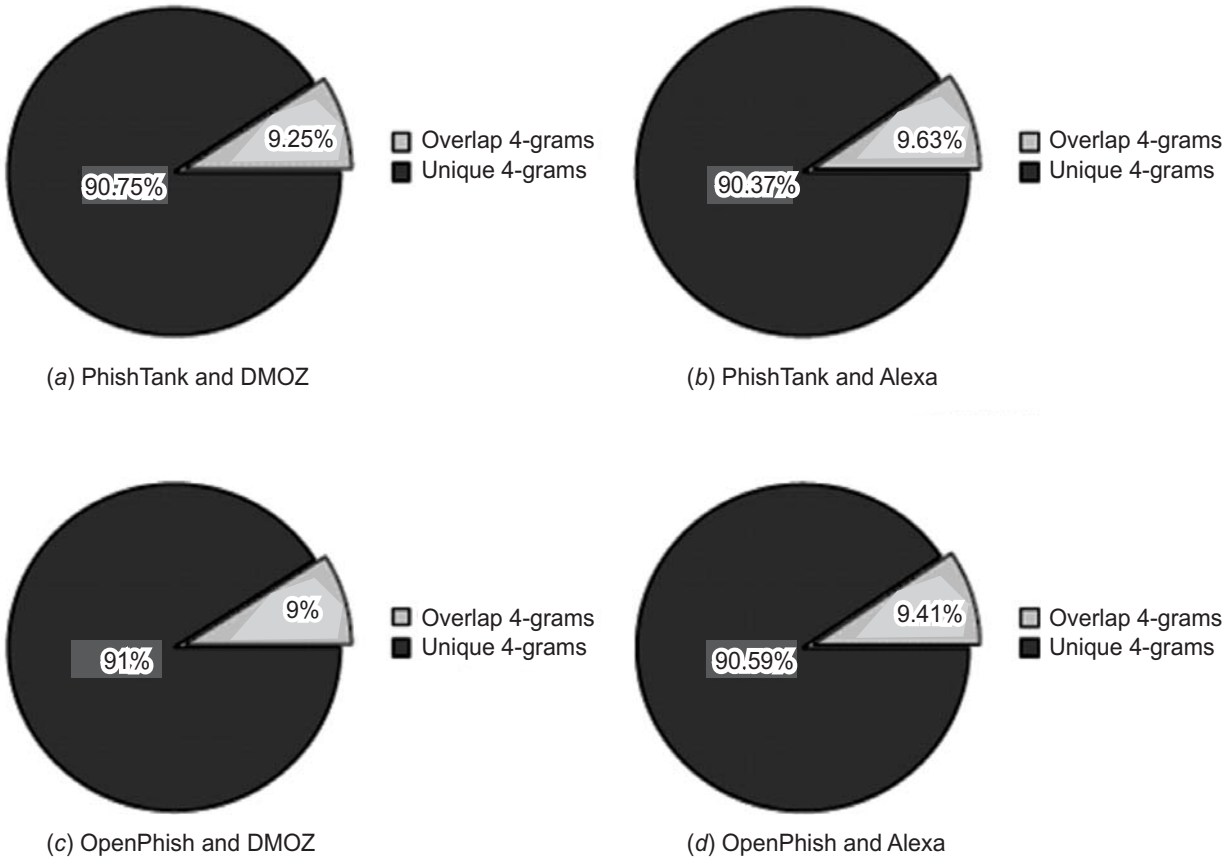
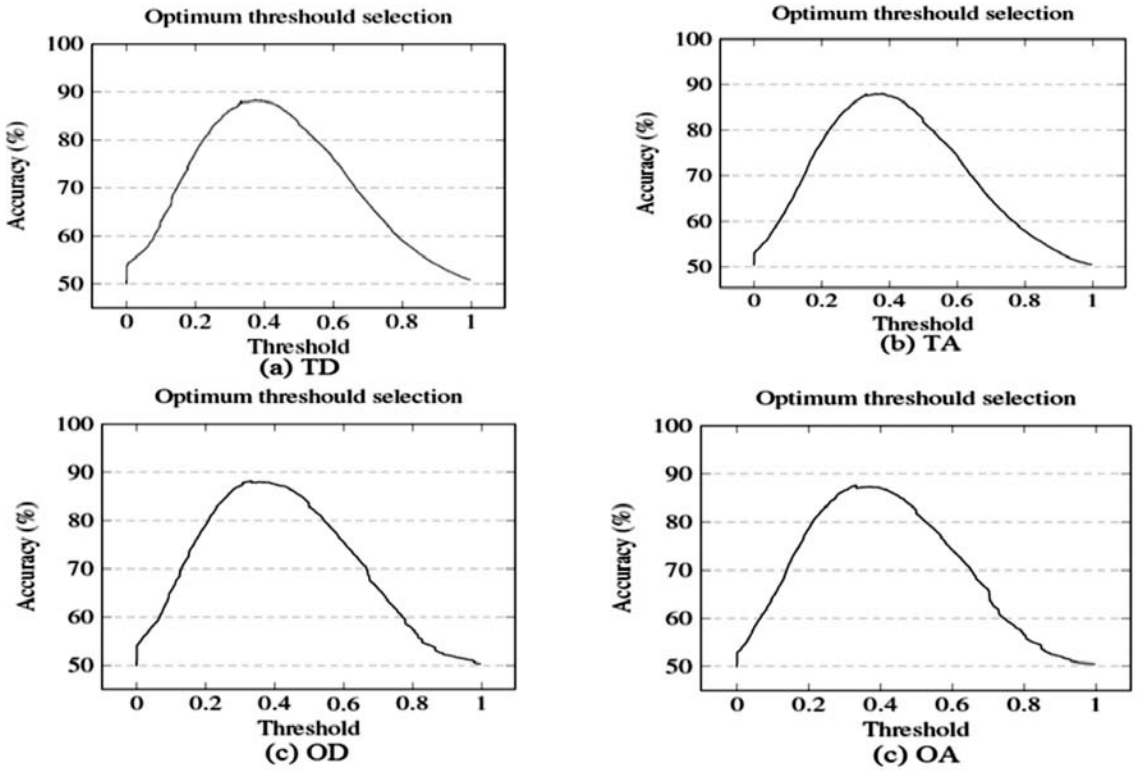


Figure 5: Overlap percentage between phishing and legitimate datasets

The statistical classifier depends on the training dataset to build the classification model then the testing dataset is used to evaluate the generated classifier. As response to that, each dataset is separated into 70% training portion and 30% as testing samples to evaluate the classifier. For each dataset, the optimal threshold is explored by applying thresholds between 0 and 1 with 0.001 increment. The process is repeated for all datasets and optimal threshold is reported according to the maximum accuracy achieved. Figure. 6, depicts how the classifier accuracy behaves as the threshold is changed.



Figur 6: Optimum threshold selection for each dataset

Figure 7 shows the optimal threshold for each dataset and the corresponding accuracies. The accuracies are not differ significantly with average accuracy 87% because the overlap percentage between the phishing datasets and each of the legitimate URLs source is close to each other.

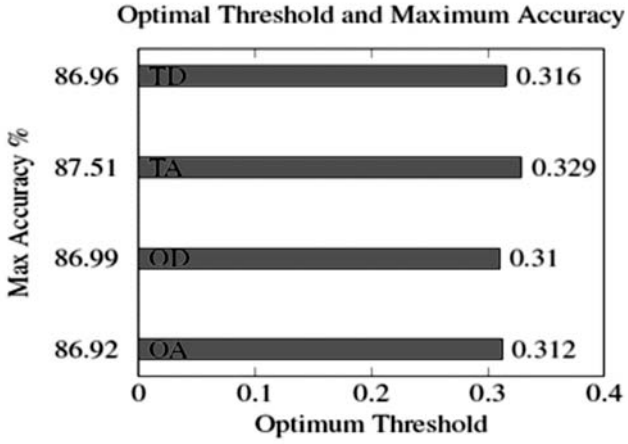


Figure 7: Optimal threshold and maximum accuracy on each Dataset

For a close look at the detailed performance metrics of the statistical classifier on each dataset with optimal threshold, Table IV presents the results of TPR, TNR, FPR, and FNR. The results show that the highest TPR of 88.46% using OD dataset while the highest TNR is 86.83% observed on TA dataset. In general, TPR is higher than TNR on all datasets this is because of the higher percentage at which phishing 4-grams are reused. Also, FPR is higher than FNR which means that more legitimate URLs are miss-classified as phishing class than classify phishing samples as legitimate URLs. Next, the out of sample test based on this method is presented to explore the classifier by training and testing using different datasets.

Table 4
Classifier Performance Metrics

Dataset	*TPR*	*FPR*	*TNR*	*FNR*
TD	88.36%	14.43%	85.57%	11.64%
TA	88.19%	13.17%	86.83%	11.81%
OD	88.46%	14.48%	85.52%	11.54%
OA	87.85%	14.01%	85.99%	12.15%

The statistical classifier results of using mismatched datasets for training and testing are shown in Table 5. Based on the results and as expected because of 4-gram overlap percentage, the error rates are better when training and testing using the same dataset (as shown in the diagonal of Table 5) compared to when mismatched datasets are used for training and testing. When using any combination of phishing and legitimate URLs in the training phase and testing by mismatched phishing URLs only (*e.g.*, TD, OD), the error rates increased because of the high FN. When the phishing URLs are mismatched and because of the nature of the used classifier, more unseen phishing 4-grams will be in the testing phase which makes the classifier miss classifying alot of phishing URLs as a legitimate class. The highest error rate observed in this category is 8.55%. In case of legitimate URLs are mismatched only (*e.g.*, OA and OD) in training and testing, the error rates are rising up mostly contributed by FP with the worst value of 3.63 %. Finally, when both sources are mismatched (*e.g.*, TD and OA) this leads to more unseen 4-grams in testing phase which makes the error rates increased rapidly. The highest error rates are observed in this category with max value reached to 12%.

Table 5
Overall rate of errors using mismatched datasets

		Training			
		TD	*TA*	*OD*	*OA*
Testing	TD	2.42%	3.02%	7.81%	11.17%
	TA	3.63%	2.62%	11.50%	8.36%
	OD	7.42%	10.57%	2.40%	2.92%
	OA	12%	8.55%	3.49%	2.62%

5. CONCLUSION

This study analyses N-gram distributions in both phishing and legitimate URLs collected from different sources. The results show that the dictionary of phishing 4-gram is smaller than the dictionary of legitimate 4-gram. Generally, 4-grams overlap between phishing and legitimate URLs is small. But, the overlap rate between different phishing sources is more than compared with legitimate overlap percentage. However, this technique can be effective if the training and testing using the same dataset but in case of out of sample test the error rates increased. We believe combine this method with high rank lexical features can be the next research step to improve the overall performance.

REFERENCES

[1] K. Mahmoud, I. Youssef, and J. Andrew "Phishing detection: a literature survey," *IEEE Communications Surveys & Tutorials, vol. 15,* pp. 2091-121, Dec. 2013.

[2] Bozkir, S. Ahmet, and A. Ebru, "Use of HOG descriptors in phishing detection," in *2016 4th International Symposium on Digital Forensic and Security (ISDFS)*. IEEE, 2016.

[3] G. Aaron and R. Rasmussen, "Global phishing survey: Trends and domain name use in 2h2014," Anti-Phishing Working Group, 2014.

[4] (2016) The antiphishing website. [Online]. Available: http://www.antiphishing.org/

[5] G. Aaron and R. Rasmussen, "Phishing activity trends report for the months of January-march 2016," Anti-Phishing Working Group, 2016.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] S. Marchal, J. Franc¸ois, 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.

[12] 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.

[13] Peng, Fuchun, Xiangji Huang, Dale Schuurmans, and Shaojun Wang. "Text classification in Asian languages without word segmentation." In*Proceedings of the sixth international workshop on Information retrieval with Asian languages-Volume 11*, pp. 41-48. Association for Computational Linguistics, 2003.