Lightweight Phishing URLs Detection Using N-gram Features

Ammar Yahya Daeef [#1], R. Badlishah Ahmad[#2], Yasmin Yacob [#3],Mohd. Nazri Bin Mohd [#4]
[#1, 2, 3,4]School of Computer and Communication Engineering,
Universiti Malaysia Perlis (UniMAP), Perlis, Malaysia
[#1]Middle Technical University, Baghdad, Iraq
[1] ammaryahyadaeef@gmail.com
[2] badli@unimap.edu.my
[3] yasmin.yacob@unimap.edu.my
[4]nazriwarip@unimap.edu.my

***Abstract*—Phishing is a kind of attack that belongs to social engineering and this attack seeks to trick people in order to let them reveal their confidential information. Several methods are introduced to detect phishing websites by using different types of features. Unfortunately, these techniques implemented for specific attack vector such as detecting phishing emails which make implementing wide scope detection system crucial demand. URLs analysis proved to be a strong method to detect malicious attacks by previous researches. This technique uses various URL features such as host information, lexical, and other type of features. In this paper, we present wide scope and lightweight phishing detection system using lexical features only. The proposed classifier provides accuracy of 93% with 0.12 second processing time per URL.**

Keyword - Phishing, Classifier, Machine learning, Lexical features.

I. INTRODUCTION

Phishing is a social engineering attack that exploits user' ignorance during system processing [1]. For example, an unsuspecting system users may leak their password to an attacker by clicking on a URL link sent by the attacker asking them updating their password or account even if the system is technically secure against password theft. These acts ultimately threaten overall security of the system.

Attackers use different attacks vector to execute their phishing such as emails, websites, websites advertising, or even through phone calls [2]. Although, the wide scope of attacks vector the common feature among them is the utilization of a fake link to direct victims to phishing websites. This fact motivates us to propose wide scope classifier using URLs features only.

Numerous works by researchers to fight phishing attacks implemented during the last few years and a lot of ideas are presented in this field [3-8]. Blacklist is one of the earliest and dominant method which deployed in Web filtering, browsers toolbars and search engines. In this method, a third party service collects the names of bad websites which labeled by feedback from users, Web crawling and critical analysis of the website content then distributes the list to the subscribers. Blacklist based systems offer incomplete protection against phishing attacks because there is no comprehensive and up-to-date blacklist, but provide the minimum query overhead [9]. Alternatively, some detection systems intercept and download the full contents of website for analyzing which can provide high detection accuracy with much more runtime overhead in comparison with blacklist method. In addition, it might accidentally provide more threats to users they look to keep safe from it. These techniques are called heuristic approaches which depend on extracting features from the website to decide whether it phishing or benign. Such techniques can detect freshly created phishing websites (zero- day attacks) in contrast to blacklist techniques [10]. Other techniques use URL features as a combination of host information and lexical features [10,11]. Hosting information needs external server beside the huge feature vector generated by a bag of word method poses high latency preventing the application of such method for real time. However, URL based methods proved to be a good option to fight phishing attacks [12]. Mostly, URL features are used to train a machine learning algorithms (ML) to generate a classifier to detect unseen URLs.

Internet users impatient if they got a delay during surfing the Internet so that the development of any detection system must be fast enough in result delivery. However, nowadays the highest accuracy system the highest time consumption. Although that previous studies employed the URLs lexical features, no one tries to build phishing detection system using a pure lexical features to provide wide scope, lightweight, and highly accurate classifier.

II. RELATED WORKS

Several methods are proposed to detect phishing attacks most of them depend on phishing features extractions. The features can be extracted from the webpage contents and can provide satisfactory meaning to detect the attacks. However, this method loading the full webpage contents which in turn expose the users to more security threats due to malicious codes. As a result, to avoid this problem, Garera [13] proposed extracting the features from URLs only depending on the truth that the users directly use URLs to surf the internet so that phishers employ several obfuscation techniques to produce legitimate looking URLs.

The study in [14] analyses the characters properties of phishing URLs, from the findings domains of phishing usually contain the brand name of the target and use vowels and different characters. In addition, long URLs and short domain names are good features of phishing. Accordingly, several techniques are produced depending on lexical features of URLs only [15-17] such as dots numbers, domains numbers, URL length, and etc. The great benefits from this technique, first lexical features extraction is lightweight which means no time consumes. Second, avoiding latency and the threats of webpage loading. Generally, the features extracted from URLs are employed to train a machine learning classifiers to build the model of phishing detection.

One of the first attempts of using the bag of words representation of lexical features is presented in [18]. The work uses machine learning technique and produces 95% detection accuracy. This result of high accuracy confirmed by other researchers such in [19] which proved that using lexical features with machine learning lead to provide good meaning to fight phishing attacks besides high accuracies can be achieved.

An automated classification technique is presented by Marchal [20]. The technique name is PhishStorm which uses lexical URLs analysis in real time environment. PhishStorm is a central classifier positioned in front of the email server. Searching engines are used in order to extract 12 features from URLs and after this stage the machine learning classifier is used to detect the phishing URLs. The accuracy of the system is 94.91% with 1.44% false positive. Nevertheless, due to the use of searching engines this system is time consumed.

There are some methodologies use lexical features in combination with other types of features (i.e. host information and webpage content) such work is presented in [11]. This real time system to detect phishing URLs can provide 91% accuracy and the time required to classify a single URL is 5.54 seconds. This long processing time leads to bother Internet users and make such technique not the optimum choice for a real time environment.

As it evidenced by previous works, the URLs features with machine learning classifiers can present accuracies more than 90% in general, needs only little information, and wide scope covering of phishing attack vectors. In spite of all of these advantages, detection phishing attacks in real time by means of lightweight processing systems is still a hot research area.

III. RESEARCH APPROACH

Our methodology comprises three phases each phase output is input for the next phase. Datasets collection from different sources is presented in the first phase, pre-processing of dataset and language module building besides extraction of n-gram features are discussed in phase 2. Phase three is the classifiers results in term of evaluation metrics beside the comparison among the classifiers to identify the best one.

A. Phase 1: Collection of Datasets

The first phishing dataset is collected from Phishtank with 46,5461 phishing URLs compiled since January 2008. We suggest a new method to separate Phishtank URLs according to its listing years. In this way, we got three different datasets namely D2015, D2014, and D2013. In order to track the evolving of URLs phishing features with mimic the real-world environment, second dataset from different source is collected from OpenPhish with 4647 phishing URLs. Due to OpenPhish is launched recently, to our knowledge none of the works in literature had used this phishing dataset in pure phishing detection system.

The legitimate datasets are collected to cover the legitimate websites diversity. In this context, we gathered two datasets from publicly sources (webcrawler.com and DMOZ.org). From webcrawler, 10,275 legitimate URLs are collected and we named this dataset as WebCrawler. DMOZ dataset comprises of 10,275 randomly chosen URLs.

We paired legitimate URLs from DMOZ and WebCrawler with Openphish and PhishTank datasets and we named these datasets as D2013-DMOZ (D13DM), D2013- WebCrawler (D13WC), D2014-DMOZ (D14DM), D2014- WebCrawler (D14WC), D2015-DMOZ (D15DM), D2015- WebCrawler (D15WC), OpenPhish-WebCrawler (OWC) and OpenPhish-DMOZ (ODM). The method of dividing and merging the datasets is shown in Fig. 1.

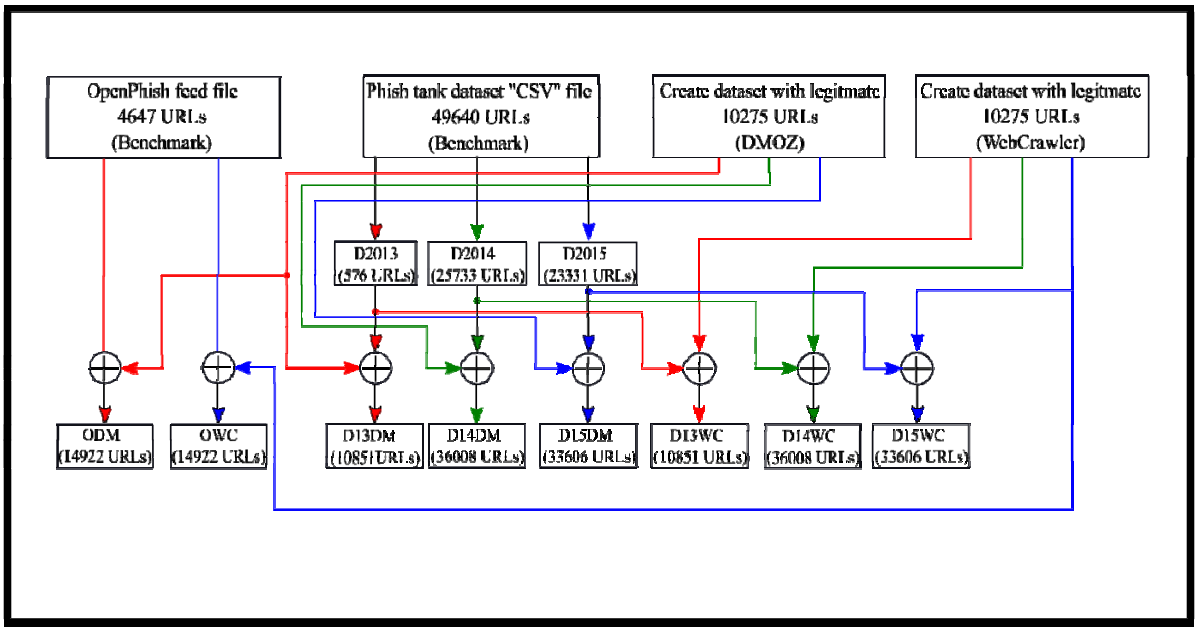

Fig. 1. The methodology of dataset division and merging

B. Phase 2: Datasets Pre-processing, N-gram model, and Features Extraction

In order to get an understandable format from the real life raw datasets, datasets pre-processing is critical due to the interested features lacking, incompleteness, and inconsistently. We make processing on our datasets to convert them into a format suitable for further processing. Phishtank dataset comprises a number of columns such as, the brand name target, phishing verified, and phishing URL. On the other hand, more columns are exit in Openphish dataset for instance: IP, isotime, sector, country code, host, etc. At the end of this stage, all unnecessary columns are removed and the combined datasets are converted into MySQL database to facilitate the next steps.

The main purpose of the language model is to assign a probability for each string in any language and the best model is the one give high probabilities for common strings and the reverse for uncommon strings. Depending on this rule, in this paper we try to build the URLs language model. Generally, the language models are contracted by parsing grams of n size where n is an integer. Markov chains is used where this method depends on gram probability in which each gram probability depends only on the probability of previous gram.

The n-gram model is used to compute the characters probability occurs in URLs whereby several works state that this method is the best method to construct the language model [21].Although, using n-gram model to construct the language model of URLs is done in literature [18] and employed to detect malicious websites in [22], to author's knowledge this is the first time the n-gram method used for pure URL phishing detection. In our work, the fourgrams, trigrams, bigrams, and unigram is constructed. The proposal in this paper to create the n-gram model using only the phishing URLs to answer the research question: could be the n-gram phishing based model differentiate between the legitimate and phishing URLs besides providing high accuracy?

For mathematical representation, w is assumed to be a gram of l length where l takes values from 1 to 4 in order to represent the grams. We constructed URL n-gram model with all occurrences of unique l size strings. Thus, the training dataset can be mapped as showed in Equ.1.

$$N - gram = w_1^l + w_2^l + w_3^l + \cdots w_n^l \quad (1)$$

Where w_i is a unique n-gram of l size and n represents the total number of grams. For each w, the total number of occurrences in dataset assumed to be T then n-grams can be mapped mathematically to get γ as shown in Equ.2.

$$\gamma = T_1^l + T_2^l + T_3^l + \cdots T_n^l \quad (2)$$

In Equ.2, T_1 represents w_1 number of occurrences, T_2 is the occurrences number of w_2, and so on. For each givenT_i, probability$_i$ is computed by dividing w_i total number of occurrences by all T_n summation as shown in Equ.3.

$$\text{probability}_i = \frac{T_i^l}{\sum_{i=1}^{n} T_i^l} \quad (3)$$

Now each w_i probability can be represented using the p set as in Equ. 4.

$$p = p_1^l + p_2^l + p_3^l + \cdots p_n^l \quad (4)$$

C. Phase 3: Classifiers Evaluation Metrics

We used the same evaluation metrics in our work [23], where these metrics are widely used to evaluate the performance of machine learning classifiers. Finally, the best classifier is highlighted at the end of this stage. The details of the used metrics are explained as following:

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

  $$\text{FPR} = \frac{NL \rightarrow P}{NL \rightarrow P + NL \rightarrow L} \quad (5)$$

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

  $$\text{FNR} = \frac{NP \rightarrow L}{NP \rightarrow P + NP \rightarrow L} \quad (6)$$

- **Recall or True Positive rate (TPR)**: Defined as the frequency of patterns which are detected correctly by the classifier as a phish.

  $$\text{TPR} = \frac{NP \rightarrow P}{NP \rightarrow P + NP \rightarrow L} \quad (7)$$

- **True Negative rate (TNR)**: Defined as the frequency of patterns which are detected correctly by the classifier as a legitimate.

  $$\text{TNR} = \frac{NL \rightarrow L}{NL \rightarrow L + NL \rightarrow P} \quad (8)$$

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

  $$\text{Accuracy} = \frac{NP \rightarrow P + NL \rightarrow L}{NP \rightarrow P + NL \rightarrow P + NL \rightarrow L + NP \rightarrow L} \quad (9)$$

IV. EXPERIMENTAL RESULTS

Depending only on phishing URLs, the n-gram model is separately constructed for host, path, and query. Java codes are developed to calculate each gram probability occurs in the phishing dataset. For close look, Fig. 2 depicts the flow of n-gram model constructing.

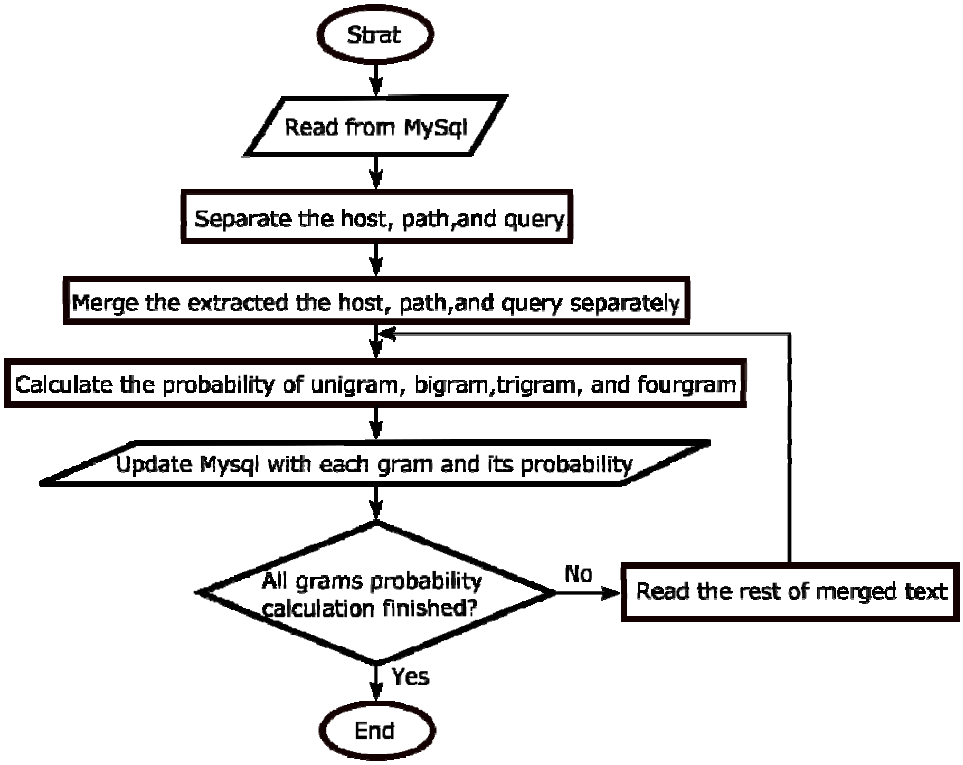

Fig. 2. N-gram model building flowchart.

The mySQL database is developed in order to extract the proposed features (12 n-gram features). Table columns of a MySQL are described in Table I.

TableI: MySQL database columns with description

Column Number	Column Name	Column Description
1	ID	Unique number of each row
2	URLs	URL of the websites which are imported from Phishingand legitimate sources.
3	UniGram_Host	The probability summation of the unigrams extractedfrom host n-gram model.
4	BiGram_Host	The probability summation of the bigrams extractedfrom host n-gram model.
5	TriGram_Host	The probability summation of the trigrams extractedfrom host n-gram model.
6	QuadGram_Host	The probability summation of the quadgrams extractedfrom host n-gram model.
7	UniGram_Path	The probability summation of the unigrams extractedfrom path n-gram model.
8	BiGram_Path	The probability summation of the bigrams extractedfrom path n-gram model.
9	TriGram_Path	The probability summation of the trigrams extractedfrom path n-gram model.
10	QuadGram_Path	The probability summation of the quadgrams extractedfrom path n-gram model.
11	UniGram_Query	The probability summation of the unigrams extractedfrom query n-gram model.
13	BiGram_Query	The probability summation of the bigrams extractedfrom query n-gram model.
13	TriGram_Query	The probability summation of the trigrams extractedfrom query n-gram model.
14	QuadGram_Query	The probability summation of the quadgrams extractedfrom query n-gram model.
15	Label	The classification of each webpage, 1 mean phishingand 0 mean legitimate.

The Java code is developed to extract URLs features by splitting the host, path, and query of each URL then extracting all grams of each part. Next step is to read each part grams in sequence and looking up a MySQL database to find the probability of each gram. In case of gram probability not found, probability with zero value is assigned. To generate the features values, the host, path, and query grams probabilities are added separately.

After finishing the features extraction process, the vector of features is converted into the format of an ARFF file. Three classifiers with the default parameter values are employed in this experiment are LR, SVM, and J4.8. These classifiers are implemented in a very famous and widely used tool for data mining the Waikato Environment for Knowledge Analysis (WEKA) [24]. Classifiers performance is evaluated using 10-fold cross-validation on each dataset and the machine used to run all the experiments has core-i7 2.57 GHz processor with 8 GB RAM.

The error rate of each classifier on all datasets is shown by the bars in Fig. 3. Overall error rates ranging from 2.74% to 12.82%. On D13WC and D13DM, the differences are not significant in accuracy rates in the range of 95% to 97% as showed in Tables II and Table III. In addition, the best TPR range from 55%-59% is obtained by LR and all classifiers excellently detected the benign URLs where SVM provided TNR of 100% on D13DM. However, on D13DM and D13WC, the worst TPR is achieved among all datasets due to the high FNR which is a result of the few number of phishing samples exit in these datasets so that the classifiers cannot recognize the benign URLs from the phishing ones.

As more phishing instances available in D15WC, D15DM, D14WC, and D14DM datasets, true positive rate values rapidly increased to reach a maximum of 96.7% as shown in Table IV, Table V, Table VI, and Table VII. J4.8 produced the highest accuracy value 93.52% followed by SVM and the worst classifier is LR.

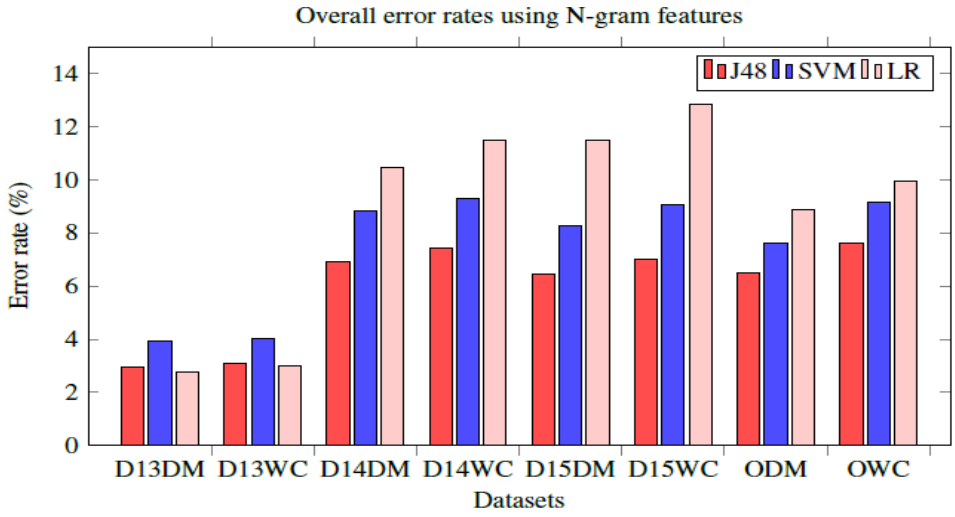

Fig. 3. Classifiers error rates.

TableII: Performance of classifierson D13DM.

Classifier	TPR	FPR	TNR	FNR	Accuracy	Error
J48	0.486	0.002	0.998	0.514	97.0417%	2.9583%
SVM	0.262	0	1	0.738	96.0741%	3.9259%
LR	0.59	0.006	0.994	0.41	97.2537%	2.7463%

TableIII: Performance of classifiers on D13WC.

Classifier	TPR	FPR	TNR	FNR	Accuracy	Error
J48	0.477	0.003	0.997	0.523	96.9035%	3.0965%
SVM	0.252	0.001	0.999	0.748	95.9543%	4.0457%
LR	0.556	0.007	0.993	0.444	97.0049%	2.9951%

TableIV: Performance of classifiers on D14DM.

Classifier	TPR	FPR	TNR	FNR	Accuracy	Error
J48	0.967	0.16	0.84	0.033	93.0988%	6.9012%
SVM	0.933	0.142	0.858	0.067	91.1575%	8.8425%
LR	0.931	0.192	0.808	0.069	89.5495%	10.4505%

TableV: Performance of classifiers on D14WC.

Classifier	TPR	FPR	TNR	FNR	Accuracy	Error
J48	0.966	0.175	0.825	0.034	92.5766%	7.4234%
SVM	0.936	0.167	0.833	0.064	90.6854%	9.3146%
LR	0.929	0.224	0.776	0.071	88.5081%	11.4919%

TableVI: Performance of classifiers on D15DM.

Classifier	TPR	FPR	TNR	FNR	Accuracy	Error
J48	0.966	0.135	0.865	0.034	93.5279%	6.4721%
SVM	0.939	0.133	0.867	0.061	91.7039%	8.2961%
LR	0.91	0.171	0.829	0.09	88.511%	11.489%

TableVII: Performance of classifiers on D15WC.

Classifier	TPR	FPR	TNR	FNR	Accuracy	Error
J48	0.966	0.151	0.849	0.034	93.0072%	6.9928%
SVM	0.939	0.158	0.842	0.061	90.9094%	9.0906%
LR	0.909	0.212	0.788	0.091	87.1719%	12.8281%

Classifiers results from OWC and ODM are presented in Table VIII and Table IX. We noted the results confirm previous observations where the best accuracy of 93.5% achieved by J4.8 with good trade off in FNR and FPR.

TableVIII: Performance of classifierson ODM.

Classifier	TPR	FPR	TNR	FNR	Accuracy	Error
J48	0.892	0.046	0.954	0.108	93.5062%	6.4938%
SVM	0.891	0.061	0.939	0.109	92.3938%	7.6062%
LR	0.799	0.038	0.962	0.201	91.1339%	8.8661%

TableIX: Performance of classifierson OWC.

Classifier	TPR	FPR	TNR	FNR	Accuracy	Error
J48	0.897	0.064	0.936	0.103	92.3737%	7.6263%
SVM	0.866	0.073	0.927	0.134	90.8256%	9.1744%
LR	0.783	0.047	0.953	0.217	90.0281%	9.9719%

From the overall results, from 6.47% to 7.62% is the lowest error rate achieved by J48 whereby SVM follows by range of error from 7.6% to 9.31% and finally the worst error rate obtained by LR with range of 8.86% to 12.82%.

Depending on the experimentally obtained results, training the classifiers with sufficient and balanced datasets is crucial to increase the classifiers performance. In spite of the promising classifiers performance achieved in our method. However, high error rate is still problem subject for improvement and in our opinion the reason of this high error rate can be due to phishing URLs have high variability features which in turn make the generated n-gram model do not have the ability to differentiate legitimate and phishing URLs.

V. TRAINING AND TESTING TIMES

The main purpose of our proposed method is to protect users from phishing websites in real time and to achieve this goal the classifiers should consume minimal processing time per URL. The feasibility of J48 classifier for real time application is explored in this experiment. We compute the time required to extract the features, training time, and testing time.

From the experimental results, our method can extract the features and transform them into classifier's input format in 0.12 second in average per URL. This superior processing time is achieved due to our method depends only on lexical features and no need for any external features.

Training and testing time measurement is obtained by splitting the D14DM into 80/20 ratio in order to perfectly testing the time required to build the model and the time consumed in the testing mode. Also in this experiment, the time required to identify a single URL is reported. After three times running the experiment the average results are shown in Table X.

TableX: Training and testing times of J48.

Samples number	Average time
28806	Train=0.848 second
7202	Test=0.021 second
1	Test=0.001 second

It is clear that model generating time is higher than time consumed for testing. However, classifier building is less frequent process and can be done offline. In order to provide real time processing, features extraction time and testing time are the critical factors and from the results very low time is consumed to classify a single URL and this amount of time can be negligible compared with the time needed to extract the features. In this context, our proposed classifier can detect a single URL in 0.12 second in average.

VI. CONCLUSIONS

In this paper, we present wide scope and lightweight phishing detection system. Three classifiers are tested and the best results are achieved by J48 classifier with 93% accuracy and a single URL can be detected in 0.12 second. In comparison with previous work, this amount of accuracy is obtained without time consuming. In spite of phishing based n-gram features can provide a good means to detect phishing websites, this method is still subject for improvements in term of the accuracies and error rates. Our future work is to construct the n-gram models by using URLs from legitimate datasets only to answer the research question: can be the legitimate based n-gram model better distinguishes the phishing URLs than phishing based n-gram based model?

REFERENCES

[1] M. Khonji, Y. Iraqi, and A. Jones, "Phishing detection: a literature survey," IEEE Communications Surveys & Tutorials, vol. 15, pp. 2091-2121. 2013.
[2] G. Liu, B. Qiu, andL. Wenyin, "Automatic detection of phishing target from phishing webpage," In 20th International Conference on Pattern Recognition (ICPR), 2010, p. 4153-4156.
[3] N. Abdelhamid, "Multi-label rules for phishing classification," Applied Computing and Informatics, vol. 11, pp. 29-46, 2015.
[4] S. Afroz, and R. Greenstadt, "Phishzoo: Detecting phishing websites by looking at them," In Fifth IEEE International Conference on Semantic Computing (ICSC), 2011,p. 368-375.
[5] I. R. A. Hamid, and J. H. Abawajy, "Profiling phishing email based on clustering approach," In 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 2013, p. 628-635.
[6] M. Khonji, Y. Iraqi, and A. Jones, "Enhancing phishing E-Mail classifiers: a lexical URL analysis approach," International Journal for Information Security Research (IJISR), vol. 2. 2012
[7] L. H. Lee, K. C. Lee, H. H. Chen, and Y. H. Tseng, "POSTER: Proactive Blacklist Update for Anti-Phishing," In Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, 2014, p. 1448-1450.
[8] G. Liu, B. Qiu, and L. Wenyin, "Automatic detection of phishing target from phishing webpage," In IEEE20th International Conference on Pattern Recognition (ICPR), 2010,p. 4153-4156.
[9] S. Sinha, M. Bailey, and F. Jahanian, "Shades of Grey: On the effectiveness of reputation-based "blacklists"," In IEEE 3rd International Conference on Malicious and Unwanted Software, 2008, p. 57-64.
[10] A. Le, A. Markopoulou and M. Faloutsos, "Phishdef: Url names say it all," In IEEE Proceedings INFOCOM, 2011, p. 191-195.
[11] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song, "Design and evaluation of a real-time url spam filtering service," In IEEE Symposium on Security and Privacy (SP), 2011, p. 447-462.
[12] 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, 2009, p. 1245-1254.
[13] 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, 2007, p. 1-8.
[14] D. K. McGrath and M. Gupta, "Behind Phishing: An Examination of Phisher Modi Operandi" LEET, 2008.
[15] 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, 2010, p. 54-60.
[16] 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, 2011, p. 109-115.
[17] 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, pp. 11-24, 2014.
[18] 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, 2005, p. 325-326.
[19] 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, 2011.
[20] S. Marchal, J. François, R. State, and T. Engel, "PhishStorm: Detecting Phishing With Streaming Analytics," IEEE Transactions on Network and Service Management, vol. 11, pp. 458-471, 2014.
[21] F. Jelinek, "Up from trigrams," Eurospeech, 1991.
[22] M. Darling, G. Heileman, G. Gressel, A. Ashok, and P. Poornachandran, "A lexical approach for classifying malicious URLs," In IEEE International Conference on High Performance Computing & Simulation (HPCS), 2015,p. 195-202.
[23] 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, 2015.
[24] 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, pp. 10-18, 2009.