Detecting Phishing Websites Using Machine Learning

Amani Alswailem
Computer Science Department
Al-Imam Muhammad Ibn Saud Islamic University
Riyadh, Saudi Arabia
amnsuwailem@sm.imamu.edu.sa

Bashayr Alabdullah
Computer Science Department
Al-Imam Muhammad Ibn Saud Islamic University
Riyadh, Saudi Arabia
baalabdollah@sm.imamu.edu.sa

Norah Alrumayh
Computer Science Department
Al-Imam Muhammad Ibn Saud Islamic University
Riyadh, Saudi Arabia
noaromaih@sm.imamu.edu.sa

Dr.Aram Alsedrani
Computer Science Department
Al-Imam Muhammad Ibn Saud Islamic University
Riyadh, Saudi Arabia
assdrinay@imamu.edu.sa

***Abstract*—Phishing website is one of the internet security problems that target the human vulnerabilities rather than software vulnerabilities. It can be described as the process of attracting online users to obtain their sensitive information such as usernames and passwords. In this paper, we offer an intelligent system for detecting phishing websites. The system acts as an additional functionality to an internet browser as an extension that automatically notifies the user when it detects a phishing website. The system is based on a machine learning method, particularly supervised learning. We have selected the Random Forest technique due to its good performance in classification. Our focus is to pursue a higher performance classifier by studying the features of phishing website and choose the better combination of them to train the classifier. As a result, we conclude our paper with accuracy of 98.8% and combination of 26 features.**

***Index Terms*—Random forest, URL, browser extension, machine learning, phishing websites, phishing features.**

I. Introduction

In today's world, technology has become an integral part of the twenty-first century. The internet is one of these technologies, which is growing rapidly every year and plays an important role in individuals' lives. It has become a valuable and a convenient mechanism for supporting public transactions such as e-banking and e-commerce transactions. That has led the users to trust it is convenient to provide their private information to the Internet. As a result, the security thieves that have started to target this information have become a major security problem. Phishing websites are considered to be one of these problems. They are using a social engineering trick, which can be described as fraudsters that try to manipulate the user into giving them their personal information based on exploiting human vulnerabilities rather than software vulnerabilities.

Statistics have shown that the number of phishing attacks keeps increasing, which presents a security risk to the user information according to the Anti-Phishing Working Group (APWG) [1] and recorded phishing attacks by Kaspersky Lab [2], which stated that it has increased by 47.48% from all of the phishing attacks that have been detected during 2016.

Recently, there have been several studies that tried to solve the phishing problem. Some researchers used the URL and compared it with existing blacklists that contain lists of malicious websites, which they have been creating, and there are others that have used the URL in an opposite manner, namely comparing the URL with a whitelist of legitimate websites. The latter approach uses heuristics, which uses a signature database of any known attacks that match the signature of the heuristic pattern to decide if it is a phishing website. Additionally, measuring website traffic using Alexa is another way that has been implemented by researchers to detect phishing websites [7].

Moreover, other researchers have used machine learning techniques. Machine learning is a field of computer science, which is also a branch of artificial intelligence (AI) that performs tasks and is capable of learning or acting in an intelligent way. It has two different types of learning: supervised learning and unsupervised learning. Supervised learning is based on training a model by giving it a set of measured features of data associated with a target label related to these data, and once the model is trained it can generate a new target label with unknown data. On the other hand, unsupervised learning is based on generating new data without giving any target label in the training process [22].

In this paper, the focus will be on the features combination that we get from Random Forest (RF) technique, as it has high accuracy, is relatively robust, and has a good performance [21].

The organization of this paper is as follows: In Section Two we give an overview in the background. Then, in Section Three, we discuss related works. After that, in Sections Four, our experiment is to be represented. Section Five presents the result and conclusion.

978-1-7281-0108-8/19/$31.00 2019 IEEE

II. Background

Phishing is one of the major problems of the information security. It can occur in two ways, either by receiving suspicious emails that lead to the fraudulent site or by users accessing links that go directly to a phishing website. However, the two methods are common in one thing, which is the attacker targets human vulnerabilities rather than software vulnerabilities. Phishing can be described as fraudsters that try to manipulate the user into giving them their personal information such as username, password, and a credit card number. These scams are leading to economic and financial crises for users [4].

In the early 90s, phishers created a false account with a fake identity and fake credit card on the America Online (AOL) company that provided a web portal and was an online service provider. In this way, the phishers could be exploiting its services without any cost to them. Since then, in the mid 90s, AOL strengthened its system to prevent phishers. Unfortunately, the phishers used another method, stealing valid accounts by acting as an AOL employee and requesting users provide their password for security purposes. This occurred either by email or via instant message services [6].

Recently, there have been several studies trying to solve the phishing problem. They can be categorized into four categories: blacklist, heuristic, content analysis, and machine learning techniques. The blacklist compares the URL with an existing database that contains a list of phishing website URLs. Because of the rapid increase of phishing websites, the blacklist approach has become inefficient in deciding whether each URL is a phishing website or not, and this kind of delay can lead to zero-day attacks from new phishing sites [4].

The heuristics approach uses the signature databases of any known attacks, to match it with the signature of a heuristic pattern. The trade-off of using heuristics is failing to detect novel attacks, as it is easy to bypass the signatures through obfuscation. Also, updating the signature database is slow considering the growth of novel attacks, especially zero-day attacks [7]. Content analysis is a content-based approach in detecting phishing websites, using well-known algorithms such as term frequency/inverse document frequency (TF-IDF). It analyses the text-based content of a page itself to decide whether the website is phishing or not. Additionally, measuring website traffic using Alexa is another method that has been implemented by researchers to detect phishing websites [7] [3].

Machine learning takes advantage of its predictive power. It learns the characteristics of the phishing website and then predicts new phishing characteristics. There are several techniques, such as nave Bayes (NB), decision tree (DT), support vector machines (SVM), RF, artificial neural network (ANN), and Bayesian net (BN). The accuracy of phishing detection varies from one algorithm to another.

III. Related work

A. Content Based Approach

Zhang et al. [3] presented the design and evaluation of CANTINA, a novel, content-based approach to detecting phishing websites, using the well-known TF-IDF algorithm. It analyses the text-based content of a page by itself. They experimented with some simple heuristics that can be applied to reduce false positives. As a result, a pure TF-IDF approach can catch about 97% of phishing sites with about 6% false positives, and with heuristics it catch about 90% of phishing sites with only 1% false positives.

Rao and Ali [5] implemented a desktop application to detect phishing websites using a novel heuristic based on URLs and website content. With the application called PhishShield, they used copyright, null footer links, zero links of the body html, links with maximum frequency domains, and whitelists to detect phishing websites. It achieved an accuracy of 96.57% with a FP of 0.035%.

B. URL Approach

A new approach that Nguyen et al. [7] had proposed to detect phishing sites is by deriving different components from the URL and computing a metric for each component. Then, the page ranking will be combined with the achieved metrics to decide whether the websites are phishing websites. The results showed that the technique can detect over 97% of phishing websites.

Jeeva and Rajsingh [8] presented a system for prediction phishing URLs by generating rules of association rule mining. They used the apriori algorithm to pick known information from frequent item set properties that were extracted from the dataset. Jeeva and Rajsingh [8] also used another algorithm that performs on hidden data to obtain the accuracy of association rules, which is a predictive apriori that engages the confidence and the support techniques that are measured in its accuracy, unlike a priori, which only mark rules that have the confidence technique. As a result, they presented significant features of the URL that distinguish if it is phishing or legitimate.

C. Machine Learning

Sanglerdsinlapachai and Rungsawang [9] added new features to heuristic features of CANTINA and used six machine learning techniques to improve blocking efficiency, and their features were able to boost detection accuracy by 15% and 20% in terms of f-measure and error rate, respectively.

Xiang et al. [10] present a layered solution CATINA+, which is an upgrade of the work of Zhang [3]. They added more features to CATINA and applied machine learning techniques, and the result was a 92% true positive rate and a 0.4% false positive rate.

Mohammad et al. [11] designed an anti-phishing tool that can predict phishing attacks at a timescale using structuring neural networks (NN). The tool has used a dataset of 600 legitimate URLs and 800 phishing URLs to apply optimal generalized performance using optimal NN structure. The accuracy when epochs of NN is 500, the testing accuracy is 92.48% using 17 features, such as, IP address, long URL, URL contains '@', misuse of HTTPs, subdomain in URL, request URL.

Pradeepthi and Kannan [12] provided a survey of research works conducted on classification techniques for phishing URL detection. They use 4500 URLs as the dataset and categorized the features into four categories: lexical features, URL based features, network based features, and domain-based features. They experimented on several machine learning techniques such as NB, multi-layer perceptron, J48 tree, Logistic Model Tree(LMT), RF, random tree, C4.5, ID 3, C-RT, and K-nearest neighbour (KNN). As a result, they concluded that the tree-based classifiers are the most suitable for the task of phishing URL classification.

Marchal et al. [13] has presented a system called PhishStorm that can detect phishing URLs based on lexical analysis of URL. The system uses 12 features, such as popularity of the registered domain, Alexa Rank, the number of related and associated words found in search engine queries, and data based on these words in URL. After applying these features to a dataset of 96,018 phishing and legitimate URLs using supervised classification, the classification resulted in 94.91% accuracy with a low false positive rate of 1.44%. With such a rate, the system could calculate the risk score of URLs on the testing dataset with an accuracy of 92.22% for legitimate URLs and 83.97% for phishing URLs.

Sirageldin et al. [14] presented a mechanism to detect phishing websites based on two categories: URL lexical analysis and page content analysis. Using DT, ANN, NB, SVM, and KNN on a dataset of 29,500 legitimate and phishing URLs, it give an accuracy of 95.12%, 96.01%, 88.47%, 93.57%, and 92.90% for these algorithms, respectively, with an approximate false positive rate of 0. The drawback of this mechanism concerning the features collection is the partial rendering method.

Verma and Dyer [15] proposed statistical analysis of website URLs with machine learning techniques by constructing a multiple machine learning classifiers comprising a short list of features, and evaluating them on four unique real datasets. At the end, they found that statistical tests for distributions of character frequencies yielded highly accurate classifiers.

Hieu Nguyen and Thai Nguyen [16] evaluated and compared a five classification of algorithms to detect phishing websites by extracting the URL features and website content information. The algorithms including J48 DT, RF, SVM, NB, and NN with 98.5%, 98.8%, 86.1%, 96.9%, and 98.4% classification accuracy, respectively.

Tahir et al. [17] had proposed a hybrid model classification that use supervised machine learning algorithms in two phases. In phase I, classification is performed on the techniques individually, for example, RF, sequential minimal optimization (SMO), J48 DT, BN, NB, and instance based learning (IBk) models. Then, they selected the best three models based on the criteria of performance and high accuracy. In phase II, the resulting models were combined with the best three individual models to make a hybrid model. The final result of the classification model combination was IBk with BN, and IBk with J48 would give a better result with an accuracy of 97.75% and an error rate of less than 0.225. The dataset contains 11055 instances to train on using 30 features, such as, prefix or suffix of '-' in the URL, age of the domain, disabling right clicks, website forwarding, Google Index, and using pop-up windows.

Mamun et al. [18] used the machine learning techniques to detect and categorize the malicious URLs according to their attack type. Their classification of data was based on two groups of algorithms, KNN algorithm, tree based classifiers C4.5, and RF. They focused on four types of malicious use of URLs such as spam, malware, phishing and defacement. Among the classifiers that were tested it appeared that random forest had the highest accuracy, while the KNN and C4.5 classifiers had approximately the same performance.

Weedon et al. [19] evaluated the random forest performance using a lexical dataset and compared it with three other algorithms, J48, NB, and logistic regression. The evaluation shows RF has higher accuracy at 86.9% and lower false negatives than the other algorithms.

IV. Experiment

A. Dataset

We collect 16000 of phishing and legitimate URLs. The phishing websites consist of 12000 phishing URLs that has been collected from PhishTank [20]. In the other hand, the legitimate websites consist of 4000 legitimate URLs that have been collected by a daily use from 10 chosen users. However, the final dataset after handling missing data and removing the duplicate is size of 6116.

B. Features extraction

The phishing websites have certain characteristics and patterns that can be considered as features. In this subsection, we cover all phishing website features that have been used in the previous researches as possible. Furthermore, while we are studying the phishing characteristics and patterns we notice some new characteristics that can be considered as features. The total number of phishing features is 36 where 3 of them are new features. We categorize them into three main categories as shown below with features in table I:

- Features can be extracted from URL.
- Features can be extracted from page content.
- Features can be extracted from page rank.

We use the number of input email and number of input password as the new features for phishing website, Since the target of the phishing website is to steal sensitive information such as email and password. We consider the number of input that have the type email or password as feature for phishing website. Another new feature is the number of button, while we are studying phishing features we noticed that a large number of phishing website doesn't use the submit button instead they use a regular button, so we consider it as feature for phishing website.

TABLE I
FEATURES THAT CAN BE EXTRACTED FROM URL, PAGE CONTENT AND PAGE RANK

Features Based on		
URL	Length of URL	Length of hostname of URL
	Length of the path of URL	Number of dot (.) in the path
	Number of dot (.) in hostname	Number of slashes (/) in URL
	Number of hyphen (-) in hostname	Number of special characters (: ; % & ? +)
	Number of at (@) in the URL	Number of digit in hostname
	Number of underscore (_) in hostname	Number of underscore (_) in path
	Number of certain keyword in URL	Number of hexadecimal with %
	Transport layer security	IP address
	Presence of www	Port redirect
	Unicode in URL	Hexadecimal characters
Page content	Number of forms	Number of forms with action ’GET’
	Number of forms with action ’POST’	Number of script
	Number of outer src script	Number of $< Iframe >$
	Number of $< Applet >$	Number of $< Embed >$
	Number of $< Frame >$	Number of link
	Number of non-link	Number of submit
	Number of input email	Number of input password
	Number of button	
Rank	Alexa rank	Age of domain

C. Methodology

We study all features to indicate the strongest, weakest and to remove the irrelevant features; the study is based on examining all possible combination of 36 features. The size of all possible combination can be specify using this formula:

$$\sum_{k=1}^{36} = \frac{n!}{k!(n-k)!}$$

where k is the number of the taken features that start from 1 to 36. and n is the number of all features which is 36.

Since the number of all possible combination is a huge number, the study will be summarized into taking the maximum and the minimum result for each k combination. In the end, the higher accuracy with the smallest number of features will be chosen for a better combination. In Fig. 1, it summarizes the process of feature selection.

The main function of the system is to decide the state of the website if it is a phishing or legitimate website. This function can be performed using the algorithm as shown in Fig. 2. This algorithm will be triggered whenever the user enters a new website, the role of the algorithm is to extract the features of the website using URL and Document Object Model (DOM) object. The URL used to extract the URL's and page rank's features. While the DOM used to extract the content page's features which is a connection between scripts and website's page that have logical structure of documents and provide accessing and manipulation for programmer to the DOM file. Afterwards, the extracted features will be sent to the classifier to produce the target label that indicates the state of the website then executes the suitable action on that.

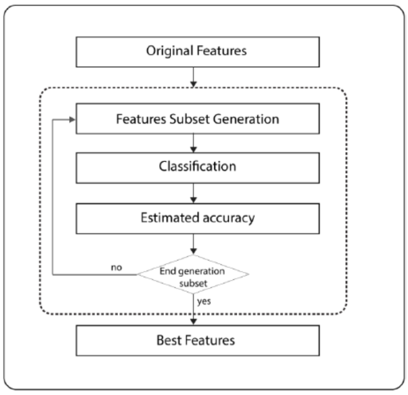

Fig. 1. The process of feature selection.

```
Algorithm 1: Website State
  Input: URL, DOM
  Output: Phishing or Legitimate website with action
  URLFeatures= Extract_URL_Features(URL)   ▷ return all features URL
  ContentPageFeatures= Extract_ContentPage_Features(DOM)   ▷ return
    all features of content page
  RankFeatures= Extract_Rank_Features(URL)   ▷ return all features of
    rank

  targetLable= Random_Forest_Classifier(URLFeatures,
    ContentPageFeatures, RankFeatures)
  Output= Decision(targetLable)
```

Fig. 2. Detecting phishing website algorithm.

We build the classifier using RF technique as in the following steps:

1) Split data into training and test dataset, which we take 80% for training and 20% for testing.
2) Train and test all possible combination of 36 features dataset to get the strongest features that arise the accuracy of detection.
3) After step two, we have numbers of features which goes to the final stage of training and testing.
4) Execute the final classifier.

V. RESULT AND CONCLUSION

We have study all 36 features in order to reduce time computation and providing high performance with the least combination of the powerful features. However, because of time shortage and hardware limitation, we chose random features to process its combination. We concluded after some observation that the combination of features computed take the shape of normal distribution curve, it starts with least combination of features with low probability of combination and time consuming, then picks up accordingly, then goes down as it reach final number of 36 features, as shown in Fig. 3.

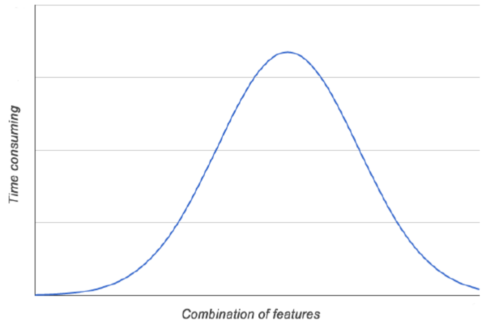

Fig. 3. Extraction features algorithm.

Nevertheless, after we are done with experimenting the combination of features listed in table V shows the maximum and minimum value for the resulted accuracy of features combination.

Combination of features	Maximum accuracy	Minimum accuracy
1	0.94281	0.564542
2	0.958333	0.556373
3	0.966503	0.544935
4	0.972222	0.549837
5	0.979575	0.539216
6	0.984477	0.547386
7	0.982843	0.553104
29	0.988562	0.913399
30	0.987745	0.908497
31	0.984477	0.914216
32	0.986111	0.916667
33	0.982026	0.933824
34	0.981209	0.939542
35	0.974673	0.940359
36	0.959967	0.959967

We notice the minimum value obtained is 0.539216, while the maximum value obtained is 0.988562. In addition, the combination of 29 features proved to have the least features combination of: *(securLayer, HEXinURL, hostLength, URLlength, pathLength, dashInHOST, dotInHOST, dotInPATH, slashInURL, DigitInHOST, NumOfCertainKeyword, numOfSpecialCharacters, underscoreInHOST, PresenceOfWWW, numGet, portRedirect, numOfHexWithePersent, NumberOfForms, numpost, numberOfOuterSRCinScript, NumberOfIframe, NumberOfFrame, NumberOflink, alexa, NumberOfNonlink, NumberOfFsubmint, NumberOfInputpassword, NumberOfFbutton, age)*. Further, Fig. 4 shows the features importance which determine how the feature is strong by computing its relative importance. We concluded from the features importance that the features *(underscoreInHOST, portRedirect, NumberOfFrame)* do not have any strength in the classifier.

Since our main objective is to archive a higher accuracy with minimal number of features, we delete these features and we achieve the same accuracy with 26 features of: *(securLayer, HEXinURL, hostLength, URLlength, pathLength, dashInHOST, dotInHOST, dotInPATH, slashInURL, DigitInHOST, NumOfCertainKeyword, numOfSpecialCharacters, PresenceOfWWW, numGet, numOfHexWithePersent, alexa, age, NumberOfForms, numpost, numberOfOuterSRCinScript, NumberOfInputpassword, NumberOfIframe, NumberOflink, NumberOfNonlink, NumberOfFsubmint, NumberOfFbutton)* as shown in Fig. 5.

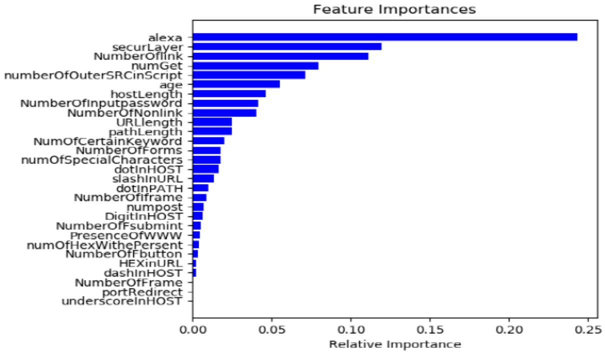

Fig. 4. The important features of 29 features.

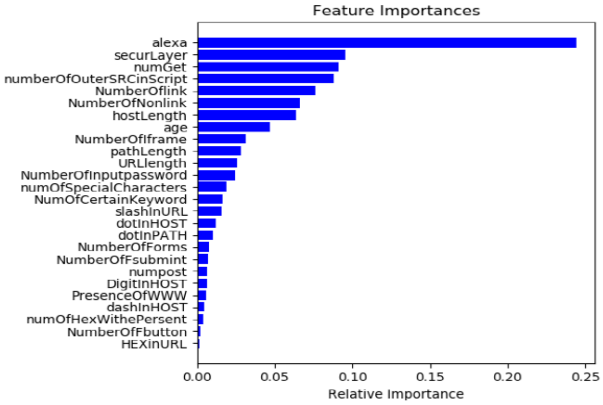

Fig. 5. The important features of 26 features.

As a result, we selected them for our Anti-phishing extension browser as final features that are used for the extension.

Acknowledgment

We would like to express our deep appreciation and our sincere gratitude to Dr.Aram Alsedrani for her valuable advice guidance and to our families and friends for continued encouragements and support.

References

[1] AO Kaspersky lab. (2017). The Dangers of Phishing: Help employees avoid the lure of cybercrime. [Online] Available: https://go.kaspersky.com/Dangers-Phishing-Landing-Page- Soc.html [Oct 30, 2017].

[2] ”Financial threats in 2016: Every Second Phishing Attack Aims to Steal Your Money” Internet: https://www.kaspersky.com/about/press-releases/2017_financial-threats-in-2016. Feb 22, 2017 [Oct 30, 2017].

[3] Y. Zhang, J. I. Hong, and L. F. Cranor, ”Cantina: A Content-based Approach to Detecting Phishing Web Sites,” New York, NY, USA, 2007, pp. 639-648.

[4] M. Blasi, ”Techniques for detecting zero day phishing websites.” M.A. thesis, Iowa State University, USA, 2009.

[5] R. S. Rao and S. T. Ali, ”PhishShield: A Desktop Application to Detect Phishing Webpages through Heuristic Approach,” Procedia Computer Science, vol. 54, no. Supplement C, pp. 147-156, 2015.

[6] E. Jakobsson, and E. Myers, Phishing and Counter-Measures: Understanding the Increasing Problem of Electronic Identity Theft. Wiley, 2006, pp.2–3.

[7] L. A. T. Nguyen, B. L. To, H. K. Nguyen, and M. H. Nguyen, ”Detecting phishing web sites: A heuristic URL-based approach,” in 2013 International Conference on Advanced Technologies for Communications (ATC 2013), 2013, pp. 597-602.

[8] Z. Zhang, Q. He, and B. Wang, ”A Novel Multi-Layer Heuristic Model for Anti-Phishing,” New York, NY, USA, 2017, p. 21:1-21:6.

[9] N. Sanglerdsinlapachai and A. Rungsawang, ”Web Phishing Detection Using Classifier Ensemble,” New York, NY, USA, 2010, pp. 210-215.

[10] G. Xiang, J. Hong, C. P. Rose, and L. Cranor, ”CANTINA+: A Feature-Rich Machine Learning Framework for Detecting Phishing Web Sites,” ACM Trans. Inf. Syst. Secur., vol. 14, no. 2, pp. 21:1-21:28, Sep. 2011.

[11] R. M. Mohammad, F. Thabtah, and L. McCluskey, ”Predicting phishing websites based on self-structuring neural network,” Neural Comput & Applic, vol. 25, no. 2, pp. 443-458, Aug. 2014.

[12] Pradeepthi K V and Kannan A, ”Performance study of classification techniques for phishing URL detection,” in 2014 Sixth International Conference on Advanced Computing (ICoAC), 2014, pp. 135-139.

[13] S. Marchal, J. Franois, R. State, and T. Engel, ”PhishStorm: Detecting Phishing With Streaming Analytics,” IEEE Transactions on Network and Service Management, vol. 11, no. 4, pp. 458-471, Dec. 2014.

[14] A. Sirageldin, B. B. Baharudin, and L. T. Jung, ”Malicious Web Page Detection: A Machine Learning Approach,” in Advances in Computer Science and its Applications, Springer, Berlin, Heidelberg, 2014, pp. 217-224.

[15] R. Verma and K. Dyer, ”On the Character of Phishing URLs: Accurate and Robust Statistical Learning Classifiers,” New York, NY, USA, 2015, pp. 111-122.

[16] H. H. Nguyen and D. T. Nguyen, ”Machine Learning Based Phishing Web Sites Detection,” in AETA 2015: Recent Advances in Electrical Engineering and Related Sciences, V. H. Duy, T. T. Dao, I. Zelinka, H.-S. Choi, and M. Chadli, Eds. Cham: Springer International Publishing, 2016, pp. 123-131.

[17] M. A. U. H. Tahir, S. Asghar, A. Zafar, and S. Gillani, ”A Hybrid Model to Detect 76 Phishing-Sites Using Supervised Learning Algorithms,” in 2016 International Conference on Computational Science and Computational Intelligence (CSCI), 2016, pp. 1126-1133.

[18] M. S. I. Mamun, M. A. Rathore, A. H. Lashkari, N. Stakhanova, and A. A. Ghorbani, ”Detecting Malicious URLs Using Lexical Analysis,” in Network and System Security: 10th International Conference, NSS 2016, Taipei, Taiwan, September 28-30, 2016, Proceedings, J. Chen, V. Piuri, C. Su, and M. Yung, Eds. Cham: Springer International Publishing, 2016, pp. 467- 482.

[19] M. Weedon, D. Tsaptsinos, and J. Denholm-Price, ”Random forest explorations for URL classification,” in 2017 International Conference On Cyber Situational Awareness, Data Analytics And Assessment (Cyber SA), 2017, pp. 1-4.

[20] ”PhishTank — Join the fight against phishing.” [Online]. Available: https://www.phishtank.com/. [Accessed: 29-Nov-2017].

[21] L. Breiman, ”Random Forests,” Machine Learning, vol. 45, no. 1, pp. 5-32, Oct. 2001.

[22] J. VanderPlas, Python data science handbook, 1st ed. 1005 Gravenstein Highway North, Sebastopol, CA 95472.: OReilly Media, Inc., 2016, pp. 331–515.