URL Phishing Detection using Machine Learning Techniques based on URLs Lexical Analysis

Conference Paper · May 2021

DOI: 10.1109/ICICS52457.2021.9464539

4 authors, including:

Mohammad Ababneh
Princess Sumaya University for Technology
19 PUBLICATIONS **147** CITATIONS

SEE PROFILE

Khaled Mahmoud
Princess Sumaya University for Technology
20 PUBLICATIONS **147** CITATIONS

SEE PROFILE

Sherenaz W. Al-Haj Baddar
University of Jordan
41 PUBLICATIONS **212** CITATIONS

SEE PROFILE

URL Phishing Detection using Machine Learning Techniques based on URLs Lexical Analysis

Mohammed Abutaha[*,1], Mohammad Ababneh[*,2], Khaled Mahmoud[*,3], Sherenaz Al-Haj Baddar[*,§,4]
[*]*Computer Science Department*
Princess Sumaya University for Technology
Amman, Jordan

[§]*Computer Science Department*
The University of Jordan
Amman, Jordan
moh20188026@std.psut.edu.jo[1], m.ababneh@psut.edu.jo[2], k.mahmoud@psut.edu.jo[3],sh.alhajbaddar@psut.edu.jo[4]

2021 12th International Conference on Information and Communication Systems (ICICS) | 978-1-6654-3351-8/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICICS52457.2021.9464539

***Abstract*—Phishing URLs mainly target individuals and/or organizations through social engineering attacks by exploiting the humans' weaknesses in information security awareness. These URLs lure online users to access fake websites, and harvest their confidential information, such as debit/credit card numbers and other sensitive information. In this work, we introduce a phishing detection technique based on URL lexical analysis and machine learning classifiers. The experiments were carried out on a dataset that originally contained 1056937 labeled URLs (phishing and legitimate). This dataset was processed to generate 22 different features that were reduced further to a smaller set using different features reduction techniques. Random Forest, Gradient Boosting, Neural Network and Support Vector Machine (SVM) classifiers were all evaluated, and results show the superiority of SVMs, which achieved the highest accuracy in detecting the analyzed URLs with a rate of 99.89%. Our approach can be incorporated within add-on/middleware features in Internet browsers for alerting online users whenever they try to access a phishing website using only its URL.**

***Keywords*—Neural Network, Random Forest, SVM, GBC, Machine Learning, URL Analysis, Phishing Detection.**

I. INTRODUCTION

With the steady acceleration in information technology, we are no longer immune to being victims of cybercrime. The use of the Internet has become essential in the modern era and an integral part of technological development, which leads to discoveries and reduction of time, effort, and costs. Nevertheless, this provides a fertile ground for piracy expansion in exploiting the weaknesses to determine private and public interests.

Although cybercrime does not differ much from traditional crimes in terms of its perpetrators' goal, because these crimes are based on unlawful targets, cybercrime has become more widespread than traditional crimes. It has become a core part in the world of digitization, as intercontinental crimes within cyberspace. Digital cybercrimes have no limits and are easy to implement.

Creating a paperless environment has become a major focus in most countries worldwide, increasing dependency on these channels. On the other hand; unprotected websites may allow fake announcement exploits under circumstances that occupy public opinion (for instance new Corona pandemic (COVID-19)). This leads the victim to a phishing website. In this context, individuals' lack of awareness in information security plays a key role in increasing the number of victims of this crime.

This work focuses on a URL phishing attack that directly depends on social engineering and specifically targets individuals by deluding them with fake websites in which the victim falls prey to these hackers [1], give his sensitive information such as e-mail account and other sensitive information related to credit card details and confidential information that may affect the reputation of the individual or institution. The various communication channels such as e-mail, social media, and forums have contributed to speed up the distribution of phishing URLs.

Various forms of attacks are categorized based on their activity, such as Phishing, Session Hijacking, Malware/Trojan, Screenlogger/Keyloggers...etc. [2]. Edge security tools are unable to deter such attacks since they target end-users rather than systems.

Phishing attacks start when the attacker copies content or page design from legitimate sites then reconstructs the scamming webpage. Once it is ready, the phisher sends a URL to the victim, luring him to fill the scam webpage with confidential information. Finally, the phisher steals the victim's information to access the original site [3]; the phishing lifecycle is illustrated in Fig. 1.

There have been many studies recently attempting to come up with a suitable solution for detecting phishing URLs. These solutions can be grouped into commonly four classifications: predefined list, signature-based, content-based, and machine learning.

In [4] [5], i.e., the Blacklist approach, every URL will be compared with a predefined list of phishing URLs. This method is inadequate to assess whether the URL is phishing or not because of the complexity of keeping the blacklist updated in real time due to the exponential rise in phishing websites. Detecting zero-day attacks is not applicable using this technique.

On the other hand, the heuristic-based approach [6], anti-phishing tools, or intrusion detection systems use a predefined signatures list of known attacks to decide whether the website is genuine or malicious based on the matching result between website heuristic patterns with the known attacks signatures. The main drawback of this technique is the weakness in detecting novel attacks such as inability to defend against attackers' different bypassing techniques; for instance, the signatures bypassing using the obfuscation technique.

The third approach, the Content-Based approach. The analysis for the website content is mainly conducted with the aid of different techniques for the textual content analysis; for example, the document frequency analysis using the TF-IDF algorithm. Then, the resulted analysis is used to decide the phishing gestures that are possibly hidden

in the given URL. Such a step is achieved depending on the text further NLP analysis [7] [8]. However, this approach requires a correct reference from which the text similarities and analysis should be compared against.

Finally, the advanced analysis achieved by the machine learning analytic forms the fourth approach for URL phishing techniques. The machine learning approach mainly depends on the ability to learn the characteristics of the websites that are listed under the phishing category, then implementing the prediction ability to distinguish the legitimate websites from the faked ones in the means of the different machine learning techniques (Prediction, Classification, Clustering … etc.) Notably, this approach is strong in terms of the analysis it provides. However, the approach adopters should decide in advance what the algorithm is. This depends on the availability of data and its nature [9]-[16].

Following the introduction, the rest of this article is organized as follows: Section II presents the related work overview for the URL's phishing classifications using machine learning. Section III introduces the dataset and presents the methodology adopted to conduct the URL phishing classification analysis. Section IV provides the experimental setup and results for the adopted methodology. Finally, section V provides the discussion of the results along with the future work and the conclusion.

II. Related Work

Nowadays, many anti-phishing techniques are proposed, but still, there is a challenge to get high accuracy detection with a low ratio of false-positive detection. In this section, a review of related work techniques and their features is presented. Moreover, a summarization of previous works and their techniques is illustrated in Table 1.

The approach proposed in [9] is a real-time detection system using URL features only; a dataset of 46,5461 URLs was used with three classifiers (J48, SVM, and Logistic Regression), which were implemented using WEKA software; the highest accuracy was 93% which was gained by J48 classifier. Authors et al. [10] implemented a middleware system to detect phishing websites. Multiple algorithms, including Random Forest, SVM, and K-Nearest Neighbor (KNN); a dataset of 11055 URLs were collected from UCI and narrowed down to contain 22 features, the highest accuracy (96%) was obtained using RF algorithm.

Another model proposed by the authors in [11] using a URL identification strategy utilizing the Random Forest algorithm. A dataset was gathered from PISHTANK [12]; only 8 out of 30 features were used for analysis. Finally, an accuracy of 95% was achieved by this model. Where authors in [12] proposed a system PHISH-SAFE using SVM and Naïve Bayes (NB) classifiers; the results show the highest accuracy 90% with the SVM.

From another perspective, authors in [13] proposed a technique through content analysis and URL features extraction. Artificial Neural Network, Decision Tree, Support Vector Machine, K-Nearest Neighbor and Naive Bayes algorithms were used in this approach. The highest accuracy (96.01%) was obtained using Artificial Neural Network algorithm.

Authors et al. [14] used random forest algorithm and compared the result with (Logistic Regression, J48, and Naïve Bayes) algorithms. Random Forest algorithm gained the best result with accuracy of (86.9%). Where authors in [15] proposed a new design called Extreme Learning Machine (ELM) based on the RF algorithm using 30 URL features; ELM detecting accuracy was 95.34%.

A system called PhishStorm was implemented in [16] to detect phishing URLs based on 12 features by utilizing supervised classifiers. The accuracy gained by this system was (94.91%).

According to the previous works, applying machine learning approaches on URL lexical features will gain a high accuracy in malicious / phishing URLs detection. Also, as URLs have been employed with attackers, a wide range of phishing attacks could be detected by those approaches. However, the accuracy achieved by this work outperformed the others in terms of accuracy.

TABLE 1
SUMMARIZATION OF PREVIOUS WORKS AND THEIR TECHNIQUES

Author	Used Algorithms	Accuracy
[9]	J48, SVM and LR	93% using J48
[10]	RF, SVM and kNN	96% using RF
[11]	RF	95%
[13]	SVM and NB	90% using SVM
[14]	DT, ANN, NB, SVM and kNN	96% using ANN
[15]	RF, J48, NB and LR	86.9% using RF
[16]	RF	95.34%
[17]	Supervised Classifiers	94.91%

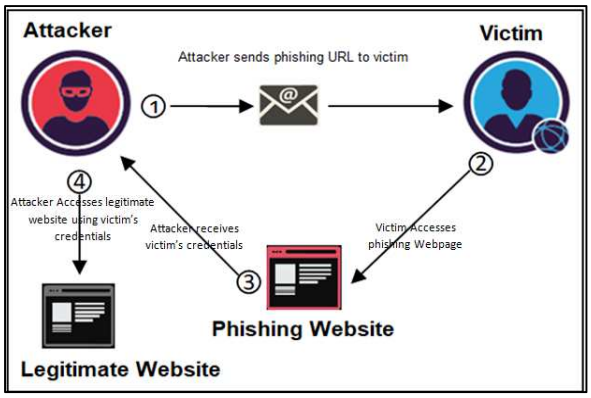

Fig 1. Website Phishing Lifecycle

III. METHODOLOGY

Fig. 2 describes the high-level methodology and the following steps in this work. Also, for each step in the figure, the following subsections explain the details.

A. Dataset

Firstly, we used the dataset published by the author in [17]. According to the author, the dataset contains only labeled URLs that are collected from many different sources. It originally contains 1056937 URLs that are unprocessed. Besides, the labels are encoded with 0 and 1 that represent the “Not-Malicious” and “Malicious” classes respectively.

B. Data Preprocessing

Data preprocessing starts with checking the duplicate URLs listed in the dataset. Almost 14,786 duplicate records were found and dropped. Next, a check for missing values

was conducted, but no missed values were found to handle.

C. Feature Extraction

The steps followed in engineering the given URLs dataset's features mainly depend on analyzing the URLs lexically. In other words, the URLs parsed and analyzed to generate further features that reflect the nature of the URLs contained in the given dataset. Accordingly, the features generated from this step resulted in 22 features, excluding the label and the URL itself. Such features are as the following:

1. URL Length: This feature represents the URL length in characters.
2. Hostname Length: This feature represents the hostname length included in the URL.
3. Dots Number: This feature represents the number of dots in the given URL.
4. URL/Domain/Path Tokens Count: Those three features represent the tokens' counts represented in the URL, the Domain, and the path.
5. URL/Domain/Path Tokens Average Length: Those three features represent the average length of the tokens represented in the URL, the Domain, and the path. Such an average is calculated by finding the tokens' length sum divided by the number of tokens.
6. URL/Domain/Path Largest Token Length: Those three features represent the largest token founded in the URL, the Domain, and the path.
7. Secure Tokens Count: This feature represents the number of sensitive keywords founded in the given URL. Such sensitive keywords are (submit, login, confirm, banking, account, secure, eBay, sign in …etc).
8. IP Address: This binary feature check if the given URL is having an IP address or not.
9. @ Sign: This binary feature checks whether the given URL is having a '@' sign or not.
10. URL Depth: This feature counts the number of the forward slashes presented in the given URL.
11. URL Redirection: This binary feature checks if the given URL is having double forward slashes ('//'), which aid in deciding if the given URL is having redirection or not.
12. URL HTTP Protocol: This binary feature checks if the given URL contains the secure protocol, i.e., the HTTPS or not.
13. URL Shortening: This binary feature checks if some shortening service produces the given URL or not (such as AdFly [18]).
14. URL Executables: This binary feature checks if the given URL contains executable resources with the extension ".exe".

D. Features Selection and Reduction

Exploring the data also took place for the produced features set. Firstly, the data distribution among the given classes is discovered, as depicted in Figures 3 and 4. The dataset appeared to have not balanced classes, i.e., imbalance classes' problem [19]. Therefore, addressing this issue is required in order to guarantee correct data modeling results.

Accordingly, we selected the randomized under-sampling technique to solve the imbalanced classes' problem. This technique uses the majority class to pick sample instances randomly with or without replacement for the sake of deleting them in order to make a balanced dataset. In this sense, the resulted instances number for each class founded to be 25212. While the original numbers for class 0 are 600078, and for class 1 is 25212.

Two approaches were considered, starting with features reduction; the first of which used a low amount of variance to detect weak features and hence prune them. The second approach utilized the multicollinearity between the features, where mathematical and statistical relationships were discovered using the Spearman Correlation Coefficient.

In this view, the Multi-collinearity between independent features were discovered and removed [20]. This should yield better modeling results. Accordingly, the correlation matrix was founded for the generated features. Next, highly correlated features were dropped from the features set.

On the other hand, features with low amounts of variance should be dropped, because they have less prediction power [21]. Accordingly, one feature was found with a low amount variance and dropped (IP Address).

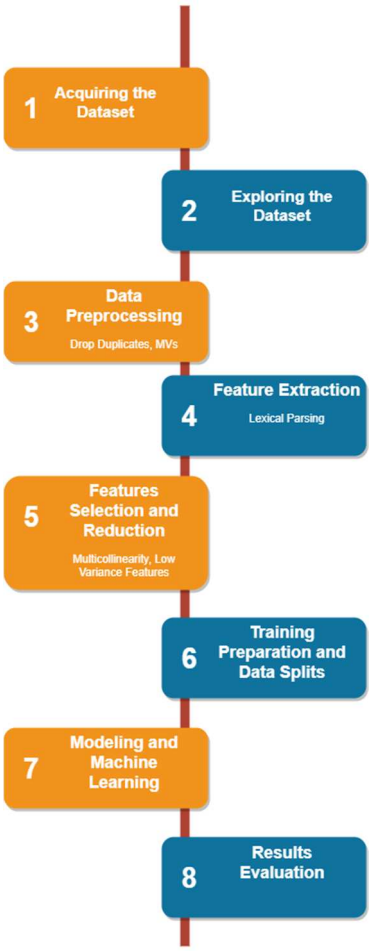

Fig 2. Work Methodology

The final selected features are shown in the correlation matrix depicted in Fig. 3.

E. Dataset Training and Testing Splits

A dataset was prepared for the training and testing using adopted machine-learning approach, where 60% of its instances were used for training, and the remaining 40% for testing.

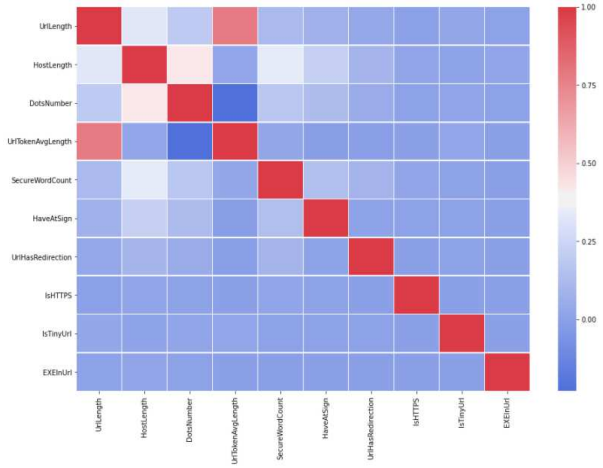

Fig 3. Reduced Features Set – After Multi-Collinearity Removal

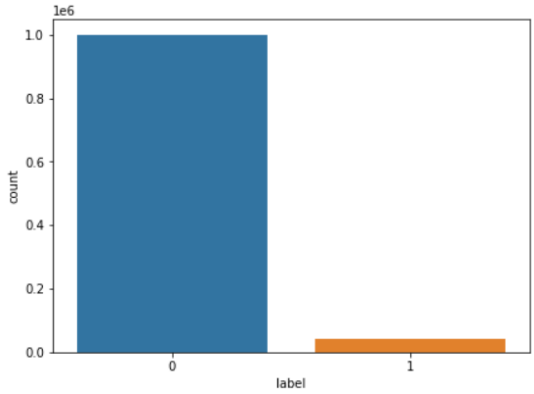

Fig 4. Imbalanced Data Problem

IV. EXPERIMENTS

A. Gradient Boosting Classifiers

In the Gradient Boosting Classifier (GBC), a group of several weak learning models are combined to produce a strong classifier (i.e., Ensemble Learning) [23]. Table 2 shows the results achieved for the GBC classification experiment. Moreover, Fig. 5 summaries the resulting (confusion) matrix.

B. Support Vector Machine

Support Vector Machine (SVM) is a supervised learning approach with the classified learning ability. It is designed well for doing either classification or regression in machine learning [22]. Table 2 shows the results achieved for the SVM classification experiment conducted. Moreover, Fig. 6 summaries the resulted evaluation matrix.

C. Random Forest

Random Forests [24], another ensemble-learning algorithm, was adopted to conduct the binary classification process on the URL dataset at hand. The result achieved in this experiment is illustrated in Table 2. Moreover, Fig. 7 summaries the resulting (Confusion) matrix.

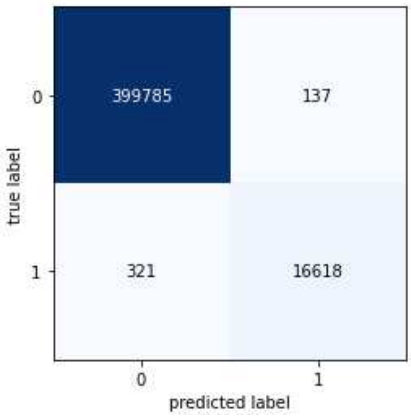

Fig 5. Imbalanced Data Problem

D. Neural Network (Adam)

Furthermore, neural network modeling is also included in the experiments. Neural network architecture is selected according to a stochastic search from 360 different models. Those are trained and evaluated with the Adam optimization algorithm [25]. Table 3 shows the results achieved for the neural networks' top 5 classifiers with their architectures.

E. Evaluation Metrics

Table 4 shows the evaluation criteria of the model during the training process. Rows denote the predicted class, while the actual class is denoted by the column. A typical evaluation matrix can be depicted from this table where TP (true positive) and TN (true negative) values represent the number of items that are properly classified. While, FP (false positive) and FN (false negative) values represent the number of misclassified items. By using the evaluation matrix, equations (1) to (3) can be calculated easily.

In this context, the precision (1) expresses the measure of actual positive items that correctly identified out of all the positive items; recall (2) provides an indicator for finding the relevant items. Finally, F-Measure (3) can be used as a harmonic mean between recall and precision. However, many other measures still exist. But they all can be used to calculate the classification and hence judging the quality of the model in the classifications processes [26].

$$\text{precision (p)} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad (1)$$

s

$$\text{recall (r)} = \frac{\text{TP}}{\text{TP} + \text{FN}} \qquad (2)$$

$$sF - meature = \frac{2 \text{ x p} * \text{r}}{\text{p} + \text{r}} \qquad (3)$$

F. Experimental Machine

All previous analysis and experiments were carried out on computer machine that has Intel core i9 CPU and 64GB of RAM.

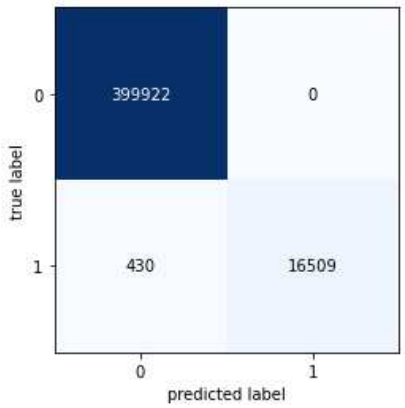

Fig 6. Confusion Matrix - SVM

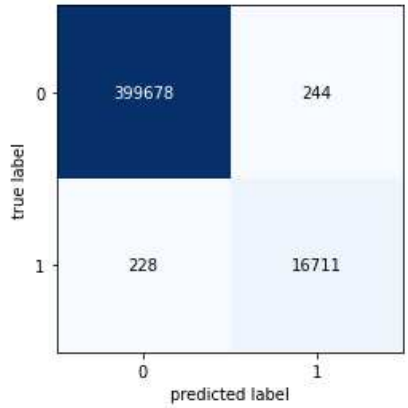

Fig 7. Confusion Matrix - RF

V. Results and Discussion

The experiments show high results and the effectiveness of the approach adopted in handling the imbalanced dataset presented. Moreover, no over fitting for the data trained and tested was presented in all experiments conducted.

In general, the results showed that SVM outperforms the other classifiers as its accuracy achieved 99.896%. Nevertheless, the other classifiers also achieved reasonably high accuracies. Also, the neural network achieved the lowest results among all the other classifiers, reaching an accuracy rate of almost (97%) using two hidden layers architecture.

The accuracy result, which measures the percentage of true decisions among all tested items, is calculated using the following equation (4):

$$Accuracy = \frac{TN+TP}{\text{TN+TP+FN+FP}} \times 100\% \qquad (4)$$

VI. Conclusion

In this paper, the URLs' lexical analysis approach was applied, where different machine learning models were trained and tested on several features sets. This approach seems to provide valuable benefits for phishing attacks detection and prevention, and only depends on the URLs included in the web requests' headers. Moreover, machine learning techniques also prove their robustness in the security domain, as reflected by the results illustrated in this work.

In the future, the work shall continue to dig deeper into involving the web content features along with the real time learning capabilities. This would help further in designing strong preventive security appliances that can learn and work simultaneously.

TABLE 3
NN (ADAM) EXPERIMENT RESULT

#	Accuracy		NN Structure
	Training	Testing	
1	0.964492	0.970380	(4 X 17)
2	0.964353	0.964984	(14 X 4)
3	0.962155	0.964962	(4 X 2)
4	0.965312	0.964588	(10 X 13)
5	0.957934	0.963909	(8X 10)

TABLE 4
EVALUATION (CONFUSION) MATRIX

	Real benign (Actual Positive)	Real malicious (Actual Negative)
Predicted (Benign)	TP	FP
Predicted (Malicious)	FN	TN

TABLE 2
SVM, GBC, RF EXPERIMENT RESULTS

Classification	Label	Precision		Recall		F1-score		FPR	Accuracy	
		Training	Testing	Training	Testing	Training	Testing	Testing	Training	Testing
SVM	0	0.97	1.00	1.00	1.00	0.99	1.00	0	0.98609	0.99896
	1	1.00	1.00	0.97	0.97	0.99	0.99			
GBC	0	0.98	1.00	1.00	1.00	0.99	1.00	0.00817	0.98937	0.99890
	1	1.00	0.99	0.98	.0.98	0.99	0.99			
RF	0	0.99	1.00	1.00	1.00	0.99	1.00	0.0143	0.99260	0.99886
	1	1.00	0.99	0.99	0.99	0.99	0.99			

Bibliography

[1] A. Almomani, B. B. Gupta, S. Atawneh, A. Meulenberg and E.Almomani, "A Survey of Phishing Email Filtering Techniques," in *in IEEE Communications Surveys & Tutorials, vol. 15, no. 4, pp. 2070-2090, Fourth Quarter 2013, doi: 10.1109/SURV.2013.030713.00020*.

[2] G. J. W. Kathrine, P. M. Praise, A. A. Rose and E. C. Kalaivani, "Variants of phishing attacks and their detection techniques," in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2019, pp. 255-259, doi: 10.1109/ICOEI.2019.8862697*, 2019.

[3] Jain, A. &. Gupta and B. B., "Phishing Detection: Analysis of Visual Similarity Based Approaches," in *Security and Communication Networks. 2017. 1-20. 10.1155/2017/5421046.*, 2017.

[4] P. Prakash, M. Kumar, R. R. Kompella and M. Gupta, "PhishNet: Predictive Blacklisting to Detect Phishing Attacks," in *2010 Proceedings IEEE INFOCOM, San Diego, CA, 2010, pp. 1-5, doi: 10.1109/INFCOM.2010.5462216*.

[5] J. Kang and D. Lee, "Advanced White List Approach for Preventing Access to Phishing Sites," in *2007 International Conference on Convergence Information Technology (ICCIT 2007), Gyeongju, 2007, pp. 491-496, doi: 10.1109/ICCIT.2007.50*.

[6] 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), Ho Chi Minh City, 2013, pp. 597-602, doi: 10.1109/ATC.2013.6698185*.

[7] Y. Zhang, J. Hong and L. Cranor, "Cantina: a content-based approach to detecting phishing web sites," in *International World Wide Web Conference, WWW 2007, May 8–12, 2007, Banff, Alberta, Canada.*.

[8] S. Afroz and R. Greenstadt, "PhishZoo: Detecting Phishing Websites by Looking at Them," in *2011 IEEE Fifth International Conference on Semantic Computing, Palo Alto, CA, 2011, pp. 368-375, doi: 10.1109/ICSC.2011.52*.

[9] A. Y. Daeef, R. B. Ahmad, Y. Yacob and N. Y. Phing, "Wide scope and fast websites phishing detection using URLs lexical features," in *2016 3rd International Conference on Electronic Design (ICED), Phuket, 2016, pp. 410-415, doi: 10.1109/ICED.2016.7804679*.

[10] A. Desai, J. Jatakia, R. Naik and N. Raul, "Malicious web content detection using machine leaning," in *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, 2017, pp. 1432-1436, doi: 10.1109/RTEICT.2017.8256834*.

[11] S. Parekh, D. Parikh, S. Kotak and S. Sankhe, "A New Method for Detection of Phishing Websites: URL Detection," in *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, 2018, pp. 949-952, doi: 10.1109/ICICCT.2018.8473085*.

[12] Jain A.K. and Gupta B.B., " PHISH-SAFE: URL Features-Based Phishing Detection System Using Machine Learning," in *Bokhari M., Agrawal N., Saini D. (eds) Cyber Security. Advances in Intelligent Systems and Computing, vol 729. Springer, Singapore*, 2018.

[13] Sirageldin A., Baharudin B.B. and Jung L.T, "Malicious Web Page Detection: A Machine Learning Approach," in *Jeong H., S. Obaidat M., Yen N., Park J. (eds) Advances in Computer Science and its Applications. Lecture Notes in Electrical Engineering, vol 279. Springer, Berlin, Heidelberg*, 2014.

[14] 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), London, 2017, pp. 1-4, doi: 10.1109/CyberSA.2017.8073403*.

[15] Sönmez Y.,Tuncer T., Gökal H. and Avc E., "Phishing web sites features classification based on extreme learning machine," in *2018 6th International Symposium on Digital Forensic and Security (ISDFS), Antalya, 2018, pp. 1-5, doi: 10.1109/ISDFS.2018.8355342*.

[16] S. Marchal, J. François, R. State and T. Engel, "PhishStorm: Detecting Phishing With Streaming Analytics," in *IEEE Transactions on Network and Service Management, vol. 11, no. 4, pp. 458-471, Dec. 2014, doi: 10.1109/TNSM.2014.2377295*.

[17] RLILOJR, "rlilojr/Detecting-Malicious-URL-Machine-Learning," [Online]. Available: https://github.com/rlilojr/Detecting-Malicious-URL-Machine-Learning. [Accessed 14 November 2020].

[18] Adfly, "AdFly - The URL shortener service that pays you! Earn money for every visitor to your links.," [Online]. Available: https://adf.ly/. [Accessed 15 January 2021].

[19] Kotsiantis, Sotiris, Dimitris Kanellopoulos and Panayiotis Pintelas, "Handling imbalanced datasets: A review," in *International Transactions on Computer Science and Engineering 30.1 (2006): 25-36*.

[20] Katrutsa, Alexandr and Vadim Strijov, "Comprehensive study of feature selection methods to solve multicollinearity problem according to evaluation criteria," in *Expert Systems with Applications 76 (2017): 1-11*.

[21] P. e. a. Mehta, "A high-bias, low-variance introduction to machine learning for physicists," in *Physics reports 810 (2019): 1-124*.

[22] Cervantes, Jair and et al., "A comprehensive survey on support vector machine classification: Applications, challenges and trends," in *Neurocomputing 408 (2020): 189-215*.

[23] Sagi, Omer and Lior Rokach., "Ensemble learning: A survey," in *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 8.4 (2018): e1249* .

[24] Resende, Paulo Angelo Alves and André Costa Drummond, "A survey of random forest based methods for intrusion detection systems," in *ACM Computing Surveys (CSUR) 51.3 (2018): 1-36*.

[25] Reddi, Sashank J., Satyen Kale and Sanjiv Kumar, "On the convergence of adam and beyond," in *arXiv preprint arXiv:1904.09237 (2019)*.

[26] Hossin m. and Sulaiman m.n., "a review on evaluation metrics for data classification evaluations," in *International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.5, No.2, March 2015*.