Impact of Current Phishing Strategies in Machine Learning Models for Phishing Detection

Impact of current phishing strategies in machine learning models for phishing detection

M. Sánchez-Paniagua[1,2], E. Fidalgo[1,2], V. González-Castro[1,2], and E. Alegre[1,2]

[1] Department of Electrical, Systems and Automatics Engineering, University of León, Spain,
[2] Researcher at INCIBE (Spanish National Institute of Cybersecurity), León, Spain
{msancp,eduardo.fidalgo,victor.gonzalez,enrique.alegre}@unileon.es
http://gvis.unileon.es//

Abstract. Phishing is one of the most widespread attacks based on social engineering. The detection of Phishing using Machine Learning approaches is more robust than the blacklist-based ones, which need regular reports and updates. However, the current Supervised Learning approaches also have some drawbacks as the datasets used for creating the models.These datasets only have the landing page of legitimate domains and they do not include the login forms from the websites, which is the most common situation in a real case of Phishing. As we show in this work, when a model is trained with old datasets, the performance in today's Phishing attacks decreases meaningfully, especially when it is tested using login pages.

In this paper, we demonstrate that a machine learning model trained with datasets collected some years ago, could have high performance when tested with the same outdated datasets, but its performance decreases notably with current datasets, using in both cases the same features. We also demonstrate that, among the commonly applied machine learning algorithms, SVM is the most resilient to the new strategies used by the current phishing attacks.

To prove these statements, we created a new dataset, Phishing Index Login URL dataset (PILU-60K), containing 60K URLs from legitimate index and login URLs, together with Phishing samples. We evaluated several machine learning methods with the known datasets PWD2016, Ebbu2017 and also with two subsets of PILU, PIU-40K and PLU-40K, which contains only index pages and only login pages respectively, showing that the accuracy decreases remarkably. We also found that Random Forest is the recommended approach among all the evaluated methods with the newly created dataset.

Keywords: Phishing Detection, URL, Machine Learning, NLP

1 Introduction

In the last years, phishing has become one of the most frequent cyber-attacks on the internet [1], boosted by the growing use of socio-technical strategies related to persuasion principles and also because spam emails lead into phishing websites which tries to steal user's credentials mainly from services like webmail, payment platforms or e-commerce [2]. According to the Anti-Phishing Working Group [3] phishing attacks on 3rd quarter of 2019 have increased up to $266,387$ websites detected, 68% of them are hosted under https, transmitting security to users while stealing their data. The only barriers standing between users and phishing attacks are the blacklist based systems like Google SafeBrowsing [3], PhishTank [4] and SmartScreen [5].

Blacklists detect phishing URLs that have been previously reported [5]. However, this approach presents two major issues: (i) they need to be updated continuously with newer data and (ii) they are not able to detect new phishing URLs. Because of these limitations, users reach the login form and could introduce their credentials, sending them straight to a server under criminal's control.

Due to the high number of phishing attacks and the low capabilities of blacklists based systems to detecting unreported phishing websites, researchers have implemented several methods based on machine learning to fight new phishing attacks [6] [7]. URL datasets collected for this purpose use the land page URLs from well-known websites as legitimate ones. However, this approach does not exactly correspond with the real world problem, where it is necessary to determine if a *login form* of a website is legitimate or phishing.

Attackers evolve their methods and change the used URLs [18] with the time, decreasing the performance of models trained with outdated datasets. APWG [4] reported that in the third quarter of 2017, less than 25% of phishing websites were under HTTPS while in 2019 up to 70% of phishing websites were under HTTPS [3]. Most of the works rely on the HTTPS feature, so their performance will decrease with recent phishing samples, as we will demonstrate later.

We discovered that recent machine learning proposals obtain high accuracy using outdated datasets, that typically are from 2009 to 2017. For that reason, we will recommend at the end of this work that a Phishing detector which intends to be used in a real situation should be trained with *legitimate recent login* websites instead of landing pages.

In this paper, we introduce Phishing Index Login URL (PILU-60K), a dataset with URLs from legitimate login websites that we consider a more representative real-world scenario for Phishing detection.

We obtain a baseline for PILU-60K computing the 38 features proposed by Sahingoz et al. [8], extracted from the URLs, and training five different classifiers. Using the same pipeline, we question how models withstand the passage of time. We evaluate the models trained with *old datasets* against the test data of more recent ones, indicating which one is the classifier that generalizes better with newer and unknown Phishing samples. Finally, we make a recommendation about what classifier and parameter configuration perform the best with the 38 URL features [8] mentioned earlier.

[3] https://safebrowsing.google.com/

[4] https://www.phishtank.com/

[5] https://bit.ly/2OJDYBS

The organization of the paper is as follows. Section 2 presents a review of the literature and related works. Then, in Section 3 we describe the used features and metrics, Section 4 describes the proposed dataset and its content. Experiments procedures are covered in Section 5. Section 6 discusses the training and test results from the experiments. Section 7 closes the paper by explaining the main contributions, the recommendations and outlining future work.

2 State of the art

In the literature, researchers have focused on phishing detection following two approaches, based on *Lists* or *Machine Learning*.

2.1 List-based

List-based approaches are widely known on phishing detection methods **??** [11] [10]. They can be whitelists or blacklists depending if the list stores legitimate URLs or phishing ones, respectively. Jain and Gupta [10] developed a whitelist-based system which blocks all websites which are not on that list.

Blacklist methods like Google Safe Browse of PhishNet [11] are the most common since they provide a zero false-positive rate, i.e. no legitimate website is classified as phishing. However, an Attacker making small changes on an URL can bypass the blacklist approach since, usually, new URLs are not registered with the necessary frequency. Furthermore, the list-based methods are not a robust solution due to the high number of new phishing websites and their short life: e.g. an average of 61 lifetime hours for non-DNS attacks [12].

2.2 Machine learning

A machine learning approach uses a number of features extracted from a website to create a model that will determine if a determined website is phishing. Based on the source of the features, the machine learning approaches could be URL-based or content-based, but usually, the last one also includes URL features.

URL based Moghimi and Vorjani [13] proposed a system independent from third services. They defined two groups of features. On the one hand, legacy features include if a website was under HTTPS, black-list words, among others. On the other hand, the group with newly proposed features, including typo-squatting detection. They obtained a 98.65% accuracy using SVM.

Shirazi et al. [14] presented a phishing detection based only on domain name features, avoiding URL dataset bias and obtaining a 97.74% accuracy with seven features.

Buber et al. [15] implemented a system composed by 209 word vector obtained from the tool "StringToWordVector" from Weka [6], and 17 NLP (Natural Language Processing) features, obtaining 97.20% accuracy on Weka's RFC (Random Forest Classifier).

[6] https://www.cs.waikato.ac.nz/ml/weka/

Sahingoz et al. [8] defined three different features set: Word vectors, NLP and a hybrid set. They obtained a 97.98% accuracy on RF (Random Forest) using only the 38 NLP features. We used those NLP features since they reported state-of-the-art performance in the last studies.

Content-based and hybrid One of the first Content-based works was CANTINA [16], which consist of a heuristic system based on TF-IDF. CANTINA extracts a sign of a document with five words, and introduced them into Google search engine. If domain is within the n first results, the page will be legitimate; either way, it will be phishing. They obtained an accuracy of 95% with a threshold $n = 30$. Due to the use of external services like WHOIS [7] and the high false-positive rate, authors proposed CANTINA+ [17], that achieved a 99.61% F1-Score including two filters: (i) comparison of hashed HTML tags with known phishing structures and (ii) the discard of websites with no form

Adebowale et al. [18] created a browser extension to protect users by extracting features from the URL, the source code, the images and third-party services like WHOIS. Those features were introduce into an ANFIS (Adaptive Neuro-Fuzzy Inference System) and combined with SIFT (Scale-Invariant Feature Transform), obtaining an accuracy of 98.30% on Rami et al. dataset [19], collected on 2015. Rao and Pais [20] used features from the URL, the source code like hyperlinks and 3rd parties like WHOIS, the domain age and the page rank on Alexa, reaching 99.31% accuracy with Random Forest. Li et al. [21] proposed a stacking model using URL and HTML features. For the URL features, they used some of the legacy features combined with sensitive vocabulary words among others. Combining this URL features with the HTML they reached 97.30% accuracy with a staking model composed by GBDT, XGBoost and lightGBM.

3 Methodology

First, we run the feature extraction on the datasets with Sahingoz et al. [8] descriptors. The feature file is introduced into a Python3 script which uses scikit learn to split the dataset, to train, to test the models and obtain the results measured with both accuracy and F1-Score. Last, we compare the results between them.

In this work, we have extracted from each URL the 38 descriptors proposed by Sahingoz et al. [8], comprising URL rules and NLP features: the number of symbols in the URL ’/’, ’-’, ’.’, ’@’, ’?’, ”, ’=’ y ’_’, digits in the domain, the subdomain and in the path, lengths of its different parts, subdomain level, domain randomness, known TLD, www or com on other places different than the TLD, words metrics like maximum, minimum, average, standard deviation, number of words, compound words, words equals or similar to a famous brand

[7] https://www.whois.net/

or a keyword like ’secure’ or ’login’, consecutive characters in the URL and punycode.

Regarding the model’s evaluation, we used the data of the oldest dataset included in this study, i.e. PWD2016. We trained five models with the five most-used classifiers in the literature [8] [13] [20] : Random Forest (RF), Support Vector Machines (SVM), k-Nearest Neighbours (kNN), Naïve Bayes (NB) and Logistic Regression (LR). We empirically assigned the parameters that returned the best accuracy. We evaluated the five trained models on PWD2016 dataset against a test set of newer datasets, i.e. PWD2016, 1M-PD and PIU-40K. To generate the baseline for our custom dataset, PILU-60K, we use the same five algorithms, returning the averaged result of 10-fold cross-validation.

We report the results using the accuracy (Eq.1) as the main metric due to the used of balanced datasets and its common use on phishing detection works [8] [18] [20], together with the F1-Score (Eq.2).

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{1}$$

$$F1 - Score = 2 * \frac{Precision * Sensitivity}{Precision + Sensitivity} \tag{2}$$

4 The dataset: Phishing Index Login URL

Most of the reported phishing websites try to steal user data through login forms, not from their landing page. We create Phishing Index Login URL (PILU-60K) dataset to provide researchers with an updated phishing and legitimate URL dataset that includes legitimate login URLs. We believe that PILU-60K adequates to the underlying problem: differentiate between legitimate login forms and the phishing ones 1.

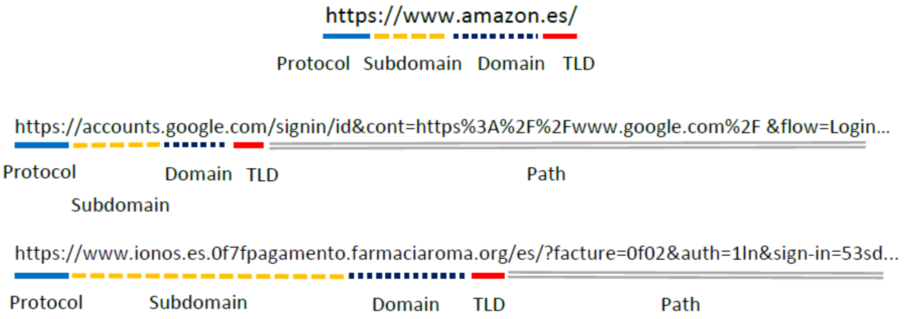
https://www.amazon.es/
Protocol
Subdomain
Domain
TLD
https://accounts.google.com/signin/id&cont=https%3A%2F%2Fwww.google.com%2F &flow=Login...
Protocol
Domain
TLD
Path
Subdomain
https://www.ionos.es.0f7fpagamento.farmaciaroma.org/es/?facture=0f02&auth=1ln&sign-in=53sd...
Protocol
Subdomain
Domain
TLD
Path

Fig. 1. Types of URLs in PILU-60K and its parts. A land page URL (up), a login page URL (middle) and a phishing URL (down). The variation between a legitimate land page and a phishing one is minimum

Legitimate URLs were taken from the Top Million Quantcast [8], which provides the most visited domains for the United States. We could not use the list as provided since only contain the name of the domains, so we revisited them all to extract the whole URL.

To reach the login from a website, we used Selenium web driver [9] combined with Python, and checked buttons or links that could lead us into the login form web page. Once we found the presumptive login, we inspect if the form had a password field, if so we added it to the dataset.

We took verified Phishing URLs at Phishtank[10], from November 2019 to January 2020.

PILU-60K is composed of 60K samples, divided into three groups, represented in Figure 1[11]: 20K legitimate index samples, 20K legitimate login samples and 20K phishing. It is worth to highlight two details about the samples of PILU-60K, which makes it a challenging dataset for Phishing detection. On the one hand, 22% of the legitimate sign-in forms URLs do not have a path, i.e. login forms are on the landing pages, matching its URL structure with the index samples. On the other one, 16% of the phishing samples do not have path, so they will also match with the legitimate index samples, increasing the challenge of classification, even for skilled humans.

5 Experimentation

5.1 Datasets

We evaluated the five trained models with the test set of Phishing Websites Dataset 2016 (PWE2016) [22], 1M-PD (1 Million Phishing Dataset) from 2017 [23], Ebbu2017 from 2017 [8] and two subsets of PILU-60K: (i) PIU-40K, containing 20K legitimate index URLs and 20K phishing URLs collected on 2020 and (ii) PLU-40K, which contains 20K legitimate login URLs and the 20K phishing URLs. Table 1 shows the distribution of samples structure within the datasets.

On the one hand, most of the legitimate URLs on PWD2016, 1M-PD and the subset PIU-40K do not have paths, since they only have index URLs. On the other hand, Ebbu2017 was created from URLs within legitimate websites so most of them have a path. Finally, PLU-40K was created with login URLs so they are expected to have a path.

In our experimentation, we also checked the capability of a model trained with legitimate index URLs to classify login URLs. The results obtained would help to decide if those models can be used on real world implementations. Finally, we examined if models trained with *recent* data (2019−2020) kept their performance when classifying *outdated* samples (< 2019).

[8] https://www.quantcast.com/products/measure-audience-insights/?redir=topsites

[9] https://selenium.dev/projects/

[10] a verified URL on Phishtank needs five people to visit the URL and vote to be real phishing. This increases the reliability of these samples

[11] URL to be added under paper acceptance

Table 1. Phishing URLs datasets analysis

	PWD2016	1M-PD	PIU-40K	Ebbu2017	PLU-40K
Legitimate URLs without a path	**10548** **(84.00%)**	**471728** **(94.35%)**	**17621** **(88.11%)**	1684 (4.62%)	4461 (22.31%)
Legitimate URLs with path	2002 (16.00%)	28272 (5.65%)	2379 (11.89%)	**34716** **(95.38%)**	**15539** **(77.69%)**
Phishing URLs	15000	500000	20000	37175	20000

5.2 Experimental Setup

The experiments have been carried out with Python3 on a PC with a $9th$ gen i3 CPU with 16GB of DDR4 RAM PC. We have used the scikit learn library [12] for data splitting - training 70%, test 30% - and algorithms evaluation.

To evaluate how the current phishing strategies affect to the performance of machine learning models, we trained the initial models with the Phishing Websites Dataset 2016 (PWE2016) [22]. Through empirical experimentation, we set the best parameters for each classifier: RF with 10 trees, kNN with $k = 1$, SVM $\gamma = 0.1$ and $C =' auto'$, LR with *LBFGS* solver and Naïve Bayes Bernoulli.

For the evaluation of login legitimate URLs, we used Ebbu2017 [8] for training the base models. We empirically set RF with 250 trees, keeping the same parameter selection as in the previous experimentation. We tested them with the PLU-40K subset which contains the legitimate login URLs collected in 2020.

Finally, in the evaluation with PILU-60K, we carried out several tests. First, we empirically obtained the best parameters for the five machine learning algorithms, that are: 350 trees for RF, $k = 3$ for kNN, $\gamma = 0.1$ and $C =' auto'$ for SVM, LBFGS solver for LR and Bernoulli for NB. Second, we obtained their results using the described features. On a hypothetical real-world implementation, models deal with login URL classification, so we tested the scenario when the models are trained with the subset PIU-40K (index pages) and takes the 20K login URLs as input. Finally, we made a throwback test, where models were trained with PIU-40K subset and tested with the old PWD2016 dataset. Results are shown in the next section in the same order.

6 Results and discussion

Table 2 shows that the accuracy of all models decreases when they are trained with old phishing attacks but tested with the new ones. On PWD2016 base models, kNN was the best one with 97.35% accuracy, but its performance was the most affected over time, decreasing to 82.85%, probably due to chose the first neighbor, $k = 1$, as parameter. On the other hand, SVM had a lower accuracy on the base dataset but prevailed the most. Using Ebbu2017 as base training dataset, results show a significant performance reduction. This could be because

[12] https://scikit-learn.org/

on Ebbu2017 legitimate URLs do not have keywords like *secure* or *login*, but on PLU-40K the most part of legitimate login URLs have those keywords, creating a bias on those descriptors.

Table 2. Phishing detection accuracy evolution over time. Training set represents 70% of the samples. PWD2016 and Ebbu2017 were evaluated with 30% of their samples to simulate their release date performance

Training set	PWD2016			Ebbu2017	
Test set	PWD2016	1M-PD	PIU-40K	Ebbu2017	PLU-40K
RF	97.31%	90.64%	86.04%	**95.85%**	67.94%
kNN	**97.35%**	87.22%	82.85%	93.77%	64.77%
SVM	95.62%	**91.91%**	**88.73%**	93.17%	**71.11%**
NB	88.97%	86.51%	85.84%	87.61%	65.30%
LR	93.46%	88.73%	85.96%	79.51%	64.42%

Table 3 shows that RF has the best performance on PILU-60K dataset. Classifying index URLs and legitimate login URLs from phishing results in an accuracy of 94.59% and 92.47%, respectively. Detecting old samples from PWD2016, RF also obtains the best performance with a 91.22% accuracy. Finally we verified that models trained with index URLs performed poorly on classifying legitimate login URLs, where LR got the best result with a low 67.60% accuracy. This might be consequence of using length features, since the model could determine that short URLs are legitimate and long ones are phishing.

Most phishing websites try to steal data from users commonly through a login form, and attackers replicate legitimate login websites. The main issue with the current state-of-the-art phishing URLs detection is that legitimate URLs on datasets do not represent those login websites but the landing page.

Table 3. URL detection performance for the proposed dataset

	PIU-40K		PLU-40K		PIU-40K throwback		PIU-40K vs login URLs
	F1	Acc.	F1	Acc.	F1	Acc.	Acc.
RF	**94.60%**	**94.59%**	**92.49%**	**92.47%**	**92.03%**	**91.22%**	54.06%
SVM	93.80%	93.85%	90.59%	90.68%	89.70%	88.66%	61.49%
kNN	92.99%	93.13%	89.40%	89.63%	86.15%	85.36%	64.28%
LR	92.45%	92.55%	85.11%	85.40%	89.02%	87.02%	**67.60%**
NB	86.63%	87.60%	72.57%	74.34%	87.25%	86.91%	58.22%

7 Conclusions and future works

On this work we tested, with different datasets, how Phishing URL detection systems withstand the passage of time and also a hypothetical real world implementation performance. Results showed that those systems lose accuracy over time because of the new strategies used by the current phishing attacks, therefore it is necessary to use recent datasets to train the models. Within this performance loss, SVM is the algorithm which performance endures better over time, dropping from 65.62% accuracy to 88.73% after four years. Models trained with index URLs showed bad performance on classifying legitimate login URLs, up to 67.60% accuracy on a LR. To provide these URLs to researchers we built PILU-60K dataset. Tests on this dataset showed that RF obtained the best result classifying the index URLs with a 94.59% accuracy and a 92.47% classifying login URLs. These results prompt that phishing detection through URLs is still feasible by using recent datasets and login URLs for training.

In future works, we will enlarge our dataset, bringing more information within the samples, like the source code of the website or a screenshot, so researchers can rely on recent data no matter what kind of features they want to use. Later on, we will endeavor on finding new features to effectively detect phishing websites, since attackers try to bypass phishing detectors changing their phishing techniques.

Acknowledgement This research was funded by the framework agreement between the University of León and INCIBE (Spanish National Cybersecurity Institute) under addendum 01.

References

1. Ferreira, A., Teles, S. (2019). Persuasion: How phishing emails can influence users and bypass security measures. International Journal of Human Computer Studies, 19–31. https://doi.org/10.1016/j.ijhcs.2018.12.004
2. Patel, P., Sarno, D. M., Lewis, J. E., Shoss, M., Neider, M. B., Bohil, C. J. (2019). Perceptual Representation of Spam and Phishing Emails. Applied Cognitive Psychology, acp.3594. https://doi.org/10.1002/acp.3594
3. Anti-Phishing Working Group. (2019). Phishing Activity Trends Report - 3Q.
4. Anti-Phishing Working Group. (2017). Phishing Activity Trends Report - 3Q.
5. Chanti, S., Chithralekha, T. (2020). Classification of Anti-phishing Solutions. SN Computer Science, 1(1). https://doi.org/10.1007/s42979-019-0011-2
6. Halgas, L., Agrafiotis, I., Nurse, J. R. C. (2019). Catching the Phish: Detecting Phishing Attacks using Recurrent Neural Networks (RNNs). (August). Retrieved from https://link.springer.com/chapter/10.1007/978-3-030-39303-8_17
7. Rao, R. S., Pais, A. R. (2019). Jail-Phish: An improved search engine based phishing detection system. Computers and Security, 83, 246–267. https://doi.org/10.1016/j.cose.2019.02.011
8. Sahingoz, O. K., Buber, E., Demir, O., Diri, B. (2019). Machine learning based phishing detection from URLs. Expert Systems with Applications, 117, 345–357. https://doi.org/10.1016/j.eswa.2018.09.029

9. Cao, Y., Han, W., Le, Y. (2008). Anti-phishing based on automated individual white-list. Proceedings of the ACM Conference on Computer and Communications Security, 51–59. https://doi.org/10.1145/1456424.1456434
10. Jain, A. K., Gupta, B. B. (2016). A novel approach to protect against phishing attacks at client side using auto-updated white-list. Eurasip Journal on Information Security, 2016. https://doi.org/10.1186/s13635-016-0034-3
11. P. Prakash, M. Kumar, R. Rao Kompella y M. Gupta (2010). PhishNet: Predictive Blacklisting to Detect Phishing Attacks. 2010 Proceedings IEEE INFOCOM, 2010.
12. Moore, T., Clayton, R. (2007). Examining the impact of website take-down on phishing. ACM International Conference Proceeding Series, 269, 1–13. https://doi.org/10.1145/1299015.1299016
13. Moghimi, M., Varjani, A. Y. (2016). New rule-based phishing detection method. Expert Systems with Applications, 53, 231–242. https://doi.org/10.1016/j.eswa.2016.01.028
14. Shirazi, H., Bezawada, B., Ray, I. (2018). Kn0w thy doma1n name: Unbiased phishing detection using domain name based features. Proceedings of ACM Symposium on Access Control, SACMAT, 69–75. https://doi.org/10.1145/3205977.3205992
15. Buber, E., Diri, B., Sahingoz, O. K. (2018). NLP Based Phishing Attack Detection from URLs. https://doi.org/10.1007/978-3-319-76348-4
16. Yue, Z., Hong, J., Cranor, L. (2007). CANTINA: A Content-Based Approach to Detecting Phishing Web Sites. ACM Transactions on Information and System Security, 14(2), 1–28. https://doi.org/10.1145/2019599.2019606
17. Xiang, G., Hong, J., Rose, C. P., Cranor, L. (2011). CANTINA+: A feature-rich machine learning framework for detecting phishing web sites. ACM Transactions on Information and System Security. https://doi.org/10.1145/2019599.2019606
18. Adebowale, M. A., Lwin, K. T., Sánchez, E., Hossain, M. A. (2019). Intelligent web-phishing detection and protection scheme using integrated features of Images, frames and text. Expert Systems with Applications, 300–313. https://doi.org/10.1016/j.eswa.2018.07.067
19. Rami, M., Thabtah, F. A. T. L.(2015). Phishing Websites Dataset Available at: (Accessed: 10 September 2016) http://eprints.hud.ac.uk/24330/9/Mohammad14JulyDS
20. Rao, R. S., Pais, A. R. (2018). Detection of phishing websites using an efficient feature-based machine learning framework. Neural Computing and Applications, 31(8), 1–23. https://doi.org/10.1007/s00521-017-3305-0
21. Li, Y., Yang, Z., Chen, X., Yuan, H., Liu, W. (2019). A stacking model using URL and HTML features for phishing webpage detection. Future Generation Computer Systems, 94, 27–39. https://doi.org/10.1016/j.future.2018.11.004
22. Chiew, K. L., Chang, E. H., Tan, C. L., Abdullah, J., Yong, K. S. C. (2018). Building Standard Offline Anti-phishing Dataset for Benchmarking. International Journal of Engineering Technology, 7(4.31), 7–14.
23. Yuan, H., Yang, Z., Chen, X., Li, Y., Liu, W. (2019). URL2Vec: URL modeling with character embeddings for fast and accurate phishing website detection. Proceedings - 8th IEEE International Conference on Big Data and Cloud Computing, 11t, 265–272. https://doi.org/10.1109/BDCloud.2018.00050