Feature Engineering Based Malicious URLs detection models using Bi-directional RNN

Jyoti Singh[1], Prof. Anjali Singh[2]
[1]Dept of CSE
[2]Assistant Professor Dept of CSE
[1, 2] Aditya College of Technology and Science, Satna, Madhya Pradesh, India.

Abstract- *Threats, including phishing, are still a widespread and growing problem in today's digital society because fraudsters use increasingly complex techniques to deceive users and gain illegal access to sensitive information, effective phishing detection systems are essential for protecting individuals and enterprises.*

In the modern digital world, phishing attacks have become a persistent threat, jeopardizing the security and privacy of individuals and organizations, demanding prompt action to avoid excessive costs imposed to individuals and businesses. The recent research in this field has shown that machine learning and deep learning are promising in designing phishing attack detection systems.

In this paper, we first propose a feature engineering approach to extract useful features from the URL and create machine learning models that effectively recognize the patterns of phishing URLs using these features with highest accuracy.

Keywords- URL, Phishing Detection, Feature Selection, Machine Learning, Metrics.

I. INTRODUCTION

According to the website Siteefy [1], there are over 1.11 billion websites in the world, and this number has increased exponentially in recent years. Approximately 252,000 new websites are created every day (please check this out). By May 9, 2023, the number of web pages will exceed 50 billion. While most websites are created with good intentions, many are malicious [2]. Malicious websites are designed to harm users in some way, such as stealing personal information or installing malware on their computers. They can be used to spread malware, phishing, spam, or cause denial of service [3]. According to a comprehensive study by google, there are approximately 12.8 million malicious websites on the internet [4]. Also, as the authors state in [5], 18.5 million web servers have faulty code. This number is constantly changing as new malicious websites are created and old ones are shut down. Since they can learn features from web data, they can extract good features without the support of manual architecture. Convolutional neural network (cnn) [6], recurrent neural network (rnn) [7] and tracking techniques are malware detection techniques. Many deep learning models are built on features extracted from web content. However, due to the nature of the website, the use of intrusion prevention devices to prevent unauthorized access, and the constant change of online threats, deep learning models are made up of large and diverse data from the content potential of the web. For difficulty. Some websites require users to log in and authenticate to access content. Accessing these websites may involve simulating user interactions, including logging in. Websites frequently change their design and layout, and regular maintenance and login script updates are required to ensure they continue to function well. Also, extracting proxy pages from web content may not be useful for resource-constrained devices such as iot devices. While content-based signatures can be used to identify various types of threats, relying on web content to identify malicious websites is inefficient and ineffective.

Url-based functionality seems to be a good option for website functionality. Many researchers have compared the performance of the models created by these two, and in all cases, url-based features always win. However, most existing studies only rely on lexical features extracted from urls. The semantic information of lexical features is limited, leading to the compact design of feature vectors. Some surveys combine URL attributes with digital certificates to improve discoverability. Malicious websites often do not have valid certificates or do not use self-signed certificates, making the verification certificate a trust value. Checking digital certificates can reveal whether a website uses encryption, which is a common practice among reputable websites. However, not all websites use digital certificates, and some may use self-signed certificates or certificates issued by lesser-known certificate authorities (cas). Extracting important and useful features from machine learning models can be challenging, and selecting the right features is critical for successful search results. Furthermore, digital certificates can be misconfigured, expire, and change frequently, leading to high levels of vulnerability. In summary, current solutions for detecting web vulnerabilities by analyzing web content often struggle due to complex feature extraction, large data texts, changing attack models, and limitations of traditional classifiers. It turns out that relying on lexical url features alone is not enough and will lead to misclassification.

1.2 Working model of malicious web-page attack

Attacks usually take the form of malicious pages, phishing, and internet viruses [8]. In malicious website attacks, attackers use camouflage technology to push malicious websites to users and lure users to malicious spam websites. Phishing is an attack in which an attacker sends fake internet links to users or sends emails to trick them into clicking on a link. As a result, users' personal information (such as passwords) can be leaked. Attackers also use scripts to write malicious code to create viruses on the network and to inject viruses into vulnerable parts of the browser. The virus is displayed immediately when the user enters the website. Malicious programs are used to add, delete and modify files on the local computer, or even to shut down the system or format the disk. Web pages containing malicious text are called Trojan horse pages. The purpose of the attack is to exploit vulnerabilities in the user's browser to be successful [9]. These attacks harm the data security of users. Blocking access to malicious pages is done by identifying malicious pages using static or dynamic methods. Static detection includes two methods: faulty link detection and static analysis based on the content of the web page. The former is done by identifying only phishing and Trojan links. The latter attempts to identify the web page source code based on the characteristics of the malware. Generally speaking, the static detection method [10] [11] uses analytical tools to analyze the static characteristics and functions of the negative code. Dynamic detection is usually based on the difference between interaction behavior that is, monitoring the status of the interaction between the browser and the web server when the user enters. If the status is abnormal, the website is considered a malicious page. Dynamic discovery is usually used in sandboxes or honeypots. The basic principle of honeypot is to use virtual machines to trick attackers into invading, and then protect the local computer by monitoring the chatting and typing behaviors of those who oppose the challenge. The function is a markov decision process (mdp), so machine learning (ml), such as decision trees, can obtain the best solution through training and testing if the decision tree is deep enough [12] [13]. Since the ml model is used to identify bad pages, both mdp and decision tree ml techniques can accurately classify web pages without the need to detect errors and correct their weights. Compared with decision trees, mdp combined with decision trees, called markov exploration trees, can represent various states of web pages based on the URL relationship between web pages, thus enabling more automatic decision making for each web page. Therefore, we propose a search method based on mdp and decision tree to improve the accuracy and efficiency in the process of web page classification. In order to ensure that users can safely browse the internet and avoid various network attacks, a way is needed to detect the vulnerabilities of web pages from a large number of web pages. Both malicious and benign pages contain related content and code snippets. In this paper, we investigate machine learning techniques to identify web pages based on key features. Especially on bad pages, we can analyze the important information of each page by using special words and finally decide whether the page is bad or not.

1.1 URL Attack Technology

Attack technology is a method or technique used by attackers to obtain illegal information or damage the underlying system. Attackers can use malicious URLs to carry out these attacks. Malicious URLs can be classified as spam, phishing, malware, or doctored URLs. Most cyber attacks occur when users click on malicious URLs. When URLs are used for purposes other than accessing legitimate Internet resources, they pose threats to information integrity, confidentiality, and availability. Different types of malicious URLs are discussed below [13].

A. Spam URL attacks: These attacks occur when spammers create web pages to trick browser engines into thinking they are legitimate when in fact they are not. Spammers hope to trick users into thinking they are legitimate through illicit promotion and drive more users to their spam websites [14]. Spammers send spam emails containing spam URLs to corrupt and infect victims' systems using spyware and adware [15].

B. Phishing URL attack: Attackers use phishing URLs to try to gain access to users' computers in order to trick users into opening fake websites and steal personal information such as credit card numbers, phone numbers, and other personal information. Non-professional users can be tricked into visiting phishing sites by adding rare misspellings to the URL, such as changing www.facebook.com to www.facebo0k.com, making it easier to profile users under [15].

C. Malware URL Attacks: These attacks often redirect users to malicious websites that install malware on the user's device, leading to illegal logging, keystroke logging, and even theft. Malware is malicious software that can steal a person's personal information and harm a computer. An example of malware is a drive-by download, which is when a user is tricked into visiting a malicious website and the malware is unintentionally downloaded [16]. Additional examples include ransomware, keyloggers, Trojans, spyware, threats, computer worms, and viruses.

D. Modified URL Attack : This type of attack redirects the user to a malicious website that has been modified by the hacker in one or more ways, such as the visual appearance or some of the website content. Hacktivists attempt to take down websites for a variety of reasons. This form occurs when an attacker finds vulnerabilities in a website and uses these vulnerabilities to compromise the website and change the content of the website without the owner's permission, known as web access [16]. The classification of malicious URL attacks by ML techniques can be binary, such as malicious or not. In contrast, the classification is not limited to any category other than more than two categories, such as benign, phishing, suspicious, malware, spam, etc. The working of URL attack model is shown below in figure 1.1.

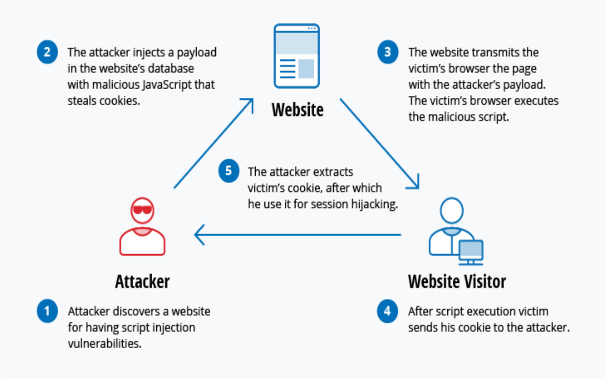

2 The attacker injects a payload in the website's database with malicious JavaScript that steals cookies.
Website
3 The website transmits the victim's browser the page with the attacker's payload. The victim's browser executes the malicious script.
5 The attacker extracts victim's cookie, after which he use it for session hijacking.
Attacker
Website Visitor
1 Attacker discovers a website for having script injection vulnerabilities.
4 After script execution victim sends his cookie to the attacker.

Figure 1.1: URL Attack Model.

II. RELATED WORK

The paper [17] presented a Factorization Machine (FM), a form of deep learning algorithm for identifying malicious URLs. This method discovers the latent relationship between lexical characteristics. To minimize the ambiguity of URL tokens, position embedding is implemented for token vectorization. It means a Temporal Convolution Network (TCN) is employed to learn the long-distance dependence between URL characters.

Precise Phishing Detection with Recurrent Convolutional Neural Networks (PDRCNN) method presented in [17], suggests a rapid approach for detecting malicious URLs that relies solely on lexical features. It turns the URL's data into a two-dimensional tensor and feeds the tensor into a newly built neural network for classification. First, extract features from the built tensor and assign all string information to each character in the URL using a bidirectional LSTM network. Second, employ CNN to automatically determine which characters are critical for detecting malicious URLs, extract the URL's major elements, and compress the collected classes into a fixed-length vector space. The PDRCNN achieves a detection accuracy of 97% on a dataset with 245,385 valid URLs.

Deep learning approaches have made great progress in detecting malicious URLs over the last few years. Many machine learning problems have been overcome, but there are still a number of major issues remaining. Massive volumes of URLs needed to be used for training to create a suitable detection method with acceptable levels of accuracy for deep learning [18], [19]. This problem becomes much worse when new URLs are available and the method need to retrained [20]. Furthermore, interpretability of method does not disclose the details and specifics of the method's prediction and feature selection, which often behave like black boxes. Interpretability can lead to some drawbacks, such as vulnerability to potential novel attacks. Due to a lack of knowledge of rules developed by machines, which prevents upgrading and optimizing the rules by the developers [21]. Moreover, the detection method's reliability and level of accuracy are entirely dependent on the quality of the dataset [22]. Lastly, an issue of note is the feature selection contradiction, with the majority of research still involving manual classification of features.

In this study [23], we introduce an innovative framework for malicious URL detection based on **predefined static feature** classification by allocating priority coefficients and feature evaluation methods. Our feature classification encompasses 42 classes, including blacklist, lexical, host-based, and content-based features. To validate our framework, we collected a dataset of **5000 real-world URLs** from prominent phishing and malware websites, namely **URLhaus and PhishTank**. We assessed our framework's performance using three supervised machine learning methods: Support Vector Machine (SVM), Random Forest (RF), and Bayesian Network (BN). The results demonstrate that our framework outperforms these methods, achieving an impressive detection accuracy of 98.95% and a precision value of 98.60%.

In this study [24], a malicious domain names detection algorithm based on lexical analysis and feature quantification is proposed. To achieve efficient and accurate detection, the method includes two phases. The first phase checks an observed domain name against a blacklist of known malicious uniform resource locator (URLs). The observed domain name is classified as being definitely malicious or potentially malicious based on its edit distances to the domain names on the blacklist. The second phase further evaluates a potential malicious domain name by its reputation value that represents its lexical feature and is calculated based on an N-gram model. The top 100,000 normal domain names in Alexa are used to obtain a whitelist substring set using the N-gram method in which each domain name excluding the top-level domain is segmented into substrings with the length of 3, 4, 5, 6 and 7. The weighted values of the substrings are calculated according to their occurrence counts in the whitelist substring set. A potential malicious domain name is segmented by the N-gram method and its reputation value is calculated based on the weighted values of its substrings. Finally, the potential malicious domain name is determined to be malicious or normal based on its **reputation value**. The effectiveness of the proposed detection method has been demonstrated by experiments on public available data.

The adoption of Quick Response (QR) codes with malicious URLs is a growing concern and is an open security issue. The existing QR link detection scanner applications mostly utilize the blacklist method to detect malicious URLs, which is not the optimal method for detecting new websites. Recently, machine learning methods have gained popularity as a means of enhancing the detection of malicious URLs. However, these methods are entirely data-dependent, and a large and updated dataset is required for the training to create an effective detection method. This research [25] proposes QsecR, a secure and privacy-friendly QR code scanner, according to a malicious URL detection framework. QsecR is an Android QR code scanner based on predefined static feature classification by employing 39 classes of blacklist, lexical, host-based, and content-based features.

In this paper [26], we compare machine learning and deep learning techniques to present a method capable of detecting phishing websites through URL analysis. In most current state-of-the-art solutions dealing with phishing detection, the legitimate class is made up of homepages without including login forms. On the contrary, we use URLs from the login page in both classes because we consider it is much more representative of a real case scenario and we demonstrate that existing techniques obtain a high false-positive rate when tested with URLs from legitimate login pages. Additionally, we use datasets from different years to show how models decrease their accuracy over time by training a base model with old datasets and testing it with recent URLs. Also, we perform a frequency analysis over current phishing domains to identify different techniques carried out by phishers in their campaigns. To prove these statements, we have created a new dataset named Phishing Index Login URL (PILU-90K), which is composed of 60K legitimate URLs, including index and login websites, and 30K phishing URLs. Finally, we present a Logistic Regression model which, combined with Term Frequency - Inverse Document Frequency (TF-IDF) feature extraction, obtains 96.50% accuracy on the introduced login URL dataset.

III. PROPOSED WORK

3.1 Proposed model Solutions

The 1,250% year-over-year increase in "very new" malicious domains is not a statistic to ignore. Attackers are using AI and other techniques to lure unsuspecting individuals into their information-stealing schemes. Organizations, in turn, must use the capabilities of artificial intelligence and machine learning to detect malicious domains faster than humans alone can. Machine learning finds more patterns of malicious behavior across every threat category, and it does it faster. Protective solutions using machine learning and Deep Learning set organizations on the path to greater security.

The architecture of proposed model using machine learning and deep learning is shown below for malicious URLs detection.

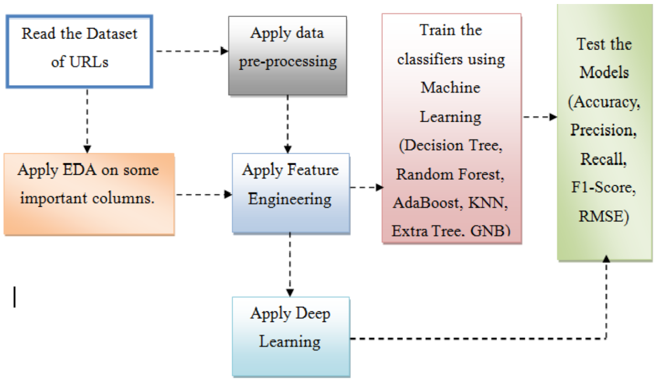

Read the Dataset of URLs
Apply data pre-processing
Train the classifiers using Machine Learning (Decision Tree, Random Forest, AdaBoost, KNN, Extra Tree, GNB)
Test the Models (Accuracy, Precision, Recall, F1-Score, RMSE)
Apply EDA on some important columns.
Apply Feature Engineering
Apply Deep Learning

Figure 3.1: Proposed model architecture.

Some of the common features that malicious URLs have are mentioned below.

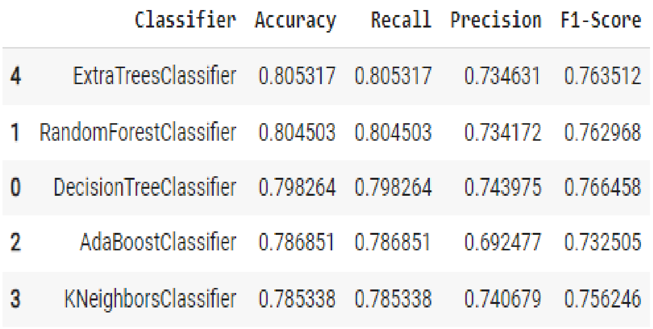
1. Malicious URLs don’t have hyphens or symbols in their domain name. So in our model, by checking special characters and symbols we can check for this malicious URL. For example www.google.com is not same as www.google-search.com.
2. Not having https in their names.
3. Missing of Legit Contact Information.
4. Websites without this important information are more likely to be fraudulent. Also, a fictional or vague address may signify a phishing site. So the “url-region” property check will help us in detecting malicious urls.
5. 5. Poor Backlink profile analysis report. A backlink is a URL that leads from one website to another. A website with many backlinks is featured on many other

pages, proving its trustworthiness. Getting the root domain will help in backlink analysis of URLs.

6. Counting of Dashes in URL link will help detecting malicious urls because they have more in numbers as compared to legitimate urls.
7. Malicious URLs generally have longest domain names. So the URL shortening services will help in detecting these features.
8. The lexical features based on the words that appear in the URLs capture the dynamic nature of the links. The static nature of the links is captured by the descriptive features, which rely on the assumption that the characteristics between legitimate and malicious links rarely varied. For instance, phishing websites sometimes utilized related symbols or letters, such as representing the lower case of letter ‘L′ with the digit ‘1′ in order to mislead the target legitimate users. Thus, the websites may have certain statistical information, such as the consecutive relationship of digits and alphabets. Using this assumption, some lexical and descriptive features may be extracted from URLs and use to train classification algorithms.

IV. RESULTS AND DISCUSSION

After applying hash encoding, the dataset is divided into train-set and test-set with test size 30 percent and train-set 70 percent.



	Classifier	Accuracy	Recall	Precision	F1-Score
4	ExtraTreesClassifier	0.805317	0.805317	0.734631	0.763512
1	RandomForestClassifier	0.804503	0.804503	0.734172	0.762968
0	DecisionTreeClassifier	0.798264	0.798264	0.743975	0.766458
2	AdaBoostClassifier	0.786851	0.786851	0.692477	0.732505
3	KNeighborsClassifier	0.785338	0.785338	0.740679	0.756246

Figure 4.1: Results Comparisons of Machine Learning Classifiers.

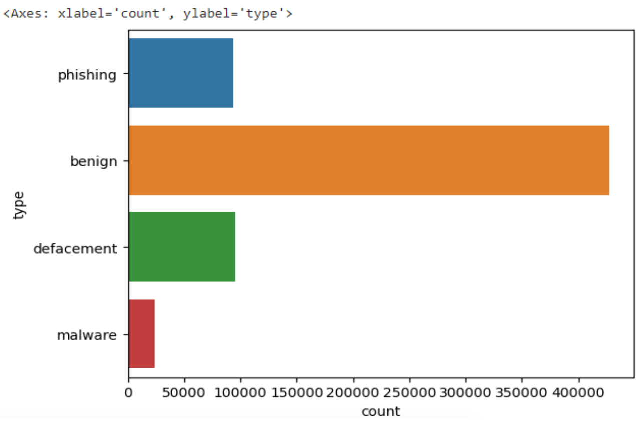
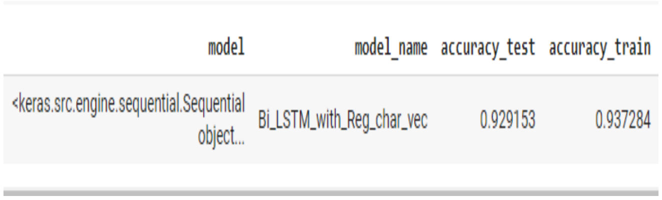
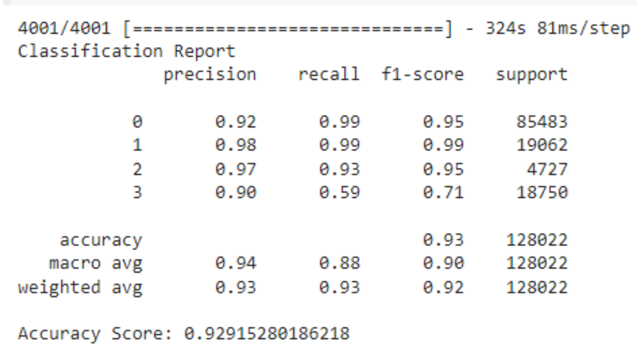


Figure 4.2: Results classification of Machine Learning Classifiers.



	model	model_name	accuracy_test	accuracy_train
	<keras.src.engine.sequential.Sequential object...	Bi_LSTM_with_Reg_char_vec	0.929153	0.937284

Figure 4.3: Accuracy of proposed Bi-LSTM model.



```
4001/4001 [==============================] - 324s 81ms/step
Classification Report
                    precision    recall  f1-score   support

           0        0.92      0.99      0.95     85483
           1        0.98      0.99      0.99     19062
           2        0.97      0.93      0.95      4727
           3        0.90      0.59      0.71     18750

    accuracy                            0.93    128022
   macro avg        0.94      0.88      0.90    128022
weighted avg        0.93      0.93      0.92    128022

Accuracy Score: 0.92915280186218
```

Figure 4.4: Classification report of proposed Bi-LSTM model.

The comparison of accuracies is shown below in table.

Model Name	**Accuracy**
KNN	78.53
AdaBoost	78.68
Decision Tree	79.82
Random forest	80.45
ExtraTree	80.53
Bi-LSTM	**92.91**

V. CONCLUSION

Machine Learning algorithms are efficient to do binary classification and to detect the malicious URLs. URL detection using a machine learning model is that it accepts the URL as user input and detects and classifies it as benign or malicious one. Model does binary classification with 99%

accuracy. This model can be used in the cyber security domain and also to avoid digital attacks by knowing the malicious and benign URLs in prior. Safety measures can be taken if the URL is found malicious.

REFERENCES

[1] NJ. (2023). How Many Websites are There in the World? Accessed: Sep. 10, 2023. [Online]. Available: https://siteefy.com/how-manywebsites- are-there/

[2] M. Liu, B. Zhang, W. Chen, and X. Zhang, ‘‘A survey of exploitation and detection methods of XSS vulnerabilities,’’ IEEE Access, vol. 7, pp. 182004–182016, 2019.

[3] J. Jang-Jaccard and S. Nepal, ‘‘A survey of emerging threats in cybersecurity,’’ J. Comput. Syst. Sci., vol. 80, no. 5, pp. 973–993, Aug. 2014, doi: 10.1016/j.jcss.2014.02.005.

[4] N. Provos, D. McNamee, P. Mavrommatis, K. Wang, and N. Modadugu, ‘‘the ghost in the browser: Analysis of web-based malware,”HotBots, vol. 7, p. 4, Apr. 2007.

[5] K. Townsend, ‘‘18.5 Million websites infected with malware at any time,’’ Wired Bus. Media, Security Week, Boston, MA, USA, Tech. Rep. Q4 2017, 2022. Accessed: Feb. 1, 2022. [Online]. Available: https://www.securityweek.com/185-million-websites-infected-malwareany-time.

[6] S. Wang, Z. Chen, Q. Yan, K. Ji, L. Peng, B. Yang, and M. Conti, ‘‘Deep and broad URL feature mining for Android malware detection,’’ Inf. Sci., vol. 513, pp. 600–613, Mar. 2020, doi: 10.1016/j.ins.2019.11.008.

[7] R. Vinayakumar, K. P. Soman, and P. Poornachandran, ‘‘Evaluating deep learning approaches to characterize and classify malicious URL’s,’’ J. Intell. Fuzzy Syst., vol. 34, no. 3, pp. 1333–1343, Mar. 2018, doi: 10.3233/JIFS-169429.

[8] I. Vayansky and S. Kumar, ‘‘Phishing—Challenges and solutions,’’ Comput. Fraud Secur., vol. 2018, no. 1, pp. 15–20, Jan. 2018.

[9] Y. Cohen, D. Hendler, and A. Rubin, ‘‘Detection of malicious webmail attachments based on propagation patterns,’’ Knowl.-Based Syst., vol. 141, pp. 67–79, Feb. 2018.

[10] M. Aljabri and S. Mirza, “Phishing attacks detection using machine learning and deep learning models," in Proc. 7th Int. Conf. Data Sci. Mach. Learn. Appl. (CDMA), Mar. 2022, pp. 175_180, doi:10.1109/cdma54072.2022.00034.

[11] M. Aljabri, F. Alhaidari, R. M. A. Mohammad, S. Mirza, D. H. Alhamed, H. S. Altamimi, and S. M. B. Chrouf, “An assessment of lexical, network, and content-based features for detecting malicious URLs using machine learning and deep learning models," Comput. Intell. Neurosci. vol. 2022, pp. 1_14, Aug. 2022, doi: 10.1155/2022/3241216.

[12] Extracting Feature Vectors from URL Strings for Malicious URL Detection. Accessed: Dec. 3, 2021. [Online]. Available: https://towardsdatascience.com/extracting-feature-vectors-from-urlstrings-for-malicious-url-detection-cbafc24737a.

[13] T. Manyumwa, P. F. Chapita, H. Wu, and S. Ji, “Towards fighting cybercrime: Malicious URL attack type detection using multiclass classification," in Proc. IEEE Int. Conf. Big Data (Big Data), Dec. 2020, pp. 1813_1822, doi: 10.1109/BIGDATA50022.2020. 9378029.

[14] M. Alsaleh and A. Alarifi, “Analysis of web spam for non-english content: Toward more effective language-based classifiers," PLoS ONE, vol. 11, no. 11, Nov. 2016, Art. no. e0164383, doi: 10.1371/journal.pone.0164383.

[15] C. Johnson, B. Khadka, R. B. Basnet, and T. Doleck, “Towards detecting and classifying malicious URLS using deep learning," J. Wireless Mob. Netw. Ubiquitous Comput. Dependable Appl., vol. 11, no. 4, pp. 31_48, Dec. 2020, doi: 10.22667/JOWUA.2020.12.31.031.

[16] M. Cova, C. Kruegel, and G. Vigna, “Detection and analysis of drive-by-download attacks and malicious JavaScript code," in Proc. 19th Int. Conf. World Wide Web (WWW), 2010, pp. 281_290, doi:10.1145/1772690.1772720.

[17] Y. Liang, Q. Wang, K. Xiong, X. Zheng, Z. Yu, and D. Zeng, ‘‘Robust detection of malicious URLs with self-paced wide & deep learning,’’ IEEE Trans. Depend. Secure Comput., vol. 19, no. 2, pp. 717–730, Mar. 2022.

[18] A. C. Bahnsen, E. C. Bohorquez, S. Villegas, J. Vargas, and F. A. González, ‘‘Classifying phishing URLs using recurrent neural networks,’’ in Proc. APWG Symp. Electron. Crime Res. (eCrime), Apr. 2017, pp. 1–8.

[19] M. Khonji, Y. Iraqi, and A. Jones, ‘‘Phishing detection: A literature survey,’’ IEEE Commun. Surveys Tuts. vol. 15, no. 4, pp. 2091–2121, 4th Quart. 2013.

[20] N. A. ALfouzan and N. C, ‘‘A systematic approach for malware URL recognition,’’ in Proc. 2nd Int. Conf. Comput. Inf. Technol. (ICCIT), Jan. 2022, pp. 325–329.

[21] K. H. Park, H. M. Song, J. D. Yoo, S.-Y. Hong, B. Cho, K. Kim, and H. K. Kim, ‘‘Unsupervised malicious domain detection with less labeling effort,’’ Comput. Secur. vol. 116, May 2022, Art. No. 102662.

[22] S. Afzal, M. Asim, A. R. Javed, M. O. Beg, and T. Baker, ‘‘URLdeepDetect: A deep learning approach for detecting malicious URLs using semantic vector models,’’ J. Netw. Syst. Manage., vol. 29, no. 3, pp. 1–27, Jul. 2021.

[23] A. S. Rafsanjani, N. Binti Kamaruddin, M. Behjati, S. Aslam, A. Sarfaraz and A. Amphawan, "Enhancing

Malicious URL Detection: A Novel Framework Leveraging Priority Coefficient and Feature Evaluation," in IEEE Access, vol. 12, pp. 85001-85026, 2024, doi: 10.1109/ACCESS.2024.3412331.

[24] H. Zhao, Z. Chang, W. Wang and X. Zeng, "Malicious Domain Names Detection Algorithm Based on Lexical Analysis and Feature Quantification," in IEEE Access, vol. 7, pp. 128990-128999, 2019, doi: 10.1109/Access.2019.2940554.

[25] A. S. Rafsanjani, N. B. Kamaruddin, H. M. Rusli and M. Dabbagh, "QsecR: Secure QR Code Scanner According to a Novel Malicious URL Detection Framework," in IEEE Access, vol. 11, pp. 92523-92539, 2023, doi: 10.1109/ACCESS.2023.3291811.

[26] M. Sanchez-Paniagua, E. F. Fernandez, E. Alegre, W. Al-Nabki and V. Gonzalez-Castro, "Phishing URL Detection: A Real-Case Scenario Through Login URLs," in IEEE Access, vol. 10, pp. 42949-42960, 2022, doi: 10.1109/Access.2022.3168681.