Detection of Malicious URLs through an Ensemble of Machine Learning Techniques

Shreya Venugopal
Department of Computer Science
PES University South Campus
Bangalore, India
shreyavenugopal@pesu.pes.edu

Shreya Yuvraj Panale
Department of Computer Science
PES University South Campus
Bangalore, India
shreyapanale@pesu.pes.edu

Manav Agarwal
Department of Computer Science
PES University South Campus
Bangalore, India
manavagarwal@pesu.pes.edu

Rishab Kashyap
Department of Computer Science
PES University South Campus
Bangalore, India
rishabkashyap@pesu.pes.edu

U Ananthanagu
Assistant Professor
Department of Computer Science
PES University South Campus
Bangalore, India
ananthanagu@pes.edu

***Abstract*—This paper aims to classify URLs and web pages into legitimate and malicious sites to alert users and allow safer browsing through the internet. Through this process we have found various points of interest and attributes that bring to light the characteristics of these malicious sources, allowing us to be aware of and prevent any damage it might cause. These attributes relate to the domain registration of the URLs, the URL text, the structure of the web page and its contents. The application of models such as BERT, LSTM, Decision Trees and their amalgamation as an ensemble result in a pragmatic solution to the problem in the form of an ensemble giving an accuracy of 95.3%. It also uses concepts such as webpage reputation, Internal Links and External Links of a web page. The method of classification used in this paper where both Natural Language Processing techniques and Machine Learning models with such a vast variety of features have been combined has not been implemented earlier. We conclude the paper by suggesting methods to improve to solve the problem.**

***Index Terms*—Malicious URLs and Web Pages, Machine Learning Models, BERT, LSTM, Decision Trees**

I. Introduction

There are approximately 4.66 billion users who utilize the internet and web search engines today. Being active all the time, threats to it can cause serious damage, which include social engineering attacks such as Trojans or key-loggers, malware through phishing, worms and viruses. Such attacks could be devastating to the user and can cause a computer to crash or personal data being stolen. One of the most common methods used to do this is through Uniform Resource Locators or URLs, since they are the most accessible forms of links apart from other sources such as emails. Classical methods of detecting unsafe web pages such as blacklists[1] are still prevalent today. However, with the technological advancements, it has become easier to cause disruptions while evading detection. Through the course of this paper, we aim to introduce a new method complementing the technology available today to safely classify these URLs in a more efficient manner.

The main contributions of the paper are as follows:

1) A novel ensemble classifier model to detect web pages that are malicious using a varying variety of attributes and unprecedented combination of models that are extracted from the URL of the web page.
2) The importance of each attribute in the respective model it belongs to.

II. Literature Survey

The classification of URLs requires a list of testing links, which in our case has been obtained from various sites[2][3] as well as extraction from different sources[4][5][6]. Through research we have uncovered six most common ways of classifying malicious URLs[7] which include Blacklists, Search Engine methods, domain based, visual similarity, heuristic machine learning methods, and finally proactive phishing. Blacklists have been the most prevalent method to prevent the access to malicious URLs but the need to improve over these static lists gave rise to new techniques. Search Engine methods are based on the reputation of a web page using metrics such as page ranks[8] and their variants[9] introduced a new perspective of looking into such issues[10], since it connected all the pages across the internet together based on its accessibility as well as their importance[11][12]. Variations into this led to domain based searches[13][14] which look into various features of each URLs domain names and their properties such as its validity, time to live and so on. The proactive phishing method looks into the text of the URL[15] and identifies the lexemes and the pattern with which it is framed. Visual similarity looks into the HTML and CSS pages[16][17] to distinguish legitimate website structures from malicious ones. Finally, a recent and growing advancement in this field includes the heuristic machine learning techniques[18][19] which make use of a range of Machine Learning models to classify URLs based on the previous techniques discussed. In one of the most recent surveys[20] conducted on the application of ML models to

this problem, the techniques for doing so are categorized into Natural Language Processing, Machine Learning and Deep Learning methods. According to the survey, although these methods have been applied and detect malicious web pages to a certain extent, the problem still persists. It recommends the use of more advanced models of this field such as extremely modern neural network models. Similar recent and complex models along with ensemble techniques to achieve better efficiency have been used here.

III. Proposed Methodology

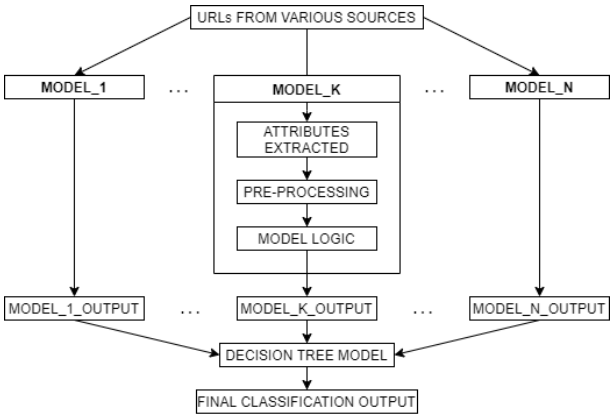

Fig. 1. Proposed Methodology Flow Diagram

As indicated by Fig. 1, the solution to the problem is divided into various models. Each model performs either a natural language processing, machine learning or artificial intelligence function. The attributes of a model belong to a specific aspect of URL classification such as one set of attributes for the characters present in the URL text, another set relating to the domain name of a URL and so on. To build this system, first the relevant URLs are taken from various sources. Then, features are extracted by pre-processing them in different ways depending on the functionality of each model. These extracted features are grouped into the models they belong to which uses them to determine whether the page is malicious or not. The results of these models are combined using a decision tree to come up with the final output displayed at the front-end section where the input from the user is a URL which is provided in a text box on the website interface. Besides selecting the attributes and grouping them for different models, the appropriate model for the attributes, the hyper-parameters of the model and the methodology to compare each model output is completely data-driven and hence has minimal human bias.

IV. Data Collection and Transformation

The datasets used are listed in references [2] to [6]. These include standard datasets along with data extracted with the help of continuous web scraping. This is done using the python library BeautifulSoup[21] along with the cron technique[22] which extracts the various URLs periodically. These URLs are then labeled with zeros for legitimate and ones for malicious

TABLE I
List of Datasets

Dataset Name	Extraction & Transformation Steps	Number of Examples
Special Characters Dataset	Perform Lexical Analysis to extract special characters	295364
HTML Tags Dataset	From the markup obtained from cron extract the tags of the webpage and remove columns and rows that had only zeroes	5615
Domain Related Information Dataset	Extract the duration in days for which the domain name of the URL is valid, the number of internal and external links it points to via web scraping with BeautifulSoup	4000
Web Page Text Dataset	Extract the information enclosed in the HTML tags that represent the text via web scraping with BeautifulSoup	5615
DOM Tree Dataset	Get a depth-first-search traversal of the tags of the markup via web scraping with BeautifulSoup	232
BERT Dataset	URLs from [2] to [6]	295364
Alexa Dataset	Contains the most popular domain names which are highly unlikely to be malicious	1000000
Ensemble Dataset	Classifier outputs along with the results for determining how the models can be pieced together	3005

to create a binary classification for the same. New datasets are derived based on the attributes extracted from the existing ones. Table 1 shows these derived datasets, the steps taken to obtain their features as well as their size. Each of these datasets are cleaned by removing duplicates and empty or null values. Fig. 2, shows the composition of the derived datasets with respect to those that have been collected.

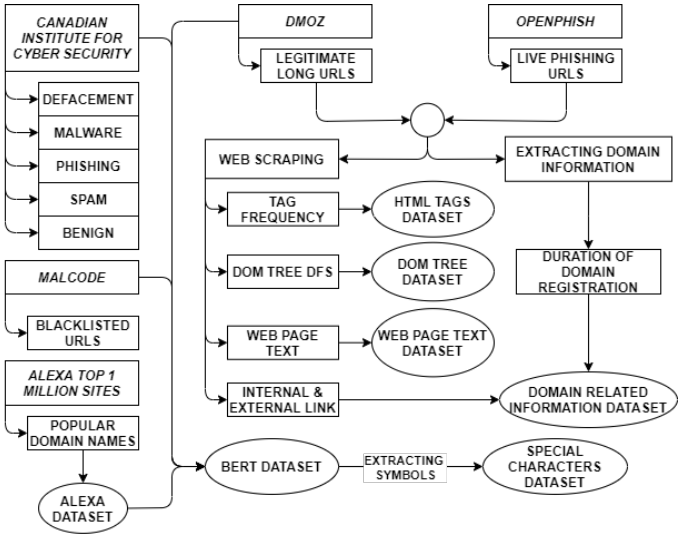

Fig. 2. Desription of Datasets

V. Models and Interfaces

In this paper, a variety of models have been used and integrated together to form a single functioning unit. Each

model pertains to a specific functionality performed to classify the URL and plays a crucial role in determining the legitimacy of a URL, which are displayed to the user by the frontend.

A. Long Short Term Memory

The Long Short Term Memory (LSTM) is a recurrent neural network which works on the principle of feedback data points allowing it to process multiple inputs simultaneously making it a viable option for URL classification[23]. As part of the pre-processing of this model, tokenization is done in order to feed the textual data in the appropriate numeric form to the neural network. In this work, the LSTM is performed as a supervised method for datasets containing attributes with a significant amount of sequential data.

1) Text Classification of the Web Pages: Here, the LSTM is used for the Web Page Text Dataset. This consists of what a user will read when looking at a web page. The text in a web page determines the topic that it addresses as well as the relevance of the topic under scrutiny. It is classified as malicious if the topics it describes fall under something that is commonly regarded as a source of threat. So, the tokenization here involves, converting the text of a web page into tokens of text in a numeric form with padding that can be processed at a time. Fig. 3 describes the complete process. The LSTM model here, has been trained on 5 epochs, with Adam optimizer and binary cross-entropy activation function. Early-stopping has been applied on compilation and a dropout mechanism has been applied on the neural network architecture to reduce overfitting.

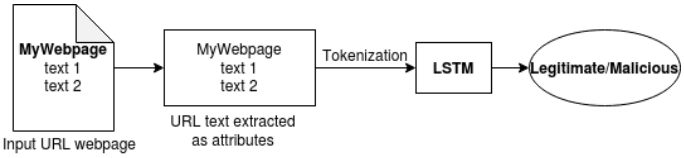

Fig. 3. LSTM model for text classification of web pages

2) DOM Tree Classification of the Web Pages: A Document Object Model(DOM) defines the structure of HTML pages. It depicts the structure of the entire web page in the form of a tree. In this paper, the extraction of these trees allow us to compare similarly organized pages and classify them into their respective classes which is the legitimate or malicious class. The DOM tree compares these tree structures[24] in order to identify any deviation from normal behaviour. This can be done by mapping the HTML tags of both malicious and legitimate classes in order to create this structure and thus creating a comparative study. Since, an awareness of the complete sequence of the tree traversal is required for the machine to learn the behavior of web pages in each of the classes an LSTM is used as it works very well with sequential data. Fig. 4 describes the complete process. The tokenization here would primarily group the tag names and their sequences, if commonly found, as they frequently appear in the corpus. Following a similar architecture to the Text Classification model, this LSTM is trained over a smaller dataset over 8 epochs to produce the final result.

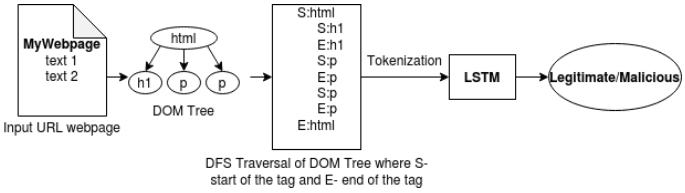

Fig. 4. LSTM model for DOM tree classification of web pages

B. The BERT model

The Bidirectional Encoding Representational Transformer[25] provides one of the most sophisticated and accurate ways of determining whether a URL formatting structure is feasible or not by the use of transfer learning. The advantage of a pre-trained model such as BERT is that it performs lexical analysis from left to right direction and right to left direction as well. This allows it to capture dependencies that might not be possible to do in most machine learning models. First tokenization is performed, which encodes the URLs by converting them into numeric tokens with unique identities, where each token represents one or more characters from the URL. This allows the model to analyze the attention, or the components of the tokenized URL together. Fig. 5 shows the mechanism of applying the BERT model in this paper.

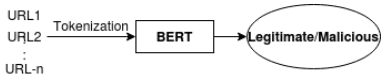

Fig. 5. BERT model for URL classsification

C. Decision Trees

Decision trees are very popular and powerful tools that are used for machine learning as they have a very good ability to handle non-linear data. In this work, decision trees are used when the number of attributes are large and the attributes involved are discrete.

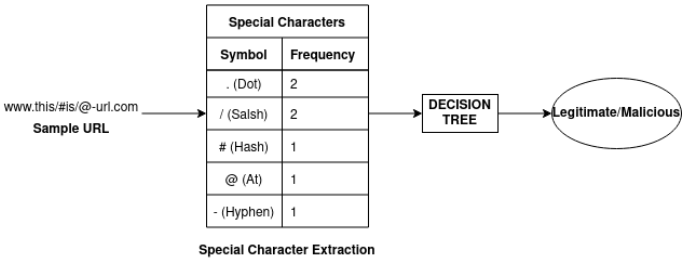

Fig. 6. Decision Tree for Special Symbols

1) Special Symbols: This decision tree uses the 'Special Characters Dataset' that consists of the frequency of various special characters present in the URL[26]. There are 30 different special characters taken into consideration. It is seen that the URLs that are malicious tend to have an unnaturally large frequency of special characters. To test this hypothesis, this dataset is created and is used for classification. The decision tree thus formed involves branching out depending on the frequency of a certain special character processed at a time. The final classification will hence be characterized by a frequency range of a specific combination of special characters that are present in the URL. Fig. 6 shows the process of how it has been created.

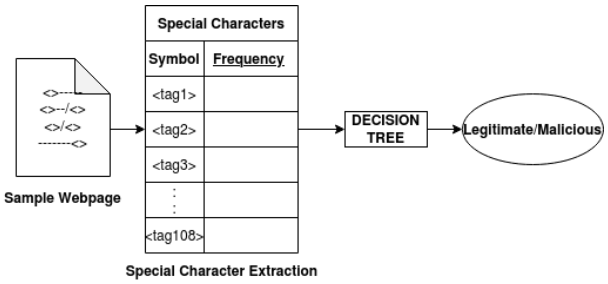

Fig. 7. Decision Tree for HTML Tags

2) HTML Tags: This decision tree uses the 'HTML Tags Dataset' which will give the frequency of each HTML tag in the markup of the URL as shown in Fig. 7. It is possible that the usage of some tags more frequently such as an anchor tag or too many input tags increases the probability of the URL being a malicious one. Thus the classification in this case, similar to that of the previous decision tree would involve a combination of some HTML tags that determines the legitimacy of the URL.

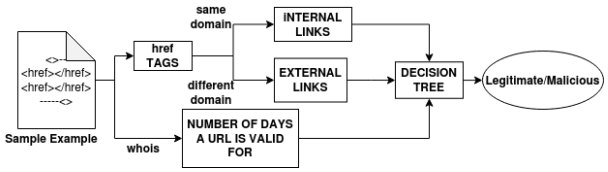

Fig. 8. Decision Tree for Domain Name specific features

3) Domain Name specific features: This decision tree uses the 'Domain Related Information Dataset' which consists of the number of internal and external links of the URL along with the duration in days for which the domain name of the URL is valid[27]. A reasonable URL is pinpointed by the decision tree based on the ratio between these two types of links. Reputed domains tend to have a registration for a longer period of time than malicious ones which try to remain latent from Blacklists. Hence, the number of days for which a domain name is valid is also included in this model. Fig. 8 shows the process of implementing this model.

4) Ensemble Model: It is the final model that collectively integrates and computes an aggregate of the predictions made by all the previous models to determine the final outcome of the URL as shown in Fig. 9. It also provides a comparative study of the performance of each of the models used. The ensemble construct allows us to obtain a non-biased multi-source contribution to classify each URL as legitimate or malicious. The Decision tree takes the outputs of the above models and creates multiple binary decisions that leads to the final conclusion about the URL to be displayed on the web page to the user.

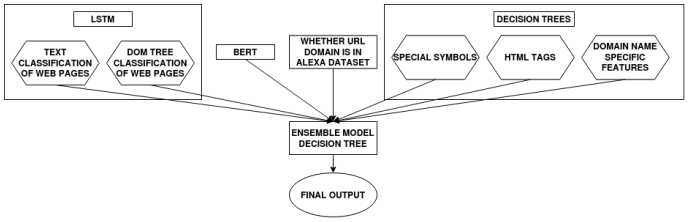

Fig. 9. Decision Tree for ensemble model

D. Front End Display

The result of the ensemble model is delivered to the user via a web page, where the user can enter the desired URL to obtain this result. The model is hosted via Flask[28] which not only displays to the user whether the given URL is legitimate or not, but also the outcomes of each individual model.

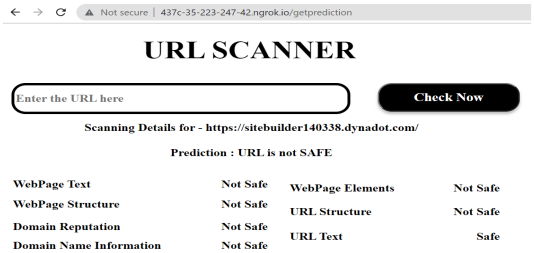

Fig. 10. Front End Output

VI. Observations

Table II shows the accuracies obtained for each model for classification along with the contribution of each model into the ensemble indicated by the importance in the table. It is seen that the LSTM for text classification has the highest accuracy, however this is not considered significant as an independent model as a predictor because it will be very difficult to extract the contents of a URL. Moreover, the time taken for a prediction increases greatly with an increase in the size of a web page. The BERT model exhibits a small scale of overfitting, and similar to LSTM cannot act as an independent model for the reason that the structure of a URL alone does not provide enough proof for a website being safe or not. Similar reasons apply to every model that make the ensemble when used individually. However, because each model looks into a different facet of a web page in an exhaustive manner the ensemble is the best solution.

TABLE II
MODEL ACCURACIES

Model Name	Dataset Used	Accuracy
LSTM for Text Classification of the Web Pages	Web Page Text Dataset	99.98
LSTM for DOM Tree Classification of the Web Pages	DOM Tree Dataset	78.26
BERT Model	BERT Dataset	98.6
Decision Tree for Special Symbols	Special Characters Dataset	73.87
Decision Tree for HTML Tags	HTML Tags Dataset	87.6
Decision Tree for Domain Name Specific Features	Domain Related Information Dataset	89.2
Decision Tree for Ensemble Model	**Ensemble Dataset**	**95.3%**

TABLE III
SOME IMPORTANT SPECIAL CHARACTERS AND THEIR IMPORTANCE IN THE MODEL

Symbol	Importance %
+	3.0302
_	8.9676
.	14.7311
-	18.6924
=	42.2431

Further, based on the architecture of the various decision trees the relative importance of the attributes involved can be observed. From these observations certain interpretations can be made. Table III shows the important special characters that influence the safeness of a URL the most according to the Decision Tree for Special Symbols. This may be because these symbols do not appear very often in a legitimate URL.

Table IV shows the most important HTML tags that influence the classification according to the Decision Tree for HTML Tags. The ”input” tag has the greatest influence and this may be because many phishing sites try to take details from users such as their banking details and for doing so they would need ”input” tags. ”a” tags also influence the classification because many malicious URLs redirect to some other pages through ”href”s present in the a tags. Following this ”h1” and ”table” tags also influence the classification if not as significantly as the ”a” and ”input” tags. This can be because the ”h1” tag text catches the eye of the user and ”table” tags make the forms that take in inputs in HTML look

TABLE IV
SOME IMPORTANT SPECIAL CHARACTERS AND THEIR IMPORTANCE IN THE MODEL

HTML tags	Importance %
table	5.4450
script	5.7386
h1	6.0793
a	19.8721
input	20.8373

TABLE V
IMPORTANCE OF DOMAIN RELATED FEATURES IN THE MODEL

Feature	Importance %
duration	82.1588
internal links	10.3852
external links	7.4559

better structured. Further, malicious URLs that don't have very complex functionality tend to have ”script” tags within their markups.

Table V shows that the most important domain related feature that is influencing the Decision Tree for Domain Name Specific Features is the duration in days for which the domain is registered. This may be because the malicious URLs want to escape the attention of the browser and hence try to remain hidden and short-lived.

VII. DISCUSSION

The methods employed by people creating the malicious URLs is continuously evolving. Hence, this topic will have constant research for a long time to come. This work, has tried to create a system that will continue to catch malicious URLs even as they evolve since the ensemble looks into almost all components in web page creation. The ensemble enables better modularity, error handling and it also reduces the overfitting. Besides, each of these models are the best for the given data as they have performed better in comparison to other models such as SVM, KMeans and so on.

VIII. CONCLUSION

This work gives an ensemble of models with different attributes to classify whether a URL is malicious or legitimate with an accuracy of 95.3%. A method which takes into account so many different types of attributes in one framework is rare and will also make sure that a hacker cannot easily infiltrate this system and become successful.

REFERENCES

[1] ”Google Blacklist classification” : https://patchstack.com/google-blacklist/
[2] ”DMOZ dataset” https://www.kaggle.com/shawon10/url-classification-dataset-dmoz?select=dmoz.csv
[3] ”ISCXURL Canadian Institute of Cybersecurity list of datasets” https://www.unb.ca/cic/datasets/index.html
[4] ”Alexa top 1 million websites” https://www.alexa.com/topsites
[5] ”Openphish phishing sites” https://openphish.com/
[6] ”Malcode website” http://malc0de.com/dashboard/
[7] Varshney, Gaurav, Manoj Misra, and Pradeep K. Atrey. ”A survey and classification of web phishing detection schemes.” Security and Communication Networks 9.18 (2016): 6266-6284.
[8] Zoltán Gyöngyi, Hector Garcia-Molina, and Jan Pedersen. 2004. Combating web spam with trustrank. In Proceedings of the Thirtieth international conference on Very large data bases - Volume 30 (VLDB '04). VLDB Endowment, 576–587.
[9] Krishnan, Vijay, and Rashmi Raj. ”Web spam detection with anti-trust rank.” AIRWeb. Vol. 6. 2006.
[10] Huh J.H., Kim H. (2012) Phishing Detection with Popular Search Engines: Simple and Effective. In: Garcia-Alfaro J., Lafourcade P. (eds) Foundations and Practice of Security. FPS 2011. Lecture Notes in Computer Science, vol 6888. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-27901-0_15

[11] Gaurav Varshney, Manoj Misra, Pradeep K.Atrey, “A phish detector using lightweight search features”, Computers & Security vol.62 2016, Article ID VARSHNEY2016213, pages 213-228, 2016. https://doi.org/10.1016/j.cose.2016.08.003

[12] Palse, V., Hangarage, S., Mithapelli, R., Arsul, S., & Dhawase, N. (2017). Real Time Phishing Detection on Generated URLs. International Journal of Engineering Research and, 6.

[13] Gopinath Palaniappan, Sangeetha S, Balaji Rajendran, Sanjay, Shubham Goyal, Bindhumadhava B S, Malicious Domain Detection Using Machine Learning On Domain Name Features, Host-Based Features and Web-Based Features, Procedia Computer Science, Volume 171, 2020, Pages 654-661, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2020.04.071.

[14] Khulood Al Messabi, Monther Aldwairi, Ayesha Al Yousif, Anoud Thoban, and Fatna Belqasmi. 2018. Malware detection using DNS records and domain name features. In Proceedings of the 2nd International Conference on Future Networks and Distributed Systems (ICFNDS ’18). Association for Computing Machinery, New York, NY, USA, Article 29, 1–7. DOI:https://doi.org/10.1145/3231053.3231082

[15] M. Sameen, K. Han and S. O. Hwang, ”PhishHaven—An Efficient Real-Time AI Phishing URLs Detection System,” in IEEE Access, vol. 8, pp. 83425-83443, 2020, doi: 10.1109/ACCESS.2020.2991403.

[16] Jian Mao, Jingdong Bian, Wenqian Tian, Shishi Zhu, Tao Wei, Aili Li, Zhenkai Liang, “Detecting Phishing Websites via Aggregation Analysis of Page Layouts”, Procedia Computer Science, Volume 129, 2018, Pages 224-230, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2018.03.053.

[17] S. Haruta, H. Asahina and I. Sasase, ”Visual Similarity-Based Phishing Detection Scheme Using Image and CSS with Target Website Finder,” GLOBECOM 2017 - 2017 IEEE Global Communications Conference, 2017, pp. 1-6, doi: 10.1109/GLOCOM.2017.8254506.

[18] Rao, R.S., Pais, A.R. & Anand, P. A heuristic technique to detect phishing websites using TWSVM classifier. Neural Comput & Applic 33, 5733–5752 (2021). https://doi.org/10.1007/s00521-020-05354-z

[19] Sankhwar, S, Pandey, D, Khan, RA, Mohanty, SN. An anti-phishing enterprise environ model using feed-forward backpropagation and Levenberg-Marquardt method. Security and Privacy. 2021; 4:e132. https://doi.org/10.1002/spy2.132

[20] Said Salloum, Tarek Gaber, Sunil Vadera, Khaled Shaalan, ”Phishing Email Detection Using Natural Language Processing Techniques: A Literature Survey”, Procedia Computer Science, Volume 189, 2021, doi: 10.1016/j.procs.2021.05.077.

[21] BeautifulSoup Documentation https://www.crummy.com/software/BeautifulSoup/bs4/doc/

[22] CronTab Documentation https://man7.org/linux/man-pages/man5/crontab.5.html

[23] Bahnsen, Alejandro Correa, et al. ”Classifying phishing URLs using recurrent neural networks.” 2017 APWG symposium on electronic crime research (eCrime). IEEE, 2017.

[24] Documentation for ”Document Object Model Trees”, https://en.wikipedia.org

[25] Devlin, Jacob, et al. ”Bert: Pre-training of deep bidirectional transformers for language understanding.” arXiv preprint arXiv:1810.04805 (2018).

[26] D. Shete, S. Bojewar and A. Sanghvi, ”Survey Paper on Web Content Extraction & Classification,” 2021 6th International Conference for Convergence in Technology (I2CT), 2021, pp. 1-6, doi: 10.1109/I2CT51068.2021.9417947.

[27] ”Internal Links” https://moz.com/learn/seo/internal-link

[28] ”Flask Documentation”: https://flask.palletsprojects.com/en/2.0.x/

[29] Mehdi Babagoli, Mohammad Pourmahmood Aghababa, and Vahid Solouk. 2019. Heuristic nonlinear regression strategy for detecting phishing websites. Soft Comput. 23, 12 (June 2019), 4315–4327. DOI:https://doi.org/10.1007/s00500-018-3084-2

[30] Ping Yi, Yuxiang Guan, Futai Zou, Yao Yao, Wei Wang, Ting Zhu, ”Web Phishing Detection Using a Deep Learning Framework”, Wireless Communications and Mobile Computing, vol. 2018, Article ID 4678746, 9 pages, 2018. doi.org/10.1155/2018/4678746

[31] Zou Futai, Gang Yuxiang, Pei Bei, Pan Li and Li Linsen, ”Web Phishing detection based on graph mining,” 2016 2nd IEEE International Conference on Computer and Communications (ICCC), 2016, pp. 1061-1066, doi: 10.1109/CompComm.2016.7924867.

[32] Chen J-L, Ma Y-W, Huang K-L. Intelligent Visual Similarity-Based Phishing Websites Detection. Symmetry. 2020; 12(10):1681. doi.org/10.3390/sym12101681

[33] Mao, J., Bian, J., Tian, W. et al. Phishing page detection via learning classifiers from page layout feature. J Wireless Com Network 2019, 43 (2019). doi.org/10.1186/s13638-019-1361-0

[34] Chavan, A., Iyer, R., Ramtirthakar, A., Guru, M. S. K., & Khude, M. P. Identification of Fraudulent Phishing Emails Based On CSS Standard Technique to Explore Similarities in Web.

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

[36] Suman Bhattacharyya, Chetan kumar Pal, Praveen kumar Pandey, “Detecting Phishing Websites, a Heuristic Approach”, International Journal of Latest Engineering Research and Applications (IJLERA) ISSN: 2455-7137, Volume – 02, Issue – 03, March – 2017, PP – 120-129.

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

[38] A. S. Manjeri, K. R., A. M.N.V. and P. C. Nair, ”A Machine Learning Approach for Detecting Malicious Websites using URL Features,” 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), 2019, pp. 555-561, doi: 10.1109/ICECA.2019.8821879.

[39] Al-Ahmadi, Saad, A Deep Learning Technique for Web Phishing Detection Combined URL Features and Visual Similarity (2020). International Journal of Computer Networks & Communications (IJCNC) Vol.12, No.5, September 2020, Available at SSRN: https://ssrn.com/abstract=3716033

[40] Abdelnabi, Sahar & Krombholz, Katharina & Fritz, Mario. (2020). VisualPhishNet: Zero-Day Phishing Website Detection by Visual Similarity. 1681-1698. 10.1145/3372297.3417233.

[41] Muhammad Taseer Suleman, Shahid Mahmood Awan, Optimization of URL-Based Phishing Websites Detection through Genetic Algorithms. Aut. Control Comp. Sci. 53, 333–341 (2019). https://doi.org/10.3103/S0146411619040102

[42] Afzal, S., Asim, M., Javed, A.R. et al. URLdeepDetect: ”A Deep Learning Approach for Detecting Malicious URLs Using Semantic Vector Models”, J Netw Syst Manage 29, 21 (2021). doi.org/10.1007/s10922-021-09587-8

[43] R. Chiramdasu, G. Srivastava, S. Bhattacharya, P. K. Reddy and T. Reddy Gadekallu, ”Malicious URL Detection using Logistic Regression,” 2021 IEEE International Conference on Omni-Layer Intelligent Systems (COINS), 2021, pp. 1-6, doi: 10.1109/COINS51742.2021.9524269.

[44] J. Acharya, A. Chuadhary, A. Chhabria and S. Jangale, ”Detecting Malware, Malicious URLs and Virus Using Machine Learning and Signature Matching,” 2021 2nd International Conference for Emerging Technology (INCET), 2021, pp. 1-5, doi: 10.1109/INCET51464.2021.9456440.

[45] Ch Rupa, Gautam Srivastava, Sweta Bhattacharya, Praveen Reddy, and Thippa Reddy Gadekallu. 2021. A Machine Learning Driven Threat Intelligence System for Malicious URL Detection. In The 16th International Conference on Availability, Reliability and Security (ARES 2021). Association for Computing Machinery, New York, NY, USA, Article 154, 1–7. DOI: 10.1145/3465481.3470029

[46] Saleem Raja, R. Vinodini, A. Kavitha, Lexical features based malicious URL detection using machine learning techniques, Materials Today: Proceedings, 2021, ISSN 2214-7853, doi.org/10.1016/j.matpr.2021.04.041.

[47] J. Yuan, G. Chen, S. Tian and X. Pei, ”Malicious URL Detection Based on a Parallel Neural Joint Model,” in IEEE Access, vol. 9, pp. 9464-9472, 2021, doi: 10.1109/ACCESS.2021.3049625.

[48] Kumar, M. S., & Indrani, B. (2020). Frequent rule reduction for phishing URL classification using fuzzy deep neural network model. Iran Journal of Computer Science. doi:10.1007/s42044-020-00067-x

[49] Xu, Pingfan. ”A Transformer-based Model to Detect Phishing URLs.” arXiv preprint arXiv:2109.02138 (2021).

[50] Chen Zuguo, Liu Yanglong, Chen Chaoyang, Lu Ming, Zhang Xuzhuo, ”Malicious URL Detection Based on Improved Multilayer Recurrent Convolutional Neural Network Model”, Security and Communication Networks, Hindawi, doi.org/10.1155/2021/9994127

[51] ”Percent-Encoding in URLs” https://en.wikipedia.org/wiki/Percent-encoding /wiki/Document_Object_Model