Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Machine Learning and NLP –Based Approaches for Real-Time Malicious Website Detection

Shaheetha L[1, *], Vadivazhagan K[2]

[1]Research Scholar, Department of Computer and Information Science, Annamalai University. Tamil Nadu, India. Email: shahee.aasc@gmail.com
** Corresponding Author*
[2]Assistant Professor, Department of Computer and Information Science, Annamalai University. Tamil Nadu, India

Abstract: Websites that damage or exploit users are malicious. Frequently, they comprise code or content that is intentionally crafted to entice users into performing actions that pose a risk to their systems or confidential information. Websites can be malicious in various ways., such as Phishing websites, Malware distribution websites, Scam websites, Drive-by download websites, and Rogue security software websites. It requires constant monitoring and preparedness to protect internet users. Researchers have used machine learning, deep learning, and rules-based models to classify harmful websites. These techniques are based on feature generation and selection. The majority of research works employ features derived from URL, Domain Name Server, Website Content and External Server Rank. Among these, website content analysis gets less attention because it is risky nature. But it has many features to help classify it better. This paper focuses on website content, particularly textual content within the <meta><para> element of websites. Natural language processing methods like Hashing vectorizer is used to encode textual content. The experiment makes use of seven distinct machine learning methods in order to get more accurate classification. The outcomes demonstrate that the accuracy is enhanced when the hashing vectorizer is combined with random forest.

Keywords: Malicious websites, Natural language Processing, Content Analysis, Hashing Vectorizer, Phishing.

1. Introduction

The internet is crucial to modern life, promoting trade, communication, and the dissemination of knowledge. Cybersecurity threats, especially from rogue websites, are a real concern on the internet, despite its many benefits [1]. Many malicious programs and phishing schemes are hidden on these websites with the intention of tricking visitors, stealing their personal information, or damaging their devices [2].Use of heuristics, URL-based feature extraction, or URL-based blacklisting are common in traditional methods for identifying compromised websites. But there are certain harmful websites that these approaches can't detect or block [3].

In URL-based feature extraction, characteristics including domain repute, URL length, and keyword occurrence are extracted from the URL itself. While this approach might provide light on a webpage's structure, it could miss important details about the content's semantic meaning or context [4]. In URL-based blacklisting, websites with matching URLs are blocked. This

method stores known harmful URLs in a database. Although it does work to a certain degree, this method does not deal with newly generated malicious webpages or ones that produce URLs dynamically. More importantly, thieves may simply avoid detection by employing URL obfuscation methods or routinely changing URLs. Potentially harmful websites can be identified using heuristic methods that depend on previously established rules or patterns. These characteristics might be based on URL structure, HTML content, or behavior. Certain types of malicious behaviour can be detected by heuristic approaches; however, these methods frequently have large false positive rates and could miss complex threats that don't fit into preset patterns [5].

By utilizing the most popular machine learning techniques, we test how well the suggested approach works. Random forest with hashing vectorizer leads to better accuracy.

This is the remaining section of the paper's outline: The significance of the study and how it differs from previous studies are discussed in Section 1. In Section 2, survey prevailing state of the art in malicious webpage identification by diving into the available research works in the field. To comprehend completely the suggested method, it is necessary to review the background material provided in Section 3. Delineating its methodology and complications in depth, Section 4 expounds upon the proposed strategy. Experiment findings demonstrating the effectiveness and performance metric are shown in Section 5. Section 6 concludes the investigation by summarizing its results and insights and offering final recommendations.

2. Related Works

A technique for identifying malicious URLs was disclosed by Saleem et al. [6]. As an alternative to the blacklist method, the suggested technique makes use of the URL's lexical properties. Algorithms utilizing k-nearest neighbour (k-NN) and random forest (RF) achieved 99% and 98% accuracy, respectively, in detecting fraudulent URLs. Using natural language processing (NLP) methods, Jeyakumar et al. [7] vectorize URL keywords and then categories them using ML and DL algorithms. The experimental variables are D1 and D2. There are three separate ways that URL text is vectorized. Random Forest (RF) with TF-IDF vectorizer and Decision Tree (DT) with count vectorizer obtained 92.4% accuracy on the D1 dataset. On D2, the DT secured the accuracy with the TF-IDF vectorizer is 99.5%. Its accuracy with the D1 dataset is 89.6 percent, and with the D2 dataset it rises to 99.2 percent.

In order to enhance malicious URL identification, Saleem et al. [2] use machine learning to examine structural trends in URLs. For increased accuracy, it highlights the need of feature extraction and categorization. It emphasizes how crucial it is to combine URL characteristics for accurate identification. Using machine learning, Saleem et al. [4] use the lexical characteristics of URLs, including length and unusual characters, to identify risks. The results highlight the accuracy with which lexical analysis can detect harmful linkages. MUDHR, a heuristic rule-based framework for recognizing malicious URLs, is proposed by Saleem et al. [5].

Cho, Hoa, and Tisenko [3] have investigated many machine learning models' ability to identify dangerous URLs. Its primary objective is to improve trustworthiness by combining rule-based detection with automated learning. Zamir et al. [15] investigate the use of a variety of machine learning methods, such as SVM, Neural Networks and Decision Trees in the identification of phishing websites. Use HTML properties, JavaScript analysis, and URL characteristics to assess these models' performance. According to their research, incorporating several algorithms can reduce false positives and improve the accuracy of phishing detection. Shaheetha et. al.[25] emphasized that cybercriminals the assault online security in a number of

ways. In accordance with recent cyber security assessments, there will be 17% more security breaches in 2021 than in 2020. The majority of categorization efforts centre on URLs. Some studies use the number of tags on a page to generate characteristics. Websites with several categorization features tend to have less emphasis on their content. Table 1 shows consolidate report of related works.

Table 1:
Table 1, shows consolidated report of related works

SNo	***Author(s)***	***Techniques Used***	***Application***	***Remarks***
1	B. Janet; Ankur Nikam; Joshua Arul Kumar R[8]	lexical and semantic features	Twitch Chatroom.	Feature extraction is an efficient and secure method to safeguard data in real time without requiring a lot of computational power.
2	Saleem Raja A[9]	Machine learning based detection method. SVC, LR, k-NN,NB, RF	Web Services,URL Detection in Ecommerce, Mcommerce	Random forest classification method outperforms the accuracy.
3	Srinivasan S., Vinayakumar R., Arunachalam A., Alazab M., Soman K [10]	Character level encoding	Web Services, URL Detection	The Deep URL Detect system encrypts URLs at the character level, uses a hidden layer deep learning structure to extract characteristics, and finally, uses these features to identify dangerous URLs from Benign.
4	Vinayakumar R, Sriram S, Soman KP, and Mamoun Alazab[11]	DeepURLDetect (DUD), hybrid CNN and RNN	Web site content, registry keys.	There were five models that were utilized. There are two CNN-based models, two RNN-based models, and one CNN-LSTM hybrid model. 93–98% of malicious URLs were detected, with a 0.001 false positive rate.
	Ashish Kumar Luhach[12]	Random forest models and gradient boosting classifier	Web Services, Web Pages	Accuracy with random forest as 98.6%.
6	Zamir, A., Khan, .U., Iqbal, T., Yousaf, N., Aslam, F., Anjum, A. Hamdani, M[13]	Begging, Naïve Bayes, kNN, SVM, RF, NN) with PCA.	Model is trained with features using 10 folds validation. Web Services	Stacking 1 (NN+RF+Bagging) gives 97.4% accuracy than other classifiers.
7	Junaid Rashid; Toqeer Mahmood; Muhammad Wasif Nisar; Tahira Nazir [14]	SVM RF	Ecommerce, Mcommerce, Online Payment Sector, webmail	Support vector machine classifier outperforms all other methods, correctly distinguishing 95.66 percent of fraudulent websites from legitimate ones.
8	Ali Aljofey 1,2, Qingshan Jiang 1,*, Qiang Qu 1, Mingqing Huang 1 and Jean-Pierre Niyigena [15]	Deep Learning, CNN	Hand-crafted, character embedding	Achieved accuracy of 98.58 with existing phishing URL models.

9	Cho Do Xuan1, Hoa Dinh Nguyen1, Tisenko Victor Nikolaevich3[16]	Support vector machine (SVM) and Random forest (RF)	information security,Online Payment Sector, webmail	Free tool has been developed [20] to identify harmful URLs on browsers.
10	Ozgur Koray Sahingoz, Ebubekir Buber, Onder Demir, Banu Diri[17]	NB,RF, kNN, Adaboost, K-star, SMO and Decision Tree classification algorithms	Web Services,URL Detection	Random Forest with the accuracy of 97.98%
11	Hafiz. Junaid, Niyaz, Devabhaktuni, Guo, Shaikh[18]	Voting classifier	Ecommerce, Mcommerce	Voting classifier that combines several machine learning algorithms on those selected features gives better results.
12	Sudhanshu Gautam, Kritika Rani and Bansidhar Joshi[19]	Naïve Bayes and PART algorithms. associative classification	Phishing website detection	PART algorithm has a higher accuracy detection
13	Purvi Pujara, M. B.Chaudhari[20]	Anti-phishing techniques are there such as blacklist, heuristic, visual similarity and machine learning	Online Payment Sector, webmail, and financial institution, file hosting or cloud storage	Tree-based classifiers in machine learning approach is best suitable
14	Waleed Ali[21]	BPNN,RBFN, SVM, naïve Bayes classifier (NB), decision tree (C4.5), RF, and (kNN). Wrapper features selection method	Website detection	The wrapper-based features selection outperformed the machine learning classifiers
15	S Liaquathali, V Kadirvelu[26]	KNN,NB,SVM,RF,NB	Website detection	Random forest with the accuracy of 93.46

3. Background

Protecting users from harmful websites is an predominant part of cybersecurity, and machine learning techniques are essential for this objective. An improvement in the overall efficacy of threat detection systems is achieved by the distinct capabilities and techniques offered by each algorithm in recognizing potentially hazardous online material. In particular, Web page categorization is made extremely efficient by Support Vector Machines (SVM), which use hyperplanes in high-dimensional space to separate webpages into discrete groups. Support vector machines (SVMs) dramatically improve the detection of sophisticated online harmful actions by spotting complicated patterns and anomalies.

Logistic Regression excels at using input information to estimate the likelihood of a website being harmful. Logistic Regression (LogR) helps cybersecurity professionals identify the elements that contribute to website risks by adjusting the data to correspond with a logistic function. The findings are interpretable. Decomposing the classification process into a sequence of binary decisions based on characteristics like URL structure, content, and behaviour, Decision Trees (DT) provide an approach to malicious webpage detection that is both clear and easy to understand. An additional helpful approach for malicious webpage identification is K-Nearest Neighbours (KNN), which uses the similarity of sites to identify them. In order to correctly categorize new occurrences, KNN looks at the features of nearby webpages and determines if they are threatening or not.

To increase generalization performance, Random Forests (RF) aggregate several trees and reduce overfitting, making them an extension of Decision Trees. Algorithms in the Gradient Boosting (GB) family, such as XGBoost and Gradient Boosting (GB), create strong predictive models for detecting harmful online content by iteratively correcting mistakes by building an ensemble of weak learners sequentially.

Machine learning uses vectorizers to transform unprocessed textual data into numerical representations that models can use. Vectorizers are crucial for deriving significant patterns from text in the context of natural language processing and phishing detection. The fastest and most memory-efficient tool for text vectorization is the Hashing Vectorizer. In a comparable vein a hashing algorithm is used to hash tokens, converting text input into a fixed-dimensional sparse matrix. Hashing Vectorizers are perfect for managing big datasets or systems that operate immediately as they eliminate vocabulary creation.

4. Proposed Approach

The proposed approach for malicious webpage detection consists of several stages, each aimed at extracting, cleaning, vectorizing, and classifying web content to enhance detection accuracy.

A. Dataset

Some prominent datasets used in the experiment were UNB [23], phistank [24], and the URL dataset (ISCX-URL2016) [22]. One common issue in machine learning is imbalanced data, when one class has much more samples than the others. Because of this inequality, models may be skewed in favour of the dominant class, which can have a negative impact on minority group performance [10]. In order to address this problem and guarantee unbiased results, an equal number of benign and malicious URLs were carefully selected for the experiment. A concise summary of every malicious and benign URL that was employed throughout the investigation is given in Table 2.

Table 2, *shows dataset summary*

No	Type	Count
1	Benign	5530
2	Malicious	5882

B. Information Extraction

Examining the webpage for text, paying close attention to paragraph (para), division (div), and meta tags, is the main goal of the information extraction step. The use of requests and Beautiful Soup, two Python tools that simplify web scraping and HTML parsing, makes this task easier to do. The first step is to use the requests package to send an HTTP request to the specified webpage and get its HTML content. Afterwards, Beautiful Soup is used to traverse the HTML structure and extract relevant tags that contain content. The raw textual information is methodically obtained through repetitive processing of each extracted tag, preparing it for later analysis and processing.

C. Data Cleaning

When the textual content of each webpage is preprocessed, it undergoes a number of alterations to make it ready for processing. To begin, we separate the words from the content of each website by tokenizing them. Later on, all tokens are changed to lowercase and any

tokens that aren't alphabetical are taken out. At last, a filtering procedure gets rid of any tokens that are part of a previously established list of stop words, maintaining simply a standardized and clear representation of each page's text.

D. Training and Testing

Logistic Regression, Decision Tree, Support Vector Machine (SVM), k-Nearest Neighbours (KNN), Gradient Boosting, Random Forest, and Extreme Gradient Boosting (XGBoost) are the seven machine learning techniques that will now be used. These algorithms were selected individually because of the variety of Classification methods they employ and the variety of data features they might potentially collect. An extensive assessment of these algorithms' efficacy in harmful webpage identification is accomplished by assessing a variety of performance parameters across them, including accuracy, precision, recall, and F1-score.

5. Experimental Results

The system configuration used to conduct the experiments was a 2 GHz CPU running Windows 15. Jupyter Notebook, an interactive platform for coding and analysis, aided the experimental environment. The machine learning tasks in Python were carried out using the scikit-learn package, which is an extensive library. Using this configuration, we tested how well different machine learning algorithms detected fraudulent websites. As part of the review procedure, we tested both the combined text from all tags and the text that was extracted from specific HTML elements, such as paragraph (Para), meta (meta), and division (Div). Accuracy, precision, recall, and F1-score are some of the performance measures displayed in the individual tables containing the results of these tests (Tables 3, 4, 5, and 6). The efficiency of each method for malicious webpage identification was rigorously evaluated using repeated stratified k-fold cross-validation, which ensures robustness and dependability.

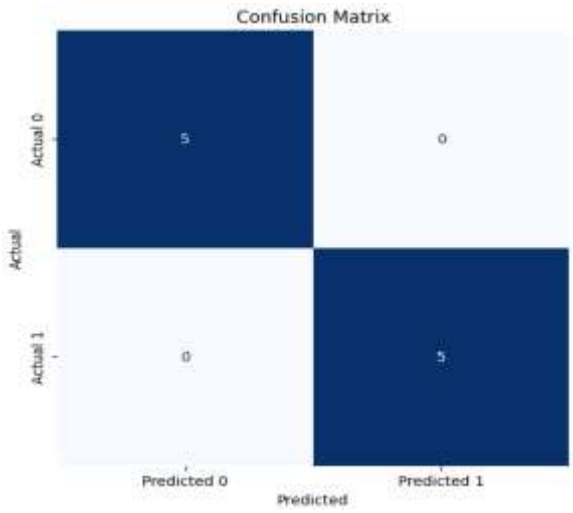
Table 3:
Performance of textual content of Paragraph Tag

Algorithm	Accuracy%	Precision%	Recall%	F1-Score%
LogR	85.9	84.6	90.2	87.0
NB	82.2	86.7	82.4	89.7
kNN	75.8	72.2	96.4	81.5
DT	85.7	83.2	91.8	87.1
RF	**90.1**	**90.7**	**90.6**	**90.5**
GB	81.3	77.9	91.9	83.8
XGB	**80.5**	**76.9**	**93.2**	**83.6**

The experimental results demonstrate that utilizing textual contents extracted from paragraph tags yields notable improvements in accuracy, achieving 90.1%, along with a commendable F1-score of 90.5% for Random Forest. Additionally, Figure 1 illustrates the heatmap generated for the para tag, providing a visual representation of the classification results. Figure 2 presents the ROC-AUC curve specifically for the Random Forest algorithm applied to the div tag, offering insights into its performance characteristics.

Figure 1
Confusion matrix of Random Forest Algorithm for Para Tag

Figure 2
ROC-AUC of Random Forest Algorithm for Para Tag.



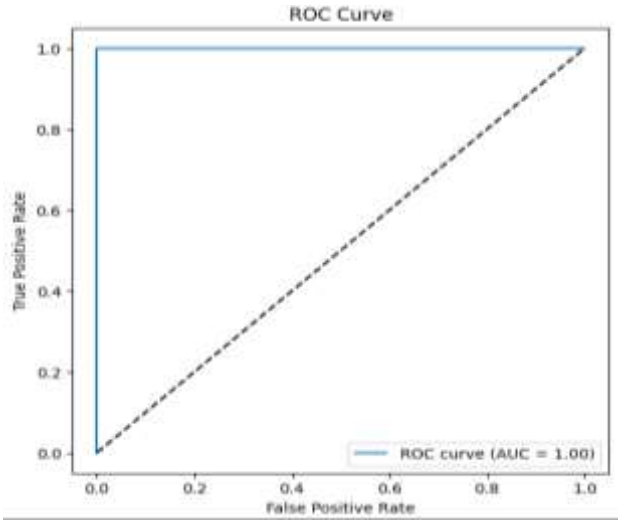


Table 4:
Performance of textual content of Div Tag.

Algorithm	Accuracy%	Precision%	Recall%	F1-Score%
LogR	86.6	84.9	91.1	87.7
NB	82.9	85.1	82.4	83.3
kNN	87.3	82.1	90.3	89.7
DT	88.7	83.2	91.8	87.1
RF	**89.0**	**88.2**	**91.6**	**89.7**
GB	82.7	79.4	92.1	84.9
XGB	87.5	81.5	84.4	86.0

Figure 3
Confusion matrix of Random Forest Algorithm for Div Tag.

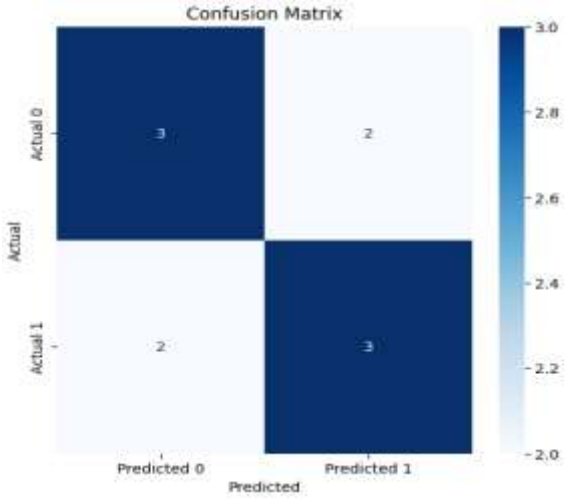
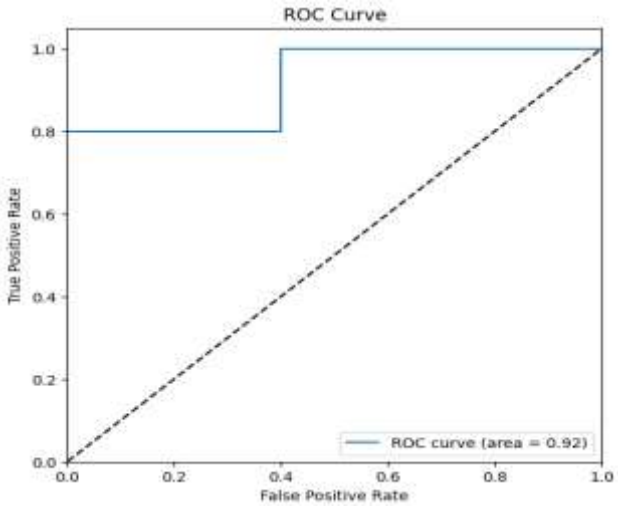


Figure 4
ROC-AUC of Random Forest Algorithm for Div Tag.



The experimental findings demonstrate that textual contents extracted from the div tag showcase performance, achieving an accuracy of 89.0% and an F1-score of 89.7% in the Random Forest classifier. Figure 3 demonstrate the heatmap generated for the para tag, providing a visual representation of the classification results. Figure 4 presents the ROC-AUC curve specifically for the Random Forest algorithm applied to the div tag.

Table 5:

Performance of textual content of Meta Tag.

Algorithm	Accuracy%	Precision%	Recall%	F1-Score%
LogR	85.7	85.6	88.0	86.5
kNN	81.6	63.9	88.8	73.5
NB	82.4	84.8	81.5	82.6
DT	**88.0**	**85.8**	**92.8**	**89.0**
RF	**90.4**	**89.6**	**92.6**	**90.9**
GB	80.5	76.9	93.2	83.6
XGB	84.8	82.3	92.1	88.6

Figure 5
Confusion matrix of Random Forest Algorithm for Meta Tag.

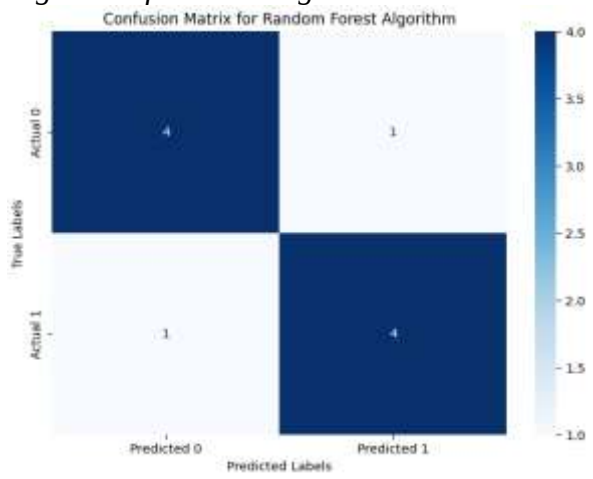
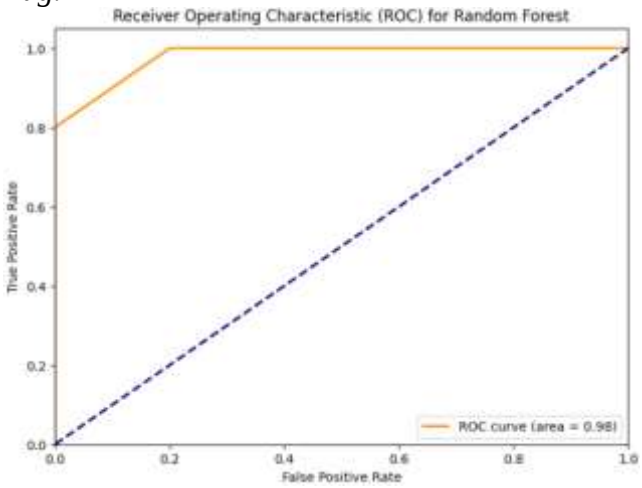


Figure 6
ROC-AUC of Random Forest Algorithm for Meta Tag.



Results of the experiment Prominent review reveal the performance metrics for textual contents derived from meta tags. The Random Forest method achieved a remarkable accuracy of 90.4% and an F1-score of 90.7%, whilst the Decision tree approach achieved an accuracy of 88.0% and an F1-score of 92.8%. The heatmap visualization presented in Figure 6 offers a graphical representation of the performance metrics associated with meta tags. ROC-AUC curve in Figure 2 illustrates specifically for the Random Forest algorithm, unveiling into its discriminative capabilities based on meta tag content.

6. Conclusion and Future work

In light of ever-changing nature of cyber threats, it is of the utmost importance to have reliable procedures that can identify dangerous information on the internet. Traditional signature-based methods frequently fall short in detecting advanced malware and phishing attacks, highlighting the need for more proactive and sophisticated detection techniques. Detection based on web content is essential, as malicious actors skillfully disguise their activities within ostensibly legitimate webpages. Examining the underlying content allows for the identification of subtle patterns and anomalies that suggest malicious intent. The proposed method utilizes a pretrained model alongside seven distinct machine learning classifiers to enhance the detection of malicious webpages. Classifiers trained on these embeddings have positive results: the Random Forest classifier obtains 90.4% accuracy and 90.7% F1-score, while the Decision tree classifier achieves 88.8% accuracy and 92.8% F1-score. More complex context-aware pretrained models might be incorporated into the suggested method to increase detection accuracy.

References

1. Borky, Bradley. Protecting Information with Cybersecurity. Effective Model-Based Systems Engineering, 9, pp.345–404, 2018.https://doi.org/10.1007/978-3-319-95669-5_10.
2. Saleem, Peerbashab, Iqbal, Sundarvadivazhagan, Surputheen. Structural Analysis of URL for Malicious URL Detection using Machine Learning. Journal of Advanced Applied Scientific Research, 5(4), 28–41, 2023. https://doi.org/10.46947/joaasr542023679.
3. Cho, H., & Tisenko. Malicious URL detection based on machine learning. International Journal of Advanced Computer Science and Applications (IJACSA), 11(1) 2020. https://doi.org/10.14569/IJACSA.2020.0110119
4. Saleem, Vinodini, Kavitha, Lexical features based malicious URL detection using machine learning techniques, Materials Today: Proceedings, vol 47, part 1, pp. 163-166 2021, ISSN 2214-7853, https://doi.org/10.1016/j.matpr.2021.04.041.
5. Saleem, Pradeepa, Arul, MUDHR: Malicious URL Detection Using Heuristic Rules based Approach, AIP Conference Proceedings, vol 2393, 1, 2022. https://doi.org/10.1063/5.0074077
6. Saleem, Pradeepa, Justin, Madhubala, Hariraman, Vinodhini, SmishGuard: Leveraging Machine Learning and Natural Language Processing for Smishing Detection, International Journal of Advanced Computer Science and Applications, vol. 14, no. 11, 2023. doi: 10.14569/IJACSA.2023.0141160
7. Kavita ,HashingVectorizer vs. CountVectorizer, [Internet]. https://kavita-ganesan.com/hashingvectorizer-vs-countvectorizer/ (Last access: 2 Apr 2024)
8. Janet, B., Nikam, A., & Kumar R., J. A. (2022). Real-time malicious URL detection on Twitch using machine learning. In *Proceedings of the International Conference on Electronics and Renewable Systems (ICEARS)*. IEEE.
9. Saleem Raja A. (2021). Lexical features-based malicious URL detection using machine learning techniques. *Elsevier Journal*.
10. Srinivasan, S., Vinayakumar, R., Arunachalam, A., Alazab, M., & Soman, K. (2021). Malicious URL detection using deep learning-based character-level representations. In *International Conference on Malware Analysis Using Artificial Intelligence and Deep Learning*. Springer, Cham.
11. Vinayakumar, R., Sriram, S., Soman, K. P., & Alazab, M. (2021). Malicious URL detection using deep learning. *IEEE Journal*.
12. Luhach, A. K. (2021). Artificial intelligence paradigms for smart cyber-physical systems. *Engineering Science Reference*, IGI Global.
13. Zamir, A., Khan, H. U., Iqbal, T., Yousaf, N., Aslam, F., Anjum, A., & Hamdani, M. (2020). Phishing website detection using diverse machine learning algorithms. *The Electronic Library*, 38(1), 65-80.
14. Rashid, J., Mahmood, T., Nisar, M. W., & Nazir, T. (2020). Phishing detection using machine learning techniques. In *Proceedings of IEEE Conference* (pp. xx-xx). IEEE.
15. Aljofey, A., Jiang, Q., Qu, Q., Huang, M., & Niyigena, J.-P. (2020). An effective phishing detection model based on character-level convolutional neural network from URL. *Electronics*, 9(1514). https://doi.org/10.3390/electronics9091514
16. Do Xuan, C., Nguyen, H. D., & Nikolaevich, T. V. (2020). Malicious URL detection based on machine learning. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 11(1).
17. Sahingoz, O. K., Buber, E., Demir, O., & Diri, B. (2019). Machine learning-based phishing detection from URLs. *Expert Systems with Applications*, 117, 345-357.
18. Junaid, H., Niyaz, Q., Devabhaktuni, V., & Guo, S. (2019). Identifying generic features for malicious URL detection system. In *Proceedings of the IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)* (pp. xx-xx). IEEE.
19. Gautam, S., Rani, K., & Joshi, B. (2018). Detecting phishing websites using rule-based classification algorithm: A comparison. In *Springer Nature Singapore Conference* (pp. xx-xx).
20. Pujara, P., & Chaudhari, M. B. (2018). Phishing website detection using machine learning: A review. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 3(7).
21. Ali, W. (2017). Phishing website detection based on supervised machine learning with wrapper features selection. *International Journal of Advanced Computer Science and Applications (IJACSA)*.
22. Manu, Malicious URLs dataset, [Internet]. https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset?resource=download (Last access: 2 Apr 2024)
23. URL dataset, [Internet]. https://www.unb.ca/cic/datasets/url-2016.html (Last access: 2 Apr 2024)

24. PhishTank, [Internet]. https://www.phishtank.com/developer_info.php (Last access: 2 Apr 2024)
25. Shaheetha, L., & Vadivazhagan, K. (2022, December). Detection of Malicious Domains in the Cyberspace using Machine Learning & Deep Learning: A Survey. In 2022 11th International Conference on System Modeling & Advancement in Research Trends (SMART) (pp. 1540-1543). IEEE.
26. Liaquathali, S., & Kadirvelu, V. (2024). WCA: Integration of Natural Language Processing Methods and Machine Learning Model for Effective Analysis of Web Content to Classify Malicious Webpages. Journal of Advanced Research in Applied Sciences and Engineering Technology, 47(1), 105-122.

ORCID

Shaheetha L*[1,*]*. , https://orcid.org/0009-0005-7035-6998

- Corresponding Author

Vadivazhagan K*[2] *: https://orcid.org/0000-0003-0709-0769