UNSYS digital

The Journal of Instrumentation, Automation & Systems
Vol.11, No.1

A Machine Learning Framework for Detecting Phishing Websites

Danat Gutema, Ravikumar S., S. Vinoth Kumar

Department of Computer Science & Engineering
Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, India.

Abstract—

Background: Identifying legitimate websites from fake ones has become increasingly difficult as phishing attempts have become a major hazard in the digital age. A recent poll conducted by Intel found that 97% of security professionals were unsure of how to differentiate between the two. Machine learning, with its many methods for URL classification, presents a viable answer to this problem.

Objectives: The major purpose of this research is to examine how well machine learning performs at identifying potentially fake websites. Different machine learning approaches have been developed and compared in this study to achieve this end. Additionally, the study tries to assess the most accurate model against current solutions in the literature.

Methods: To do this, we use a large dataset that includes both legal and counterfeit websites to train machine learning algorithms. These algorithms can learn to distinguish between the two types if they are presented with examples from both. As a result, advanced phishing detection systems may be built that can immediately warn users when they visit malicious websites.

Statistical Analysis: The algorithms utilized, datasets, and performance indicators for the machine learning methods used will all be presented and analyzed statistically in this part. An impartial assessment of the procedures will be provided.

Findings: The outcomes of using machine learning methods to identify fraudulent websites will be summed up in this section. Comparisons of the accuracy and efficiency of several algorithms for detecting phishing websites will be highlighted.

Applications and Enhancements: Here, we'll look at how the research can be used in the real world to make the Internet safer for everyone. To further develop the machine learning models for improved accuracy and user safety, we will also address potential enhancements and future research initiatives.

Keywords— website phishing, random forest, decision tree, dataset, identity theft.

DOI: 10.21535/w4xj7b79

Corresponding author: Ravikumar S. (ravikumars@veltech.edu.in)
This paper was submitted on November 11th, 2023; revised on December 20th, 2023, and accepted on January 3rd, 2024.

I. Introduction

The most widespread form of cyberattack is the phishing assault. Involves stealing a person's or a company's essential credentials; first used in the 1920s. In a 2021 report by CISCO, it was noted that approximately 86% of organizations had at least one person who clicked on a phishing link. In the other report by APWG, in 2022 APWG observed a total of over one million reported cases for the first quarter and it became the worst quarter ever observed. The vulnerability of different industries to phishing attacks as Statista report shows that the financial sector is the most affected at 23.6%, followed by software-as-a-service at 20.5%, E-commerce at 14.6%, and social media at 12.5%. Apart from this, one of the most damaging phishing attacks that occurred in 2022 included hacking between Russia and Ukraine resulting in data breaches, blackouts, and the release of malware. Phishing attacks come in various forms, including spear phishing, vishing, HTTP phishing or website phishing, pharming, and other types. This paper will specifically focus on phishing websites and the implementation of machine-learning models for detecting fake websites.[1]

A. Phishing website

A phishing Website are designed with the intent to deceive victims into revealing their credentials for theft or engaging in financial fraud. Fake websites can be created using spoofed or lookalike domains or as part of compromised legitimate websites by a method called water-holing which is one of the social engineering techniques, and makes people click on phishing links mostly through malicious email.[2] Phishing attacks targeting websites involve more than just sending emails to potential victims and relying on them to click on malicious links or open malicious attachments. Attackers employ various techniques such as:

- URL spoofing or Link manipulation
- homograph spoofing
- Link shortening
- Graphical rendering
- Covert redirect, and
- Chatbots.

Phishing attackers employ a range of tactics to deceive users. They may utilize JavaScript to superimpose an authentic URL's image onto a web browser's address bar, obscuring the real URL. Only when a user hovers over a linked element is the true URL exposed, and even that can also be manipulated using JavaScript. These malicious actors often create websites with URLs that closely resemble or imitate well-known and trusted sites, causing users to mistake them for the real thing. Another common tactic is that attackers frequently employ URL shortening services like Bitly to shrink URLs, making it difficult to discern whether the link is legitimate or fraudulent.

In other cases, a malicious intermediary page is linked between the victim's browser and the genuine site; this page then phishes the user for login information before delivering it to a third-party site created by the phishers.[3] The APWG study found that a single phishing site might use hundreds of different URL variations to market itself. There were a massive 1,350,037 phishing website assaults registered in the fourth quarter of 2022.

Experts recommend using antivirus software, desktop and network firewalls, anti-spyware software, an anti-phishing toolbar installed in web browsers, a Web security gateway, Spam filters, and Phishing filters from vendors like Microsoft, even though it is difficult to prevent phishing messages from reaching end users.[4]

II. Related Works

In this section, a few of the research works that deploy the methods of phishing website identification and techniques are summarized. Reference [5] has categorized phishing site detection techniques into three main groups. The first category, which involves heuristic and machine learning methods, relies on supervised and unsupervised learning techniques. These approaches necessitate the availability of features or labels to train a model for making predictions. The second category, referred to as proactive phishing URL detection, bears similarities to machine learning approaches. However, it involves the processing of URLs and supports a system that can predict whether a URL is legitimate or malicious. Lastly, there are the blacklist and whitelist approaches, which represent traditional methods for identifying phishing sites. However, the ever-expanding number of web domains has led to a decline in the effectiveness of these conventional techniques.

Apart from this using various assessments of URL attributes, determining the legitimacy of the website based on its hosting and management details, and relying on visual appearance-based analysis to authenticate websites are ways of avoiding being a victim of fake sites.[6] In reference [7], an examination of the suspicious page's visual attributes was conducted through the application of gestalt theory. This approach incorporated the utilization of the "super signal" concept, treating each web page as an individual entity within the overall detection system. Subsequently, these web pages underwent a thorough analysis employing compression algorithms to ascertain their similarity. Evaluation was carried out using metrics such as Normalized Information Distance (NID) and Normalized Compression Distance (NCD). Although the existing visual similarity-based methods have demonstrated promising results in phishing website detection, they do come with certain drawbacks.

The data used in the analysis cited in reference [8] came from the work of Chiew et al. It was made of 48 different characteristics learned from 10,000 websites, 5,000 of which were malicious phishing sites and 5,000 of which were legitimate legal resources. The phishing web sites were got from respectable sources such as Phish-Tank and Open-Phish, and the real internet pages were gathered from trusted databases such as Alexa and Common Crawl. The dataset was split into three categories according to the characteristics it contained: capabilities accessible via the address bar, capabilities specific to Abnormal, and HTML/JavaScript features.

Address bar-based characteristics relate to the attributes connected with the URL (Uniform Resource Locator) of a web page, including elements such as the length of the URL and the port number.

Second, abnormal-based features are non-standard page actions like loading items from other domains.

Third, features based on HTML and JavaScript include characteristics that refer to HTML and JavaScript methods embedded in the page's source code.

Not all HTTPS links ensure a secure connection and the confidentiality of sensitive data, as SSL certificates can be issued by unauthenticated or self-signed entities. Relying solely on an SSL certificate is inadequate for determining the security of an HTTPS link. It is imperative to verify the legitimacy of the certificate issuer to establish the email's authenticity. Therefore, if the SSL certificate is not issued by a reputable and trusted authority like GoDaddy, Comodo, or Symantec, the email should be regarded as suspicious.[9]

III. Methodology

A. Data Collection

The initial phase involves data collection, in which the necessary datasets for developing algorithms and detecting phishing websites are identified.[10] This paper made use of a Kaggle dataset, specifically the "malicious_phish.csv" dataset, which is rich in properties and can be categorized into four distinct types, each with its respective number of instances. There are 428,103 instances of Benign, 96,457 instances of Defacement, 94,111 instances of phishing, and 32,520 instances of malware available in the selected dataset. This dataset serves as the foundation for our model training and further analysis. For a selected dataset 20% of the data will be used for testing, and the remaining 80% will be used for training.

Table 1 Weak Dataset Used in the Experiments

Types of URL	Number of URL
Benign	428103
Defacement	96457
Phishing	94111
Malware	32520

Table 1 describes the content of the malicious dataset utilized in the experiment. This malicious dataset comprises 651,191 URLs which can be categorized into four groups Benign, Defacement, phishing, and Malware.

B. Feature

The dataset contains a variety of URLs with distinct features that play a crucial role in training and testing our machine-learning model, enabling us to determine whether a given site is benign or phishing.[11] Some of the key features used during the model training process include:[12]

1. The Google index functionality evaluates the presence of a URL within Google's search panel.
2. Phishing or malware websites sometimes utilize numerous sub-domains within their URLs, as seen by the presence of more than three dots. These characteristics give rise to distrust.
3. Quantify the occurrence of "www": It is commonly observed that secure websites generally incorporate a solitary instance of "www" within their uniform resource locators (URLs). Instances in which the established pattern deviates, such as the absence of the "www" prefix or the presence of several occurrences, may potentially signify malevolent intentions.
4. Directory Quantity: The presence of numerous directories within a URL may indicate the potential presence of a dubious website.
5. The shortening service function assists in the identification of URLs that utilize URL shortening services such as bit.ly, goo.gl, go2l.ink, and others. These services are frequently linked to phishing activities.
6. Malicious URLs often refrain from utilizing the HTTPS protocol due to its requirement of user credentials and indication of a secure website. The inclusion or exclusion of HTTPS in the URL serves as a noteworthy indicator.
7. Phishing or malicious websites may exhibit many instances of "HTTP" within their URL, whereas secure websites often have only a singular occurrence.
8. In the context of URL encoding, it is important to note that spaces are not allowed in URLs. To address this limitation, the practice of URL encoding is employed, whereby spaces are substituted with the "%" symbol. In general, secure websites tend to exhibit a lower number of spaces, whereas harmful websites tend to display a higher frequency of spaces, resulting in an increased prevalence of "%" symbols.
9. Quantifying the Question Mark: The use of the "?" symbol within a Uniform Resource Locator (URL) serves as an indicator of a query string, perhaps suggesting a dubious URL, particularly when many occurrences of "?" are observed.
10. Cybercriminals commonly add hyphens ("-") as either prefixes or suffixes to brand names in order to give the illusion of validity in the URL. As an example of a suitable website, one may choose www.flipkart-india.com.
11. The inclusion of the "=" symbol within a URL indicates the transmission of changeable values between form pages, a practice that carries inherent risks. The alteration of these variables has the ability to affect the webpage, so serving as a possible indication of a malicious Uniform Resource Locator (URL).
12. Numerical Quantification: Suspicious Uniform Resource Locators (URLs) frequently incorporate numerical digits, but secure URLs typically lack such numerical representations. The ability to determine the number of digits included in a URL is a significant attribute that can be utilized to identify potentially harmful URLs.
13. The length of the URL, including letters and numbers, is an important factor in the detection of fake URLs. The lengthening of URLs by attackers to hide the domain name is a crucial factor that must be carefully considered.

The combination of these features together enhances the discriminatory capability of our model in distinguishing between genuine websites and those that may provide potential harm.

C. Machine Learning Classification

In order to differentiate between phishing and benign websites, a range of machine-learning approaches can be employed. This study employs the random forest (RF) classification algorithm, specifically the Random Forest (RF) variation, as the preferred strategy.[13] The Random Forest (RF) variant is described as follows:

One of the most well-known algorithms for this purpose is called Random Forest (RF). The Random Forest algorithm is an unpruned collection of numerous individual decision trees that together form an ensemble learning method. The capacity of ensemble models, such as Random Forests, to reduce the effects of bias and variation, has gained widespread recognition. Large training datasets and a large number of input variables (up to hundreds or thousands) appear to be particularly well-suited for these models. The Random Forest algorithm excels mostly because of its ability to efficiently process numerous input variables. This is achieved by iteratively choosing subsets from the pool of accessible variables. A Random Forest model often has a substantial number of decision trees, often spanning from tens to hundreds. The Random Forest (RF) methodology involves the development of numerous decision trees, with an evaluation of the mistakes for instances that were not utilized in the construction of each individual tree. The procedure is commonly referred to as Out-of-Bag (OOB) error estimation.

IV. Experiment

A. Evaluation Metrics

The Confusion Matrix was employed to compute various metrics, including accuracy, sensitivity (true positive rate or TPR), specificity (true negative rate or TNR), false positives (FP), true positive (TP), true negative (TN), and false negative (FN).

Accuracy (A) is the fraction of the whole test dataset for which your model produced accurate predictions,

$$A = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Precision (P) is the degree to which a favorable forecast is accurate. In other words, it refers to the degree of certainty with which a projected favorable outcome may be assumed. The formula for its determination is:

$$P = \frac{TP}{TP + FP} \quad (2)$$

Recall, also known as the true positive rate (TPR), is the percentage of correct predictions relative to the total number of positives in the training set. Sensitivity (S) is another name for this quality:

$$S = TPR = \frac{TP}{TP + FN} \quad (3)$$

To calculate specificity (SS), divide the number of properly identified positive records by the total number of positive records:

$$SS = TNR = \frac{TN}{TN + FP} \quad (4)$$

False Positive Rates are

$$FPR = \frac{FP}{FP + TN} \quad (5)$$

Table 2 Description for Parameters FP, FN, TN, TP.

Parameters	Descriptions
TP (True Positive)	The quantity of phishing websites that were accurately recognized.
FP (False Positive)	The quantity of websites that were erroneously classified as phishing websites while not engaging in phishing activities.
TN (True Negative)	The quantity of benign websites identified as benign websites.
FN (False Negative)	The quantity of phishing websites that were erroneously seen as authentic websites.

Table 2 illustrations the description of different parameters used in the experiment for classifying purposes. In order to evaluate the value of matrices, knowing the value of TP, FP, TN, and FN is crucial. We have assigned the phishing URLs as the results which are positive for the test, and Benign as the results which are negative for the test.

B. Experimental results

This study examines a total of 651,191 websites sourced from the Kaggle dataset, focusing on certain predetermined essential criteria. The implementation in this context has utilized the Jupiter notebook. The random forest, decision tree, and K-nearest neighbors (KNN) algorithms have been applied to the dataset. The evaluation metrics used to assess their performance include accuracy, true positive rate (TPR), false positive rate (FPR), precision, and recall. The values have been computed utilizing provided performance measurements and employing a confusion matrix on datasets designated for testing purposes.

The experimental findings of the Random Forest method are presented in Table 3. The Random Forest algorithm is a machine learning methodology that functions by generating numerous decision trees during the training phase and subsequently aggregating their outcomes to enhance the accuracy of predictions or classifications.

Table 3 Performance of Random Forest

Classes	Precision	True positive rate (sensitivity, recall)	True negative rate (specificity)	F1-score
Benign	97%	98%	93.61%	97%
Defacement	95%	98%	99.08%	96%
Phishing	96%	92%	99.80%	94%
Malware	88%	81%	98.19%	84%

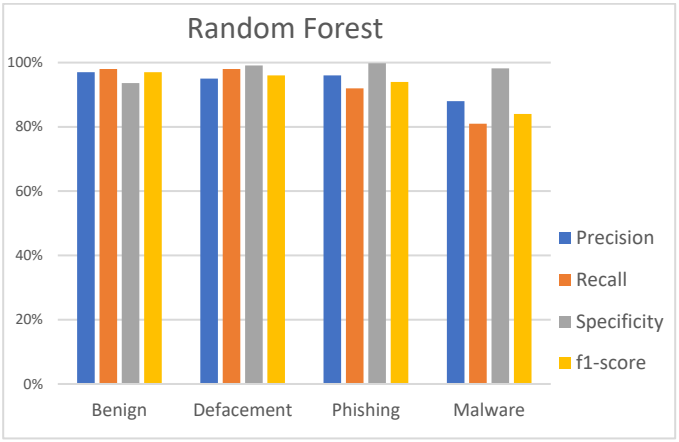

Figure 1 Random Forest Algorithm

This ensemble approach enhances the model's overall performance and reduces overfitting, making it a popular choice for various supervised learning tasks such as regression and classification. In this experiment, the Jupiter notebook was utilized, and a selected malicious dataset was classified as Benign, Defacement, Phishing, and Malware. The results are found based on parameters like precision, true positive rate (Recall), true negative value, and f1-score.

Table 4 Performance of Decision Tree

Classes	Precision	True positive rate (sensitivity, recall)	True negative rate (specificity)	F1-score
Benign	96%	98%	92.96%	97%
Defacement	94%	97%	98.83%	96%
Phishing	94%	92%	99.67%	93%
Malware	87%	78%	98.09%	83%

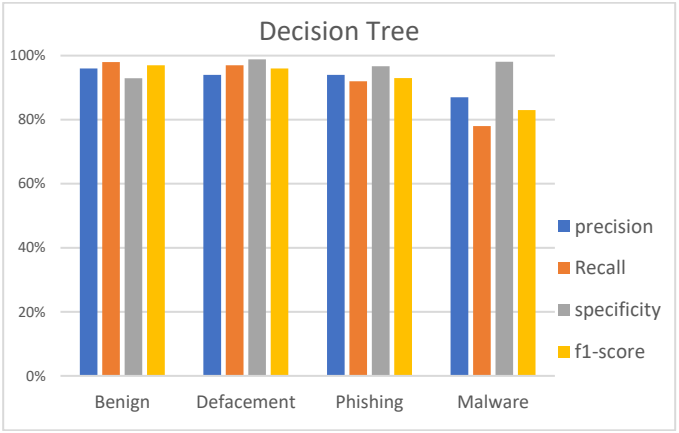

Figure 2 Decision Tree Algorithm

The experimental results gained from the use of the Decision Tree method are summarized in Table 4. The decision tree algorithm is a machine learning strategy that requires building a tree-like hierarchical structure to depict choices and their outcomes. Both classification and regression applications make heavy use of the method. For this experiment, we utilized a Jupiter notebook and categorized a harmful dataset as benign, defacement, phishing, or malware. The outcomes are determined by metrics such as accuracy, recall, specificity, specificity, and specificity, and f1-score.

Table 5 displays the results of experimental evaluations of the k-nearest neighbors method. In machine learning, the k-nearest neighbors (KNN) algorithm is used for both classification and regression. Using the k closest data points in the training dataset, a prediction is made about the class or value of a new data point. The suggested technique is non-parametric, meaning that it makes use of just distance metrics to make predictions. It is important to note, however, that when working with massive datasets, the processing requirements of this method may grow dramatically. In this project, we used the Jupiter notebook to classify a harmful dataset into benign, defacement, phishing, and malicious categories. The outcomes are determined by metrics such as accuracy, recall, specificity, specificity, and specificity, and f1-score.

Table 5 Performance of KNN

Classes	Precision	True positive rate (sensitivity, recall)	True negative rate (specificity)	F1-score
Benign	94%	97%	87.36%	95%
Defacement	91%	94%	98.33%	93%
Phishing	94%	88%	99.69%	91%
Malware	86%	70%	98.00%	77%

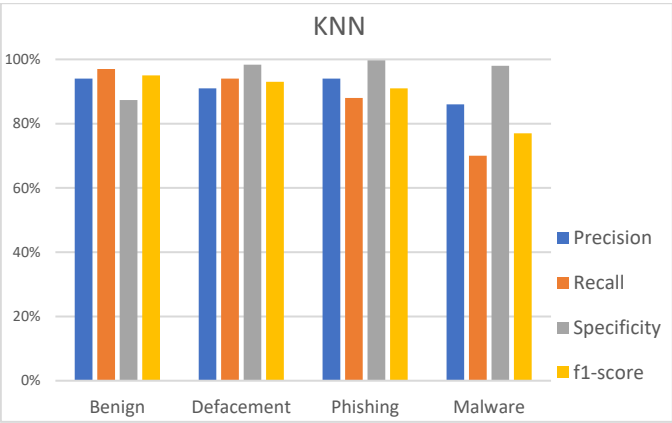

Figure 3 KNN Algorithm

Table 6 presents a comparative analysis of methods utilized for the phishing websites dataset during the testing phase, which was executed using the Jupiter notebook software. The dataset under consideration consists of four distinct classes: Benign, Defacement, Phishing, and Malware. For each of these classes, precision, recall, F1-score, and true negative values are computed. This evaluation is performed using three distinct machine learning techniques. The Random Forest algorithm outperforms other algorithms across various parameters. It attains the highest accuracy at 95.30%, surpassing the accuracy of other algorithms, which are 94.73% for Decision Trees (DT) and 92.34% for K-Nearest Neighbors (KNN).

Table 6 Accuracy of Different Algorithms

Techniques	Accuracy
Random Forest	95.3%
Decision Tree	94.73%
KNN	92.34%

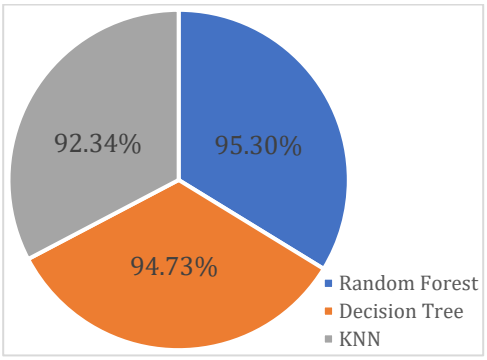

Figure 4 Accuracy

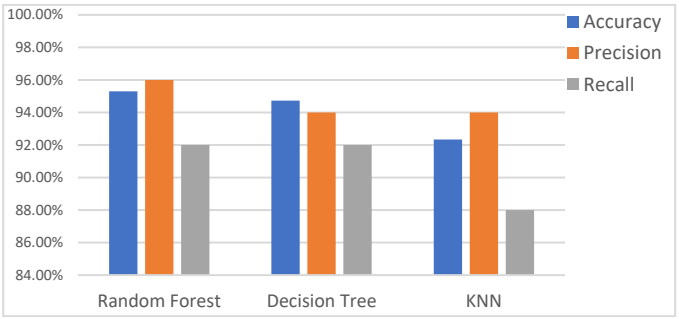

Figure 5 Comparison of Algorithms

The Random Forest method has superior performance in several parameters when compared to alternative algorithms, as depicted in Figure 4 and Figure 5. The algorithm in question demonstrated the highest level of accuracy, reaching 95.30%. In comparison, the DT algorithm earned an accuracy of 94.73%, while the KNN method achieved an accuracy of 92.34%.

V. Conclusion

The evolution of phishing threats remains an ongoing challenge, persisting despite the progress made in intelligence and security measures. Ensuring the safeguarding of persons from becoming targets of such fraudulent schemes is of utmost importance. This study compares various machine learning algorithms using a malicious dataset obtained from Kaggle datasets. The findings indicate that the random forest approach outperforms others in terms of accuracy, precision, and other relevant metrics. The investigation demonstrated that the Random Forest algorithm exhibits superior performance compared to alternative methods in terms of accuracy, precision, and other pertinent evaluation parameters. In order to assess the performance of the metrics, the algorithm under

consideration utilized essential components like true positives, true negatives, false positives, and false negatives. This was achieved by the implementation of a multi-class confusion matrix.

REFERENCES

[1] Ahmed, Dalia Shihab, Assist Prof Dr Karim Q. Hussein, and Hanan Abed Alwally Abed Allah. "Phishing Websites Detection Model based on Decision Tree Algorithm and Best Feature Selection Method", *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*13.1 (2022): 100-107.

[2] Ahmed, Kahkasha, and Sameena Naaz. "Detection of phishing websites using machine learning approach", *Proceedings of International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM), Amity University Rajasthan, Jaipur-India*. 2019.

[3] Almseidin, Mohammad, et al. "Phishing detection based on machine learning and feature selection methods", (2019): 171-183.

[4] Alnemari, Shouq, and Majid Alshammari. "Detecting phishing domains using machine learning", *Applied Sciences*13.8 (2023): 4649.

[5] Dutta, Ashit Kumar. Detecting phishing websites using machine learning technique", *PloS one*16.10 (2021): e0258361

[6] https://apwg.org/trendsreports/

[7] https://wisdomml.in/malicious-url-detection-using-machine-learning-in-python/

[8] https://www.egress.com/blog/phishing/how-to-identify-a-phishing-website#:~:text=Another%20indication%20that%20you%20may,omitted%2C%20treat%20it%20with%20suspicion.

[9] https://www.egress.com/blog/phishing/phishing-statistics-round-up

[10] https://www.techtarget.com/searchsecurity/definition/phishing#:~:text=Fake%20websites%20are%20set%20up,the%20user%20to%20respond%20quickly

[11] Mehanović, Dželila, and Jasmin Kevrić. "Phishing Website Detection Using Machine Learning Classifiers Optimized by Feature Selection", *Traitement du Signal* 37.4 (2020).

[12] Salahdine, Fatima, Zakaria El Mrabet, and Naima Kaabouch. "Phishing Attacks Detection A Machine Learning-Based Approach", *2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. IEEE, 2021.

[13] Saravanan, Priya, and Selvakumar Subramanian. "A framework for detecting phishing websites using GA based feature selection and ARTMAP based website classification", *Procedia Computer Science* 171 (2020): 1083-1092.