IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

# URL BASED PHISHING WEBSITE DETECTION USING MACHINE LEARNING MODELS

**Ms.A.Jayasmruthi[1], C.Srinivasan[2],S.Syed Razeem[3], R.Satish[4]**

[1]Ms.A.Jayasmruthi, Assistant Professor (Sr. Gr.) CSE, Sri Ramakrishna Institute of Technology, Coimbatore, India
[2]C.Srinivasan, CSE, Sri Ramakrishna Institute of Technology, Coimbatore, Tamil Nadu, India
[3]S.Syed Razeem, CSE, Sri Ramakrishna Institute of Technology, Coimbatore, Tamil Nadu, India.
[4]R.Satish, CSE, Sri Ramakrishna Institute of Technology, Coimbatore, Tamil Nadu, India

**Abstract:**
The project has been developed with the goal of enhancing online security by identifying and mitigating phishing threats. Phishing attacks have been identified as a major security concern, and traditional methods of detection have proven insufficient in handling sophisticated attacks. In this project, machine learning models have been employed to analyze URLs and classify them as either legitimate or phishing. Various features extracted from URLs, such as the presence of suspicious keywords, domain age, and the structure of the URL, have been considered for accurate classification. Datasets containing labeled URLs have been used for training and testing the models, with performance metrics evaluated to ensure optimal detection rates. The models have been optimized and validated using standard evaluation techniques, and their efficiency in real world scenarios has been demonstrated. By implementing this approach, the risks associated with phishing websites have been significantly reduced, and the reliability of the proposed solution has been confirmed

## 1. Introduction

Phishing websites, also known as "spoofed" websites, are often designed to replicate the appearance of trusted platforms, making them difficult for users to differentiate from authentic ones. These fraudulent websites are commonly distributed via deceptive emails, SMS messages, or social media posts, leading unsuspecting users to enter their private information, which is then exploited by attackers. According to various cybersecurity reports, the number of phishing attacks continues to rise globally, posing significant risks to individuals and organizations alike.

### 1.1 General Introduction

Its an open, anonymous, and unregulated nature, the Internet is a prime target for cyberattacks, which can endanger both individual users—even those with more experience—and the security of networks. It is impossible to completely stop people from falling for phishing schemes, even though vigilance and user experience are crucial elements (Greene, Steves, & Theofanos, 2018). In order to make phishing attacks more successful, attackers frequently take advantage of end users' personality attributes, especially when targeting even users with a fair amount of expertise (Curtis, Rajivan, Jones, & Gonzalez, 2018). These targeted cyberattacks have the potential to cause billions of dollars in damages annually in terms of money and sensitive personal data (Shaikh, Shabut, & Hossain, 2016). The analogy of "fishing" for victims is where the name "phishing" originates. Because of its efficacy and attractiveness to attackers, or phishers, In recent years, scholars have given this strategy a lot of attention. They develop fake websites that closely resemble well-known and trustworthy websites on the Internet. The Uniform Resource Locators (URLs) of these phony pages are usually different, despite the fact that they frequently resemble the real websites. By looking at the URLs, a knowledgeable user can identify these rogue websites. However, many users do not always examine the entire address of the active webpage, which may be connected from other websites, social networks, or simply received via email, due to the fast-paced nature of modern life. For cybersecurity, identifying phishing URLs is essential. We can enhance the detection of phishing URLs . Blacklists include known malicious URLs, yet XGBoost works well with complicated data. This hybrid strategy shields clients from phishing attacks while improving accuracy.

### 1.2 Problem Statement

The issue of phishing assaults, in which rogue websites are designed to trick visitors into disclosing private information, has been widely documented. These phishing websites have been designed to look exactly like authentic websites, making it challenging to identify them using traditional techniques. Such fraudulent tactics have deceived users, causing significant data and financial losses. In order to tackle this problem, machine learning models have been investigated for URL-based phishing detection techniques that seek to effectively and efficiently identify phishing threats.

### 1.3 Existing Methodology

The existing systems for phishing website detection have largely relied on traditional methods such as blacklists and heuristic-based detection. Blacklist-based approaches have been used, where URLs identified as malicious were stored in a database and matched against incoming URLs. However, this approach has been found to be ineffective against newly created phishing sites, as these blacklists were often outdated and failed to recognize new threats. Heuristic-based detection methods have been employed, which utilized rule-based techniques to identify patterns commonly associated with phishing websites. Nevertheless, these methods have been prone to high false positive rates, causing legitimate websites to be wrongly flagged as phishing. Some existing systems have incorporated browser extensions to alert users about potential threats, but these have had limited scope and were unable to dynamically learn from new data. Security tools have also been developed that depended on user feedback to update phishing databases, yet this reactive strategy often left users exposed to new threats before proper updates were implemented. Overall, these systems have struggled to keep up with the evolving tactics of cyber attackers, highlighting the need for more adaptive and intelligent solutions, such as those offered by machine learning models that have been designed to address these shortcomings.

## 2 Disadvantages

- High False Positive Rates in Heuristic-based Detection: Heuristic methods, which rely on rule-based detection of phishing patterns, often flag legitimate websites as malicious, leading to false positives. This can reduce user trust and cause unnecessary disruptions.

- Limited Scope of Browser Extensions: While browser extensions can warn users about potential phishing websites, their scope is often limited. They can't dynamically adapt to new phishing tactics or threats, leaving users exposed to more advanced phishing techniques.

- Lack of Adaptivity to Evolving Threats: Traditional systems are often not adaptive enough to keep up with the rapidly evolving techniques used by cyber attackers. This lack of flexibility can make existing methods outdated and less effective over time, requiring more intelligent, adaptive solutions like machine learning models to better address these challenges.

### 2.1 Proposed Methodology

A robust system for detecting phishing websites using machine learning models has been proposed to address the increasing threat of online fraud. The system has been designed to analyze and classify URLs based on several extracted features. These features, such as domain age, URL length, presence of special characters, and abnormal usage of HTTPS, have been identified and processed to build an effective predictive model. Data preprocessing techniques have been implemented to prepare a comprehensive dataset for training and testing purposes. The suggested method is set up to automatically identify and warn dubious URLs in real time, giving consumers instant protection. To find out which aspects have affected the model's performance the most, a feature importance analysis has also been conducted. In order to confirm the system's dependability . The goal of the suggested solution has been to be flexible and scalable so that it can be improved continuously when new phishing techniques are created. All things considered, this solution has been designed as a proactive way to strengthen cybersecurity defenses against phishing attacks.

### 2.2 Advantages

The URL-based phishing detection system using machine learning models has been acknowledged for offering several significant advantages in combating online threats. High accuracy in detecting phishing websites has been achieved, as machine learning models have been trained on vast datasets, enabling precise identification of malicious URLs. Real-time detection has been provided, ensuring users are alerted instantly when attempting to access a suspicious site, reducing the risk of data compromise. Compared to traditional rule-based detection systems, adaptability has been improved, as machine learning models have been designed to learn from evolving phishing techniques, making them more robust against new and sophisticated attacks. The system has also been automated, minimizing the need for human intervention and allowing for seamless integration into existing cybersecurity frameworks, such as web browsers and email filtering services.

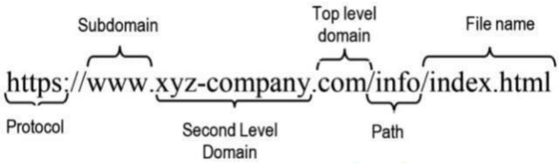
### 4.1 Machine Learning-Based Detection Systems

Machine learning algorithms are among the most widely used techniques for identifying fraudulent websites. This method views the detection of phishing attacks as a classification issue. The training data must include a number of characteristics of both authentic and fraudulent websites in order to create a learning-based detection system. It is simpler to find previously unseen or unclassified URLs utilizing a dynamic method when a learning algorithm is used. The CANTINA+ model is an enhanced version of this model that incorporates 15 HTML-based characteristics. Despite the 92% accuracy rate of the system, a significant number of false positives were also produced.Collecting keywords from the dubious website is the first step. The second step is to use a search engine to find potential target domains using these keywords. In the third step, the system assesses the validity of the webpage being accessed. The Online Perceptron, Confidence Weighted, and Adaptive Regularization of Weights algorithms were utilized for online classification, while Support Vector Machines were employed for offline classification.

### 4.2 URLs and Attackers Techniques

To avoid being discovered by system administrators and security measures, attackers use a variety of tactics. Some of these techniques are examined in this section. Understanding the elements of URLs is essential to understanding how attackers function. The protocol used to access the webpage, which usually identifies the business hosting the website, is the first element of a URL's standard structure. The Top-Level Domain comes next, which aids in locating the domain inside the DNS root zone of the Internet. These elements work together to form the webpage's hostname, often known as its domain name. The URL's other components are more adaptable and changeable. Cybersecurity companies actively identify fake domain names used in phishing in order to block related IP addresses.Cybercriminals use strategies such as random characters, word combination, cybersquatting, and typosquatting to improve their attacks. For detection measures to continue to work, these techniques must be taken into consideration.

#### 4.2.1 URL Components



##### 4.2.1.1 Dataset

We first looked for a well used dataset to compare the suggested solution, but we couldn't find one that satisfied our needs. Therefore, it was imperative to create a thorough and equitable dataset. Two categories of URLs—legitimate and phishing—had to be included in this collection. We mostly used information from PhishTank (2018) for the phishing URLs. Nevertheless, PhishTank's website does not offer a free dataset. We developed a script to download a lot of malicious website addresses in order to fix this. Simultaneously gathering reliable websites was essential. In order to retrieve the top-ranked web sites, we first created a specific list of query phrases and submitted them to the Yandex Search API. Given that fraudulent URLs usually do not score well in search results because of their limited duration. There are 73,575 URLs in the dataset that other researchers utilized (Ebbu2017PhishingDataset,2017), of which 36,000 are valid URLs and 37,200 are phishing URLs.

##### 4.2.1.2 Data Pre-processing

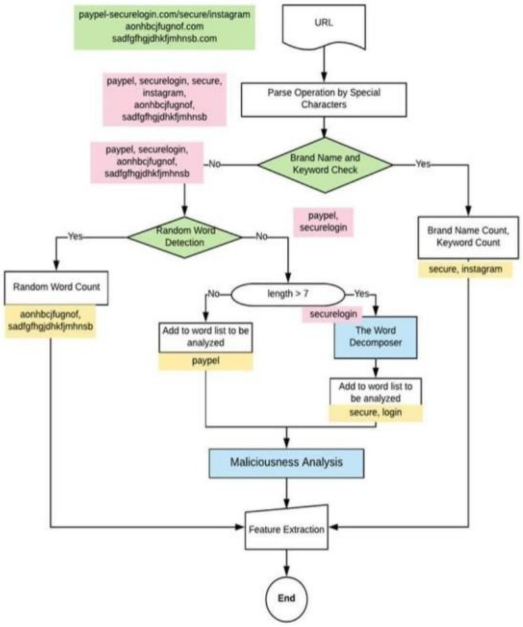
Both meaningful and nonsensical words, along with special characters that divide its key components, make up a URL. Before the data can be used for machine learning, it must be cleaned and organized through data preparation. There are multiple steps in this process: A dot ("."), for instance, separates the TLD from the SLD. In the same way, this character separates domain names from subdomain names. The "/" symbol separates the folders in the URL's path.12 Furthermore, as seen in the example xyz_company.com, each component of the URL may also have additional separators like ".", "_," etc. In the file path section, other characters like "=", "?", and "&" might also show up.

Therefore, the data preprocessing involves the following steps:

1. Extract each word from the URL.

2. Add the extracted words to the word list for further analysis during execution.

3. Detect the similarity between these words and those from the most targeted websites, as well as randomly generated words.


paypel-securelogin.com/secure/instagram
aonhbcjfugnof.com
sadfgfhgjdhkfjmhnsb.com
URL
paypel, securelogin, secure, instagram, aonhbcjfugnof, sadfgfhgjdhkfjmhnsb
Parse Operation by Special Characters
paypel, securelogin, aonhbcjfugnof, sadfgfhgjdhkfjmhnsb
No
Brand Name and Keyword Check
Yes
Brand Name Count, Keyword Count
secure, instagram
Yes
Random Word Detection
No
paypel, securelogin
Random Word Count
aonhbcjfugnof, sadfgfhgjdhkfjmhnsb
No
length > 7
Yes
securelogin
Add to word list to be analyzed
paypel
The Word Decomposer
Add to word list to be analyzed
secure, login
Maliciousness Analysis
Feature Extraction
End

#### 4.2.2 Word Decomposer Module (WDM)

Incoming URLs with a lot of words are broken up into individual words or objects by the Word Decomposer Module. The first step is to eliminate any digits from the original words because attackers often employ numeric values to confuse the location. A dictionary is then used to check the remaining string for the presence of any of the terms. If a word appears in the dictionary, it is instantly added to the list of words. To identify potential adjacent phrases that are also included in the list of words, a word that is not in the dictionary is broken down into substrings. In Figure 5, the Word Decomposer's execution sequence is shown. It is crucial to confirm a word's existence in the dictionary before beginning the extraction process. No additional parsing is required if the word is located. Dictionary words that can be written in a contiguous manner can be distinguished by the decomposer module. It uses a package named Enchant, which is openly accessible, to achieve this (Pyenchant, 2017).

#### 4.2.3 Random Word Detection Module (RWDM)

Phishing URLs are difficult to identify because they usually contain randomly generated words. One helpful method for identifying these potential random words is to examine their length. In order to address this, we developed the Random Word Detection Module , which is based on the GitHub16 open-source Gibberish Detector . In this work, random words are found using the Markov Chain Model. Our approach first trains the system using texts from a control language. During the training phase, the probability of two consecutive characters is established. This value becomes an essential part of our system. Since this computation only considers alphabetical characters and spaces, there is no need to assess the probabilities of other character kinds. In order to determine whether a word is random, we examine its subsequent character pairings throughout the test phase. The likelihood of these letter pairs is then multiplied to generate a fitness value. This fitness rating allows the system to identify whether the word is random. A high fitness value suggests that the term is most likely real. Conversely, a low value implies that the word is arbitrary

### 4.2.4 Maliciousness Analysis Module (MAM)

To ascertain whether the words in a certain URL are being used fraudulently, the Maliciousness Analysis Module was developed. This module's primary objective is to detect typosquatting, also known as URL hijacking. Typosquatting is the practice of users writing wrong internet addresses by accident, which often redirects them to malicious websites designed to steal personal information.

### 4.2.5 Random Forest Classifier

It generates several decision trees, each constructed from a randomly selected subset of the training data. The final classification is derived from the sum of the predictions made by each tree. This ensemble technique successfully reduces overfitting, a significant issue with individual decision trees.Random forests are powerful because they average multiple decision trees' predictions, mitigating the instability of individual trees. This approach helpsthe model generalize better, especially when dealing with noisy or unbalanced data.

$$\hat{y} = \frac{1}{N}\sum_{i=1}^{N} f_i(x)$$

### 4.2.5.1 Support Vector Machine (SVM)

Using the kernel method, the Support Vector Machine (SVM), a potent classification tool, can locate non-linear decision boundaries in feature space. One effective training technique for SVMs that streamlines the optimization process is Sequential Minimal Optimization (SMO). It accomplishes this by decomposing the main issue into smaller, easier-to-manage subproblems that can be successfully resolved. SMO is one of the most widely used algorithms for SVM training due to its ease of use .

$$f(x) = \text{sign}\left(\sum_{i=1}^{m} \alpha_i y_i K(x_i, x) + b\right)$$

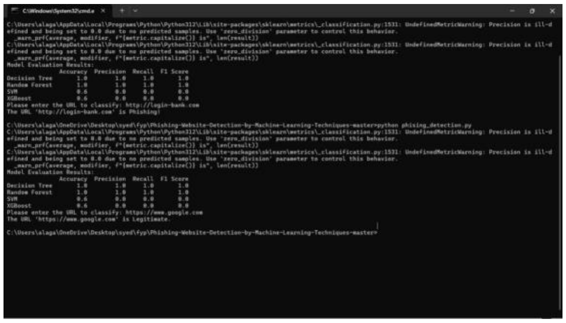
### 4.2.5.2 Decision Tree Classifier

The Decision Tree algorithm is a well-liked supervised learning technique that performs well in both classification and regression tasks. By splitting the training dataset into smaller subgroups recursively, it locates decision boundaries in a structure resembling a tree. Each decision node in the tree divides the data based on a particular attribute, and each leaf node is assigned a class, which is often determined by calculating the probability that each class will appear in that leaf. Because of their structure, decision trees are interpretable models; yet, they may be prone to overfitting if not properly adjusted.

$$H(D) = -\sum_{k=1}^{K} p_k \log_2(p_k)$$

$$G(D) = 1 - \sum_{k=1}^{K} p_k^2$$

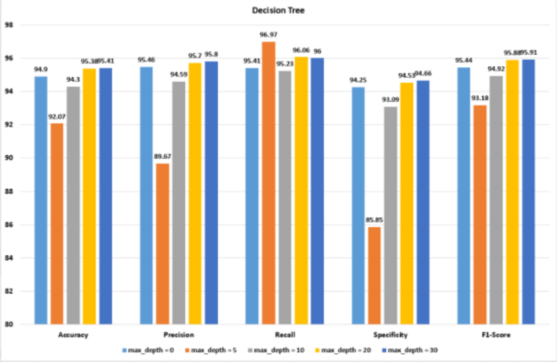
## 5.Results & Analysis



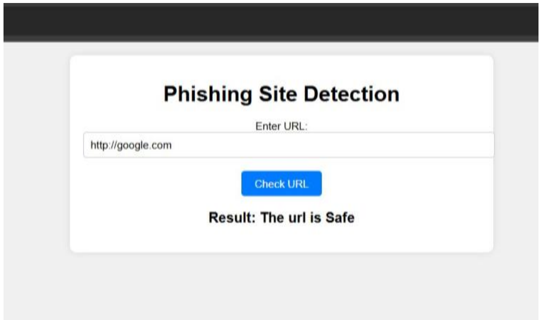
```
C:\Users\alaga\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\metrics\_classification.py:1531: UndefinedMetricWarning: Precision is ill-d
efined and being set to 0.0 due to no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\alaga\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\metrics\_classification.py:1531: UndefinedMetricWarning: Precision is ill-d
efined and being set to 0.0 due to no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
Model Evaluation Results:
               Accuracy  Precision  Recall  F1 Score
Decision Tree       1.0        1.0     1.0       1.0
Random Forest       1.0        1.0     1.0       1.0
SVM                 0.6        0.0     0.0       0.0
XGBoost             0.6        0.0     0.0       0.0
Please enter the URL to classify: http://login-bank.com
The URL 'http://login-bank.com' is Phishing!

C:\Users\alaga\OneDrive\Desktop\syed\fyp\Phishing-Website-Detection-by-Machine-Learning-Techniques-master>python phising_detection.py
C:\Users\alaga\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\metrics\_classification.py:1531: UndefinedMetricWarning: Precision is ill-d
efined and being set to 0.0 due to no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\alaga\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\metrics\_classification.py:1531: UndefinedMetricWarning: Precision is ill-d
efined and being set to 0.0 due to no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
Model Evaluation Results:
               Accuracy  Precision  Recall  F1 Score
Decision Tree       1.0        1.0     1.0       1.0
Random Forest       1.0        1.0     1.0       1.0
SVM                 0.6        0.0     0.0       0.0
XGBoost             0.6        0.0     0.0       0.0
Please enter the URL to classify: https://www.google.com
The URL 'https://www.google.com' is Legitimate.

C:\Users\alaga\OneDrive\Desktop\syed\fyp\Phishing-Website-Detection-by-Machine-Learning-Techniques-master>
```

**Figure 5.1**

| max depth | Accuracy | Precision | Recall | Specificity | F1-score |
|---|---|---|---|---|---|
| 0 | 94.9 | 95.46 | 95.41 | 94.25 | 95.44 |
| 5 | 92.07 | 89.67 | 96.97 | 85.85 | 93.18 |
| 10 | 94.3 | 94.59 | 95.23 | 93.09 | 94.92 |
| 20 | 95.38 | 95.7 | 96.06 | 94.53 | 95.88 |
| 30 | 95.41 | 95.8 | 96 | 94.66 | 95.91 |



Lack of access to a well-known dataset was one of the difficulties in testing the suggested approach. This led to the creation of a custom dataset, which is described in Section 4 and released as the Ebbu2017PhishingDataset (2017). Owing to the quantity of the dataset, testing was done on 73,500 URLs, 36,390 of which were valid and 37,200 of which were phishing. WEKA was used for testing, and pre-made libraries were used. Throughout the experiments, all algorithms used default parameter values and a 10-fold cross-validation methodology. Seven distinct machine learning methods were used to assess each test set. Table 4, which shows the findings for NLP-based features.



http://google.com

Result: The url is Safe

## 6. References

1. Abu-Nimeh, S., Nappa, D., Wang, X., & Nair, S. (2007). A comparison of machine learning techniques for phishing detection. In Proceedings of the Anti-Phishing Working Group's 2nd Annual eCrime Researchers Summit, eCrime '07, ACM, New York, NY, USA (pp. 60– 69).

2. Babagoli, M., Aghababa, M. P., & Solouk, V. (2018). Heuristic nonlinear regression strategy for detecting phishing websites. Soft Computing, 1–13.

3. Buber, E., Diri, B., & Sahingoz, O. K. (2017a). Detecting phishing attacks from URLs using NLP techniques. In 2017 International Conference on Computer Science and Engineering (UBMK) (pp. 337–342).

4. Buber, E., Diri, B., & Sahingoz, O. K. (2017b). NLP-based phishing attack detection from URLs. In A. Abraham, P. K. Muhuri, A. K. Muda, & N. Gandhi (Eds.), Intelligent Systems Design and Applications (pp. 608–618). Springer International Publishing, Cham.

5. Cao, Y., Han, W., & Le, Y. (2008). Anti-phishing based on automated individual white-list. In Proceedings of the 4th ACM Workshop on Digital Identity.

6. Cook, D. L., Gurbani, V. K., & Daniluk, M. (2008). Phishwish: A stateless phishing filter using minimal rules. In Financial Cryptography and Data Security (pp. 182–186). Springer, Berlin, Heidelberg.25

7. Chiew, K. L., Yong, K. S. C., & Tan, C. L. (2018). A survey of phishing attacks: Their types, vectors, and technical approaches. Expert Systems with Applications, 106, 1–20.

8. Curtis, S. R., Rajivan, P., Jones, D. N., & Gonzalez, C. (2018). Phishing attempts among the dark triad: Patterns of attack and vulnerability. Computers in Human Behavior, 87, 174–182.

9. Feng, F., Zhou, Q., Shen, Z., Yang, X., Han, L., & Wang, J. (2018). The application of a novel neural network in the detection of phishing websites. Journal of Ambient Intelligence and Humanized Computing.

10. Fu, A. Y., Wenyin, L., & Deng, X. (2006). Detecting phishing web pages with visual similarity assessment based on Earth Mover’s Distance. IEEE Transactions on Dependable and Secure Computing, 3(4), 301–311.