Phishing Website Detection Using URL-Based Features

A Machine Learning Approach to Combat Fraudulent Links

Ayrton Joseph DiPina  
 Data Science  
 Wentworth Institute of Technology  
 Boston, Massachusetts, USA  
 dipinaa@wit.edu

ABSTRACT

Phishing attacks are a pervasive cybersecurity threat. They exploit user trust to steal sensitive information such as user credentials or financial data. This project used machine learning techniques to classify phishing and non-phishing websites. The dataset, sourced from UC Irvine, includes URL, domain, and webpage features. Multiple machine learning models, such as Random Forest, Logistic Regression, and Support Factor Machine, were used for classification.

KEYWORDS

* Phishing
* Machine Learning
* URL Features

1 Introduction

On July 19th, 2024, a faulty software update from CrowdStrike, a reputable cybersecurity firm, caused a global IT outage. [6] The update contained a defect that caused Windows computers to crash, rendering the computers unable to restart. This incident affected about 8.5 million worldwide devices. This incident impacted airports, banks, and multiple other businesses. At the time of this global panic, I was a cybersecurity intern at a water company. I was researching what was happening to see if this faulty update led to our systems being down. After confirming that CrowdStrike did not affect our computers, I noticed many links that did not look official. Then I realized that malicious users were creating fake links to try and socially engineer employees panicking to try and fix their computers that were impacted by CrowdStrike, acting as CrowdStrike Support.

The proliferation of online phishing attacks has become a critical cybersecurity concern, as these deceptive techniques exploit users into divulging sensitive information such as passwords, credit card details, or personal identification numbers. Phishing detection, therefore, plays a vital role in securing internet users and organizations from financial loss and data breaches. This project, which identifies phishing websites using URL-based features, is a significant step towards enhancing cybersecurity. It offers a lightweight and real-time approach that can augment broader cybersecurity measures, providing reassurance and confidence in the fight against phishing.

2 Data

For this analysis, the UCI ML Phishing Dataset was sourced from Kaggle and derived initially from the UCI Machine Learning Repository. It includes a mix of URL, domain, and content-based features, such as SSL certificate status, presence of IP addresses, URL Length, and domain age. These were labeled with a target variable indicating “phishing” or “non-phishing.” This well-structured dataset is primarily used for research and educational purposes in phishing detection. It offers insights into the characteristics that differentiate phishing sites from legitimate ones. Its categorical and ordinal data format makes it suitable for machine learning and classification projects.

**2.1** **Source of dataset**

The data was obtained from the Phishing Dataset from Kaggle [8], a widely used and credible platform for machine learning and data science datasets. This dataset was taken from the UC Irvine Phishing dataset and converted into a .csv file. [4] UC Irvine created this dataset using the PhishTank archive, MillerSmiles archive, and Google Searching Operators. URLs were gathered and classified as “phishing” or “non-phishing” based on various markers, such as URL patterns, HTTPS certificates, and domain registration information. These classifications were determined to be either classified manually or provided by the phishing archives used to create the dataset.

**2.2** **Characters of the Datasets**

The UCI dataset downloaded from Kaggle was converted from a .csv to a .xlsx file. The dataset size is 31 columns and 11,055 rows, including one target column, result, indicating phishing or non-phishing results. See Appendix Item 1 for the complete feature list. No significant cleaning was required for this data. The dataset was already well structured and did not contain any missing values.

3 Methodology

For this analysis, multiple machine-learning models were employed to classify websites as phishing or non-phishing. These models include Logistic Regression, Random Forest, and Support Vector Machine.

**3.1 Random Forest**

Random Forest was used to identify the most helpful URL features for spotting phishing sites (**Question 1**) and to test whether URL information alone suffices for classification (**Question 4**). [5] As an ensemble method, it combines multiple decision trees to effectively capture non-linear relationships and mixed data types. Its robustness to noise and ability to highlight feature importance make it highly suitable for these tasks. Random Forest can be computationally intensive and prone to overfitting without proper tuning. Implemented with the RandomForestClassifier function from sklearn, hyperparameters such as the number of trees (n\_estimators) and tree depth (max\_depth) were optimized using grid search to improve performance.

**3.2 Support Vector Machine**

Support Vector Machine (SVM) was employed to assess whether a more complex model is necessary to detect phishing sites accurately (**Question 3**). [1] SVM identifies a hyperplane in a high-dimensional space to separate classes with the maximum margin, leveraging kernel functions to handle non-linear relationships. This makes it suitable for addressing complex class boundaries in phishing classification. SVM requires careful tuning of parameters like C and gamma and can be computationally expensive for larger datasets. The SVC function from sklearn, the R BF kernel, and hyperparameter optimization via grid search were used to enhance its effectiveness.

**3.3** **Logistic Regression**

Logistic Regression was applied to explore whether a simple model could effectively detect phishing sites (**Question 3**). Its simplicity and interpretability make it an ideal baseline model. [3] This method assumes a linear relationship between the independent variables and the log odds of the dependent variable, making it computationally efficient and effective for linearly separable data. Limitations include reduced performance for complex, non-linear data and susceptibility to multicollinearity. Implemented using the Logistic Regression function from sklearn, adjustments such as increasing the maximum iterations (max\_iter=1000) were made to ensure convergence and accuracy improvement.

4 Results

This analysis provides insights into phishing websites using machine learning models. The following questions were answered using the methodologies detailed in this paper:

1. What URL Features are most helpful in spotting phishing sites?
2. Does using HTTPs make a phishing site look more trustworthy?
3. Can a simple model detect phishing accurately, or do we need a more complex model?
4. Is URL information enough to classify phishing, or do we need more data from the webpage?

**4.1 What URL Features are most helpful in spotting phishing sites?**

For this question, a Random Forest classifier was used to determine which URL features most indicate phishing. The model identified the top features that contributed most to accurate classification by calculating feature importance scores. The top ten features are displayed in Figure 1.

**A green bar graph with white background

Description automatically generated**

Figure 1 - Top 10 Features by Importance

SSLfinale\_State, URL\_of\_Anchor, and having\_Sub\_Domain were among the most important features for distinguishing a phishing site. The SSL feature indicates the validity of the website's SSL certificate. [4] Phishing sites often lack valid SSL certificates or use untrusted ones, making this feature a reliable indicator of legitimacy. Sites with invalid SSL certificates are more likely to be phishing attempts, as legitimate sites prioritize secure communication.

The URL\_of \_Achor feature evaluates the proportion of anchor tags that link to external URLs. Phishing frequently uses external links to redirect users to malicious sites, making this an essential feature for detecting their behavior. Many external anchors suggest suspicious activity, while legitimate sites maintain consistent internal linking.

Phishing websites often use subdomains to mimic legitimate domains. [7] For example, a phishing site might use “login.bankexample.com.mal.com” to appear as a legitimate banking site. The having\_Sub\_Domain feature captures the number of subdomains, with higher counts more indicative of phishing attempts.

These features collectively highlight the tactics employed by phishing sites, such as exploiting security vulnerabilities, manipulating links, and using deceptive domain structures. Their high importance in the Random Forest model underscores their effectiveness in identifying phishing.

**4.2 Does using HTTPs make a phishing site look more trustworthy?**

The SSLfinale\_State feature was utilized to see the impact of HTTPS on making a phishing site look more trustworthy. The data revealed the following in Figure 2.

A graph with blue and red bars

Description automatically generated

Figure 2 - Impact of HTTPS on Phishing vs Non-Phishing

A significant portion of phishing websites have an invalid or untrusted SSL. Non-phishing websites predominantly possess valid SSL certificates, and many phishing sites also use valid SSL certificates.

This indicates that while HTTPS is more common among legitimate websites, some phishing sites also employ HTTPS to appear trustworthy. HTTPS alone is not a definitive indicator of a site’s legitimacy. Users should be casual and consider additional factors beyond HTTPS when assessing a website’s trustworthiness.

**4.3 Can a simple model detect phishing accurately, or do we need a more complex mode?**

To answer this question, we used two models: Logistic Regression, our simple model, and Random Forest, our slightly complex model. Logistic regression yielded a **92.3%** accuracy, while the Random Forest yielded a **97.5%** accuracy.

[3] [5] While Logistic Regression provided a respectable yield, it was outperformed by the Random Forest model. The higher accuracy of the Random Forest suggests that capturing the complex, non-linear relationships in the data is needed to improve classification performance. A more complex model like Random Forest is preferable for accurately detecting phishing websites.

**4.4 Is URL information enough to classify phishing, or do we need more data from the webpage?**

The analysis evaluated the predictive power of URL-based features alone compared to the entire dataset. The model trained exclusively on URL features achieved an accuracy of **89.7%,** while the model incorporating all features reached a higher accuracy of **97.5%**.

[4] This details that URL information, such as having\_IP\_Address, URL\_Length, and having\_Sub\_Domain, strongly predicts phishing websites. These features capture many common tactics phishers use, including unusual URL structures, excessive subdomains, or using IP addresses instead of domain names.

5 Discussion

This project achieved strong results in identifying phishing sites using machine learning models; several limitations highlight the need for further improvement. One weakness is the dataset’s age, which was created in 2015. [2] Phishing tactics have evolved significantly since then, and modern phishing campaigns often use more sophisticated techniques, such as exploiting zero-day vulnerabilities. Features that were effective predictors in 2015 may no longer adequately capture the nuances of current phishing threats.

The dataset focuses on static URL-based and webpages features, limiting the models’ ability to detect advanced phishing techniques. [9] User interactions with a page, real-time content updates, and server-side details are absent but increasingly important in combating modern phishing. The models in this project are static and do not incorporate real-time updates or adaptive measures to respond to evolving threats. While the Random Forest model yielded high accuracy, false negatives remain risky, exposing users to potential harm. False positives can undermine trust in the detection system and disrupt legitimate websites.

Future improvements will focus on utilizing a more updated dataset to reflect current phishing techniques. This includes collecting data from recent phishing campaigns and integrating dynamic features such as server-side attributes. Leveraging advanced techniques like deep learning and adaptive models that retrain periodically with new data can enhance the system. Testing the models in live environments and evaluating real-world performance will ensure practicality. Creating detection systems with user education and warnings explaining why a website was flagged enhances the trust between us and the user.

6 Conclusion

This project highlights using machine learning to address the evolving threat of phishing attacks. Analyzing the dataset from the UCI Machine Learning Repository involved using various models to classify phishing and non-phishing websites. The study revealed the importance of features such as SSLfinale\_State URL\_of\_Anchor and having\_Sub\_Domain, which underscore the tactics commonly employed by phishing sites to deceive users.

While the models achieved high accuracy, the dataset's limitations and the models' static nature highlight opportunities for future improvement. Future work will be focused on developing adaptive models that can retrain with new data to reflect evolving phishing techniques. While these improvements are implemented, users are encouraged to stay vigilant about phishing websites.

REFERENCES

[1] 1.4. Support Vector Machines. 2007a. *Scikit-Learn*. https://scikit-learn.org/1.5/modules/svm.html.

[2] History of phishing. 2022. *History of Phishing*. <https://www.phishing.org/history-of-phishing>.

[3]Logisticregression.2007b.*ScikitLearn*.https://scikitlearn.org/1.5/modules/generated/sklearn.linear\_model.LogisticRegression.html.

[4] Mohammad, R. and McCluskey, L. 2015. Phishing websites. *UCI Machine Learning Repository*. https://archive.ics.uci.edu/dataset/327/phishing+websites.

[5]Randomforestclassifier.2007c.*Scikit-Learn*.<https://scikit-learn.org/1.5/modules/generated/sklearn.ensemble.RandomForestClassifier.html>.

[6] Singh, M. 2024. Faulty crowdstrike update causes major global it outage, taking out banks, airlines and businesses globally. *TechCrunch*. <https://techcrunch.com/2024/07/19/faulty-crowdstrike-update-causes-major-global-it-outage-taking-out-banks-airlines-and-businesses-globally/>.

[7 ]Tayar, C. 2023. Subdomain hijacking: The domain’s Silent Danger  . *Cyberint*. <https://cyberint.com/blog/research/subdomain-hijacking-the-domains-silent-danger/>.

[8 ]Yadav,S.2020. Phishing dataset UCI ML CSV. *Kaggle*. <https://www.kaggle.com/datasets/isatish/phishing-dataset-uci-ml-csv/data>.

[9] Árnason, Á.T. 2024. Social Engineering vs phishing: Understanding the differences. *Keystrike*. https://keystrike.com/social-engineering-vs-phishing-understanding-the-differences/.

Conference Name:ACM Woodstock conference

Appendix

Appendix Item 1

|  |  |  |
| --- | --- | --- |
| Feature | Description | Units |
| ID | Unique identifier for each record | Integer |
| Having\_IP\_Address | Whether the URL uses an IP address | Categorical |
| URL\_Length | Length of the URL | Categorical |
| Shortining\_Service | Use of URL shortening service | Categorical |
| having\_At\_Symbol | Presence of “@” symbol in URL | Categorical |
| Double\_slash\_redirecting | Double slashes in redirect | Categorical |
| Prefix\_Suffix | Presence of “-“ in domain | Categorical |
| Having\_Sub\_Domain | Number of subdomains | Categorical |
| SSLfinale\_State | SSL status | Categorical |
| Domain\_registeration\_length | Domain validity | Categorical |
| favicon | Whether the favicon is from another domain | Categorical |
| port | Abnormal port usage | Categorical |
| HTTPS\_token | Use of “https” in domain name | Categorical |
| Request\_URL | Requests made to external URLs | Categorical |
| popUPWindow | Presence of pop-ups | Categorical |
| Iframe | Usage of iframe tags | Categorical |
| age\_of\_domain | Domain age | Categorical |
| DNSRecord | DNS record availability | Categorical |
| Web\_traffic | Website Traffic | Categorical |
| Page\_Rank | Page rank | Categorical |
| Google\_Index | Presence in Google Index | Categorical |
| Links\_pointing\_to\_page | Number of links to page | Categorical |
| Statistical\_report | Reported by statistical engines | Categorical |
| Result | Target variable | Categorical |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Conference Short Name:WOODSTOCK’18

Conference Location:El Paso, Texas USA

ISBN:978-1-4503-0000-0/18/06

Year:2018

Date:June

Copyright Year:2018

Copyright Statement:rightsretained

DOI:10.1145/1234567890

RRH: F. Surname et al.

Price:$15.00